Supplementary Material for GraphFill: Deep Image Inpainting using Graphs

Shashikant Verma[1], Aman Sharma[2], Roopa Sheshadri[2], Shanmuganathan Raman[1]
[1] Indian Institute of Technology Gandhinagar, India
[2] Samsung R&D Institute Bangalore, India
shashikant.verma@iitgn.ac.in, {aman.sharma, roopa}@samsung.com, shanmuga@iitgn.ac.in

1. Loss Functions

In the main paper, we describe the construction of pyramidal graphs $(\mathcal{G}, \mathcal{G}')$ from the training pair $(\mathcal{I}, \mathcal{I}_m)$ using the superpixel map $\mathcal{S}$. The GraphFill algorithm takes the input graph $\mathcal{G}'$ and employs it to estimate the values of unknown nodes, resulting in the output graph $\hat{\mathcal{G}}$. At the i-th pyramid level, we denote the sub-graphs as G_i, $G_i^{'}$, and $\hat{G}_i$, representing the ground-truth, masked, and predicted sub-graphs, respectively. Where sub-graphs G_i and $G_i^{'}$ are obtained by applying the *I2G-layer* on the images $\mathcal{I}$ and $\mathcal{I}_m$ with the corresponding superpixel map S_i. To obtain their coarser representations, we employ the *G2I-layer* to unmap the sub-graphs onto image space, resulting in C_i, C_i', and $\hat{C}_i$. Using the aforementioned notations, we define the following loss functions.

1.1. GraphFill Losses

At each pyramidal level, the GraphFill network is trained using two loss functions: Mean Squared Error (MSE) loss and Perceptual Loss. These losses are applied to the coarser image space representations C_i and $\hat{C}i$ and are defined in Equation 1. Here, the total number of pyramid levels is denoted by p, the MSE loss is represented by $\mathcal{L}_{\text{MSE}}$, and the Perceptual Loss is denoted by $\mathcal{L}_{\text{PL}}$. In our experiments, we set $\lambda_1 = 1$ and $\lambda_2 = 5$.

$$\mathcal{L}_{\text{GF}}(\mathcal{C}, \hat{\mathcal{C}}) = \sum_{i=1}^{p} \Big(\lambda_1 \mathcal{L}_{\text{MSE}}(C_i, \hat{C}_i) + \lambda_2 \mathcal{L}_{\text{PL}}(C_i, \hat{C}_i) \Big) \tag{1}$$

1.2. Refine Network Losses

The predicted coarser representation after iterative filling by GraphFill at the p-th layer of the pyramid, which corresponds to the finest layer, is denoted as $\hat{C}_p$. The Refine Network takes the input $\mathcal{I}_r$, which is obtained by applying the Coarse to Masked Union operation on $\hat{C}_p$ as described in the main paper. It then predicts the final inpainting image $\hat{\mathcal{I}}$. Refine Network is trained in an adversarial setting where generator loss is computed as a combination of Mean squared error loss, Perceptual Loss, and Feature Matching Loss (denoted as $\mathcal{L}_{\text{FM}}$) as described in Equation 2. In our experiments, we set $\lambda_1 = 1$, and $\lambda_2, \lambda_3 = 5$.

$$\begin{aligned} \mathcal{L}_{\text{R}}(\mathcal{I}, \hat{\mathcal{I}}) = \lambda_1 \mathcal{L}_{\text{MSE}}(\mathcal{I}, \hat{\mathcal{I}}) \\ + \lambda_2 \mathcal{L}_{\text{PL}}(\mathcal{I}, \hat{\mathcal{I}}) + \lambda_3 \mathcal{L}_{\text{FM}}(\mathcal{I}, \hat{\mathcal{I}}) \end{aligned} \tag{2}$$

The Refine Network, in combination with GraphFill, is trained using standard Generative Adversarial Network (GAN) loss functions, as outlined in the [1]. In this setup, a discriminator is employed to distinguish between the generated inpainted image $\hat{\mathcal{I}}$ and real images, facilitating the training process. In our experiments, we use discriminator architecture similar to the one described in [3]. Now, we will proceed to provide a detailed description of each component of the loss functions discussed above.

