<Final Report: EEE4610-01>

Deep Lensless Partial Reconstruction via Convolutional Neural Network

Donggeon Bae
School of Electrical and Electronic Engineering
College of Engineering
Yonsei University

<Final Report: EEE4610-01>

Deep Lensless Partial Reconstruction via Convolutional Neural Network

Report Advisor: Seung Ah Lee

December 2022

Donggeon Bae
School of Electrical and Electronic Engineering
College of Engineering
Yonsei University

Contents

ABSTRACT

Deep Lensless Partial Reconstruction via Convolutional Neural Network

By replacing lens modules with thin masks and computation, cameras can be built at a low cost with a small form-factor. Though iterative optimization and deep learning approaches can be used in computation these days, there are more problems facing commercial point of view, such as weakness of quality and difficulty modulating systems. A representative issue is that not all lights from a scene can be in the camera's sensor for a wide field of view because of diffusing property. The problem makes the quality of the restored image worse.

Focusing on the convolution property in lensless imaging, we define the system as re-generatable with the remaining part of a measurement and an inpainting network. Our goal is to reconstruct a scene while deriving more information from a restricted raw image. In this paper, we propose a 3-staged deep neural network architecture for the reconstruction via image inpainting, predicting disappeared measurements in various situations such as boundary crop and random masking. Through a coarse-to-refine inpainting network, Wiener deconvolution, and enhancing network, we can restore the scene from the sparse measurements.

Optics Lensless-camera Deep-Learning Computer-Vision Image-Reconstruction GAN Inpainting-Network Compressive-Imaging

1. Introduction

A typical camera based on a lens module has many emerging applications such as wearables, device cameras, virtual reality, and many others with light-weight and miniaturized systems. However, the lens modules have limits in reducing those volumes and weights with a fixed cost.

Lensless cameras, replacing lens modules with thin masks and computation, have emerged recently as a new type of camera [1-2]. Using a surface-curvature mask such as a phase mask or light-weight diffusers, we can design the physical system to be ultra-compact and cost-effective imaging. Due to the lensless imaging's diffusing property, the sensor takes in light as a highly complicated measurement. The Point Spread Function (PSF) is a pattern taken by the lights from a point source passing through the mask. We can use reconstruction algorithms to solve deconvolution problems with PSF under the assumption that the system is shift-invariant. Including the usage of Deep Neural Networks (DNNs), several methods to improve the performance of reconstruction have been studied [3-5].

In particular, recovering the scene from the measurement faces challenges with the physical structure of the lensless imaging system [4]. Previous research has a non-separable problem that indicates the system is unsuitable for real-wild images. The problem arises from the property of convolution and the imperfect spatially variance of PSF. Because the sensor does not absorb all the diffused lights from the scene passing the mask, the receptive field is restricted. On the other hand, any small portion of dead or mistaken pixels from the sensor is treated as critical noise in the process of deconvolution. The defect in these cases affects the quality of reconstruction. Various research dealt with the above challenges by using trainable inversion with an additional padding method [4], generating parameterized PSF [6], making the diffuser as a randomly dispersed microlens [7], and stacking the measurement in a 3D manner [8].

Here, we propose DNNs for compressive lensless imaging, conducting an inpainting task in the case of restricting receptive fields or missing pixels. Our network consists of three stages: coarse-to-refine inpainting, deconvolution in an inversion manner, and enhancement. We show that the inpainting task improves performance in compressive imaging, in a way the previous studies have not attempted.

Background

2-1. Lensless imaging

A lensless camera physically uses a thin mask such as a phase mask or a weak diffuser in front of a sensor [1-2]. Point spread function (PSF) is a pattern that casted light from a point source because of the mask's small curvature surface, and generally we suppose it is shift-invariant, thus can model the imaging system as a convolution between the scene and the PSF. By deconvolution, we can get a competent output from the measurement.

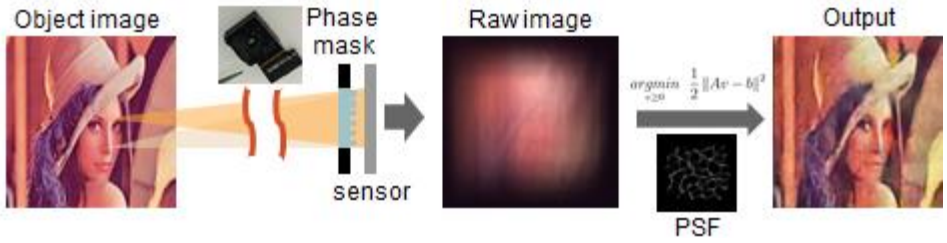

Figure 1. Overview of the lensless imaging system.

2-1-1. Wiener deconvolution

We can use wiener deconvolution to untie convolutional combinations [9]. It is an application of the wiener filtering to the noise, attempting to minimize noises in the process of deconvolution. Since we formulate the lensless system as the convolution between a scene and the PSF, we can define the problem to be deconvolutional filtering to the measurement. Now we can write the equation both for the measurement and the target:

$$\text{measurement}(x, y) = (\text{PSF} * \text{scene})(x, y) + \text{noise}(x, y) \quad (1)$$

$$\hat{X}(f) = G(f)Y(f) \quad (2)$$

Where each X(f) and Y(f) are Fourier transforms of the scene, and the measurement. We find G(f) to get a desired estimation of the target. The operation of the wiener filtering equation follows as we set H to the Fourier transform of the PSF:

$$G(f) = \frac{1}{H(f)}\left[\frac{1}{1 + 1/(|H(f)|^2 \text{SNR}(f))}\right] \quad (3)$$

Here, 1/H(f) is the inverse of the Fourier transform of the PSF, SNR=S(f)/N(f) is the signal to noise ratio, using the mean power of spectral density of the scene S(f) and the noises N(f).

