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

Abstract

We present a novel coarser-to-finer approach for deep graphical image inpainting that utilizes GraphFill, a graph neural network-based deep learning framework, and a lightweight generative baseline network. We construct a pyramidal graph for the input-masked image by reducing it into superpixels, each representing a node in the graph. The proposed pyramidal approach facilitates the transfer of global context from coarser to finer pyramid levels, enabling GraphFill to estimate plausible information for unknown node values in the graph. The estimated information is used to fill in the masked region, which a Refine Network then refines. Furthermore, we propose a resolution-robust pyramidal graph construction method, allowing for efficient inpainting of high-resolution images with relatively fewer computations. Our proposed GAN-based network is trained in adversarial settings on Places365 and CelebA-HQ datasets and demonstrates competitive performance compared to existing methods while using fewer learning parameters. We conduct thorough ablation studies to evaluate the effectiveness of each component in the GraphFill Network for improved performance. Our proposed lightweight model for image inpainting is efficient in real-world scenarios, as it can be easily deployed on mobile devices with limited resources.

1. Introduction

Image inpainting entails generating realistic content to fill in missing areas within an image. These missing regions may have been deliberately masked to remove unwanted objects from the image. During the early stages of research, various classical approaches were proposed to address the problem of image inpainting. In [2][3][4], the authors have presented patch-based and exemplar-based region-filling with suitable textures synthesized from the surrounding pixel information. Advancements in parallel computational capabilities have significantly increased the development of deep learning-based solutions for various computer vision problems, including image inpainting. Neural architectures of deep learning frameworks used for image inpainting can be broadly categorized into Generative Adversarial Networks (GAN) [7], Autoregressive Modeling [14], and Denoising Diffusion Probabilistic Models (DDPM) [10]. The image inpainting problem is ill-posed and lacks a unique solution, which motivates one to explore multiple solutions.

Due to the spatially shared convolutional filters, simple convolution-based deep generative models for image inpainting have inherent limitations. These filters treat all input pixels or features as equally valid, making the models unsuitable for accurately filling in the missing image information. Partial convolutions, as proposed in [17], address the limitation of simple convolution-based deep generative models for image inpainting by using masked and normalized convolutions that are conditioned only on valid pixels, followed by a rule-based mask updation step. Building on this approach, [46] proposed gated convolutions using a dynamic feature gating mechanism for each channel and spatial location. The work presented in [46] integrates contextual attention [45]. Large masked regions can still challenge these approaches, resulting in poor inpainting results. To alleviate this challenge, it is essential to have a large, effective receptive field to comprehend the global context of the image for generating high-quality inpainting of the missing regions. In contrast, [32] proposed the usage of Fast Fourier Convolutions to increase the receptive field and improve the aggregation of the global context in the image.

We propose GraphFill, an image inpainting method that employs a Graph Neural Network (GNN) on a graphical representation of the masked image to learn coarser inpainting of the unknown region, which is then refined using a Refine Network. Our approach robustly captures global information in the image by learning coarser inpainting on a pyramidal graphical representation of the input image. Additionally, our graphical approach significantly reduces computational overhead for high-resolution image inpainting. Moreover, our model is very lightweight and has substantially fewer learnable parameters than the current state-of-the-art methods, making it ideal for mobile device deployment. While many studies [46][45][21] have explored the coarser to finer approach, our method is the first to em-

*A part of the research presented in this article was conducted during an internship at Samsung R&D Institute Bangalore, India.

ploy graph neural networks for the task of image inpainting, to the best of our knowledge.

Our major contributions are two-fold: (1) We introduce a novel pyramidal graph construction scheme to represent images as graphs for learning. Additionally, we extend this method and propose an efficient approach for processing high-resolution images for inpainting. (2) We demonstrate the effectiveness of graph neural networks for image inpainting, which has not been explored before, and show that our GraphFill Network effectively captures global information to improve robustness in filling missing regions.

2. Related Work

Our proposed work draws inspiration from Graph Convolutional Networks (GCNs) [15], which are convolutional as the filter parameters are usually shared across all locations in the graph. GCNs are particularly effective when dealing with data represented as graphs or network structures. They have been extensively used in a variety of problems related to graphical formats such as point-cloud or mesh analysis [26][6][28], social network analysis [33], and recommendation tasks [9]. Graph-based analysis of images has gained attention in various computer vision tasks such as image segmentation, detection, and recognition. Several studies, such as [35][38][43], have shown that these approaches can achieve competitive or even better results compared to Convolutional Neural Networks (CNNs).