Mean Squared Error (MSE) Loss. MSE Loss provides a natural interpretation of the average squared difference between predictions and ground truth values. Moreover, MSE Loss penalizes larger prediction errors more heavily due to the squaring operation. In our approach, we opt to project the graphs G_i and $\hat{G}_i$ to coarser image space representations C_i and $\hat{C}_i$, respectively, instead of directly employing MSE loss on the node values. This projection is advantageous because a node is comprised of collections of pixels within a superpixel of the map S_i. By projecting the graphs, we ensure that each pixel contributes equally to generating the loss. We employ pixel-wise MSE Loss at i-th pyramid level as illustrated in Equation 3.

$$\mathcal{L}_{\text{MSE}}(C_i, \hat{C}_i) = \frac{1}{N} \sum_{i=1}^{N} \|C_i - \hat{C}_i\|^2 \tag{3}$$

Perceptual Loss. While Mean Squared Error (MSE) focuses on pixel-level differences, Perceptual Loss measures differences between higher-level visual features extracted from a pre-trained deep neural network for perceptually meaningful transformations, capturing aspects such as structural similarity, texture, or overall visual style. We

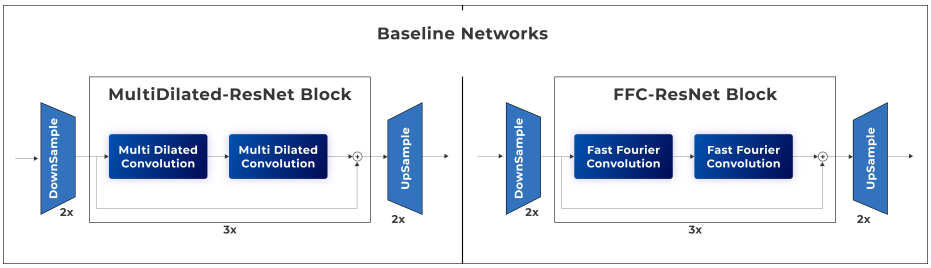

Figure 1. The architecture of the shallow baselines, Pix2Pix [4] and FFC-ResNet [3], is depicted. The deep counterparts of these baselines consist of $9\times$ the respective blocks.

Pyramid Level	Model	MSE $\times 10^{-2}$		Model	MSE $\times 10^{-2}$	
		PCI	MR		PCI	MR
1	GraphFill	1.904	0.772	GraphFill	1.884	0.725
2	(Non-Iterative)	1.488	0.823	(Iterative)	1.475	0.779
3	Depth: 2	1.449	0.916	Depth: 2	1.412	0.854
Mean	#Pars: 8.5K	1.614	0.837	#Pars: 8.5K	1.59	0.786
1	GraphFill	1.844	0.737	GraphFill	1.723	0.723
2	(Non-Iterative)	1.414	0.777	(Iterative)	1.289	0.755
3	Depth: 4	1.359	0.856	Depth: 4	1.244	0.831
Mean	#Pars: 41.3K	1.539	0.79	#Pars: 41.3K	1.419	0.77
1	GraphFill	1.851	0.733	GraphFill	1.703	0.701
2	(Non-Iterative)	1.416	0.783	(Iterative)	1.271	0.725
3	Depth: 6	1.376	0.874	Depth: 6	1.205	0.78
Mean	#Pars: 172K	1.548	0.797	#Pars: 172K	1.393	0.736
1	GraphFill	1.859	0.749	GraphFill	1.892	0.74
2	(Non-Iterative)	1.452	0.789	(Iterative)	1.452	0.765
3	Depth: 8	1.402	0.874	Depth: 8	1.376	0.82
Mean	#Pars: 696K	1.571	0.804	#Pars: 696K	1.573	0.775

Table 1. Ablation study on the depth of the GraphFill architecture, evaluating the mean squared error (MSE) between the predicted coarser image (PCI) and ground truth, as well as the MSE incurred only at the masked region (MR).

define loss at i-th level of the pyramid as demonstrated in equation 4, where $\phi(\cdot)$ denotes the feature extraction function of a pre-trained ResNet Model and $|\cdot|^2$ represents the squared Euclidean distance between the extracted feature representations. We determine total loss as summation on N feature layers of ResNet model $\phi(\cdot)$.

$$\mathcal{L}_{\text{PL}}(C_i, \hat{C}_i) = \frac{1}{N}\sum_{j=1}^{N}\left\|\phi(C_i)_j - \phi(\hat{C}_i)_j\right\|^2 \qquad (4)$$