2-1-2. Partial reconstruction

Because measurement stands for convolution operation between a scene and a camera's PSF, information from the scene spreads all over the measurement so we can derive the original even if there are holes in the measurement. However, another problem arises from this property of diffusing. Because the sensor does not absorb all the diffused lights from the scene passing the mask, the receptive field is restricted. On the other hand, any small portion of dead or mistaken pixels from the sensor is treated as critical noise in the process of deconvolution.

The two points of view conflict, but still they can be handled. With only an advantage of the system's property, we let a network learn to refill the missing parts with the global information of the measurement.

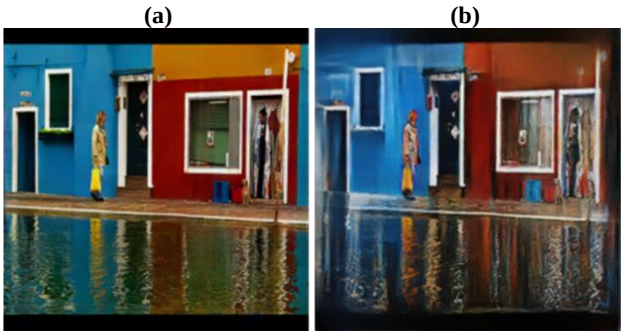

Figure 2. Performance degradation derived from a cropped measurement: (a) output from our network with a non-crop measurement of size 600×600; (b) cropped measurement of size 300×300.

2-2. Deep lensless imaging

Previous methods such as Wiener deconvolution or iterative optimizations have limits with long inference time and fatal errors in spatially varying systems like partial reconstruction. Deep neural networks can handle these limits. DNNs are studied to solve lensless imaging problems in various systems [3-6]. For lensless image reconstruction, researchers have managed to take features from the measurements by using networks highly observative for the features and by combining iterative optimization techniques [5] or deconvolution [4].

2-2-1. Image reconstruction

Typically, Encoder-Decoder type of neural networks [10] is used for restoration in lensless cameras, simply using end-to-end modules [3]. U-Net [10], called encoder-decoder, is a fully

convolutional network combined with upsampling, downsampling, and skip connection as you can see in Figure 3. Typically, the encoding path captures the input image's features by adding channels and decreasing feature dimension. The decoding path reconstructs encoded data desirable by reversing them. However, steps like these lose certain positional information in a part of channel decrease, and we cannot get the proper information in the decoding path since they use low dimensional features only. The model solves this problem by adding the skip connection, which concatenates each layer's features we get from an encoding path to a layer of a decoding path.

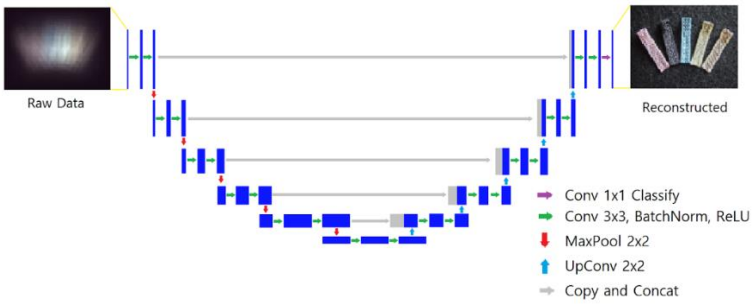

Figure 3. Network architecture of vanilla U-Net for deep lesnelss imaging [3]. Note that the architecture table is attached in Table 4.

2-2-2. Partial reconstruction via DNNs

In the previous research [4], combining the deconvolution part, reconstruction also can be improved using a trainable filtering system and denoisers. Parameterizing the deconvolution part with the U-Net denoiser improves overall performance.

We can choose replicated padding for the cropped measurements improving deconvolution quality in the random crop condition as shown in Figure 4. This approach for imaging in the restricted environment makes the model separable from the camera [4].

2-3. DNNs for image inpainting

Nowadays, inpainting networks [11-19] are typically based on Generative adversarial network (GAN) [20]. GAN is a powerful generative model that generator network and discriminator network learn in an adversarial manner, improving each itself. Generating for the missing parts with GAN can get advantages of diversity and highly adaptable to the input.

