Structure First Detail Next: Image Inpainting with Pyramid Generator

Shuyi Qu∗†
School of Information Science and Technology
Northwest University
Xi’an, China

Zhenxing Niu
School of Computer Science and Technology
Xidian University
Xi’an, China

Jianke Zhu
College of Computer Science
Zhejiang University
Hangzhou, China

Kaizhu Huang‡
Data Science Research Center
Duke Kunshan University
Suzhou, China

Abstract—Recent deep generative models have achieved promising performance in image inpainting. However, it is still challenging for a neural network to generate realistic image details and textures due to its inherent spectral bias. We suggest adopting a ‘structure first detail next’ workflow for image inpainting by knowing how artists work. Thus, we propose to build a Pyramid Generator by stacking several sub-generators, where lower-layer sub-generators focus on restoring image structures. In contrast, the higher-layer sub-generators emphasize image details. Our model progressively restores the input through the entire pyramid in a bottom-up fashion. Notably, our approach has a learning scheme of progressively increasing hole size, which allows it to restore large-hole images. In addition, our method could fully exploit the benefits of learning with high-resolution images and hence is suitable for high-resolution image inpainting. Extensive experimental results on benchmark datasets have validated the effectiveness of our approach compared with state-of-the-arts.

Index Terms—Image inpainting, GAN, Multi-scale model

I. INTRODUCTION

Image inpainting aims at restoring corrupted images with appropriate and relevant contents, which has wide real-world applications in image restoration, object removal, and photo editing. In recent years, deep generative models are broadly adopted to solve this problem with impressive results [1]–[3]. However, it is still very challenging to generate coherent and realistic image details [4], [5]. Such challenges probably stem from the neural networks’ spectral bias [6], i.e., neural networks are biased towards learning low-frequency components instead of high-frequency details. Most state-of-the-art methods adopt a coarse-to-fine framework to alleviate this problem. Taking the DeepFillv2 [4] as an example, inputs are first restored with a coarse network, and then the details are further refined at the second stage.

Furthermore, there is a conflict in modelling image global structures and local details [5]. To tackle this problem, another two-stage model StructureFlow [7] adopt the edge-preserved smooth images as supervision in their structure reconstructor. The smoothed image discards the high-frequency details [8] and help the structure reconstructor without disturbance of textures. However, modelling image structures from the smoothed image only is not enough. The global information often prefer to learn from large image regions or even the whole image, while modelling image details prefers to learn from small image patches. This dilemma requires different receptive field sizes, and it is hard to satisfy in a single neural network.

By our understanding, image inpainting looks like artists drawing a picture for a scene. Artists usually first draw the global structures/sketches of the scene and then refine its local details [9]. Inspired by that, we suggest explicitly separating the restoration of image structures from that of details and following the philosophy of ‘structure first detail next’. Thus, we propose to engage distinct sub-generators to restore image structures and details, respectively.

In particular, we build a Pyramid Generator for image inpainting. The pyramid generator is constructed by stacking several sub-generators from the bottom to the top layer. Inputs are downsampled into different resolutions and are fed into corresponding sub-generators. The bottom sub-generators focus on restoring the image global structures, while the top sub-generators focus on restoring image local details. Within the pyramid, we firstly construct the image structure under the supervision of the edge-preserved smooth image. Then, we
refine the reconstruction on the higher layer through the global information learning on the pixel level. At last, higher levels are responsible for learning image details. Our novel inpainting approach takes the ‘structure first detail next’ workflow.

Note that the previous coarse-to-fine frameworks in [4] only cascade two similar sub-generators (i.e., they are trained with inputs of the same resolution) and fail to separate the image structures restoration and image details restoration effectively. On the contrary, with the multi-scale multi-layer stacking framework, our pyramid generator can not only effectively separate the structures and details modelling and indeed practise restoring structures before details.