Feature Matching Loss Along with higher-level features similarity that is captured by Perceptual loss, at the refiner stage we also employ feature-matching loss. Feature matching loss aims to minimize the differences between feature statistics of the generated and target images at intermediate layers of discriminator network $\mathcal{D}$. Let the final inpainted image from Refiner Network be represented as $\hat{\mathcal{I}}$ and the corresponding ground truth be $\mathcal{I}$, we calculate feature matching loss as shown in Equation 5. Here, the sum

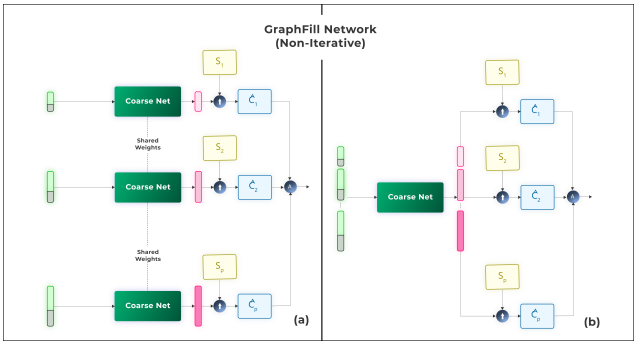

Figure 2. Graph-Fill Architecture for Image inpainting using Non-iterative Graph-filling. (Best viewed in Zoom)

is taken over N discriminator layers.

$$\mathcal{L}_{\text{FM}}(I, \hat{I}) = \frac{1}{N}\sum_{j=1}^{N}\left\|\mathcal{D}(\phi(I)_j) - \mathcal{D}(\phi(\hat{I})_j)\right\|^2 \qquad (5)$$

2. Baseline Network

We apply the GraphFill algorithm to two shallow baseline models: Pix2Pix [4] and FFC-ResNet [3]. The architecture of our shallow baselines, as depicted in Figure 1, consists of three Multi-Dilated ResNet Blocks for the Pix2Pix variant and FFC-ResNet Blocks for the FFC variant.

3. Non-Iterative GraphFill

In the case of Iterative Graph Filling, a coarser estimation is needed from the subsequent lower pyramid layer to iteratively refine the output. However, in the Non-iterative graph-filling approach, all sub-graphs can be processed in a single step by GraphFill using the adjacency matrix $\mathcal{A}$, as explained in the main paper. This approach offers faster image inpainting compared to the Iterative Graph-Filling approach, although it may result in slightly degraded performance, as demonstrated quantitatively in Table 5 of the main manuscript. At the final output stage of all pyramidal levels, the outputs are averaged and passed to the Refine

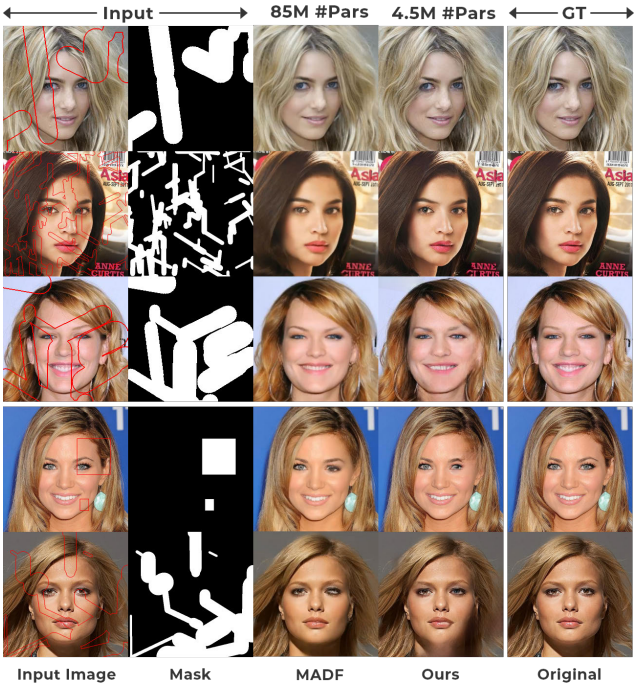

Figure 3. Qualitative comparison of inpainting on CelebAHQ [2] by our approach and MADF [6].