The proposed work introduces a novel end-to-end trainable deep-learning method for image inpainting to learn coarse inpainting of the masked region in the image by utilizing its pyramidal graphical representation. Subsequently, a shallow Pix2Pix Refine Network is employed to improve the coarse inpainted region and generate the final inpainted output. The following paragraphs provide an overview and analysis of existing approaches in the field of image inpainting, with an emphasis on GAN-based methodologies.

GAN-based Approaches. Generative Adversarial Networks (GANs) [7][8] have gained popularity for their effectiveness in generating realistic textures. Therefore, generative networks have been extensively used for image inpainting problems. Among generative methods for image inpainting, the general approach uses an encoder-decoder architecture for the generator coupled with an adversarial training strategy. This method was first proposed by [23], and subsequent follow-up works [36][46][50][21][18][48][19][44][51][50][41][54] have achieved impressive results. GAN-based architectures that rely solely on simple convolutional layers often face challenges in generating semantically meaningful inpainted regions due to their small receptive fields. Various methods have been proposed in the literature to capture global and high-level semantic context. [12] use Dilated Convolutions to increase the receptive field of the network. [17] propose Partial Convolutions, while [46] introduce Gated Convolutions addressing limitations of [17] to guide convolutional kernels according to the masked region. Furthermore, [32] utilize Fourier Convolutions, which allow for a wide receptive field and improved results. The method by [42] leverages the relationship between the contextual regions in the encoder and the hole region in the decoder to enhance image inpainting outcomes. Subsequent works on contextual attention by [31][49][45] have further improved the method by incorporating global context for better inpainting results. Additionally, [21][39][40] employ edge maps, and [11][22] use segmentation maps for guidance in generation.

Other Approaches. Several approaches based on Variational Autoencoders (VAEs) have been proposed to address the lack of diversity in GAN-based image inpainting methods. [52][24][47] introduced large-scale VAEs with conditional prior networks, a hierarchical sampling method, and a bidirectional autoregressive transformer, respectively. However, VAE-based methods may produce blurry images and fail to preserve fine details, affecting the overall quality of results. Some alternative methods for diverse image inpainting include utilizing deep image priors and transformers [27][34][16][5], and [29][30][20] use Denoising Diffusion Probabilistic Models (DDPMs).

3. Approach

In this section, we outline our approach for image inpainting, covering the problem statement, our architecture (see Figure 1), and the training loss functions used.

3.1. Problem Statement

Suppose a portion of the image $\mathcal{I} \in \mathbb{R}^{3\times H\times W}$ is masked using a binary mask $\mathcal{M} \in \mathbb{R}^{H\times W}$, resulting in a masked image $\mathcal{I}_m \in \mathbb{R}^{3\times H\times W}$. The task of image inpainting is to fill in the masked region of $\mathcal{I}_m$ with plausible information to obtain an inpainted image $\hat{\mathcal{I}}$. We can represent $\mathcal{I}$ and $\mathcal{M}$ using graphs $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{F})$ and $\mathcal{G}_m = (\mathcal{V}, \mathcal{E}, \mathcal{F}_m)$, respectively, where $\mathcal{V}$ represents the set of nodes corresponding to pixels or superpixels, $\mathcal{E} \subseteq \mathcal{V}^2$ is the set of edges connecting neighbouring pixels or superpixels, $\mathcal{F}$ is the node-wise feature matrix of $\mathcal{G}$, and $\mathcal{F}_m$ is a binary vector containing 0, for every node $v \in \mathcal{V}$ that belongs to the masked region and 1, otherwise. Now, we can represent the masked image $\mathcal{I}_m$ using a graph $\mathcal{G}' = (\mathcal{V}, \mathcal{E}, \mathcal{F}')$, where $\mathcal{F}' = \mathcal{F} \odot \mathcal{F}_m$, $\odot$ denoting element-wise multiplication. Note that graphs $\mathcal{G}, \mathcal{G}_m$, and $\mathcal{G}'$ share same number of nodes $\mathcal{V}$ and edge connectivity $\mathcal{E}$. Our objective in coarser-to-finer image inpainting is to obtain the final inpainted image $\hat{\mathcal{I}}$ by refining the coarse inpainted image, which is obtained from recovering the original graph $\mathcal{G}$ from $\mathcal{G}'$.

