Digital Image Processing and Computer Vision Final Project GDIP: A Fast Deep Image Prior Framework with Guided Filtering

Yo-Yu Lai
Institute of Computer and Communication Engineering
National Cheng Kung University
Tainan, Taiwan
q36104195@gs.ncku.edu.tw

***Abstract*—This document is a final report for course Digital Image Processing And Computer Vision given by Institute of Data Science, National Cheng Kung University.**

***Index Terms*—Deep Image Prior, Guided Filtering.**

I. Introduction

Deep Image Prior (DIP) [1] is a famous method to deal with image inverse problem, which need only single data. Guided Filter [3] is an edge-preserving filter, enable to smooth the image without discarding the edge and suitable for upsampling the image. Fast Guided Filter [4] is the improved version of Guided Filter, enable to accelerate guided filter with about the same effect. Fast End-to-End Trainable Guided Filter (DGF) [2] proposed an end to end trainable guided filtering layer, so that the guided filter can be combine with neural network. Although DIP is able to accomplish the image inverse problem with single data, it tooks many iteracitions to run. If the image has big size or has many bands, it may lead to a lot of computing time. In order to deal with the problem to improve DIP performance, we proposed GDIP framwork. GDIP taking advantage of both DIP and DGF to reduce computing time, moreover improve the result of DIP.

II. Related Work

A. Deep Image Prior

Deep Image Prior utilizes the high noise impedance of neural network, leading excellent performance in image inverse problem. Generally, image inverse problem can be expressed as:

$$x^\star = \min_x E(x; x_0) + R(x)$$

where $E(x; x_0)$ is the data fitting term and $R(x)$ is the regularizer. Deep Image Prior replaces the regularizer $R(x)$ with the implicit prior captured by the neural network:

$$\theta^\star = \arg\min_\theta E(f_\theta(z); x_0), \quad x^\star = f_{\theta^\star}(z)$$

where f_θ represented the neural, $\theta^\star$ is the neural parameters, and z represented random noise. The different application of inverse problem need only to rewrite the loss function.

1) Denoise and generic reconstruction:

$$E(x; x_0) = \|x - x_0\|^2$$

2) Inpainting:

$$E(x; x_0) = \|(x - x_0) \odot m\|^2$$

where m repersent the mask.

B. Guided Filtering Layer

Deep Guided Filtering Layer parameters are designed to be differentiable, so that we can train the layer by using gradient descent method, the parameters are automatically updated in the training. Given a high resolution ipnut I_h, we first downsample the image I_h to get low resolution input I_l. We train neural network customized designed with I_l, after training we will get low resolution result O_l. Then we use those three image as the input of Guided Filtering Layer to train the network, finally we will get the high resolution result O_h the whole flowchart is shown in Fig. 1.

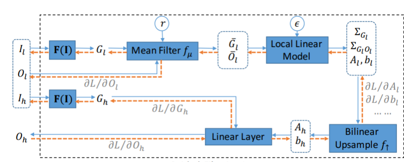


Fig. 1: Guided Filtering Layer flowchart.

III. Methodology

The training setup is same as [1]. In order to accelerate the DIP process, the only difference is that we downsample the image first. Then we use the downsample image to run DIP process, after the process we will get the downsample version of DIP. The second step is deep guided filtering, we input the orignal image, the downsample version of the original image, and the downsample version of DIP image to the guided filtering layer. After training we will get the upsample version of the DIP image, it's spatial resolution is same as the original image. The flowchart is shown in Fig. 2.

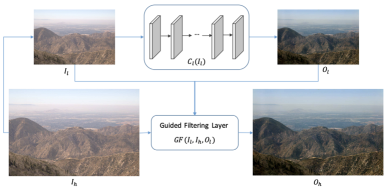


Fig. 2: Guided Filtering Deep Image Prior flowchart.

IV. Experimental setup

In order to compare with DIP, the training and testing image setup is same us [1]. For image denoising case, we use image F16 with noise strength of $\sigma = 25$ shown in Fig. 3. For image restoration case, we use image barbara with random 50% signal droped shown in Fig. 4. For image inpainting case, we use image kate with text masked shown in Fig. 5. The model used is the default skip net proposed in [1]. [5] prosed a hyperspectral version of DIP, in order to compare the result we also use the hyperspectral image (HSI) in the experiments. The HSI used in the experiment is the city area of Ottawa, Canada. We test it in denoising case with Gaussian noise strength of $\sigma = 100$ shown in Fig. 6 and inpainting case with mask of corrupted strips Fig. 7. The model used is same as [5]. The training stage is executed on Python 3.8.11 and Pytorch 1.9.1, equipped with NVIDIA RTX-3090 GPU and with Intel(R) Core(TM) i9-10900K CPU (3.70 GHz and 64 GB RAM).

