Reflection removal without training: a single image perspective

Qi Le
University of Minnesota
le000288@umn.edu

Zejun Zhang
University of Minnesota
zhan7565@umn.edu

Wenjie Zhang
University of Minnesota
zhan7867@umn.edu

Louis Wang
University of Minnesota
wangx723@umn.edu

Abstract

1. Introduction

Deep convolutional neural networks (ConvNets) currently are the best method in inverse image reconstruction problems such as denoising or single-image superresolution and ConvNets for image restoration and generation are always trained on large image datasets. But we find out that the inverse image reconstruction based on a single image is a fundamental and long-lasting topic and has made great progress in the recent three years.

Our group is interested in the application of the untrained ConvNets, especially the removal of water wave noise. We choose Deep Image Prior (DIP), which uses a randomly-initialized neural network as a handcrafted prior for standard inverse problems such as denoising, superresolution, and inpainting as our baseline method to implement the application and we focus on improving the defect of the DIP method.

We mainly have three challenges. The first challenge is to obtain the clean generated image from the test image using DIP. We find out that the DIP cannot denoise the water wave noise directly from the image. Thus, we use the image inpainting part of the DIP. The second challenge is improving the quality of the generated images. We add the total variation norm to improve the performance of the basic DIP method. The third challenge is the degradation of the generated image. By applying DIP to our dataset, we find out that the degradation of the generated image would affect the quality of the images a lot. The quality of the generated image will increase at first, but drop at some point. We propose an effective algorithm to stop the Iteration automatically to avoid the degradation of the image.This algorithm may also be helpful in other standard inverse problems.

2. Related Work

Symmetry detection is a long-lasting topic in computer vision and there are plenty of existing works about this topic. Mariola et al.[12] established a simple algorithm for finding all the axes of symmetry of symmetric and almost symmetric planar images having nonuniform gray-level. Keller et al.[8] used algebraic approach and Fourier analysis to realize symmetry detection. Elawady et al.[4] detected symmetry via textual and color histograms. Feature-based method [11, 2, 3] such as edge features, boundary points and key points were popularly implemented. Yang et al.[17] leveraged key points pairs between the direct and reflected views to decrease disparity and finally reconstructed 3D images. Kawahara et al.[7] introduced appearance and shape from water reflection to recover 3D geometry and high-dynamic range appearance of a scene. The modern approaches about water detection are based on machine learning [22]. Besides symmetry detection of a single image, water can be detected through the video or moving cameras [20, 13, 5].

As the water part is split from the frame, we want to remove the impact of water reflection. There are several works have been done to remove certain elements from images, such as water removal [1], image reflection removal [15, 14, 16], moire removal [23], bad weather removal [10]. Water reflection removal also can be regarded as image restoration or inpainting. A series of works about image restoration or inpainting are based on machine learning [9, 21, 6, 18]. CNN or other machine learning models are used as denoisers to remove unnecessary parts. However, all of them require massive dataset and thousands of training sessions.

Deep convolutional neural networks (ConvNets) are used widely in image restoration, image denoising, image super-resolution and have extraordinary performance on large dataset training. Some assume that it is their ability to learn realistic image priors that state-of-the-art ConvNets

can have great performance. However, others [19] provide opposite evidence. Thus, in our research, we consider using the generator networks to capture low-level image prior to any learning.

3. Baseline Method

3.1. Deep Image Prior

This is an image reconstruction [**?**] method based on untrained convolutional neural networks which has U-Net type "hourglass" architecture with skip-connections. No aspect of the network is learned from data previously and the weights of the network are always randomly initialized. The only prior information is in the structure of the network itself. Despite that this method performs well in denoising

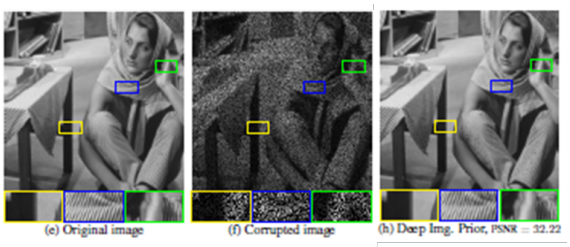

Figure 1. image reconstruction using the Deep Image Prior[**?**].

and inpainting, it has some drawbacks in performance. This method has the overfitting problem which means in some iterations, it will have the peak reconstruction quality. After it touches the peak reconstruction quality, the reconstruction quality will fall straightly to some of its primitive state, which means that the neural network restarts a new loop of image reconstruction.