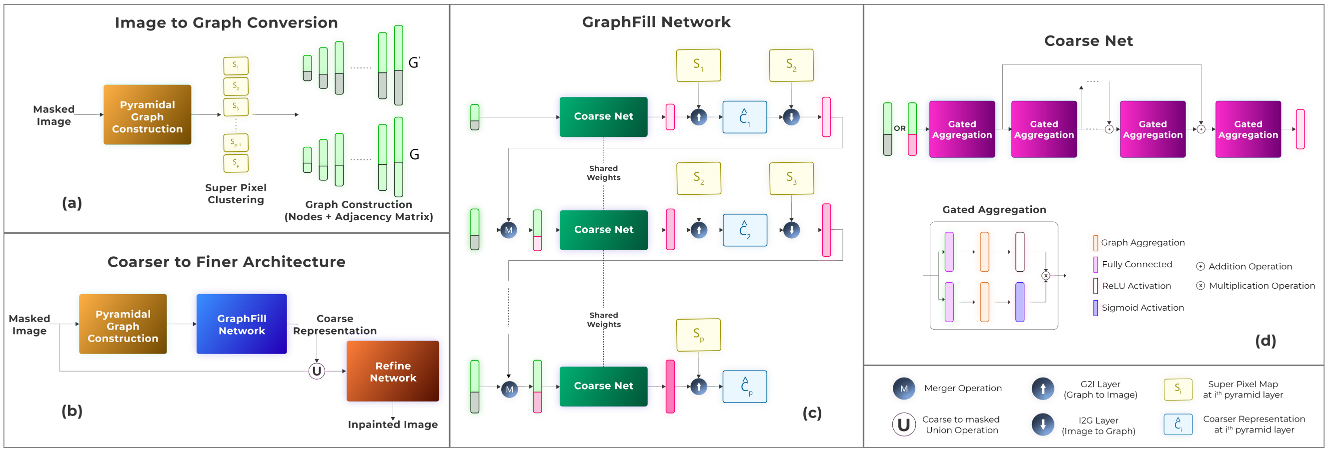

Figure 1. The proposed coarser-to-finer approach for image inpainting: (a) Pyramidal Graph Construction method based on SLIC [1]. (b) The overall architecture for image inpainting contains (c) GraphFill Network to learn a coarse representation and (d) CoarseNet Architecture with Gated Graph Aggregators as a fundamental building block.

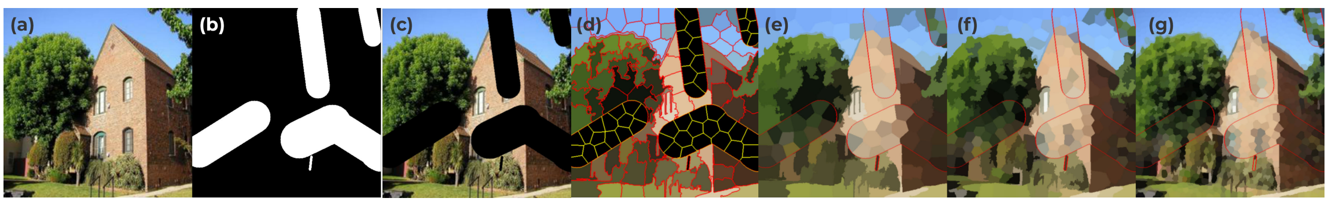

Figure 2. The image $\mathcal{I}$ (a) is subjected to pyramidal graph construction, (b) and (c) being mask $\mathcal{M}$ and masked image $\mathcal{I}_m$. Coarse images (e-g) are generated using the superpixel map at the respective pyramid level. Superpixels corresponding to foreground and background regions are indicated in (d) by red and yellow outlines, respectively.

3.2. Network Architecture and Loss Functions

We propose an end-to-end deep learning neural network for coarser-to-finer image inpainting, consisting of three main components: Pyramidal Graph Construction, GraphFill Network, and Refine Network, as illustrated in Figure 1(b). We provide a detailed description of each component and the loss functions used for training.