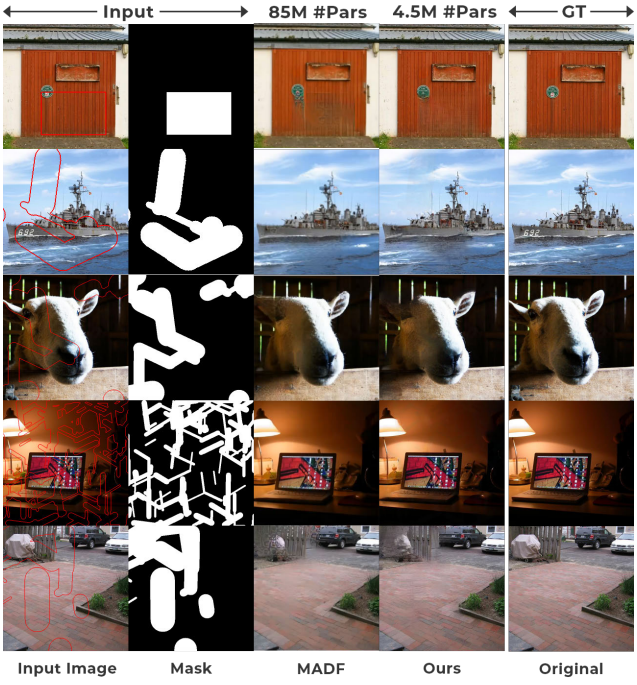

Figure 4. Qualitative comparison of inpainting on Places365 [5] by our approach and MADF [6].

Network after the masked update using the Coarse to Union operation.

4. Qualitative Results

We provide supplementary qualitative results in Figure 5, displaying the inpainted outputs obtained from both the Iterative and Non-Iterative Graph-Filling approaches. Additionally, we showcase the inpainting results achieved using the proposed Resolution-Robust Pyramidal Graph Filling approach in Figure 6. The images used for evaluation have a resolution of 512×512. Figure 7 and Figure 8 showcase qualitative results of different variants, including shallow baselines, visually representing their performance. We include qualitative results with MADF (85M learnable parameters) [6] in Figure 3 and Figure 4.

5. Ablation Studies

Table 1 provides a comprehensive analysis of the effectiveness of iterative graph-filling compared to non-iterative graph-filling, along with the results of the depth ablation study on GraphFill architecture. We compute the Mean Squared Error (MSE) across up to three pyramid levels, comparing the predicted image $\hat{I}$ to the ground truth image I at each level. As pyramid levels increase, the Mean Squared Error between the predicted and ground truth images (PCI) diminishes, illustrating the efficacy of GraphFill's coarser-to-finer strategy. Nonetheless, we note a rise in MSE within the masked region (MR) at higher pyramid levels due to the greater number of superpixels (Graph nodes). Furthermore, our findings highlight an enhanced MSE improvement in Iterative Graph Filling compared to the Non-Iterative approach. Iterative GraphFill yields a more substantial decrease in PCI and a lesser increase in MSE within the masked region, in contrast to the Non-Iterative GraphFill technique.

References

[1] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020.

[2] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017.

[3] Roman Suvorov, Elizaveta Logacheva, Anton Mashikhin, Anastasia Remizova, Arsenii Ashukha, Aleksei Silvestrov, Naejin Kong, Harshith Goka, Kiwoong Park, and Victor Lempitsky. Resolution-robust large mask inpainting with fourier convolutions. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 2149–2159, 2022.

