Defocus Deblurring Using Dual-Pixel Data Supplemental Materials

Abdullah Abuolaim[1] and Michael S. Brown[1,2]

[1] York University, Toronto, Canada
[2] Samsung AI Center, Toronto, Canada
{abuolaim,mbrown}@eecs.yorku.ca

The supplemental materials provide an ablation study of different variations of our DPDNet in Sec. S1. Sec. S2 provides a brief discussion about defocus blur and motion blur. Use cases are described in Sec. S3. Sec. S4 provides results on dual-pixel (DP) data obtained from a smartphone camera. Sec. S5 provides additional quantitative and qualitative results. Furthermore, as mentioned in Sec. 3 of the main paper, we provide videos of animated examples that show the difference between the DP views in the "animated_dp_examples" directory—located at the same directory as this pdf file.

S1 Ablation study

In this section, we provide an ablation study of different variations in training our DPDNet with: (1) an extra input image (Sec. S1.1), (2) less E-Blocks and D-Blocks (Sec. S1.2), (3) different input sizes (Sec. S1.3), (4) different ratios of homogeneous region filtering (Sec. S1.4), and (5) different data types (Sec. S1.5). This is related to Sec. 5 and Sec. 6 of the main paper.

S1.1 DPDNet with extra input image

As described in Sec. 5 of the main paper, our DPDNet takes the two dual-pixel L/R views, $\mathrm{I_L}$ and $\mathrm{I_R}$, as inputs to estimate the sharp image $\mathrm{I_S^*}$. In our dataset, in addition to the L/R views, we also provide the corresponding combined image $\mathrm{I_B}$ that would be outputted by the camera. In this section, we examine training our DPDNet with all three images, namely $\mathrm{I_L}$, $\mathrm{I_R}$, and $\mathrm{I_B}$. We refer to this variation as DPDNet($\mathrm{I_L}, \mathrm{I_R}, \mathrm{I_B}$).

Table 1 shows the results of the three-input DPDNet, DPDNet($\mathrm{I_L}, \mathrm{I_R}, \mathrm{I_B}$), vs. the two-input one, DPDNet($\mathrm{I_L}, \mathrm{I_R}$), proposed in the main paper. The results of all metrics are quite similar with a slight difference. Our conclusion is that training and testing the DPDNet with the extra input $\mathrm{I_B}$ provides no noticeable improvement. Such results are expected, since $\mathrm{I_B}$ is a combination of $\mathrm{I_L}$ and $\mathrm{I_R}$.

Method	Indoor				Outdoor				Combined			
	PSNR ↑	SSIM ↑	MAE ↓	LPIPS ↓	PSNR ↑	SSIM ↑	MAE ↓	LPIPS ↓	PSNR ↑	SSIM ↑	MAE ↓	LPIPS ↓
DPDNet($\mathrm{I_L}, \mathrm{I_R}, \mathrm{I_B}$)	27.32	0.842	**0.029**	0.191	**22.94**	0.723	**0.052**	0.257	25.07	0.781	**0.041**	0.225
DPDNet($\mathrm{I_L}, \mathrm{I_R}$)	**27.48**	**0.849**	**0.029**	**0.189**	22.90	**0.726**	**0.052**	**0.255**	**25.13**	**0.786**	**0.041**	**0.223**

Table 1: DPDNet with extra input image. The quantitative results of DPDNet($\mathrm{I_L}, \mathrm{I_R}, \mathrm{I_B}$) vs. DPDNet($\mathrm{I_L}, \mathrm{I_R}, $) using four metrics. The testing on the dataset is divided into three scene categories: indoor, outdoor, and combined. The best results are in bold numbers. The results of DPDNet($\mathrm{I_L}, \mathrm{I_R}, \mathrm{I_B}$) and DPDNet($\mathrm{I_L}, \mathrm{I_R}, $) are quite similar with a slight difference. Note: the testing set consists of 37 indoor and 39 outdoor scenes.

S1.2 DPDNet with less blocks

In this section, we train a "lighter" version of our DPDNet with less E-Blocks and D-Blocks. This is done by reducing E-Block 1 and D-Block 4. We refer to this light version as DPDNet-Light. In Table 2, we provide a comparison of DPDNet-Light and our full DPDNet that is proposed in the main paper.

Table 2 shows that our full DPDNet has a better performance compared to the lighter one. Nevertheless, the sacrifice in performance is not too significant, which implies that the DPDNet-Light could be an option for environments with limited computational resources.

