Lensless Device and Blind-deblurring

1st 曾泓硯
Dept. of Engineering and Electronics
National Tsing Hua University
Taiwan, Republic of China
tony871107@cloud.nthu.edu.tw

2nd 廖宏儒
Dept. of Engineering and Electronics
National Tsing Hua University
Taiwan, Republic of China
hankliao1998@cloud.nthu.edu.tw

***Abstract*—We are interested in how decent images are obtained by lensless camera. In our final project, we utilize a lensless camera, flashlight, and a box to project target images on lensless camera sensor. Since we adopt pinhole imaging as a substitute of designing concentric circles to focus the target, the drawbacks are that the hole may be too big to project the target on the sensor accurately. As a result, we apply both traditional optimization-based algorithm [1] and DNN-based image deblurring algorithm [2] as solutions and compare the results. Eventually, we deploy our model to embedded device, capture images by camera, and inference model to get deblurring results.**

***Index Terms*—lensless, blind-deblurring**

I. Introduction

It is important for lensless devices to focus on the target before applying any computational photography algorithms. Generally, designing concentric circles on the sensor to focus on target is a common solution (Fig.1). However, in our project, we choose to align the pinhole to our charge-coupled device (CCD) sensor, and the target will focus on the sensor preliminary. We tried the flashlight of mobile phones and headlamp, drawing our target images on normal A4 paper and thin paper. As a result, using headlamp and thin paper performs better. We also compare deblur results of traditional optimization-based deblurring algorithms and DNN-based deblurring scale-recurrent network. For the demonstration, we use Nvidia Jetson Nano Developer Kit, an embedded board with GPU, to inference our model.

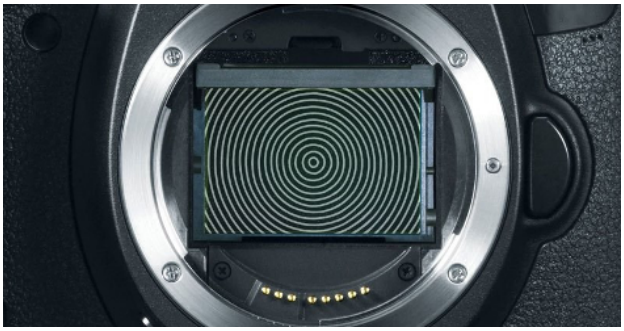

Fig. 1. Lensless Camera with Concentric Circles [link]

II. Methodology

A. System Built Up and Datasets Collection

As Fig 2 shows, we fix our camera sensor and flashlight on the optical plate. Therefore, the distance between these devices and the distance to the pinhole should be fixed. The light sources and the texture of the target is what we can adjust after fixing the system. In this part, we will show the effect of using mobile phone flashlight with A4 paper and headlamp with thin paper. Besides, we also observe that the left side of the picture looks red. Since we use a carbon box as a camera obscura to keep our experiment environment dark.

- Mobile Phone Flashlight with A4 Paper (version 1):
  We use the flashlight of the mobile phone as our light source and draw our images on the A4 paper. As you can see in the Blur Images part of Fig 3, the vertical line is more blur than the horizontal line, and it may degrade our performance. We train our network, which will be introduced later, with version 1 datasets. As the Testing part of Fig 3 shows, the model overfits on the training set and fail on the testing set.
- Headlamp with Thin Paper (version 2):
  From the failure of version 1 dataset, we find out that the vertical line of blur image should be enhanced for improvement. Therefore, we changed our light source and the texture of our target images. Eventually, we get better quality in both the vertical line and the contrast of the images as Fig 4 shows.