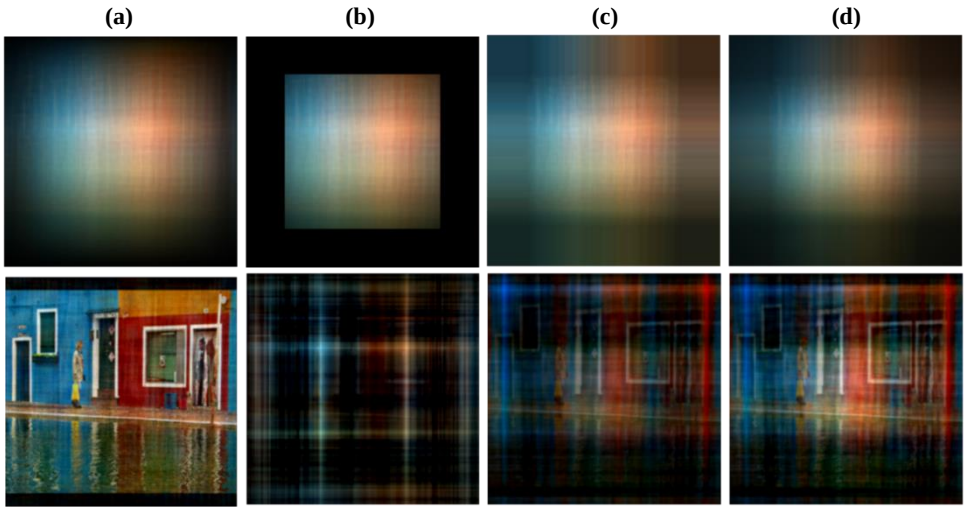

Figure 4. Deconvolution performance along the padding methods. The first row indicates measurements, and the second row reconstruction from each given measurement: (a) Full size measurement size of 600×600; (b) Zero padding for the cropped measurement size of 400×400; (c) Replicated padding for the cropped measurement; (d) Gaussian filtering after replicated padding for the cropped measurement.

2-3-1. Generative Adversarial Network

A generator makes fake data from a given latent vector. Thus, the generated data looks real. A discriminator distinguishes what is real or fake from the generator. The overall formulation is:

$$\min_{\theta,\phi}\max \mathrm{E}_{\mathrm{x}\sim \mathrm{p}_{\mathrm{data}}}\log \mathrm{D}_{\phi}(\mathrm{x}) + \mathrm{E}_{\mathrm{z}\sim \mathrm{p}(\mathrm{z})}\log\left(1 - \mathrm{D}_{\phi}\left(\mathrm{G}_{\theta}(\mathrm{z})\right)\right) \quad (4)$$

Where G is the generator, D is the discriminator, and z for a latent vector for the generator input. In the objective function above, we can see the discriminator makes the loss close to 0 for the fake data $\mathrm{G}_{\theta}(\mathrm{z})$ and 1 for the real data x, while the generator shows an adversarial behavior maximizing losses.

2-3-2. Inpainting network

Inpainting network, which generates desirable features from randomly lost measurements, uses this powerful tool. For this task, several research arise such as using contextual information from given information [11], semantic ways [12], prior information from a preceded network output [13]. There're mixed cases of discriminator and CNN, using global and local refinement network ensembles [14-16].

In lensless imaging system, inpainting task gets more complicated because a portion of changes in measurement affects the whole reconstruction. Reversely, the diffusing property also can be an advantage because we can get highly integrated information from the entire remaining area. We try this to make our tasks more highly applicable, taking high quality of finally reconstructed output.

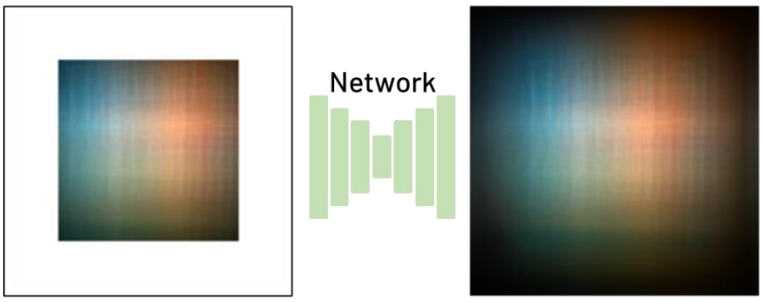

Figure 5. Inpainting network plan for the lensless imaging system.

2. Method

3-1. Network

We design a model with different receptive fields for our network to successfully make up missing parts and restore the scene from the measurement. We train three-different networks following an inpainting-then-reconstruct framework as shown in Figure 6. Firstly, we use a U-Net to a coarse network for a larger receptive field than the input image's resolution. Then, with a shallow network for a small receptive field, the model refines coarse-inpainting output. After the whole course-to-refine network output, our model deconvolves and denoises it with the complicated encoder-decoder shape of enhancing network. We denoted all the stages each as "C", "R", "Deconv", and "E".

3-1-1. Coarse inpainting network

For image inpainting, the coarse inpainting network takes U-Net architecture with the skip connection, consisting of seven downsampling and upsampling layers. The propagated information through the skip connections recovers lost during downsampling. After this encoder, the receptive field is much larger than the input to get a beneficial effect on the completion. We used 0.2 negative slope LeakyReLU in the encoder and ReLU in the decoder. Discriminator loss

with patch-based discriminator and spectral normalization reduces a whole inpainting network output's blur effect. The discriminator takes a raw image and a label image as input and takes out a 2D feature map. Each element of the 2D feature map indicates that the output of the inpainting network is real or fake compared to the ground truth.