It is well-known that image inpainting becomes much more challenging when the corrupted area is relatively large [3], [5]. Some recent work [5] illustrate that such large-hole challenges could be effectively alleviated by a progressive learning strategy, i.e., first learn to restore small hole and then learn to restore a large hole. Our approach precisely aligns with this strategy. As shown in Fig. 2, both input image and mask image are simultaneously downsampled in our pyramid. Thus the ratio of hole size to image size is identical for all layers. However, whether the hole size is large or small should be measured concerning the size of the convolutional receptive field (instead of image size). As shown in Fig. 1, since all sub-generators at different layers have the same architecture as well as the same receptive field size, the hole size is relatively small for the low-layer sub-generator. Thus, the low-layer inpainting looks like the small-hole inpainting task. In contrast, the high-layer inpainting looks like large-hole inpainting. With our bottom-up inpainting workflow, we are aligning to the hole-increasing strategy, i.e., first conduct small-hole inpainting and then large-hole inpainting. As a result, our pyramid generator has the advantage of dealing with large holes.

Finally, our pyramid generator is more suitable for high-resolution image inpainting. Many previous works have shown that increasing the resolution of training images could benefit the high-resolution image inpainting [3], [10]. Our experimental results also validate such observation. However, we find that such performance gain is still limited if we just directly train the existing models (e.g., DeepFillv2 [4]) with high-resolution images (e.g., on 512×512 images).

We argue that such observation is also related to the large-hole challenge. Since high-resolution training images often come with large holes, we have to face the large-hole challenge if we directly train a generator with high-resolution images. Nevertheless, due to the capability of handling the large-hole challenge, our approach could fully exploit the advantages of learning with high-resolution images. As a result, our approach is more suitable for high-resolution image inpainting. More detailed discussions and results about the large-hole and high-resolution inpainting are illustrated in our Supplemental material.

II. RELATED WORK

Conventional image inpainting methods often utilize low-level image statistics. PatchMatch [11] fills holes by searching similar patches from unfilled area. Although effective at textured images, these methods often generate artifacts or incoherent content when inpainting complex images with global structures.

Recently, many deep learning-based methods are proposed with great improvement [1], [3], [4], [12], [13]. Context encoder [12] presents the first attempt to apply the convolution neural network on image inpainting. Iizuka et al. [1] improve the architecture with both global and local discriminators to keep consistency. Partial convolution [13] is proposed to handle free-formed masks by using only valid pixels as conducting convolutions. Recently, DeepFillv2 [4] introduces a contextual attention module and a coarse-to-fine learning framework, which has significantly improved the inpainting performance. Another recent work uses a transformer model to learn the distribution of missing regions, and the generation is done through an encoder-decoder architecture [14]. HiFill [3] focuses on high-resolution inpainting tasks and proposes a contextual residual aggregation mechanism.

III. METHOD

In this section, we first introduce the architecture of our pyramid generator. Next, we discuss the loss function and describe how to train our pyramid generator effectively.

A. Pyramid Generator

In order to fully capture the internal statistics of an image at different scales, [15] proposed a hierarchical GAN architecture to learn the distribution within each scale separately. Inspired by that, we adopt its hierarchical framework to separate the modeling of image global structures and local details at different scales.

Fig. 2 shows the architecture of our pyramid generator which are stacked from some sub-generators. We adopt the UNet-based structure in each sub-generators, and we replace each convolution layer with gated convolution as in DeepFillv2 [4]. At different layers, the sub-generator is fed with inputs...
In the pyramid, each layer has a sub-generator $G_n$ and a corresponding discriminator $D_n$. The model is trained in a bottom-up manner: 1) $G_0$ is trained firstly by using $x_0$. Let $\tilde{I}_0$ indicate the recovered image of each layer. Thus, we have the output of bottom layer,
\begin{equation}
\tilde{I}_0 = G_0(x_0).
\end{equation}

2) The output $\tilde{I}_0$ is then upsampled and fed to sub-generator $G_1$, meanwhile $x_1$ is also fed to $G_1$. Both of them are used to train $G_1$. Both $G_1$ and $G_2$ are trained with same losses and discriminator architecture. Notably, we have different fusion strategies among our pyramid, see more details in the Supplemental material. As a result, we have the outputs,
\begin{equation}
\tilde{I}_n = G_n(x_n,Upsample(\tilde{I}_{n-1})), n > 0.
\end{equation}

Note that $\tilde{I}_2$ is regarded as the final output of our pyramid generator.