Method	Indoor				Outdoor				Combined			
	PSNR ↑	SSIM ↑	MAE ↓	LPIPS ↓	PSNR ↑	SSIM ↑	MAE ↓	LPIPS ↓	PSNR ↑	SSIM ↑	MAE ↓	LPIPS ↓
DPDNet-Light	27.08	0.824	0.030	0.225	22.81	0.701	0.053	0.309	24.89	0.761	0.042	0.268
DPDNet	**27.48**	**0.849**	**0.029**	**0.189**	**22.90**	**0.726**	**0.052**	**0.255**	**25.13**	**0.786**	**0.041**	**0.223**

Table 2: DPDNet with less blocks. The quantitative results of DPDNet-Light vs. our full DPDNet using four metrics. The testing on the dataset is divided into three scene categories: indoor, outdoor, and combined. The best results are in bold numbers. Our full DPDNet has the best results on all metrics for different categories. Nevertheless, DPDNet-Light can operate with less computational power and produce acceptable deblurring results. Note: the testing set consists of 37 indoor and 39 outdoor scenes.

S1.3 DPDNet with different input sizes

Our DPDNet is a fully convolutional network. This facilitates training with different input patch sizes with no change required in the network architecture. As such, we consider training with two different patch sizes, namely 256×256 pixels and 512×512 pixels referred to as DPDNet_{256} and DPDNet_{512}, respectively.

Table 3 shows that the two different input sizes perform similarly. Particularly, input patch size does not change the performance drastically as long as it is larger than the blur size.

Method	Indoor				Outdoor				Combined			
	PSNR ↑	SSIM ↑	MAE ↓	LPIPS ↓	PSNR ↑	SSIM ↑	MAE ↓	LPIPS ↓	PSNR ↑	SSIM ↑	MAE ↓	LPIPS ↓
DPDNet$_{256}$	27.28	0.847	**0.029**	0.195	22.86	**0.734**	**0.050**	0.257	25.01	**0.789**	**0.040**	0.227
DPDNet$_{512}$	**27.48**	**0.849**	**0.029**	**0.189**	**22.90**	0.726	0.052	**0.255**	**25.13**	0.786	0.041	**0.223**

Table 3: DPDNet with different input sizes. The quantitative results of DPDNet$_{256}$ vs. DPDNet$_{512}$ using four metrics. The testing on the dataset is divided into three scene categories: indoor, outdoor, and combined. The best results are in bold numbers. Both input sizes perform on par, in which the patch size does not change the performance drastically as long as it is larger than the blur size. Note: the testing set consists of 37 indoor and 39 outdoor scenes.

S1.4 DPDNet with different filtering ratios

Homogeneous patches are inherently ambiguous in terms of incurred blur size, and do not provide useful information for network training [5]. As a result, filtering homogeneous patches can be beneficial to the trained network. In this section, different filtering ratios are examined including: 0%, 15%, 30%, and 45%; we refer to them as DPDNet$_{0\%}$, DPDNet$_{15\%}$, DPDNet$_{30\%}$, DPDNet$_{45\%}$, respectively.

In Table 4, we present the results of different filtering ratios. The 30% filtering is a reasonable ratio that has the best quantitative results. Therefore, we filter 30% of the extracted image patches based on the sharpness energy to train our proposed DPDNet as described in Sec. 6 of the main paper.

S1.5 DPDNet with different data types

Our dataset provides high-quality images that are processed to an sRGB encoding with a lossless 16-bit depth per RGB channel. Since we are targeting dual-pixel information which would be obtained directly in the camera's hardware, in a real hardware implementation we would expect to have such high bit-depth images. However, since most standard encodings still rely on 8-bit image, we provide a comparison of training our DPDNet with 8-bit (DPDNet$_{8-bit}$) and 16-bit (DPDNet$_{16-bit}$) input data type.

Based on the numbers in Table 5, DPDNet$_{16-bit}$ has a slightly better performance. In particular, it has a lower LPIPS distance for all categories. As a result, training with 16-bit images is helpful due to the extra information embedded in, and is more representative of the hardware's data.