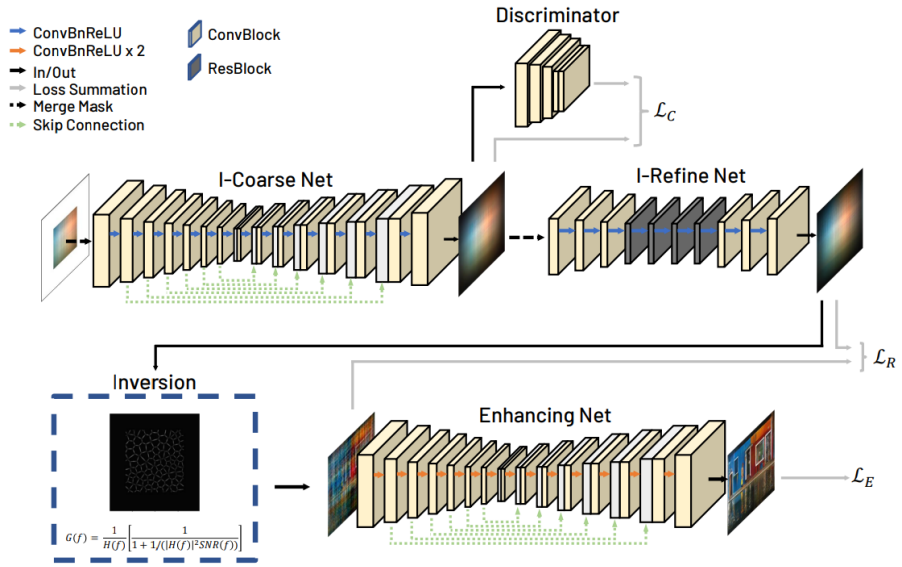

Figure 6. Overview of our proposed network, consisting of 3-stages of coarse and refine inpainting, deconvolution and enhancing.

We use image **I** and binary mask **M** as input, and as inpainting result $\mathbf{I}_{\mathbf{out}}$ as output. We can define a pixel-wise reconstruction loss and an adversarial loss for the discriminator. For ground truth image $\mathbf{I}_{\mathbf{GT}}$, $\odot$ is the element-wise production, and N(**M**) indicates the sum of the non-zero elements in **M**. With weighted factor λ, we use the loss as:

$$\mathcal{L}_{\mathrm{C}} = \lambda_{\mathrm{hole}}\mathcal{L}_{\mathrm{hole}}^{\mathrm{C}} + \lambda_{\mathrm{valid}}\mathcal{L}_{\mathrm{valid}}^{\mathrm{C}} + \lambda_{\mathrm{G}}\mathcal{L}_{\mathrm{G}}^{\mathrm{C}} \tag{5}$$

$$\mathcal{L}_{\mathrm{D}} = \frac{1}{2}\mathbb{E}_{\mathbf{I}\sim\mathbf{pdata}}(\mathbf{I})[(\mathrm{D}(\mathbf{I}_{\mathbf{GT}}) - 1)^2] + \frac{1}{2}\mathbb{E}_{\mathbf{I}_{\mathbf{out}}\sim\mathbf{pI}_{\mathbf{out}}}(\mathbf{I}_{\mathbf{out}})[\mathrm{D}(\mathbf{I}_{\mathbf{out}})^2] \tag{6}$$

$$\mathcal{L}_{\mathrm{valid}} = \frac{1}{\sum \mathrm{N}(\mathbf{1} - \mathbf{M} = 1)} \|(\mathbf{I}_{\mathbf{out}} - \mathbf{I}_{\mathbf{GT}}) \odot (\mathbf{1} - \mathbf{M})\|_1 \tag{7}$$

$$\mathcal{L}_{\mathrm{hole}} = \frac{1}{\sum \mathrm{N}(\mathbf{M} = 1)} \|(\mathbf{I}_{\mathbf{out}} - \mathbf{I}_{\mathbf{GT}}) \odot \mathbf{M}\|_1 \tag{8}$$

Finally, the total loss for the coarse network is:

$$\mathcal{L}_{\mathrm{C}} = \lambda_{\mathrm{hole}}\mathcal{L}_{\mathrm{hole}}^{\mathrm{C}} + \lambda_{\mathrm{valid}}\mathcal{L}_{\mathrm{valid}}^{\mathrm{C}} + \lambda_{\mathrm{G}}\mathcal{L}_{\mathrm{G}}^{\mathrm{C}} \tag{9}$$

Here we use factors of λ_{hole}=6, λ_{hole}=1 and λ_G=0.1.

Name	Input	Channel (in, out)	Feature (in, out)	Activation	Kernel	stride	Padding	BN
L1	cat (Masked_img, Mask)	(4, 32)	(600, 300)	None	4	2	1	N
L2	F_L1	(32, 64)	(300, 150)	LeakyReLU	4	2	1	Y
L3	F_L2	(64, 128)	(150, 75)	LeakyReLU	4	2	1	Y
L4	F_L3	(128, 256)	(75, 37)	LeakyReLU	4	2	1	Y
L5	F_L4	(256, 512)	(37, 18)	LeakyReLU	4	2	1	Y
L6	F_L5	(512, 512)	(18, 9)	LeakyReLU	4	2	1	F
LT7	F_L6	(512, 512)	(9, 4)	LeakyReLU	5	2	1	Y
LT8	F_LT7	(512, 512)	(4, 9)	ReLU	4	2	1	Y
LT9	cat (F_L6, F_LT7)	(1024, 512)	(9, 18)	ReLU	5	2	1	Y
LT10	cat (F_L5, F_LT8)	(1024, 512)	(18, 37)	ReLU	4	2	1	Y
LT11	cat (F_L4, F_LT9)	(1024, 256)	(37, 75)	ReLU	4	2	1	Y
LT12	cat (F_L3, F_LT10)	(512, 256)	(75, 150)	ReLU	4	2	1	Y
LT13	cat (F_L2, F_LT11)	(256, 64)	(150, 300)	ReLU	4	2	1	Y
LT16	cat (F_L1, F_LT12)	(128, 32)	(300, 600)	ReLU	4	2	1	Y

Table 1. Architecture of the coarse network. “BN” stands for the batch normalization

Name	Input	Channel (in, out)	Feature (in, out)	Activation	Kernel	stride	Padding	SN
L1	coarse_out	(3, 64)	(600, 300)	LeakyReLU	4	2	1	Y
L2	F_L1	(64, 128)	(300, 150)	LeakyReLU	4	2	1	Y
L3	F_L2	(128, 256)	(150, 75)	LeakyReLU	4	2	1	Y
L4	F_L3	(256, 512)	(37, 18)	LeakyReLU	4	2	1	Y
L6	F_L5	(512, 1)	(18, 18)	None	4	2	1	N

Table 2. Architecture of the discriminator. “SN” stands for the spectral normalization.