Pyramidal Graph Construction. To construct the pyramidal graph for an image $\mathcal{I}$ and its binary mask $\mathcal{M}$, we use the Simple Linear Iterative Clustering (SLIC) technique proposed by [1] to create superpixels at each level of the pyramid. Assuming p is the number of pyramid levels, we define $\mathcal{N} = \{N_f^i + N_b^i\}_{i=1}^p$ that represents the total number of superpixels in which the image $\mathcal{I}$ can be decomposed into, at i-th pyramid level. Here, N_f^i and N_b^i represent the number of superpixels for the foreground region where $\mathcal{M}(x,y) = 1$ and the background region where $\mathcal{M}(x,y) = 0$, respectively. The superpixel map $\mathcal{S}_i = \{S_i^k\}_{k=0}^{n_i}$ is defined as the collection of all superpixels, where $n_i \in \mathcal{N}$ denotes the total number of superpixels and $S_i^k \in \mathcal{S}_i$ represents the k-th superpixel in the superpixel map at i-th pyramid level. The complete set of all superpixel maps in the pyramid is represented by $\mathcal{S} = \{\mathcal{S}_i\}_{i=1}^p$. Note that the construction of superpixel map $\mathcal{S}_i$ involves clustering foreground and background regions into N_f^i and N_b^i superpixels, respectively. We represent $\mathcal{S}_i = \{\mathcal{S}_f^i\} \cup \{\mathcal{S}_b^i\}$, where $\mathcal{S}_f^i$ and $\mathcal{S}_b^i$ are superpixels corresponding to foreground and background region, respectively, as illustrated in Figure 2(d).

We apply the Image-to-Graph (I2G) layer to both the original image $\mathcal{I}$ and the masked image $\mathcal{I}_m$ using the set of superpixel maps $\mathcal{S}$, resulting in the graphs $\mathcal{G}$ and $\mathcal{G}'$, respectively. At the i-th pyramid level, we represent the sub-graph of $\mathcal{G}$ as G_i, and define the pyramidal graph $\mathcal{G}$ as $\mathcal{G} = \{G_i\}_{i=1}^p = (\mathcal{V}, \mathcal{E}, \mathcal{F})$. The set $\mathcal{V}$ contains nodes represented by superpixels $S_i^k \in \mathcal{S}_i \quad \forall \mathcal{S}_i \in \mathcal{S}$ and total number of nodes is $|V| = \sum n, \ \forall n \in \mathcal{N}$. The node features are represented by $\mathcal{F} \in \mathbb{R}^{|\mathcal{V}| \times 3}$. We can obtain graph $\mathcal{G}'$ from graph $\mathcal{G}$ by setting node feature $S_i^k = 0, \ \forall S_i^k \in \mathcal{S}_b^i$, in addition to using the I2G-layer on $\mathcal{I}_m$. This can be formulated mathematically as $\mathcal{F}' = \mathcal{F} \odot \mathcal{F}_m$, as discussed previously. At level i of the pyramid, we represent the sub-graph of $\mathcal{G}_m = (\mathcal{V}, \mathcal{E}, \mathcal{F}_m)$ corresponding to mask $\mathcal{M}$ as $G_i^m = (V_i, E_i, F_i^m)$.

After obtaining the sub-graph G_i for each pyramid level, we apply the Graph-to-Image (G2I) layer to map each graph G_i back to the image space. This results in a coarser representation of the original image, as shown in Figure 2(d-g). Figure 2(d) is obtained by projecting a sub-graph at a pyramid level of $\mathcal{G}'$, while Figures 2(e-g) are obtained by projecting sub-graphs at three pyramid levels from $\mathcal{G}$. Train-

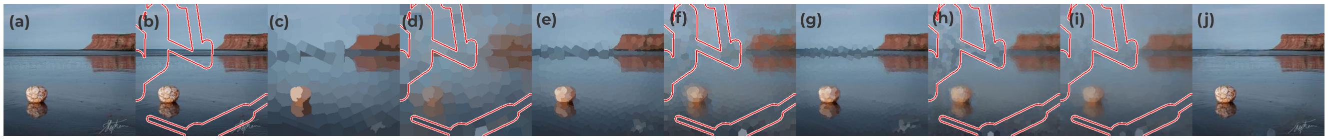

Figure 3. The inpainting process of an image $\mathcal{I}$ (a) from Places365 dataset [53], with masked region outlined in red (b). Ground truth and predicted coarser representations $(C_i, \hat{C}_i)$ are shown in (c,d), (e,f), and (g,h). Final coarse representation (i) is obtained by averaging pixel values of all $\hat{C}_i$, and the final inpainting result (j) is obtained after refinement by Refine Network.