3.2. TV norm

The Total-Variation norm (TV norm) is a denoising method first proposed by the team of Rudin [**?**]. It has the formula 1. This method is based on the principle that signals with excessive and possibly spurious detail have high total variation.

$$V(y) = \sum_{i,j} \sqrt{|y_{i+1,j} - y_{i,j}|^2 + |y_{i,j+1} - y_{i,j}|^2} \qquad (1)$$

4. Methodology

4.1. Framework

Our method to generate the clean image from the test image using DIP including three steps as shown in Fig.2: water wave noise, image inpainting, and peak stopping.

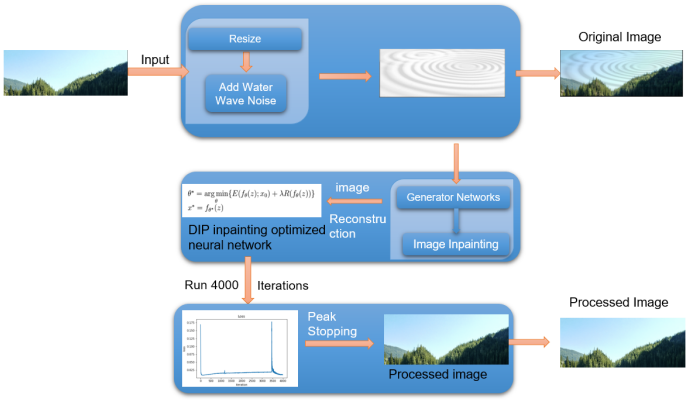

Figure 2. Overview of our method

4.1.1 Water Wave Noise

We choose a random 512*251 pixels image from our dataset as our input and we add a same size water wave noise to it to form the original image.

4.1.2 Image Inpainting

o realize image restoration, instead of searching for the answer in the image space, we now search for it in the space of neural network's parameters. A generator net work [**?**] is implemented to capture a great deal of low level image statistics prior to any learning. And we will not use a pretrained network or an image database but establish randomly-initialized neural networks as a handcrafted prior. The detailed process is as shown below:

1. Initialize the deep convolutional network F with random parameters.
2. Based on GAN, feed each fixed random vector z as the input of the network.
3. The goal of the network is to perform continuous distributed learning with the input z to obtain a picture X, so as to train the parameters of the network to realize the image restoration task.
4. The loss function selected by the model is $E(f_\theta(z); x_0) + \lambda R(f_\theta(z))$, where $\lambda R(f_\theta(z))$ is TV Norm, between the generated image and the real image.
5. The network structure used by the model is the self-encoding and decoding network structure used in the GAN network, the number of whose parameters is about 2 million.

In image inpainting, we are given an image x_0 with missing pixels corresponding to a binary mask $m \in \{0,1\}^{H \times W}$. The goal of image inpainting is to reconstruct the missing data. The corresponding data term is given by $E(x; x_0) = ||(x - x_0) \odot m||^2$, where $\odot$ is Hadamard's product. The

prior is introduced by optimizing the data term with respect to the reparametrization (formula 23):

$$\theta^* = \arg\min_{\theta}\{E(f_\theta(z); x_0) + \lambda R(f_\theta(z))\} \quad (2)$$

$$x^* = f_{\theta^*}(z) \quad (3)$$

subsubsectionPeak Stopping We calculate the difference between two nearby losses and when the difference is larger the threshold, stop training. Our threshold is defined as formula 4:

$$[threshold = 0.2 * abs(max - min)] \quad (4)$$

, where max is maximum of the first ten difference and min is the minimum value of the first ten difference.

4.2. Novelty

4.2.1 TV-Norm

In our method, we add the TV norm to the dip loss function and generated function (2). We use the MSE computed using the original image and the image we generate as part of the loss function. We set the TV norm $R(f_\theta(z))$ to be a regularizer of the loss function. A non-negative number λ is used to balance the weight between the TV norm and MSE in the loss function. Our goal is to achieve the minimum value in the loss function. This means the image generated from the neural network should be close enough to the original image and the value of the neighboring pixels in the newly generated image should be closed to each other.