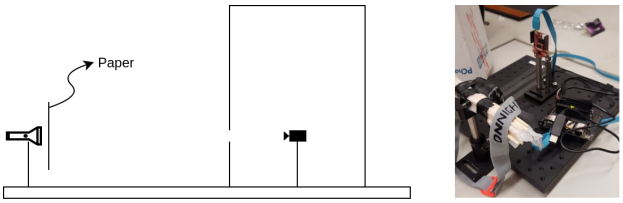

Fig. 2. System Diagram

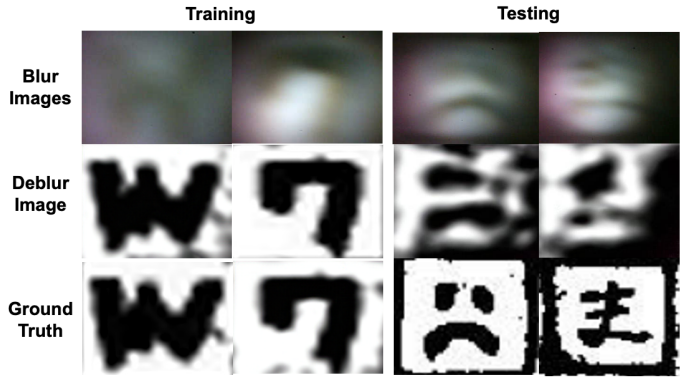

Fig. 3. Version 1 Result

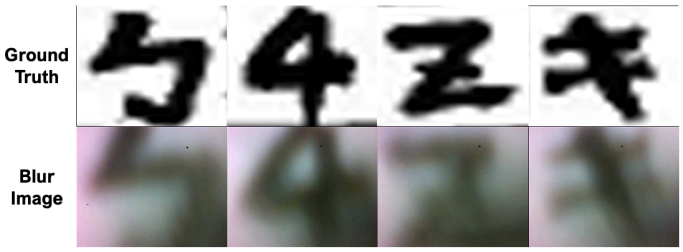

Fig. 4. Ground Truth and Blur Image (version 2)

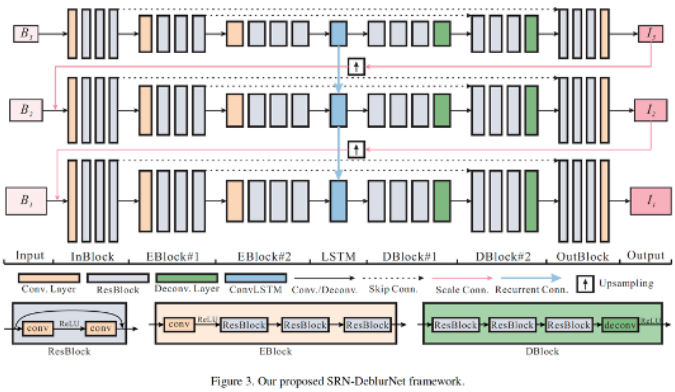

Figure 3. Our proposed SRN-DeblurNet framework.

Fig. 5. Scale-recurrent Network [2]

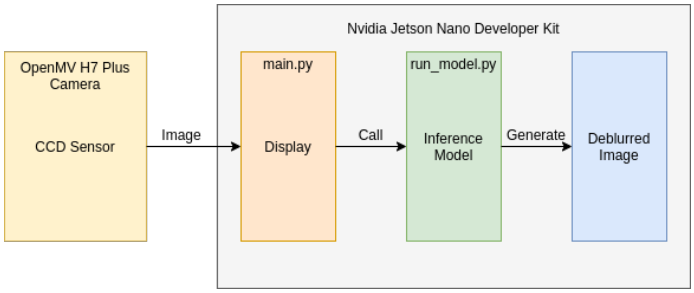

Fig. 6. Deploy System Diagram

B. Traditional Blind-deblurring Algorithm

We use the method provided by Pan, Jinshan, et al. "Blind image deblurring using dark channel prior." [1]

The blurred image can be formula like:

$$B = I \otimes k + n$$

where B, I, k, and n denote the blur image, latent image, blur kernel, and noise, and $\otimes$ is the convolution operator.

For a image I, the dark channel is defined as:

$$D(I)(x) = \min_{y \in N(x)} \left(\min_{c \in r,g,b} I^c(y) \right)$$

where x and y denote pixel locations; $N(x)$ is the neighbor of x; and I^c is the c-th color channel.

They assumed the dark channel of the original image is sparse.

In the end, they develop a method to solve the latent image I by:

$$\min_I ||I \otimes k - B||_2^2 + \mu||\nabla I||_0 + \lambda||D(I)||_0$$

and the blur kernel k by:

$$\min_k ||I \otimes k - B||_2^2 + \gamma||k||_2^2$$

where μ, λ, and γ are weight parameters.