[4] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

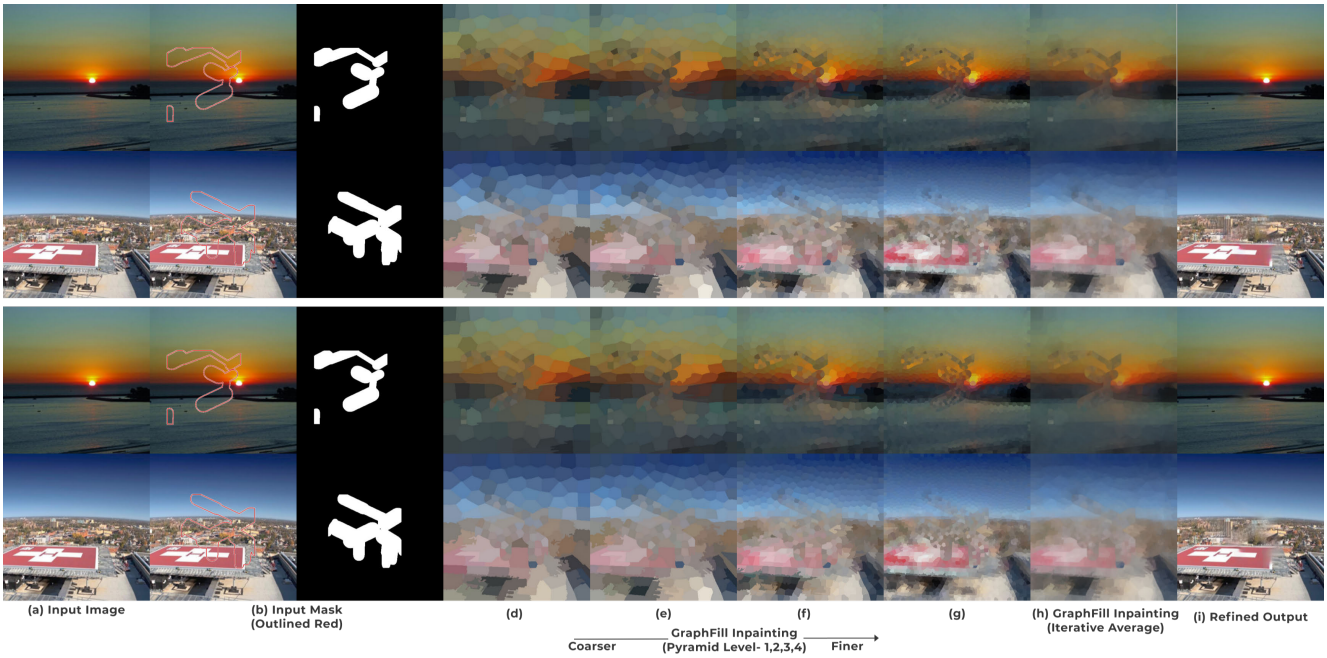

Figure 5. Qualitative result with Iterative Graph-Filling at each pyramidal layer (top two rows) and Non-Iterative Graph-Filling (bottom two rows).

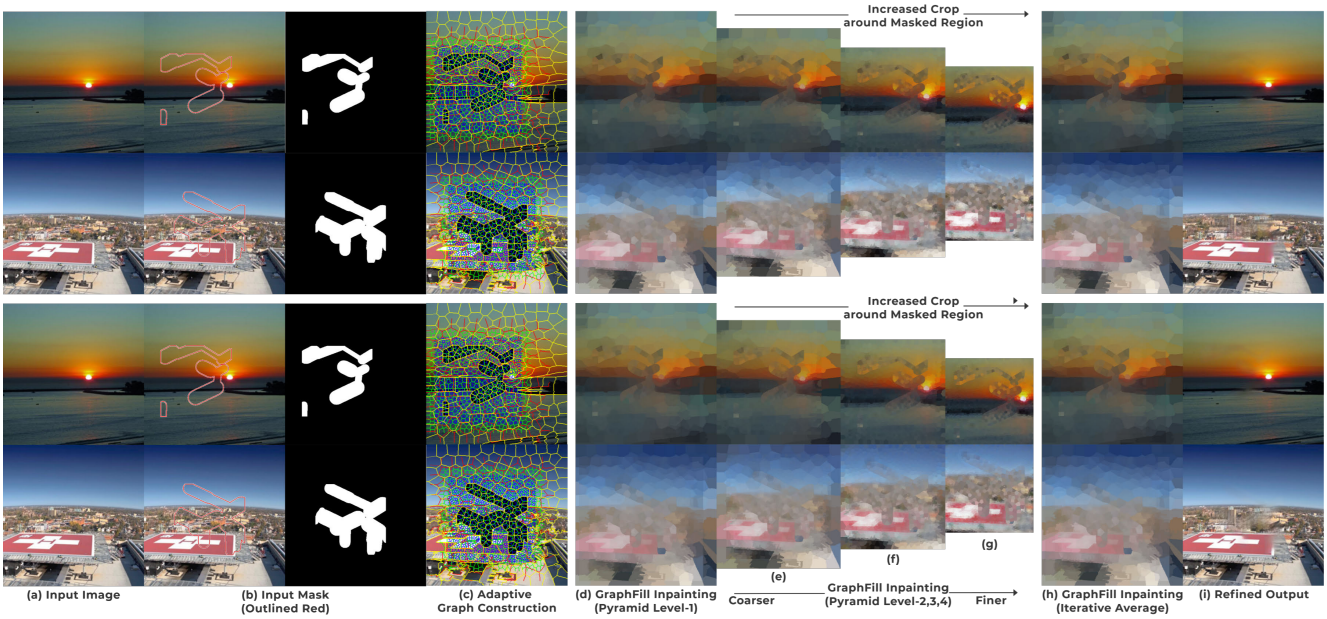

Figure 6. Qualitative result using Resolution Robust Image Inpainting approach with Iterative Graph-Filling at each pyramidal layer (top two rows) and Non-Iterative Graph-Filling (bottom two rows).