**B. Learning of Pyramid Generator**

In this section, we describe the loss function and the training details of our pyramid generator. Our model is composed of several sub-generators, where the pyramid is trained by using GAN mechanism. Notably, for the top two layers with image-level supervision, we adopt the similar loss function as [4], which consists of a reconstruction term and an adversarial term,
\begin{equation}
\mathcal{L}_n = \mathcal{L}_{adv}(G_n, D_n) + \alpha \mathcal{L}_{re}(G_n), n > 0,
\end{equation}

and we choose $\alpha = 1$ in our experiments.
The reconstruction term is defined as the $L_1$ distance between the generated output $\hat{I}_n$ and the ground-truth image $I_n$ at the pixel level:

$$L_{re}(G_n) = ||\hat{I}_n - I_n||_1, n > 0.$$  \hfill (4)

For the adversarial term, we use the hinge loss. The loss for generator is

$$L_G = -E_{z \sim p_z, y \sim p_{data}} D(G(z), y),$$

where $z$ denotes the input from the distribution $p_z$, $y$ represents the real data sampled from the distribution $p_{data}$, and for discriminator, it is

$$L_D = E_{x \sim p_{data}(x)} \text{ReLU} (\| - D(x) \|) + E_{z \sim p_z(x)} \text{ReLU} (\| + D(G(z)) \|).$$ \hfill (6)

The architecture of discriminators is identical to each other, as in [15]. Note that those discriminators are independently trained in our approach. While in the bottom structure reconstruction layer, we found that the adversarial term does not contribute to the overall training. Thus, we use Structure $L_1$ loss to penalize the smooth image:

$$L_{re}(G_0) = ||\hat{I}_0 - S||_1,$$ \hfill (7)

where $S$ and $\hat{I}_0$ are input corrupted structure and output inpainted structure, respectively.

Taking the three-layer pyramid generator as an example, its total loss can be represented by

$$L_{PG} = \lambda_0 L_0 + \lambda_1 L_1 + \lambda_2 L_2,$$ \hfill (8)

where $L_0$, $L_1$ and $L_2$ indicate $L_{re}(G_0)$, the loss of $G_1$ and $G_2$, respectively. The $\lambda_0$, $\lambda_1$, $\lambda_2$ are the weights for them. In practice, we select the weight values empirically according to experimental results, i.e., we set $\lambda_{0,1,2} = 1$.

IV. EXPERIMENTS

A. Experiment Settings

We conduct experiments on Places2 [16] and DIV2K [17] to evaluate our approach. The official train+val splits of Places2 are used to train our object-inpainting model, where each image is randomly cropped into $512 \times 512$ resolution.

Although our two inpainting models are both trained on $512 \times 512$ images, they can be used to restore an image in any resolution. Besides evaluating our approach on $512 \times 512$ testing images, high-resolution testing image are also involved into the evaluation. Particularly, we choose DIV2K dataset as our high-resolution validation dataset, which consists of 1000 images from the Internet in 2k resolution. There total data are divided into 800 train, 100 val and 100 test, respectively. These images have diverse contents in nature and are suitable for evaluations. Note that the images of DIV2K are randomly cropped into size $512 \times 512$ and $1024 \times 1024$ for evaluation, since it is memory-consuming to conduct image inpainting on 2k resolution.