4.2.2 Peak Stopping

During the training procedure, the result shows that the loss is not gradually steady. According to the Fig. **??**, the value of loss will increase rapidly. Correspondingly, the PSNR value during this period will drop greatly. Since the model leverages MSE loss, it is difficult for the model to converge on pure 0/1 noise and randomly scrambled images. And the model is continuously compared with the original image so the loss will drop again although the training is overfitting. Thus, there is no use to increase the number of iterations and we must find a method to properly stop training. Firstly, we set different parameters to figure out it is a coincidence. No matter how we change the learning rate, the loss still displays such "drop", thus we decide to use this drop to stop training. Firstly, we calculate the difference of every two nearby and set the 0.2 times difference between maximum value and minimum value of first 10 loss as the threshold. If the difference between two loss is larger than the threshold,

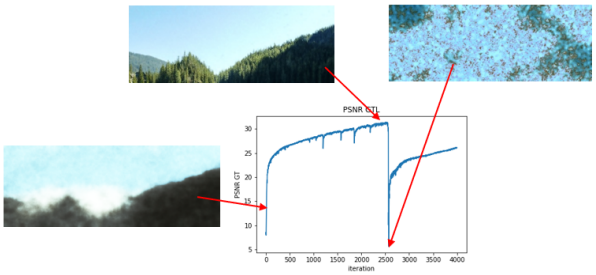

Figure 3. Image PSNR falls after it reaches the local maxima. (Image No.10 in our dataset. PSNR fall happens between iteration 2560 and 2570)

the training will be stopped.As shown in Fig. **??**. Correspondingly, the PSNR of this point will drop rapidly. Finally, we would choose the fixed images just before the intercept as our final output.

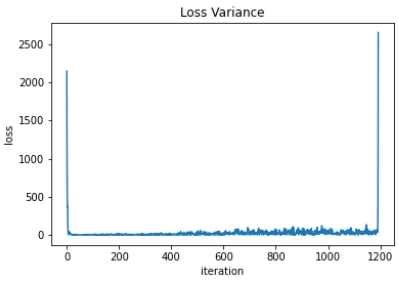

Figure 4. The different of loss before the training stops

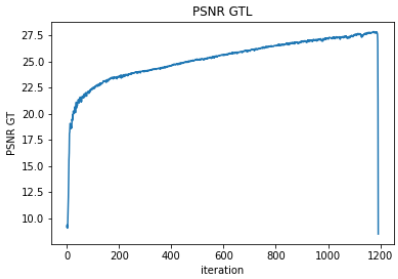

Figure 5. The plot of psnr before the training stops

5. Evaluation

5.1. Dataset

The resource of our dataset is the pexels website, which is a free HD photo site. We resize the images to 512*512 resolution, which is a high resolution because our project would focus on the image inpainting. Image inpainting needs high resolution. We currently have 28 images in the dataset repository because the baseline method is Deep Im

age Prior, which does not need a large number of example images. We segment the images into original objects and the reflection part by using the method that was implemented from detecting symmetry and symmetric constellations of features [11] and put them into the training set and testing set, respectively. The categories of our images dataset are mostly Scene and Man-made Objects.

For this report, in order to fit the adjusted methods and motivations, we randomly pick 5 pictures from our test object and resize them to 512*251 and we add a 512*251 water wave noise. The llink to final dataset refer to Appendix Section put in the final used file.[https://github.com/Qi-Le1/ComputerVisionProject]

5.2. Performance

5.2.1 Water Wave Noise

As for the water noise, we randomly choose a picture about the water wave. Then, superimpose this water wave image to the original image. The result is shown in Fig.6.

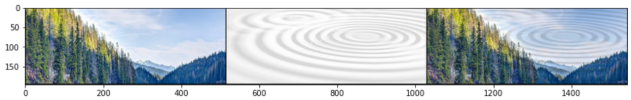

Figure 6. Three images: the orginial image, the water noise, noisy image (from left to right))

5.2.2 TV-Norm

In this subsection, we analyze the performance of DIP after adding TV norm in the loss function based on ripple removal problems. In order to get an overall result of the performance, we tested this model based on 5 different images coming from our dataset. The $lambda$ we chose are 0.01,0.1,0.5,0.9 and 2. We also used the dip model without the TV norm as the comparison set. Our analysis is majorly based on the loss, loss_abs (which is the difference of loss of two consecutive iterations), PSNR and PSNR_abs (which is the difference of PSNR of two consecutive iterations). After running all these models for 4001 Iterations, we have the following result.