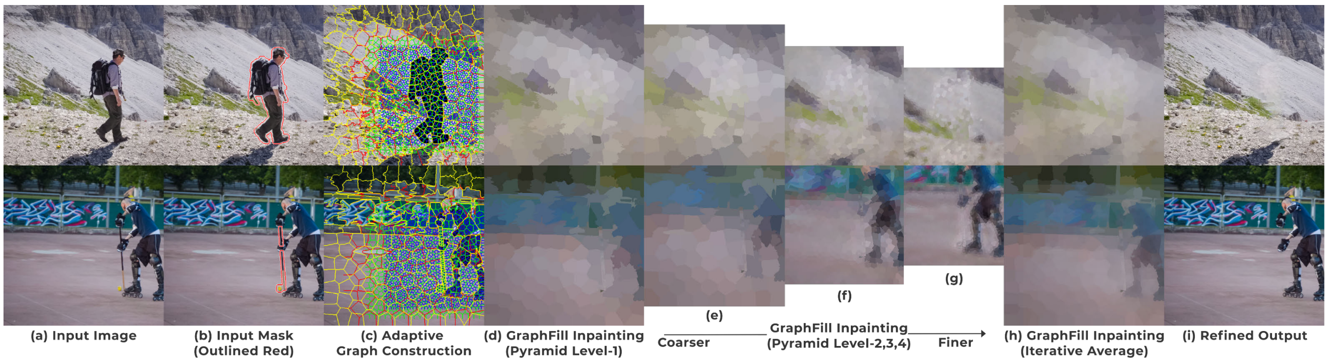

Figure 4. Resolution-Robust Pyramidal Graph construction and inpainting using our proposed approach. We utilize images and segmentation masks from the DAVIS [25] dataset for illustrative purposes.

ing pairs for our network **GraphFill** consist of both graphs $\mathcal{G}$ and $\mathcal{G}'$, along with their coarser representations at each pyramid level. An example of such a pair of coarser representations is depicted in Figures 2(d) and 2(e). The architecture of GraphFill, along with the I2G layer and G2I layer, is described below.

GraphFill Network. GraphFill performs gated graph aggregations in the graph $\mathcal{G}'$ constructed from the masked image $\mathcal{I}_m$, to obtain a coarser inpainting of the missing regions. The sub-graph of $\mathcal{G}$ formed from the superpixel map $\mathcal{S}_i$ containing the minimum number of superpixels (i.e., $\min(\mathcal{N})$) will be referred to as the coarsest sub-graph, and the one with the maximum number of superpixels (i.e., $\max(\mathcal{N})$), will be referred to as the finest sub-graph. GraphFill takes the coarsest sub-graph as input and uses CoarseNet to estimate values of unknown superpixels. It then iteratively updates the unknown superpixel values in subsequent finer sub-graphs through a merger operation, which is fed back to CoarseNet, as illustrated in Figure 1(c). Following, we provide a detailed description of the building blocks of GraphFill architecture.

Image-to-Graph (I2G) layer. The I2G layer maps an image I to a graph representation, where each superpixel $S_i^k \in \mathcal{S}_i$ corresponds to a node in the graph $G_i = (V_i, E_i, F_i)$, with V_i being the set of nodes, E_i being the set of edges and $F_i \in \mathbb{R}^{|\mathcal{S}_i| \times 3}$ being the feature matrix for each node. The node features are defined as the mean values of the pixels in P_i^k, which is the set of all pixels in the image I that superpixel S_i^k contains. An edge e_i^{mn} is added between nodes S_i^m and S_i^n if they are adjacent in the superpixel map.

Graph-to-Image (G2I) layer. The G2I layer projects the nodes of sub-graph G_i onto image space using the superpixel map $\mathcal{S}_i \in \mathbb{S}$ to obtain a coarser image representation $C_i \in \mathbb{R}^{3 \times H \times W}$ at the i-th pyramid level. Let P_i^k denote the set of all pixels in the image I contained in superpixel S_i^k. Then, each pixel in P_i^k is assigned the same value as the corresponding node S_i^k, i.e., $C_i(x) = S_i^k \ \forall x \in P_i^k$. The coarser representations in Figure 2(c-f) are obtained by projecting graphs back to image space using the G2I layer.