Method	Indoor				Outdoor				Combined			
	PSNR ↑	SSIM ↑	MAE ↓	LPIPS ↓	PSNR ↑	SSIM ↑	MAE ↓	LPIPS ↓	PSNR ↑	SSIM ↑	MAE ↓	LPIPS ↓
DPDNet$_{0\%}$	27.21	0.838	0.030	0.205	22.86	0.721	**0.051**	0.275	24.98	0.778	**0.041**	0.241
DPDNet$_{15\%}$	27.19	0.840	**0.029**	0.194	**22.94**	0.721	0.052	**0.254**	25.01	0.779	**0.041**	0.225
DPDNet$_{30\%}$	**27.48**	**0.849**	**0.029**	**0.189**	22.90	**0.726**	0.052	0.255	**25.13**	**0.786**	**0.041**	**0.223**
DPDNet$_{45\%}$	27.21	0.839	0.030	0.194	22.90	0.724	**0.051**	0.258	25.00	0.780	**0.041**	0.227

Table 4: DPDNet with different filtering ratios. The quantitative results of DPDNet$_{0\%}$ vs. DPDNet$_{15\%}$ vs. DPDNet$_{30\%}$ vs. DPDNet$_{45\%}$ using four metrics. The testing on the dataset is divided into three scene categories: indoor, outdoor, and combined. The best results are in bold numbers. The 30% filtering is a reasonable ratio that has the best quantitative results and , thus, we pick it as a filtering ratio for our proposed framework. Note: the testing set consists of 37 indoor and 39 outdoor scenes.

Method	Indoor				Outdoor				Combined			
	PSNR ↑	SSIM ↑	MAE ↓	LPIPS ↓	PSNR ↑	SSIM ↑	MAE ↓	LPIPS ↓	PSNR ↑	SSIM ↑	MAE ↓	LPIPS ↓
DPDNet$_{8-bit}$	27.37	0.834	**0.029**	0.196	**23.10**	0.723	**0.052**	0.258	**25.18**	0.777	**0.041**	0.228
DPDNet$_{16-bit}$	**27.48**	**0.849**	**0.029**	**0.189**	22.90	**0.726**	**0.052**	**0.255**	25.13	**0.786**	**0.041**	**0.223**

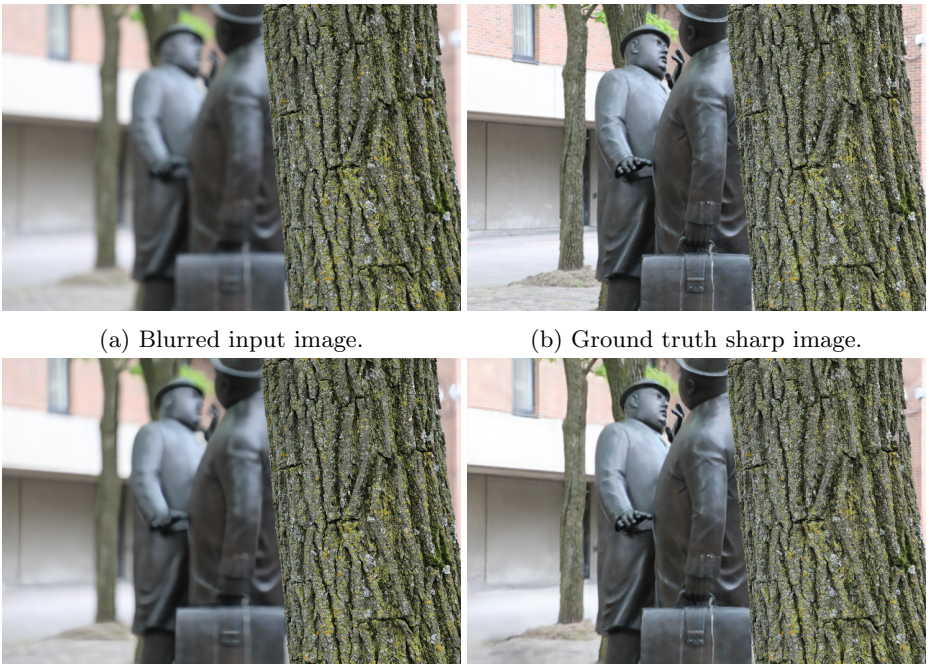
Table 5: DPDNet with different data types. The quantitative results of DPDNet$_{8-bit}$ vs. DPDNet$_{16-bit}$ using four metrics. The testing on the dataset is divided into three scene categories: indoor, outdoor, and combined. The best results are in bold numbers. DPDNet$_{16-bit}$ has a slightly better performance, in which it has a lower LPIPS distance for all categories. Note: the testing set consists of 37 indoor and 39 outdoor scenes.

S2 Defocus and motion blur discussion

One may be curious if motion blur methods can be used to address the defocus blur problem. While defocus and motion blur both produce a blurring of the underlying latent image, the physical image formation process of these two types of blur are different. Therefore, comparing with methods that solve for motion blur is not expected to give good results. However, for a validity check, we tested the scale recurrent motion deblurring method (SRNet) in [9] using our testing set. This method achieved an average LPIPS of 0.452 and PSNR of 20.12, which is lower than all other existing methods that solve for defocus deblurring. Fig. 1 shows results of applying motion deblurring network SRNet [9] to input image from our dataset.