We adopt the original mask generation algorithm in [4], which generates center square mask plus random free-form masks for each training sample. There is no normalization applied in our network. All of our experiments are trained with TensorFlow v1.15, Cuda v10.0. The final model needs 4 days to converge on two NVIDIA 2080Ti GPU with batch size of 2. We apply Adam as our optimizer with initial learning rate at 1e $-4$. For the network details, we adopted the similar $3 \times 3$ gated conv in the network. The generators and discriminators exploit a fully convolutional architecture. In the generator, the input features are downsampled twice via convolution and upsamped twice to the input scale via deconvolution. The bottlenecks are stacks of dilated convs with our adaptive dilatation strategy. The discriminators are six successive convs with the final output layer as similarly used in most patch-GAN models.

More dataset results and ablation study can be found in our Supplemental material.

B. Comparison to SOTA Methods

In this section, we compare our model with methods EdgeConnect (EC) [9], DeepFillv2 [4], PEN [10], and HiFill [3]. The original EdgeConnect and PEN are trained with $256 \times 256$ images, while HiFill is trained with $512 \times 512$ images. For more comparisons, we also re-trained DeepFillv2 on $512 \times 512$ images, noted as DeepFillv2†.

For numerical comparisons, we evaluate those methods on $L_1$ loss, structural similarity index measure (SSIM), peak signal-to-noise ratio (PSNR), and learned perceptual image patch similarity (LPIPS) [18].

Comparisons on $512 \times 512$ Images Table I and Table II are the comparison of those methods on the datasets Places2 and DIV2K (with images in $512 \times 512$ resolution). We could find that our approach outperforms most of methods on SSIM, PSNR, and $L_1$. Our approach still leads to competitive performance on LPIPS, but is slightly inferior than the best of the other methods.

Fig. 3 shows the visual comparison for examples of resolution $512 \times 512$ from dataset Places2 and DIV2K. From the results, we could find EdgeConnect is prone to producing blurry contents, especially when inpainting large corrupted area. The original DeepFillv2 could better recover content on large holes. Moreover, PEN is not good at generating realistic high-frequency details. Probably because they are trained on $256 \times 256$ images, they cannot properly handle $512 \times 512$ images, and there is an obvious performance difference on $512 \times 512$ images, and there is an obvious performance improvement. However, the restored content from DeepFillv2† is not coherent to their surrounding pixels, and HiFill tends to generate repeated patches. In contrast, our approach can not only recover image global structures but also produce coherent image details. Probably because they are trained on $256 \times 256$ images, they cannot properly handle $512 \times 512$ inpainting task.

Besides, DeepFillv2†, HiFill and our model are all trained on $512 \times 512$ images, and there is an obvious performance improvement. However, the restored content from DeepFillv2† is not coherent to their surrounding pixels, and HiFill tends to generate repeated patches. In contrast, our approach can not only recover image global structures but also produce coherent image details. Meanwhile, the transformer-based ICT [14] could synthesize reasonable pixels on free-form masks with small artifacts, but it fails on generating sharp details with color consistency.
Table I: Quantitative comparisons on the Places2 test set (with image resolution 512×512). We choose commonly used free-form mask validation settings from [13]. Up-arrow (↑) indicates higher score is better, while lower score is better for down-arrow (↓). The symbol † indicates that the model is retrained by us on 512×512 images instead of 256×256 images.