CoarseNet. CoarseNet consists of several gated aggregation blocks with skip connections that perform feature aggregation in the graph G_i' iteratively, taking a high dimensional feature vector $X_i \in \mathbb{R}^{|\mathcal{S}_i| \times k}$ extracted at a certain depth from the input feature matrix F_i', and the adjacency matrix A_i constructed from E_i. We use graph aggregation from [15] and modify gated convolutions from [46] to form a gated graph convolution block. The gated graph aggregation is defined as $g(X_i, A_i) = \sigma_r(\hat{D}_i^{-\frac{1}{2}} \hat{A}_i \hat{D}_i^{-\frac{1}{2}} X \mathcal{W}_f) \odot \sigma_g(\hat{D}_i^{-\frac{1}{2}} \hat{A} \hat{D}_i^{-\frac{1}{2}} X_i \mathcal{W}_g)$, where $\mathcal{W}_f$ and $\mathcal{W}_g$ are learnable weight matrices, $\hat{A}_i = A_i + I$ (I being the identity matrix), $\hat{D}_i$ is the diagonal node degree matrix of $\hat{A}_i$, and σ_r and σ_g are ReLU and sigmoid activation functions, respectively. The aggregation operation is shown in Figure 1(d), where we use shared weights $\mathcal{W}_f$ and $\mathcal{W}_g$ across all iterations for all sub-graphs G_i in $\mathcal{G}$.

Merger Operation. At $(i-1)$-th pyramid level, let CoarseNet estimates sub-graph $\hat{G}_{i-1}$ for input sub-graph G_{i-1}. Applying the G2I-layer on output sub-graph $\hat{G}_{i-1}$ with superpixel map $\mathcal{S}_{i-1}$ results in a coarse image denoted by $\hat{\mathcal{C}}_{i-1}$. Subsequently, the I2G-layer transforms $\hat{\mathcal{C}}_{i-1}$ to a finer sub-graph $\hat{G}_{i-1}^{\uparrow} = (V_i, E_i, \hat{F}_i^{\uparrow})$ corresponding to the

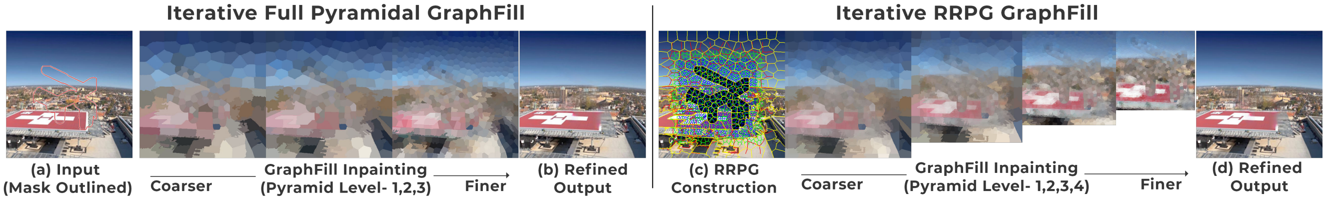

Figure 5. Visual comparison of results using the Full Pyramidal and RRP-Graph filling approach is shown in (b) and (d), respectively, with (a) as the input image. The RRPG method achieves comparable inpainting with lower computational requirements.