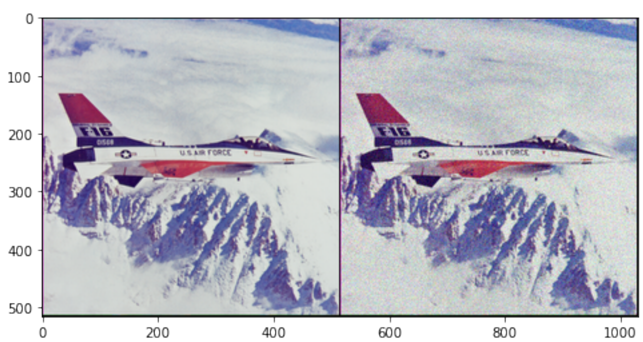


Fig. 3: Ground truth F16 (left side) and noisy F16 with noise strength of $\sigma = 25$ (right side).

Fig. 4: Ground truth barbara (left side), mask with 50% signal drop (middle), the masked barbara (right side).

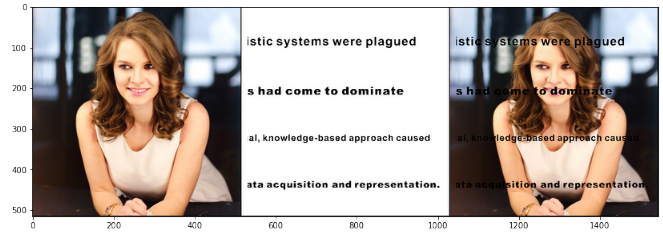


Fig. 5: Ground truth kate (left side), mask with text (middle), the masked kate (right side).

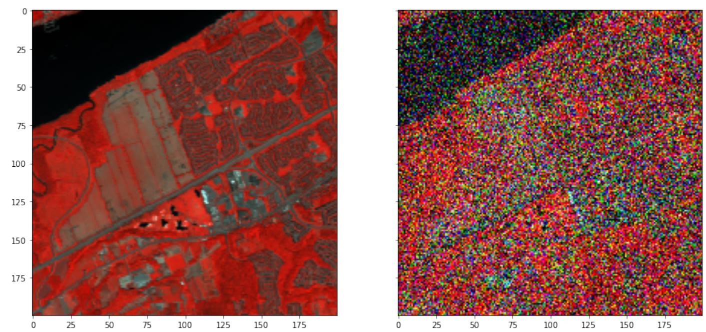


Fig. 6: Ground truth Ottawa HSI (left side) and noisy Ottawa HSI with noise strength of $\sigma = 100$ (right side).

V. Results and discussion

The whole results are shown in Table I ˜ IV and Fig.8 ˜ 12. For the image denoise task, DIP run 3000 iterations to finish the task. In the same iterations GDIP takes less time to run, but the final result is pretty bad. Using the noisy image as the guided image is really hard to achieve a good result because of the edge-preserving property.

For the image restoration task, running iteration is 11000, although GDIP is two times faster than DIP, the result is bad too. The Guided Filtering process blurs the image, leading seriously decline in PSNR.

For the image inpainting task, 6000 iteractions are took to finish the task. The inpainting task is the task that GDIP result closest to the DIP result. To GDIP the trade off of faster speed, the cost is drop of PSNR.

Now we turn to the part of HSI. HSI has much more bands than RGB image, so the image property will be very different. For HSI denoising task, we added Gaussian noise strength

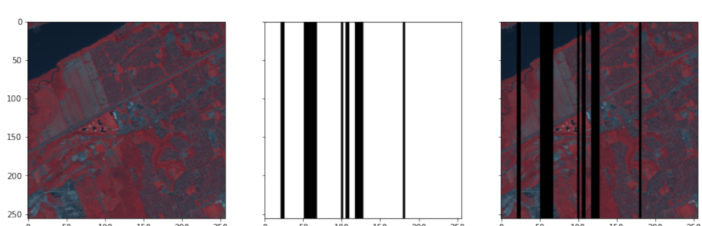
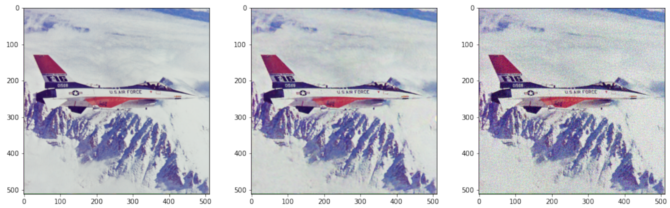
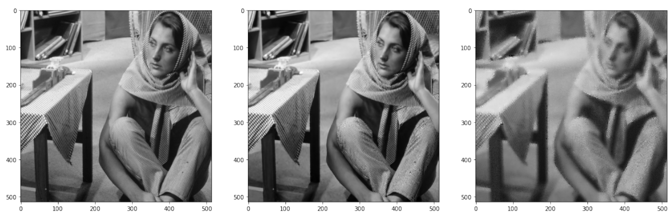


Fig. 7: Ground truth Ottawa HSI (left side), mask of corrupted strips (middle), the masked Ottawa HSI (right side).



(a) GT (b) DIP (c) GDIP

Fig. 8: Ground truth F16, F16 DIP, and F16 GDIP.



(a) GT (b) DIP (c) GDIP

Fig. 9: Ground truth barbara, barbara DIP, and barbara GDIP.

of $\sigma = 100$ to the HSI, we nearly can't see any detail in noisy HSI, but DIP is still able recover the HSI. DIP shows its ability in the inverse problem, truly surprises us. Back to the comparison, they run 1800 iterations this time. GDIP still slightly drops the PSNR compare to DIP. The result of GDIP can see more obvious edge, although there exists more noise.