[5] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

[6] Manyu Zhu, Dongliang He, Xin Li, Chao Li, Fu Li, Xiao Liu, Errui Ding, and Zhaoxiang Zhang. Image inpainting by end-to-end cascaded refinement with mask awareness. *IEEE Transactions on Image Processing*, 30:4855–4866, 2021.

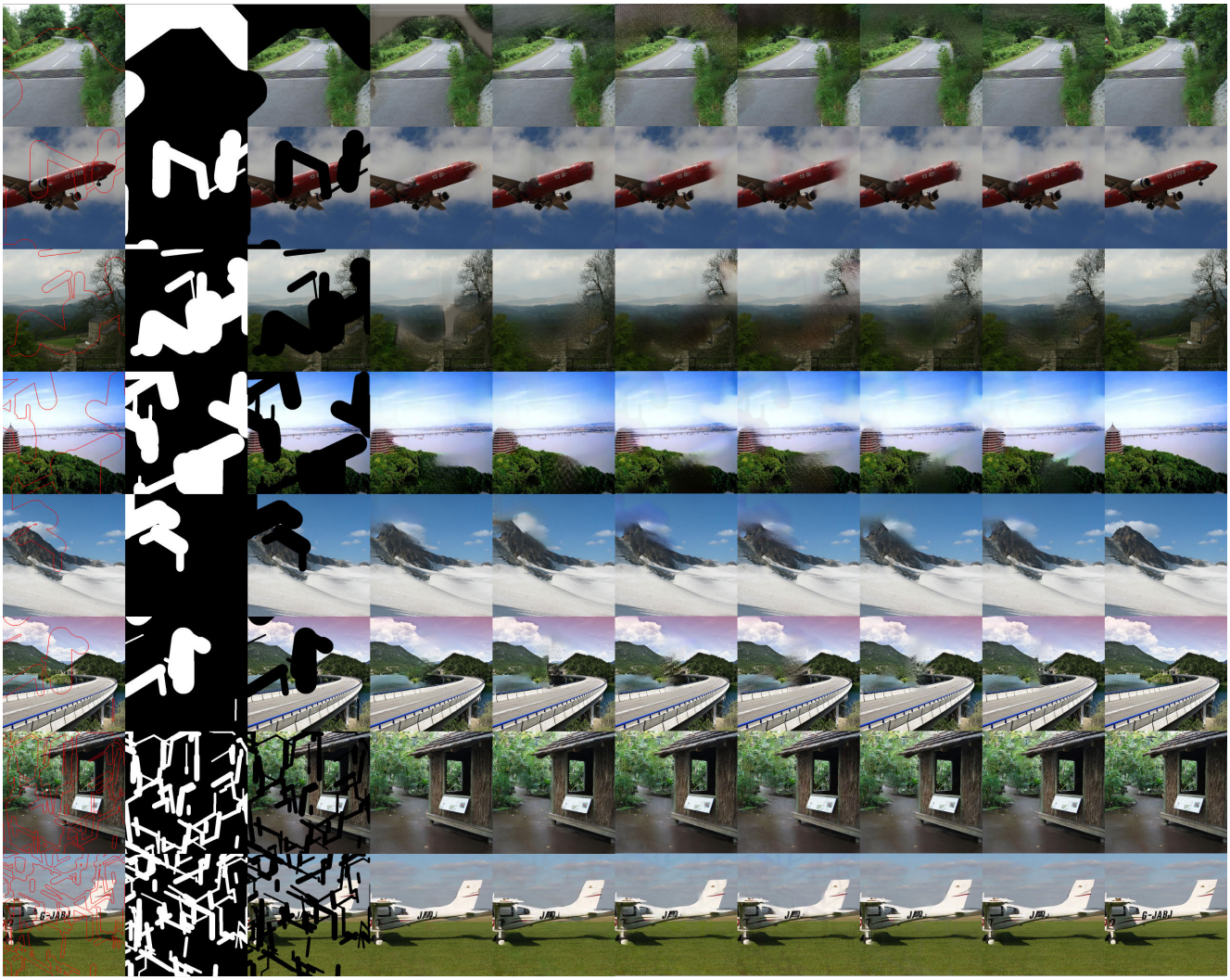

Figure 7. Qualitative comparison of various variants on Places365 Dataset[5] as proposed in Table 5 of the main paper.