The different lines in the plot of result shows that the choice of lambda could largely affect the training step of each image. This means the choice of $lambda$ can affect the weight between the MSE and TV norm and finally influence the outcome.

We also compare our results to the DIP model itself ($lambda = 0$) which is shown as the blue lines. The result shows that some of the $lambda$ values achieve better performance which generates images with higher PSNR value (In figure 7, $lambda = 0.5$ performs better than $lambda = 0$ in most of the time). $lambda = 0.5$ achieves better performance in the pictures of our task cases. Moreover, based on

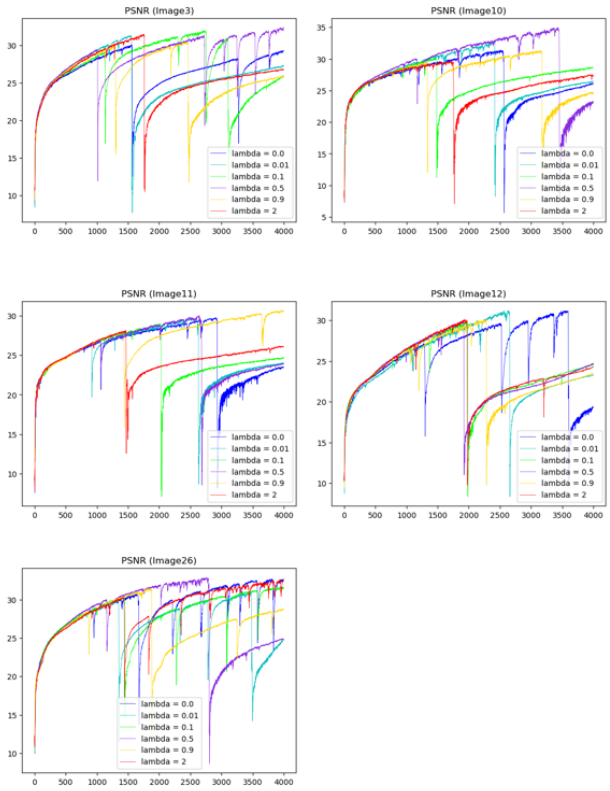

Figure 7. PSNR graph in our training step of image 3, image 10, image 11, image 12, image 17

the "restart" feature of the model, during the training step, in each iteration, there is no significant evidence of which $lambda$ value could lead to better performance.

However, there is still some limitation for our model. Firstly, after adding the TV norm to the model, the overall runtime is dramatically increased. This increase of runtime is majorly due to the TV norm where in each iteration we need to calculate the TV norm for each pixel in the 3 channels of the image we generated. Moreover, our method is still suffering from the sudden drop of the quality during the training.

5.2.3 Peak Stopping

During the training process, the loss would increase or decrease rapidly, causing several 'Peak's. We take advantage of these changes to stop training. Firstly, we calculate the difference of every two nearby loss. We set $c = 0.1, 0.2, 0.4, 0.8$ and found that when c=0.2, the first 'peak' will be recognized. If c is too high and too small, the first 'Peak' would be misrecognized or omitted.

By using threshold to stop the training, we would get the image after and before the 'Peak'. We set the image before the 'Peak' as our output. Finally, we calculate the

PSNR of the original image and output image to evaluate the performance of our model. The result is shown in Fig .??.

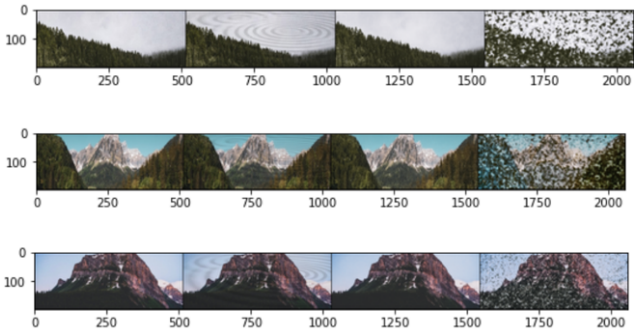

Figure 8. Four images: the orginial image, the noisy image, the output image, and the image after the 'Peak' (from left to right)). The PSNR of three examples is 27.94264,30.15872,31.99262 (from top to bottom).

6. Conclusion

7. Appendix

8. Role

9. Comments

10. Links of paper

- Link to the dataset:Github Link for our Dataset
- Link to the code:Github Link for Experiment Result
- Link to the result:Github Link for Experiment Result

References

[1] Derya Akkaynak and Tali Treibitz. Sea-thru: A method for removing water from underwater images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1682–1691, 2019.