For HSI inpainting task we use the mask to cover every band of HSI data, we run 10000 iteractions to deal with the problem. This time GDIP win in all quality index. But if we observe the result carefully, we can see that the masked place in GDIP is blurrer than the DIP. DIP tends to produce clean result without noise and high-frequency details, sometimes it will lead to lower PSNR. The result of GDIP is usually blurred, even if the PSNR is higher, the visual quality is worser than DIP.

TABLE I: Quantitative performance of DIP and GDIP denoising using image F16.

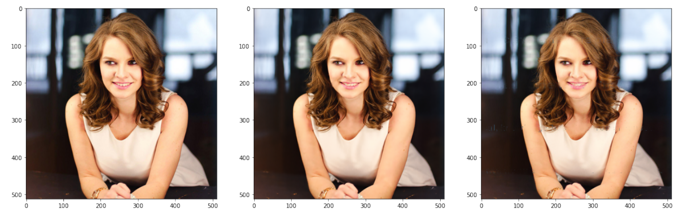
Methods	PSNR (↑)	PSNR smooth (↑)	Time (sec.)
DIP	**30.25**	**32.37**	256.5
GDIP	20.32	20.32	**181.9**

TABLE II: Quantitative performance of DIP and GDIP restoration using the image barbara .

Methods	PSNR masked (↑)	PSNR (↑)	Time (sec.)
DIP	**40.32**	**32.28**	803.8
GDIP	28.46	22.94	**344.5**

VI. Conclusion

GDIP is not good at the task with randomness, like noise or random singal drop. Due to the Guided Filter property, the result will be seriously affect by guided image, leading to the bad result. For the fixed corrupt stripes, GDIP seems to have

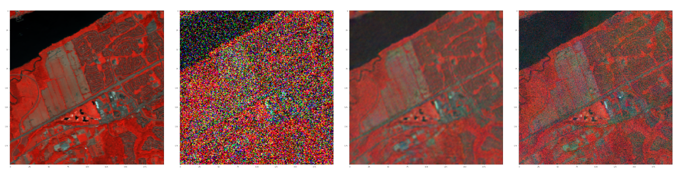


(a) GT (b) DIP (c) GDIP

Fig. 10: Ground truth kate, kate DIP, and kate GDIP.

TABLE III: Quantitative performance of DIP and GDIP inpainting using the image kate .

Methods	PSNR masked (↑)	PSNR (↑)	Time (sec.)
DIP	40.52	**39.22**	544.3
GDIP	**43.70**	37.83	**327.8**

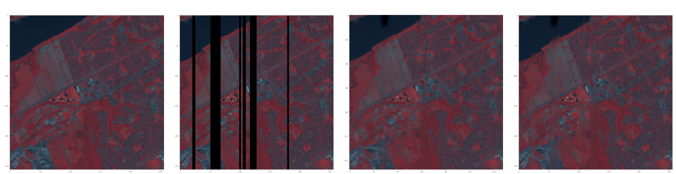


(a) GT (b) noisy (c) DIP (d) GDIP

Fig. 11: Ground truth Ottawa, noisy Ottawa with Gaussian noise strength of $\sigma = 100$, DIP Ottawa , and GDIP Ottawa.

TABLE IV: Quantitative performance of DIP and GDIP denoising using the Ottawa data.

Methods	PSNR (↑)	PSNR smooth (↑)	Time (sec.)
DIP	**20.08**	**20.23**	622.9
GDIP	17.67	17.76	**377.4**



(a) GT (b) crop (c) DIP (d) GDIP

Fig. 12: Ground truth Ottawa, crop Ottawa, DIP Ottawa , and GDIP Ottawa.

TABLE V: Quantitative performance of DIP and GDIP inpainting using the Ottawa data.

Methods	PSNR masked (↑)	PSNR (↑)	Time (sec.)
DIP	45.95	33.31	258.3
GDIP	**48.18**	**35.43**	**206.2**

better result, but still needs to check the other quality index to verify the performance. By observing the experiments reslt, we see DIP gives the better visual quality than GDIP. To practical application, GDIP still has a lot to improve.

REFERENCES

[1] Ulyanov, Dmitry, Andrea Vedaldi, and Victor Lempitsky. ”Deep image prior.” Proceedings of the IEEE conference on computer vision and pattern recognition. 2018.

[2] Wu, Huikai, et al. ”Fast end-to-end trainable guided filter.” Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018.

[3] He, Kaiming, Jian Sun, and Xiaoou Tang. ”Guided image filtering.” IEEE transactions on pattern analysis and machine intelligence 35.6 (2012): 1397-1409.

[4] He, Kaiming, and Jian Sun. ”Fast guided filter.” arXiv preprint arXiv:1505.00996 (2015).

[5] Sidorov, Oleksii, and Jon Yngve Hardeberg. ”Deep hyperspectral prior: Single-image denoising, inpainting, super-resolution.” Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops. 2019.