3-1-2. Refine inpainting network

A refinement network is a shallow network, consisting of a smaller number of downsampling and upsampling than the coarse network and residual block. Passing through the encoders, the remaining input’s information is in a small receptive field, but the lost information is replaced by the skip connection. As a result, the network reduces edge effects made during inpainting in the coarse network. We used activation as ReLU in the whole network architecture except the last layer using Tanh activation.

We use several losses for the network following previous inpainting works [17-19]. (10) Total variation (TV) loss is used for smoothing [14, 17]. (11) Perceptual loss for the textual information [21], (12) style loss for preserving features [22]. At last, (13) a mix of Mean Square Error (MSE) loss and Multi-Scale Structural Similarity (MS-SSIM) [23] loss for the deconvolution is used following [24]. Formulations of each loss are:

$$\mathcal{L}_{\text{tv}} = \|\mathbf{I_m}(\text{i}, \text{j}+1) - \mathbf{I_m}(\text{i}, \text{j})\|_1 + \|\mathbf{I_m}(\text{i}+1, \text{j}) - \mathbf{I_m}(\text{i}, \text{j})\|_1 \quad (10)$$

$$\mathcal{L}_{\text{per}} = \sum_{\text{i}} \|\mathcal{F}_{\text{i}}(\mathbf{I_{out}}) - \mathcal{F}_{\text{i}}(\mathbf{I_{GT}})\|_1 + \|\mathcal{F}_{\text{i}}(\mathbf{I_m}) - \mathcal{F}_{\text{i}}(\mathbf{I_{GT}})\|_1 \quad (11)$$

$$\mathcal{L}_{\text{sty}} = \sum_{\text{i}} \|\mathcal{G}_{\text{i}}(\mathbf{I_{out}}) - \mathcal{G}_{\text{i}}(\mathbf{I_{GT}})\|_1 + \|\mathcal{G}_{\text{i}}(\mathbf{I_m}) - \mathcal{G}_{\text{i}}(\mathbf{I_{GT}})\|_1 \quad (12)$$

$$\mathcal{L}_{\text{deconv}} = \alpha \cdot \mathcal{L}^{\text{MS\_SSIM}} + (1-\alpha) \cdot \mathcal{L}^{\ell_1} \quad (13)$$

Where $\mathcal{F}_i$ is the i_{th}-layer feature map in pre-trained VGG-16 [25] ($i \in \{5, 10, 17\}$, pre-trained on ImageNet [26]) and $\mathcal{G}_i = \mathcal{F}_i \mathcal{F}_i^T$ is the gram matrix [22]. To summarize, the objective of the refine network is:

$$\mathcal{L}_{\text{R}} = \lambda_{\text{hole}}\mathcal{L}^{\text{R}}_{\text{hole}} + \lambda_{\text{valid}}\mathcal{L}^{\text{R}}_{\text{valid}} + \lambda_{\text{tv}}\mathcal{L}^{\text{R}}_{\text{tv}} + \lambda_{\text{per}}\mathcal{L}^{\text{R}}_{\text{per}} + \lambda_{\text{sty}}\mathcal{L}^{\text{R}}_{\text{sty}} + \lambda_{\text{deconv}}\mathcal{L}^{\text{R}}_{\text{deconv}}\mathcal{L}^{\ell_1} \quad (14)$$

Here we use factors of $\lambda_{\text{hole}} = 6$, $\lambda_{\text{valid}}=1$, $\lambda_{\text{tv}} = 0.1$, $\lambda_{\text{per}} = 0.05$, $\lambda_{\text{sty}} = 120$ and $\lambda_{\text{deconv}} = 1$

Name	Input	Channel (in, out)	Feature (in, out)	Activation	Kernel	stride	Padding	BN
L1	cat (Inpainted, Mask)	(4, 64)	(600, 600)	None	7	1	3	Y
L2	F_L1	(64, 128)	(600, 300)	ReLU	3	2	1	Y
L3	F_L2	(128, 256)	(300, 150)	ReLU	3	2	1	Y
L4	F_L3	(256, 512)	(150, 75)	ReLU	3	2	1	Y
RB5	F_L4	(512, 512)	(75, 75)	ReLU	3	1	1	Y
RB6	F_RB5	(512, 512)	(75, 75)	ReLU	3	1	1	Y
LT7	F_RB6	(512, 256)	(75, 150)	ReLU	4	2	1	Y
LT8	F_LT7	(256, 128)	(150, 300)	ReLU	4	2	1	Y
LT9	F_LT8	(128, 64)	(300, 600)	ReLU	4	2	1	Y
LT10	F_LT9	(64, 3)	(600, 600)	Tanh	7	1	3	N

Table 3. Architecture of the refine network. “RB#” stands for the residual layer and “BN” stands for the batch normalization.

3-1-3. Deconvolution stage

We use Wiener Deconvolution after the whole inpainting network. As indicated in [4], we use trainable inversion with the Wiener Deconvolution, parameterizing the entire model to be learnable. PSF of the given camera taken in advance is used for this step, inverted by deconvolution for the proper adaptation to various environments. The reconstructed image in the deconvolution stage is used for deconvolution loss, comparing it to the label image.