[2] Hugo Cornelius and Gareth Loy. Detecting bilateral symmetry in perspective. In *2006 Conference on Computer Vision and Pattern Recognition Workshop (CVPRW'06)*, pages 191–191. IEEE, 2006.

[3] Hugo Cornelius, Michal Perďoch, Jiří Matas, and Gareth Loy. Efficient symmetry detection using local affine frames. In *Scandinavian Conference on Image Analysis*, pages 152–161. Springer, 2007.

[4] Mohamed Elawady, Christophe Ducottet, Olivier Alata, Cécile Barat, and Philippe Colantoni. Wavelet-based reflection symmetry detection via textural and color histograms: Algorithm and results. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 1734–1738, 2017.

[5] Xiaofeng Han, Chuong Nguyen, Shaodi You, and Jianfeng Lu. Single image water hazard detection using fcn with reflection attention units. pages 105–120, 2018.

[6] Yi Jiang, Jiajie Xu, Baoqing Yang, Jing Xu, and Junwu Zhu. Image inpainting based on generative adversarial networks. *IEEE Access*, 8:22884–22892, 2020.

[7] Ryo Kawahara, Meng-Yu Kuo, Shohei Nobuhara, and Ko Nishino. Appearance and shape from water reflection. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 128–136, 2020.

[8] Yosi Keller and Yoel Shkolnisky. An algebraic approach to symmetry detection. In *ICPR (3)*, pages 186–189, 2004.

[9] Jaakko Lehtinen, Jacob Munkberg, Jon Hasselgren, Samuli Laine, Tero Karras, Miika Aittala, and Timo Aila. Noise2noise: Learning image restoration without clean data. *arXiv preprint arXiv:1803.04189*, 2018.

[10] Ruoteng Li, Robby T Tan, and Loong-Fah Cheong. All in one bad weather removal using architectural search. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3175–3185, 2020.

[11] Gareth Loy and Jan-Olof Eklundh. Detecting symmetry and symmetric constellations of features. In *European Conference on Computer Vision*, pages 508–521. Springer, 2006.

[12] Giovanni Marola. On the detection of the axes of symmetry of symmetric and almost symmetric planar images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(1):104–108, 1989.

[13] Pascal Mettes, Robby T Tan, and Remco C Veltkamp. Water detection through spatio-temporal invariant descriptors. *Computer Vision and Image Understanding*, 154:182–191, 2017.

[14] Renjie Wan, Boxin Shi, Haoliang Li, Ling-Yu Duan, and Alex C Kot. Reflection scene separation from a single image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2398–2406, 2020.

[15] Kaixuan Wei, Jiaolong Yang, Ying Fu, David Wipf, and Hua Huang. Single image reflection removal exploiting misaligned training data and network enhancements. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8178–8187, 2019.

[16] Jie Yang, Dong Gong, Lingqiao Liu, and Qinfeng Shi. Seeing deeply and bidirectionally: A deep learning approach for single image reflection removal. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 654–669, 2018.

[17] Linjie Yang, Jianzhuang Liu, and Xiaoou Tang. Depth from water reflection. *IEEE Transactions on Image Processing*, 24(4):1235–1243, 2015.

[18] Tao Yu, Zongyu Guo, Xin Jin, Shilin Wu, Zhibo Chen, Weiping Li, Zhizheng Zhang, and Sen Liu. Region normalization for image inpainting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 12733–12740, 2020.

[19] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. 2017.

[20] Hua Zhang, Xiaojie Guo, and Xiaochun Cao. Water reflection detection using a flip invariant shape detector. In *2010 20th International Conference on Pattern Recognition*, pages 633–636. IEEE, 2010.

[21] Kai Zhang, Wangmeng Zuo, Shuhang Gu, and Lei Zhang. Learning deep cnn denoiser prior for image restoration. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3929–3938, 2017.

[22] Lili Zhang, Yi Zhang, Zhen Zhang, Jie Shen, and Huibin Wang. Real-time water surface object detection based on improved faster r-cnn. *Sensors*, 19(16):3523, 2019.

[23] Bolun Zheng, Shanxin Yuan, Gregory Slabaugh, and Ales Leonardis. Image demoireing with learnable bandpass filters. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3636–3645, 2020.