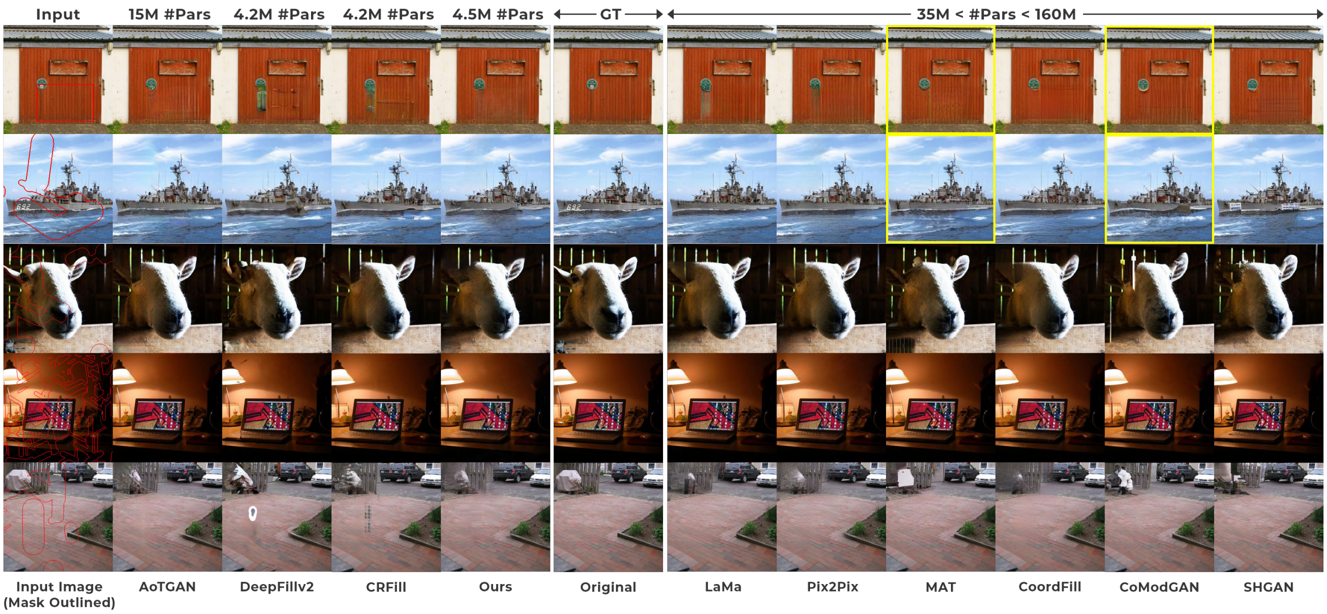

Figure 6. Qualitative comparison of our Coarser-to-Finer approach with state-of-the-art methods on Places365[53]: AOTGAN [49], DeepFillv2 [46], CRFill [50], GraphFill (Ours), LaMa [32], Pix2Pix [37], MAT[16], CoordFill [19], CoModGAN [51], and SHGAN [41].

Figure 7. Comparison of image inpainting methods on varying mask sizes. While DeepFillv2 [46] and CRFill [50] exhibit difficulties in capturing the global context, our GraphFill method demonstrates effective global context preservation.

i-th pyramid level, using the superpixel map $\mathcal{S}_i$. G_i and $\hat{G}^{\uparrow}_{i-1}$ have the same number of nodes V_i and edge connectivity E_i since they are obtained from the same superpixel map $\mathcal{S}_i$. To merge the feature matrix F'_i of G'_i with the feature matrix $\hat{F}^{\uparrow}_i$ of $\hat{G}^{\uparrow}_{i-1}$, we use a merger operation defined as $(F_i \odot F^m_i) + (\hat{F}^{\uparrow}_i \odot (1 - F^m_i))$.

Refine Network.* To refine the coarse output from the GraphFill Network for inpainting, we employ a shallow version of the GAN-based network proposed by [37]. We achieve the final inpainting outcome by refining CoarseNet's output $\hat{C}_p$ at the finest pyramid layer p combined with the masked image $\mathcal{I}_m$ using the Refine Network. We perform the combination of these two inputs through a masked update, which involves $(\mathcal{I}_m \odot \mathcal{M}) + (\hat{C}_p \odot (1 - \mathcal{M}))$, represented as *Coarse to Masked Union* in Figure 1.

Loss Functions.* At each level i of the pyramid, the CoarseNet estimates the sub-graph $\hat{G}_i$ from the input G'_i. Then, we use the G2I-layer to project $\hat{F}_i$ onto the image space and obtain a coarse inpainting $\hat{C}_i$ for the masked image $\mathcal{I}_m$. To get the corresponding ground truth C_i for this estimated $\hat{C}_i$, we apply the G2I-layer on the node features F_i obtained from sub-graph G_i of image $\mathcal{I}$. The training of the GraphFill Network involves minimizing L2 and Perceptual losses between C_i and $\hat{C}_i$ at all levels i in the pyramid. On the other hand, the Refine Network is trained using GAN loss and feature matching loss inspired by [37].

*More details included in the supplementary material

Model	#Pars	Model	#Pars
GraphFill (Ours)	175K	DeepFillv2[46]	4.2M
GraphFill-Pix (Ours)	175K + 4.4M	CRFill[50]	4.2M
Pix2Pix (Shallow)	4.4M	CoordFill[19]	34.4M
Pix2Pix[37] (Deep)	45.6M	Big LaMa[32]	45M
FFCResNet (Shallow)	3.8M	MAT[16]	62M
FFCResNet[32] (Deep)	27M	CoModGAN[51]	109M
AOTGAN[49]	15.2M	SHGAN[41]	159.6M

Table 1. Total Number of learnable parameters in GraphFill, Refine Network baselines, and other existing methods.