3-1-4. Enhancing network

As a denoiser, the architecture remains in the encoder-decoder manner. For the highly complicated reconstruction of lensless imaging, we doubled and tripled the number of layers in each downsampling and upsampling [3]. We compare the output with the label, using a loss that mix of MSE and MS-SSIM, the same as the deconvolution loss [24].

Name	Input	Channel (in, out)	Feature (in, out)	Activation	Kernel	stride	Padding	BN	# layer
L1	Inpainted	(3, 24)	(400, 200)	ReLU	7	1	3	Y	2
L2	F_L1	(24, 64)	(200, 100)	ReLU	3	1	1	Y	2
L3	F_L2	(64, 128)	(100, 50)	ReLU	3	1	1	Y	2
L4	F_L3	(128, 256)	(50, 25)	ReLU	3	1	1	Y	2
L5	F_L4	(256, 512)	(25, 12)	ReLU	3	1	1	Y	2
C6	F_L5	(512, 512)	(12, 12)	ReLU	3	1	1	Y	1
LT7	cat (F_C6, F_L5)	(1024, 256)	(12, 25)	ReLU	3	1	1	Y	3
LT8	cat (F_LT7, FL_4)	(512, 128)	(25, 50)	ReLU	3	1	1	Y	3
LT9	cat (F_LT8, FL_3)	(256, 64)	(50, 100)	ReLU	3	1	1	Y	3
LT10	cat (F_LT9, FL_2)	(128, 24)	(100, 200)	ReLU	3	1	1	Y	3
LT11	cat (F_LT10, FL_1)	(48, 3)	(200, 400)	ReLU	3	1	1	Y	3

Table 4. Architecture of the enhancing network. “BN” stands for batch normalization and “# layer” stands for the number of sets consist of convolutional layer, batch normalization and ReLU activation.

To this end, our 3-staged inpainting-deconvolution-enhancing model is trained in an end-to-end manner, and the final training loss is the sum of all losses from the networks includes discriminator, i. e. , $\mathcal{L}_{\mathrm{C}} + \mathcal{L}_{\mathrm{D}} + \mathcal{L}_{\mathrm{R}} + \mathcal{L}_{\mathrm{E}}$

3-2. Experimental setting

We conduct experiments on a Mirflickr-25000 dataset [27], widely used in the lensless imaging task [3-6]. We simulated the lensless system as a convolution between the label image and the given PSF shown in Figure 7. We used 24,000 images for the train and 1,000 images for the test. Image of size 400×400 as a label, we define the full resolution of the convolved output to be 600×600. To compare performances from full size images to highly cropped images in the model, we randomly restrict the number of pixels from above 40,000 pixels, when the full-size image contains 360,000 pixels. Experimentally almost 10% of the measurement is taken in the full-resolution image as the limit of the reconstruction. All networks are trained using Adam [28] optimizer with $\beta_1 = 0.5$ and $\beta_2 = 0.999$. We trained the model with a batch size of 32 and a learning rate of 0.0001 for 100 epochs.

3-2-1. Boundary crop case

We define a situation in which a sensor would not absorb all the lights from a scene, computationally as a boundary masking. As bigger and closer the object illuminating to the sensor, a vanished area of the measurement that cut off from the original gets larger. We simulated it with a crop situation masking it to binary boundary mask. We obtained the measurements from a full size of 600×600 to a sparse size of 200×200.

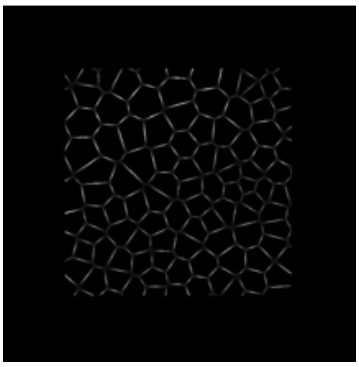

Figure 7. PSF we used in the experiment. The image of size 400×400 captured from the point source, where we use it by padding to the size of 600×600

In this situation, we compare our inpainting method with zero-padding and replicated-padding, conducted in [4] for the separable model as shown in Figure 8.

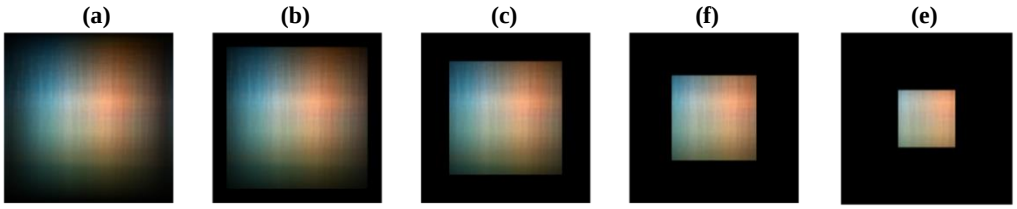

Figure 8. Boundary crop situation examples: (a) 600×600; (b) 500×500; (c) 400×400; (d) 300×300; (e) 200×200.

3-2-2. Random masking case

There can be a problem with the missing pixels in the sensor. If the sensor has physical harm, the measurement can have defective pixels, so-called dead pixels. These faults in the sensor deduce fatal errors in reconstruction because the lensless reconstruction uses all over the pixel's correlation following the deconvolutional process. Our network also can solve this problem. We simulated it with a random pixel-patch mask. With the patch size of 20×20, we selected the number of patches for the masking from 0 to 900. This factor gets the remaining measurement from a full size of 360,000 pixels to a sparse size of 40,000 pixels, as shown in Figure 9.