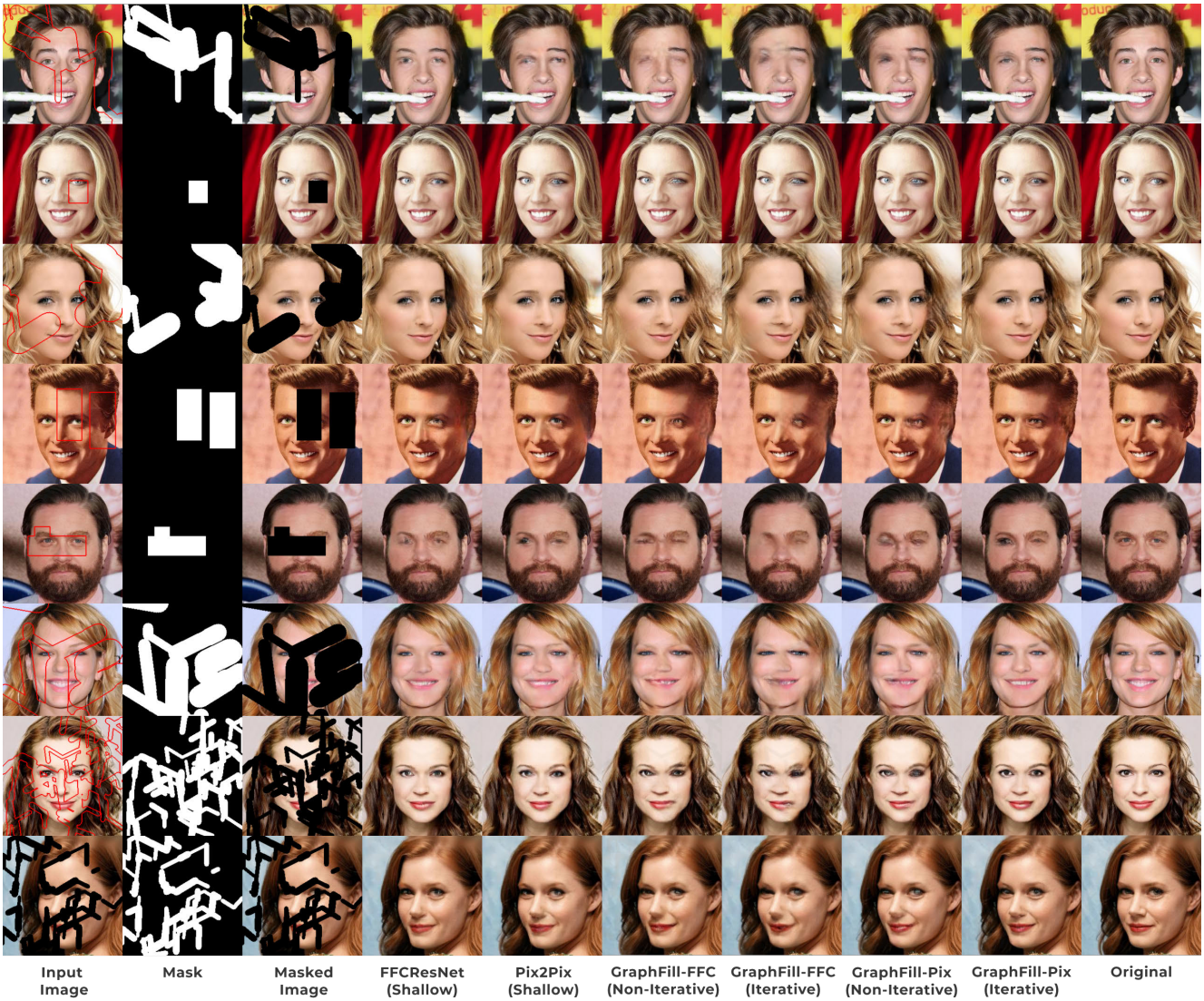

Figure 8. Qualitative comparison of various variants on CelebA-HQ Dataset[2] as proposed in Table 5 of the main paper.