C. Scale-recurrent Network

Auto-encoder has been widely used in many computer vision tasks and has been proven effectively in image segmentation, image-text translation, deblurring tasks and so on. As Fig 5 shows, this network [2] consist of three auto-encoder structure and the image size of each structure are different. We can observe that it consists of many convolution layers, and the receptive field may be large as a result, which benefits the deblurring task particularly on large motion shift. However, the more layer we add, the more trainable parameters need to be tuned, and this might increase the training time and the stability of convergence. Fortunately, they proposed the method of sharing variable between different scales of auto-encoder and decrease the training time to 1/3. (Source code)

D. Deploy Model to the Embedded Device

Nvidia Jetson Nano Developer Kit, an embedded board with GPU, runs on the modified Ubuntu, so we can deploy the model to the embedded board easily. However, the board uses the ARM architecture not the common x86 architecture. There are several packages that the original authors didn't provide the ARM architecture version, so we need to compile some of the needed packages by ourselves. Thanks to the Nvidia, it provides some common packages in the ARM architecture, such as TENSORFLOW and CUDA-TOOKIT, which lessen our effort a lot.

As we show in Fig. 6:

First, we wrote a script in the OpenMV H7 Plus Camera to make it send the image using STRUCT, a special format to pack binary data with string, with universal asynchronous receiver-transmitter (UART).

Second, we wrote another script in the Nvidia Jetson Nano Developer Kit to get the image from the OpenMV H7 Plus Camera and display the image on the screen. As you pressed 'd' (which means 'deblur'), the program will inference the model and output the deblurred image.

III. Experimental Results

Fig. 7. Training Result (DNN-based)

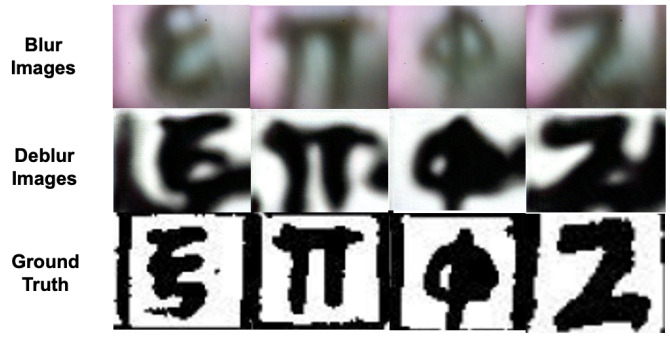

Fig. 8. Testing Result (DNN-based)

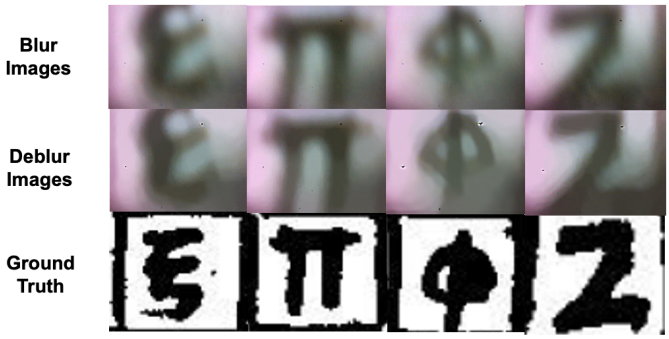

Fig. 9. Testing Result (Traditional)

TABLE I
PSNR & SSIM

Item	PSNR (dB)	SSIM
Traditional	27.85	0.435
DNN-based	29.89	0.618

IV. Conclusion

In general, we get a preliminary result and prove that using pinhole imaging as a solution to lensless camera can work. However, even some of our cases succeed to deblur, we still have a lot to improve on.

First, our Scale-recurrent Network may generate some patterns that doesn't exist. Besides, some of our testing cases fail to deblur images, and even perform worse than traditional algorithm.

Second, in our final project, we use a carbon box as a camera obscura, which fails to keep the experiment environment dark, and need some algorithms such as white balance to eliminate its drawback.

Finally, we have to find a more stable way to get the ground truth. In our project, we get the ground truth by scanning the images drawing on the paper. Obviously, this method isn't ideal, compared with taking a photo directly through the lenses. Therefore, we have to calculate the field of view of the camera and place our target at proper distance.

V. Contribution

曾泓硯：Build the system, collect the dataset, and train the model.
廖宏儒：Build the system, collect the dataset, and deploy the model to the embedded device.

References

[1] Pan, Jinshan, et al. "Blind image deblurring using dark channel prior." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016.
[2] Xin Tao, Hongyun Gao, Xiaoyong Shen, Jue Wang, Jiaya Jia. "Scale-recurrent Network for Deep Image Deblurring" CVPR 2018
[3] Ayan Sinha, Justin Lee, Shuai Li, George Barbastathis. "Lensless computational imaging through deep learning" Cs.CV 2017