S3 Use cases

As discussed in Sec. 1 of the main paper, we described how defocus blur is related to the size of the aperture used at capture time. The size of the aperture is often dictated by the desired exposure which is a factor of aperture, shutter speed,



(a) Blurred input image. (b) Ground truth sharp image.

(c) SRNet [9] output image. (d) Our DPDNet output image.

Fig. 1: Qualitative deblurring results using SRNet [9] and our DPDNet.

and ISO setting. As a result, there is a trade-off between image noise (from ISO gain), motion blur (shutter speed), and defocus blur (aperture). This trade off is referred to as the exposure triangle. In this section, we show some common cases, where defocus deblurring is required.

Moving camera. Global motion blur is more likely to occur with the moving cameras like hand-held cameras (I_1 in Fig. 2-A). One way to handle motion blur is to set a fast shutter speed and this can be done by either increasing the image gain (i.e., ISO) or the aperture size. However, higher ISO can introduce noise as stated in [6] (Fig. 2-B), and wider aperture can introduce undesired defocus blur as shown in I_3 (Fig. 2-C). For such case, we offer two solutions: apply motion deblurring method SRNet [9] on I_1 (result shown in Fig. 2-D) or apply our defocus deblurring method on I_3 (result shown in Fig. 2-E). Our defocus deblurring method is able to obtain sharper and cleaner image as demonstrated in Fig. 2-E.

Moving object. In this scenario, we have a stationary camera, with a scene object that is moving (i.e., Newton's cradle in Fig. 3). Fig. 3-A shows an image with motion blur, in which the object speed is higher than the shutter speed. In Fig. 3-B, the ISO is significantly increased in order to make the shutter speed faster, nevertheless, the pendulum speed remains the fastest and the motion blur is pronounced. Another way to increase the shutter speed is to open the

Fig. 2: Image noise, motion and defocus blur relation with a moving camera. The number shown on each image is the shutter speed. Zoomed-in cropped patches are also provided. (A) shows an image I_1 suffers from motion blur. (B) shows an image I_2 fixes the motion blur by increasing the ISO, however, I_2 has more noise. (C) shows another image I_3 handles the motion blur by increasing the aperture size, nevertheless, I_3 suffers from defocus blur. (D) shows the results of deblurring I_1 using the motion deblurring method SRNet [9]. The image in (E) is the sharp and clean image obtained using our DPDNet to deblur I_3.

aperture wider as shown in Fig. 3-C and this setting handles the motion blur. However, capturing at wider aperture introduces the undesired defocus blur. To get a sharper image, we can use the motion deblurring method SRNet [9] to deblur I_1 (result shown in Fig. 3-D) and I_2 (result shown in Fig. 3-E), or apply our defocus deblurring method on I_3 (result shown in Fig. 3-F). Our defocus deblurring method is able to obtain sharper image compared to motion deblurring method as demonstrated in Fig. 3-F.

S4 DPDNet performance for a smartphone DP sensor

In this section, we test our DPDNet on images captured with a smartphone. As we mentioned in Sec. 4 of the main paper, there are two camera manufacturers