| Mask ratio | 10-20% | 20-30% | 30-40% |
|------------|--------|--------|--------|
| Metrics    | SSIM↑  | PSNR↑  | L1↑ | LPIPS↓ | SSIM↑  | PSNR↑  | L1↑ | LPIPS↓ | SSIM↑  | PSNR↑  | L1↑ | LPIPS↓ |
| EC [9]     | 0.946  | 26.96  | 0.033 | 0.051 | 0.874  | 25.22  | 0.048 | 0.091 | 0.816  | 22.07  | 0.054 | 0.133 |
| PEN [10]   | 0.916  | 25.70  | 0.017 | 0.101 | 0.839  | 22.47  | 0.033 | 0.176 | 0.759  | 20.70  | 0.049 | 0.245 |
| DeepFillv2 [4] | 0.953  | 28.75  | 0.012 | 0.044 | 0.888  | 23.78  | 0.023 | 0.087 | 0.843  | 22.46  | 0.036 | 0.135 |
| DeepFillv2† [4] | 0.950  | 28.54  | 0.012 | 0.050 | 0.897  | 24.82  | 0.024 | 0.094 | 0.836  | 22.54  | 0.036 | 0.141 |
| HiFill [3]  | 0.904  | 25.04  | 0.021 | 0.098 | 0.826  | 22.09  | 0.038 | 0.166 | 0.748  | 20.39  | 0.054 | 0.229 |
| ICT [14]   | 0.916  | 25.05  | 0.028 | 0.080 | 0.855  | 22.84  | 0.039 | 0.128 | 0.791  | 21.28  | 0.052 | 0.180 |
| ours       | 0.954  | 28.96  | 0.012 | 0.048 | 0.903  | 25.25  | 0.023 | 0.093 | 0.846  | 22.96  | 0.035 | 0.142 |

Comparisons on 1024×1024 images Table III illustrates the comparison of high-resolution image inpainting (with images resolution 1024×1024). On the free-form mask inpainting, our approach outperforms DeepFillv2† and HiFill.

Table II: Quantitative comparisons on the DIV2K test set (with image resolution 512×512). We conduct experiments on square center masks and free-form masks. The free-form mask is generated by algorithms from [2].

| Method      | Mask Type | SSIM↑ | PSNR↑ | L1↑ | LPIPS↓ |
|-------------|-----------|-------|-------|-----|-------|
| EC [9]      | Center    | 0.773 | 20.33 | 0.066 | 0.185 |
| PEN [10]    | 0.754     | 19.36 | 0.056 | 0.228 |
| DeepFillv2 [4] | 0.766 | 19.23 | 0.053 | 0.158 |
| HiFill [3]  | 0.772     | 20.21 | 0.048 | 0.154 |
| ICT [14]    | 0.739     | 19.52 | 0.056 | 0.186 |
| ours        | 0.751     | 20.21 | 0.000 | 0.205 |
| DeepFillv2† [4] | 0.760 | 20.08 | 0.045 | 0.156 |
| HiFill [3]  | 0.715     | 19.44 | 0.074 | 0.256 |
| ours        | 0.740     | 20.07 | 0.061 | 0.203 |

Table III: Quantitative comparisons on 1024×1024 images from DIV2K val set. We apply free-form masks for validation.

| Method | free-form mask | SSIM↑ | PSNR↑ | L1↑ | LPIPS↓ |
|--------|----------------|-------|-------|-----|-------|
| DeepFillv2 [4] | 0.862 | 22.39 | 0.034 | 0.122 |
| HiFill [3] | 0.769 | 19.98 | 0.053 | 0.202 |
| ours   | 0.867 | 22.76 | 0.032 | 0.121 |

Fig. 3. Qualitative comparisons on Places2 and DIV2K (with image resolution 512×512). Best viewed by zooming-in.

Fig. 4. Qualitative comparisons on 1024×1024 example from DIV2K dataset.

The results of these methods on high-resolution image inpainting are shown in Fig. 4. Obviously, it is much more challenging to generate coherent and realistic image details on 1024×1024 images. Both DeepFillv2† and HiFill struggle to generate plausible border of the mountain. At the same time, our approach’s inpainting results look more relevant and realistic.
V. CONCLUSION

This paper proposes a ‘structure first detail next’ workflow for image inpainting. In particular, we introduce a Pyramid Generator by stacking several sub-generators, where image global structures and local details could be better separately modeled at different pyramid layers. Notice that our approach has a progressive learning scheme that allows it to restore images with large masks. In addition, our model is suitable for inpainting high-resolution images. Extensive experiments show that our approach outperforms the other state-of-the-art methods. For future work, we aim at improving our model for 2K/4K image inpainting on two aspects. Firstly, we focus on reducing the current network parameters for the larger resolution inference capability. Secondly, we believe it is reasonable to follow our current multi-scale philosophy on ultra-resolution image inpainting, and our target is building a better model that maximizes the benefit.