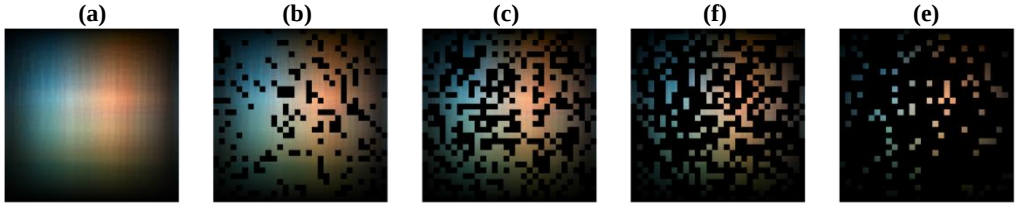

Figure 9. Random masking situation examples: (a) 600×600; (b) 500×500; (c) 400×400; (d) 300×300; (e) 200×200.

3. Result

In random masking, which we define as dead pixels in the sensor, we can reconstruct the masked image with a look-able feature, but it shows worse performance than the boundary crop condition with the same number of pixels because there's weak consistency and less center-specific-concentrated data. Thus, we decided to focus on the boundary crop situation.

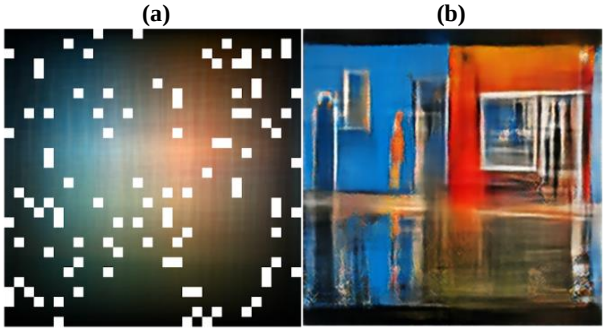

Figure 10. Random masking situation with 40,000 missing pixels: (a) given measurements; (b) our proposed network's output.

4-1. Varying crop size

We evaluate the performance of reconstructed images from the masked image. Using five crop factors in evaluation, each factor crops the measurements into 600×600 (no crop), 500×500, 400×400, 300×300, and 200×200, remaining original resolution as the full-size of 600×600. For the evaluation metrics, we adopt metrics in the reconstruction: PSNR (peak signal-to-noise ratio), LPIPS (learned perceptual image patch similarity) [29], and L1 loss

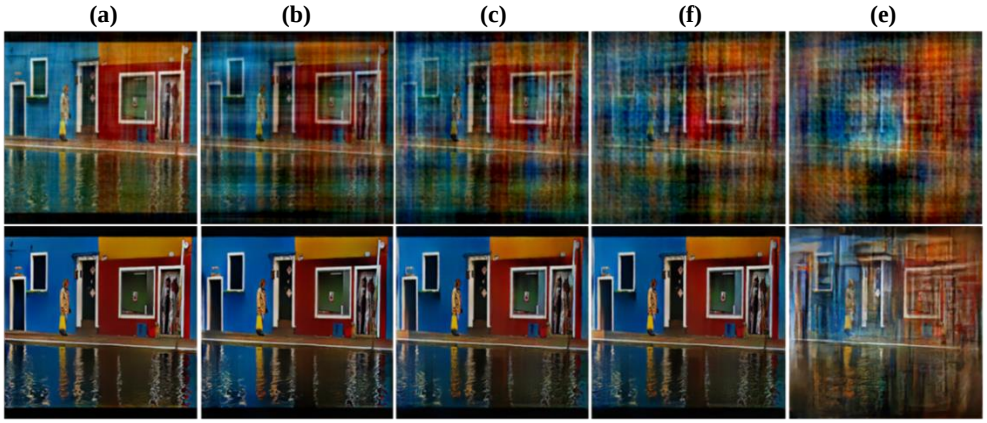

Figure 11. Reconstruction results with varying crop factors. The first row indicates deconvolution results with output from the inpainting networks and the second column indicates results from the enhancing network: (a) measurement of full-size 600×600; (b) 500×500; (c) 400×400; (d) 300×300; (e) 200×200.

As we can see in Figure 11, our proposed model shows apparent quality preservation in the boundary mask situation until it gets largely cropped.

4-2. Ablation study

We validate how successfully each part of our model operates by subtracting them in three ways. Firstly, we can compare “E”, “RP-Deconv-E” and “C-R-Deconv-E”, filling the missing area with zero padding, replicated padding, and inpainting network. Second, we compare “C-R-E” and “C-R-Deconv-E” as a form of the deconvolution stage subtracted or not. Lastly, we compare “C-Deconv-E” and “C-R-Deconv-E” as a form of the refinement network subtracted or not. All the comparisons are in Table 5. We can find that the deconvolution method with the measurement is critical to reconstruction performance. By substituting the inpainting network for replicated padding and zero padding, it shows better performances when it is deconvolved, whereas it gets worse performances without the deconvolution stage because the preceded inpainting network generates unexpected artifacts. This result indicates that the deconvolution stage guides the model to reconstruct desirably with the output of the inpainting network.

Results show that our model improves performances in terms of L1, LPIPS, and PSNR for the given conditions, where a slightly lower score for the full-size reconstruction. This degradation for the networks seems derived from the lack of consistency and data focusing, while the networks learn different sizes of masking at once, in every step.

	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**

Table 5. Comparison in reconstruction performance with varying crop factors. This includes both ablation studies.