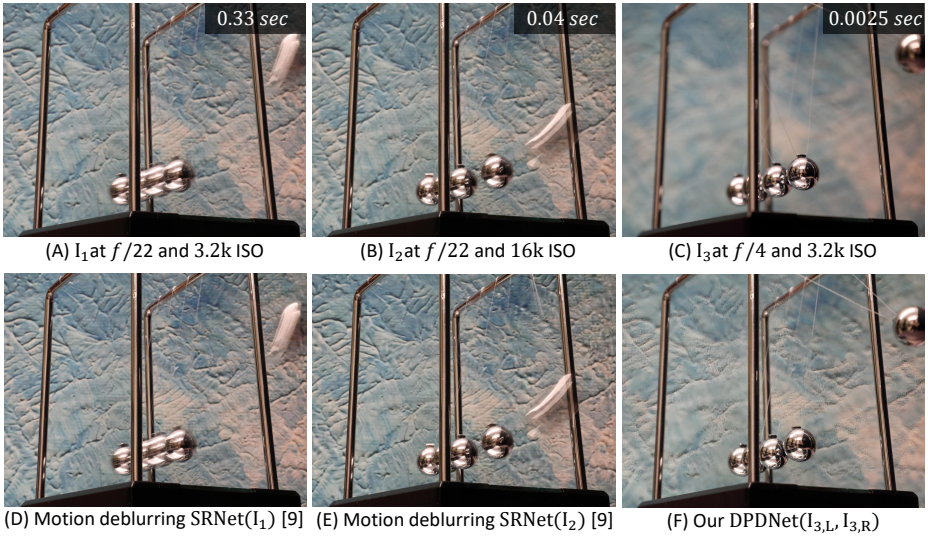


Fig. 3: Motion and defocus blur relation with a moving object. The number shown on each image is the shutter speed. (A) shows an image I_1 has a moving object that suffers from motion blur. Image I_2 in (B) tries to fix the motion blur by increasing the ISO, but the motion blur is still pronounced. I_3 in (C) handles the motion blur by setting the aperture wide, nevertheless, it introduces defocus blur. (D) and (E) show the results of deblurring I_1 and I_2, respectively, using the motion deblurring method SRNet [9]. The image in (F) is sharp and obtained by drblurring I_3 using our DPDNet.

that provide DP data, namely, Google Pixel 3 and 4 smartphones and Canon EOS 5D Mark IV DSLR. The smartphone camera currently has limitations that make it challenging to train the DPDNet with. First, the Google Pixel smartphone cameras do not have adjustable apertures, so we are unable to capture corresponding "sharp" images using a small aperture as we did with the Canon camera. Second, the data currently available from the Pixel smartphones are not full-frame, but are limited to only one of the Green channels in the raw-Bayer frame. Finally, the smartphone has a very small aperture so most images do not exhibit defocus blur. In fact, many smartphone cameras synthetically apply defocus blur to produce the shallow DoF effect.

As a result, the experiments here are provided to serve as a proof of concept that our method should generalize to other DP sensors. To this end, we examined DP images available in the dataset from [1] to find images exhibiting defocus blur. The L/R views of these images are available in the "animated_dp_examples" directory—located at the same directory as this pdf file.

To use our DPDNet, we replicate the single green channel to be 3-channel image to match our DPDNet input. Fig. 4 shows the deblurring results on images captured by Pixel camera. The image on the left is the input combined image

Method	Average LPIPS ↓	
	DP L view	DP R view
EBDB [3]	0.342	0.337
DMENet [4]	0.355	0.353
JNB [8]	0.322	0.313
Our DPDNet	**0.223**	

Table 6: Average LPIPS evaluation of a single DP view separately.

Method	Average LPIPS ↓
EBDB [3]	0.229
DMENet [4]	0.216
JNB [8]	0.207
Our DPDNet	**0.104**

Table 7: Average LPIPS evaluation of the images used to test DPDNet robustness to different aperture settings.

and the image on the right is the deblurred one using our DPDNet. Note that the Pixel android application, used to extract DP data, does not provide the combined image [2]. To obtain it, we average the two views. Fig. 4 visually demonstrates that our DPDNet is able to generalize and deblur for images that are captured by the smartphone camera. Because it is not possible to adjust aperture on the smartphone camera to capture a ground truth image, we cannot report quantitative numbers. The results of two more full images are shown in Fig. 5.

S5 More results

Quantitative results. In Table 6, we provide evaluation of other methods on a single DP view separately using the average LPIPS. Note that a single DP L or R view is formed with a half-disc point spread function in the ideal case. When the two views are combined to form the final output image; the blur kernel would look like a full-disc kernel [7]. Non-blind defocus deblurring methods assume full-disc kernel and the blur kernel of the combined image aligns more with their assumption. More details about DP view formation and modeling DP blur kernels can be found in [7].

In addition to above, we report in Table 7 the average LPIPS numbers for other methods on the images used to test DPDNet robustness to different aperture settings. Note that the LPIPS numbers here are lower than numbers in

Table 1 of the main paper. The reason is that for the robustness test we used f/10 and f/16, which results in less defocus blur compared to the images captured at f/4 (a much wider aperture than f/10 and f/16).

Qualitative results. As we mentioned in Sec. 6 of the main paper, we provide more qualitative results of the full image from the testing set. Specifically, there are 14 examples presented in Fig. 6–Fig. 19. For better visual comparisons, we also provide the corresponding animated video for each example in the "animated_results" directory—located at the same directory as this pdf file.