ACKNOWLEDGMENT

The work was partially supported by the following: Jiangsu Science and Technology Programme under no. BE2020006-4; Key Program Special Fund in XJTU under No. KSF-T-06; Key Research and Development Program of Shaanxi (No. 2021ZDLGY15-06); the International Science and Technology Cooperation Research Plan in Shaanxi Province of China (No. 2022KW-08; the Program for Changjiang Scholars and Innovative Research Team in University (No. IRT_17R87).

REFERENCES

[1] S. Iizuka, E. Simo-Serra, and H. Ishikawa, “Globally and locally consistent image completion,” ACM Transactions on Graphics (ToG), vol. 36, no. 4, pp. 1–14, 2017.
[2] J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu, and T. S. Huang, “Generative image inpainting with contextual attention,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018, pp. 5505–5514.
[3] Z. Yi, Q. Tang, S. Azizi, D. Jang, and Z. Xu, “Contextual residual aggregation for ultra high-resolution image inpainting,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020.
[4] J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu, and T. S. Huang, “Free-form image inpainting with gated convolution,” in Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 4471–4480.
[5] T. S. K. Kim, K. Aberman, M. Kim, and O. Liba, “Zoom-to-inpaint: Image inpainting with high frequency details,” in arXiv: 2012.09401, 2020.
[6] N. Rahaman, A. Baratin, D. Arpit, F. Draxler, M. Lin, F. A. Hamprecht, Y. Bengio, and A. Courville, “On the spectral bias of neural networks,” in International Conference on Machine Learning, 2020.
[7] Y. Ren, X. Yu, R. Zhang, T. H. Li, S. Liu, and G. Li, “Structureflow: Image inpainting via structure-aware appearance flow,” in Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 181–190.
[8] L. Xu, Q. Yan, Y. Xia, and J. Jia, “Structure extraction from texture via relative total variation,” ACM Transactions on Graphics (ToG), vol. 31, no. 6, pp. 1–10, 2012.
[9] K. Nazeri, E. Ng, T. Joseph, F. Qureshi, and M. Ebrahimi, “Edgeconnect: Structure guided image inpainting using edge prediction,” in Proceedings of the IEEE/CVF International Conference on Computer Vision, Oct 2019.
[10] Y. Zeng, J. Fu, H. Chao, and B. Guo, “Learning pyramid-context encoder network for high-quality image inpainting,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 1486–1494.
[11] C. Barnes, E. Shechtman, A. Finkelstein, and D. B. Goldman, “Patchmatch: A randomized correspondence algorithm for structural image editing,” ACM Transactions on Graphics (ToG), vol. 28, no. 3, p. 24, 2009.
[12] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros, “Context encoders: Feature learning by inpainting,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2016, pp. 2536–2544.
[13] G. Liu, F. A. Reda, K. J. Shih, T.-C. Wang, A. Tao, and B. Catanzaro, “Image inpainting for irregular holes using partial convolutions,” in Proceedings of the European Conference on Computer Vision, 2018, pp. 85–100.
[14] Z. Wan, J. Zhang, D. Chen, and J. Liao, “High-fidelity pluralistic image completion with transformers,” in Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 4692–4701.
[15] T. R. Shaham, T. Dekel, and T. Michaeli, “Singen: Learning a generative model from a single natural image,” in Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 4570–4580.
[16] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, and A. Torralba, “Places: A 10 million image database for scene recognition,” IEEE Transactions on Pattern Analysis and Machine Intelligence, 2017.
[17] E. Agustsson and R. Timofte, “Ntire 2017 challenge on single image super-resolution: Dataset and study,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, 2017, pp. 126–135.
[18] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, “The unreasonable effectiveness of deep features as a perceptual metric,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018, pp. 586–595.