4. Conclusion

Our proposed network can predict unknown parts of the measurement cropped by sensor FOV, and as a result, we successfully restore the missing area and reconstruct the original scene. The model can obtain the features of the scene with almost only 10 % of the image. Therefore, we made the model separable, achieving higher quality reconstruction comparing the previous method [3-4] in compressive imaging.

Research also applied to the random masks concerning robust imaging in the case of missing pixel existence. As a result of random masking, for the given unclouded area, feature distance and its consistency seem to affect the reconstruction quality. The larger the patch size and the number of patches, the worse the quality become rapidly. It is worth investigating in various scenarios under the size, shape, and number of patches.

Even though there remain limits in system-specific problems, long-time consumption, and feature collapse, our model can partially solve them with inpainting tasks. We expect that the model can be applied in much compressive lensless imaging, with the advantage of separability. Furthermore, we can improve the model by preserving data consistency following crop factors and binding the networks properly so that each of the three networks can be independent of their errors.

Reference

[1] Kuo, G., Antipa, N., Ng, R., & Waller, L. (2017). DiffuserCam: Diffuser-based lensless cameras. Optics InfoBase Conference Papers, Part F46-C(c), 2016–2018.

[2] ASIF, M. Salman, et al. Flatcam: Thin, lensless cameras using coded aperture and computation. *IEEE Transactions on Computational Imaging*, 2016, 3.3: 384-397.

[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] KHAN, Salman Siddique, et al. Flatnet: Towards photorealistic scene reconstruction from lensless measurements. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[5] MONAKHOVA, Kristina, et al. Learned reconstructions for practical mask-based lensless imaging. *Optics express*, 2019, 27.20: 28075-28090.

[6] REGO, Joshua D.; KULKARNI, Karthik; JAYASURIYA, Suren. Robust lensless image reconstruction via PSF estimation. In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 2021. p. 403-412.

[7] KUO, Grace, et al. On-chip fluorescence microscopy with a random microlens diffuser. *Optics express*, 2020, 28.6: 8384-8399.

[8] ANTIPA, Nick, et al. DiffuserCam: lensless single-exposure 3D imaging. *Optica*, 2018, 5.1: 1-9.

[9] KHIREDDINE, A.; BENMAHAMMED, Khier; PUECH, William. Digital image restoration by Wiener filter in 2D case. *Advances in Engineering Software*, 2007, 38.7: 513-516.

[10] RONNEBERGER, Olaf; FISCHER, Philipp; BROX, Thomas. U-net: Convolutional networks for biomedical image segmentation. In: *International Conference on Medical image computing and computer-assisted intervention*. Springer, Cham, 2015. p. 234-241.

[11] YU, Jiahui, et al. Generative image inpainting with contextual attention. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018. p. 5505-5514.

[12] YEH, Raymond A., et al. Semantic image inpainting with deep generative models. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017. p.

5485-5493.

[13] CAO, Chenjie; DONG, Qiaole; FU, Yanwei. Learning Prior Feature and Attention Enhanced Image Inpainting. In: *European Conference on Computer Vision*. Springer, Cham, 2022. p. 306-322.

[14] QUAN, Weize, et al. Image inpainting with local and global refinement. *IEEE Transactions on Image Processing*, 2022, 31: 2405-2420.

[15] LIU, Hongyu, et al. Coherent semantic attention for image inpainting. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019. p. 4170-4179.

[16] ZHU, Manyu, et al. Image inpainting by end-to-end cascaded refinement with mask awareness. *IEEE Transactions on Image Processing*, 2021, 30: 4855-4866.

[17] LIU, Guilin, et al. Image inpainting for irregular holes using partial convolutions. In: *Proceedings of the European conference on computer vision (ECCV)*. 2018. p. 85-100.

[18] LIU, Hongyu, et al. Rethinking image inpainting via a mutual encoder-decoder with feature equalizations. In: *European Conference on Computer Vision*. Springer, Cham, 2020. p. 725-741.

[19] LI, Jingyuan, et al. Recurrent feature reasoning for image inpainting. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020. p. 7760-7768.

[20] GOODFELLOW, Ian, et al. Generative adversarial networks. *Communications of the ACM*, 2020, 63.11: 139-144.

[21] JOHNSON, Justin; ALAHI, Alexandre; FEI-FEI, Li. Perceptual losses for real-time style transfer and super-resolution. In: *European conference on computer vision*. Springer, Cham, 2016. p. 694-711.

[22] GATYS, Leon A.; ECKER, Alexander S.; BETHGE, Matthias. Image style transfer using convolutional neural networks. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016. p. 2414-2423.

[23] WANG, Zhou; SIMONCELLI, Eero P.; BOVIK, Alan C. Multiscale structural similarity for image quality assessment. In: *The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003*. Ieee, 2003. p. 1398-1402.

[24] ZHAO, Hang, et al. Loss functions for image restoration with neural networks. *IEEE*

Transactions on computational imaging, 2016, 3.1: 47-57.

[25] SIMONYAN, Karen; ZISSERMAN, Andrew. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[26] DENG, Jia, et al. Imagenet: A large-scale hierarchical image database. In: *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009. p. 248-255.

[27] HUISKES, Mark J.; LEW, Michael S. The mir flickr retrieval evaluation. In: *Proceedings of the 1st ACM international conference on Multimedia information retrieval*. 2008. p. 39-43.

[28] KINGMA, Diederik P.; BA, Jimmy. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[29] ZHANG, Richard, et al. The unreasonable effectiveness of deep features as a perceptual metric. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018. p. 586-595.