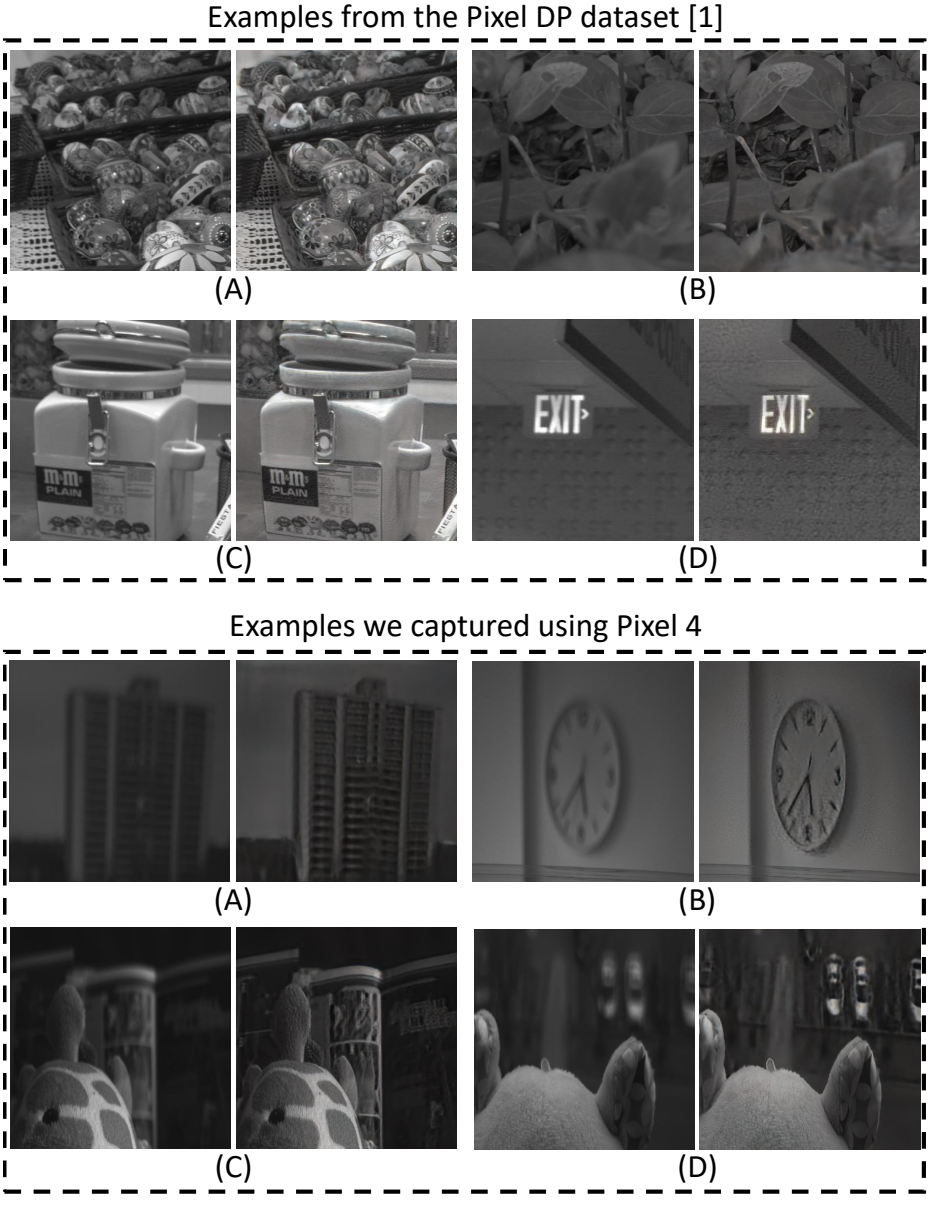


Fig. 4: The results of using our DPDNet to deblur images captured by Pixel smartphone camera. The image on the left is the combined input image with defocus blur and the one on the right is deblurred one. Our DPDNet is able to generalize well for images captured by a smartphone camera.

(a) Blurred input image.

(b) Our DPDNet output image.

(c) Blurred input image.

(d) Our DPDNet output image.

Fig. 5: Qualitative deblurring results using our DPDNet for images captured by a smartphone camera.

(a) Blurred input image.

(b) Predicted sharp image using our DPDNet.

(c) Ground truth sharp image.

Fig. 6: Qualitative defocus deblurring results using our DPDNet.

(a) Blurred input image.

(b) Predicted sharp image using our DPDNet.

(c) Ground truth sharp image.

Fig. 7: Qualitative defocus deblurring results using our DPDNet.

(a) Blurred input image.

(b) Predicted sharp image using our DPDNet.

(c) Ground truth sharp image.

Fig. 8: Qualitative defocus deblurring results using our DPDNet.

(a) Blurred input image.

(b) Predicted sharp image using our DPDNet.

(c) Ground truth sharp image.

Fig. 9: Qualitative defocus deblurring results using our DPDNet.

(a) Blurred input image.

(b) Predicted sharp image using our DPDNet.

(c) Ground truth sharp image.

Fig. 10: Qualitative defocus deblurring results using our DPDNet.

(a) Blurred input image.

(b) Predicted sharp image using our DPDNet.

(c) Ground truth sharp image.

Fig. 11: Qualitative defocus deblurring results using our DPDNet.

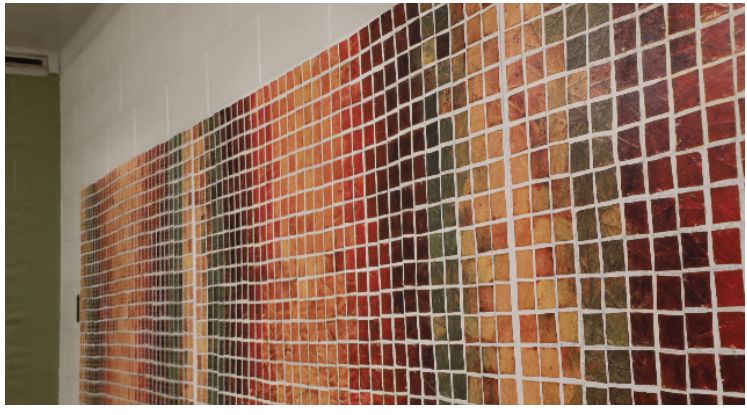
(a) Blurred input image.

(b) Predicted sharp image using our DPDNet.

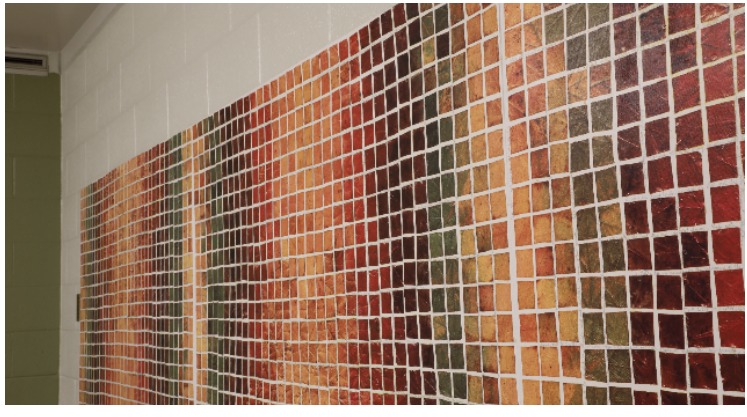
(c) Ground truth sharp image.

Fig. 12: Qualitative defocus deblurring results using our DPDNet.

(a) Blurred input image.

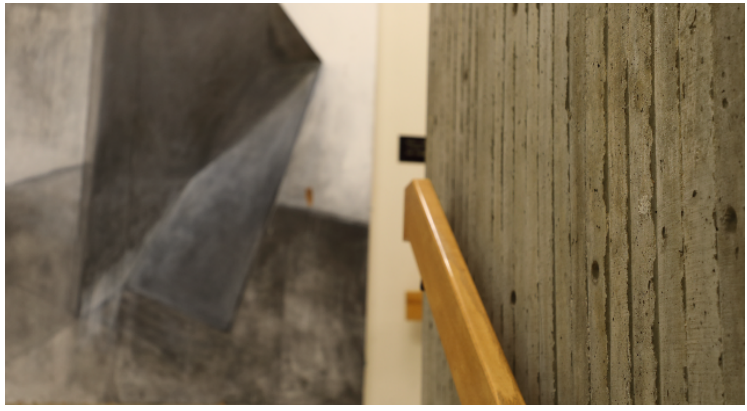


(b) Predicted sharp image using our DPDNet.



(c) Ground truth sharp image.

Fig. 13: Qualitative defocus deblurring results using our DPDNet.



(a) Blurred input image.

(b) Predicted sharp image using our DPDNet.

(c) Ground truth sharp image.

Fig. 14: Qualitative defocus deblurring results using our DPDNet.

(a) Blurred input image.

(b) Predicted sharp image using our DPDNet.

(c) Ground truth sharp image.

Fig. 15: Qualitative defocus deblurring results using our DPDNet.

(a) Blurred input image.

(b) Predicted sharp image using our DPDNet.

(c) Ground truth sharp image.

Fig. 16: Qualitative defocus deblurring results using our DPDNet.

(a) Blurred input image.

(b) Predicted sharp image using our DPDNet.

(c) Ground truth sharp image.

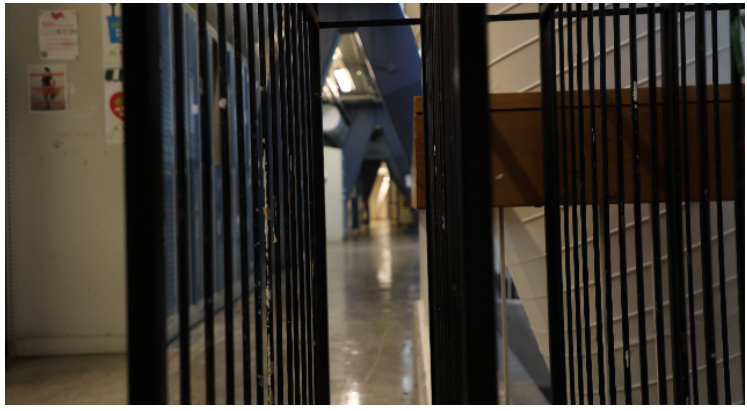
Fig. 17: Qualitative defocus deblurring results using our DPDNet.

(a) Blurred input image.

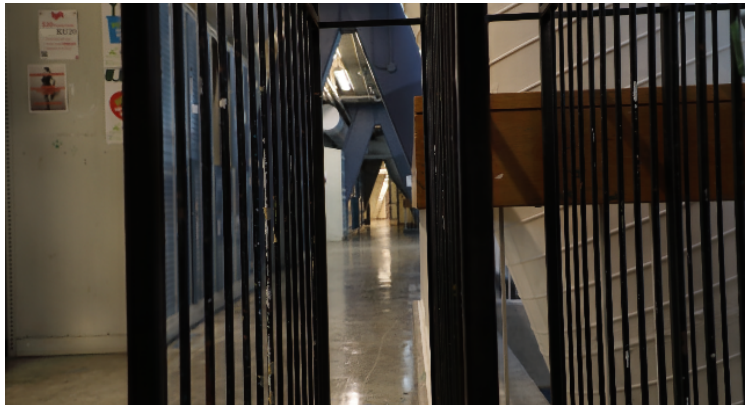
(b) Predicted sharp image using our DPDNet.

(c) Ground truth sharp image.

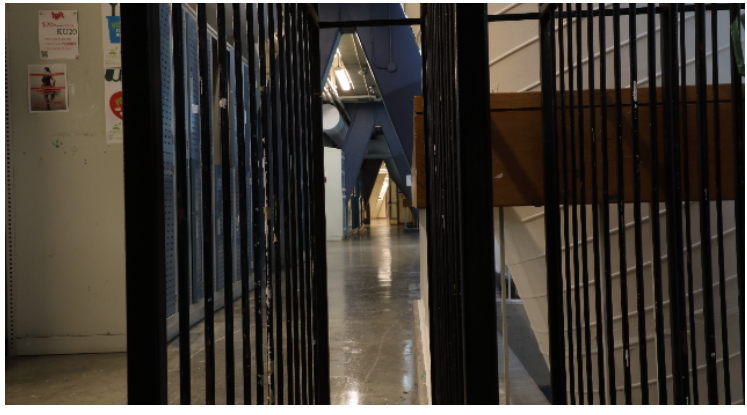
Fig. 18: Qualitative defocus deblurring results using our DPDNet.



(a) Blurred input image.



(b) Predicted sharp image using our DPDNet.



(c) Ground truth sharp image.

Fig. 19: Qualitative defocus deblurring results using our DPDNet.

References

1. Garg, R., Wadhwa, N., Ansari, S., Barron, J.T.: Learning single camera depth estimation using dual-pixels. In: ICCV (2019)
2. Google: Google research: Android app to capture dual-pixel data. https://github.com/google-research/google-research/tree/master/dual_pixels (2019), last accessed: March, 2020
3. Karaali, A., Jung, C.R.: Edge-based defocus blur estimation with adaptive scale selection. TIP **27**(3), 1126–1137 (2017)
4. Lee, J., Lee, S., Cho, S., Lee, S.: Deep defocus map estimation using domain adaptation. In: CVPR (2019)
5. Park, J., Tai, Y.W., Cho, D., So Kweon, I.: A unified approach of multi-scale deep and hand-crafted features for defocus estimation. In: CVPR (2017)
6. Plotz, T., Roth, S.: Benchmarking denoising algorithms with real photographs. In: CVPR (2017)
7. Punnappurath, A., Abuolaim, A., Afifi, M., Brown, M.S.: Modeling defocus-disparity in dual-pixel sensors. In: ICCP (2020)
8. Shi, J., Xu, L., Jia, J.: Just noticeable defocus blur detection and estimation. In: CVPR (2015)
9. Tao, X., Gao, H., Shen, X., Wang, J., Jia, J.: Scale-recurrent network for deep image deblurring. In: CVPR (2018)