Model	Metrics		
	FID ↓	LPIPS ↓	SSIM ↑
GraphFill-Pix (Iterative) with RRPG	1.509	0.0301	0.981
GraphFill-Pix (Iterative) without RRPG	1.505	0.0298	0.980
GraphFill-Pix (Non-Iterative) with RRPG	1.719	0.0308	0.976
GraphFill-Pix (Non-Iterative) without RRPG	1.704	0.0307	0.979

Table 2. Comparison of the proposed GraphFill inpainting models with and without the Resolution-robust Pyramidal Graph (RRPG). Symbol ↑ denotes larger values are better. This ablation study is validated with random masks on a reduced validation split of 5000 images from the Places365[53] dataset.

3.3. Resolution-Robust Pyramidal Graph

To address the inpainting of high-resolution images, we propose a resolution-robust pyramidal graph construction approach for inpainting using GraphFill, as illustrated in Figure 4. Since undesired objects occupy a smaller region of the overall image size, we use an adaptive cropping approach that crops the input image around the masked area (see Figure 4(c)). Our pyramidal graph construction follows a similar procedure as described in section 3.2. However, we use images with an increased crop and a larger value of $n \in \mathcal{N}$ at higher levels of the pyramid, generating finer superpixels as the level of the pyramid increases. The lowest pyramid level contains the coarsest sub-graph generated from the full-resolution image, and the highest level contains the finest sub-graph generated from the maximum possible cropping of the input image. We constrain the maximum cropping around the masked region to ensure the image size is not reduced below a certain threshold. In our experiments, we set $H_c = 224$ and $W_c = 224$. The cropping parameters are saved at each pyramid level to enable proper merger operations in CoarseNet and stitching to obtain the final inpainted image. The *Coarse to Masked Union Operation* is performed on the original image cropped to the maximum possible extent, and the coarse output $\hat{\mathcal{C}}_p$ predicted at the p-th level of the pyramid graph, p representing the total number of levels in the pyramid and contains the finest sub-graph.

To refine the sub-graphs at higher pyramid levels in the resolution-robust pyramidal graph, the *Merger Operation* relies on the cropping information saved at each level. The predicted coarse image $\hat{\mathcal{C}}_{i-1}$ is obtained by applying the G2I layer on $\hat{G}_{i-1}$ with the superpixel map $\mathcal{S}_{i-1}$. This coarse image is then cropped using the cropping parameters at the i-th level of the pyramid, resulting in $\hat{C}^{\boxdot}_{i-1}$. The I2G-layer then transforms $\hat{C}^{\boxdot}_{i-1}$ instead of $\hat{C}_{i-1}$ to obtain a finer subgraph $\hat{G}^{\uparrow}_{i-1}$ corresponding to the i-th pyramid level, using the superpixel map $\mathcal{S}_i$. The Merger Operation refines the graph and facilitates the transfer of global context from the $(i-1)$-th pyramid level to the i-th pyramid level. Figure 4(d-g) shows predicted coarse output $\hat{\mathcal{C}}^{\boxdot}_{i-1}$ at i-th level of pyramid. Figure 4(h) shows the averaged output of all $\hat{\mathcal{C}}^{\boxdot}_i$'s stitched with corresponding cropping parameters. Figure 4(i) shows the final refined output from Refine Network.

4. Results and Discussions

Datasets. Our proposed network is trained and evaluated on the Places365 [53] and CelebA-HQ [13] datasets, which have 1.8 million and 30k images, respectively, in the training split and 10k and 5k images, respectively, in the validation split. To evaluate our model and compare it to other state-of-the-art models, we adopt a similar approach to [32]. Specifically, we use pre-generated narrow (NM), medium (MM), and wide masks (WM) for each image in the validation split to ensure a fair comparison of metrics.

Results. The GraphFill Network is trained using a pyramidal graphical image representation with three levels $p = 3$. For a 256×256 resolution, the number of nodes in the foreground regions is $N_f = (100, 500, 1500)$, and in the background regions, the number of nodes is $N_b = (50, 100, 200)$. For a 512×512 resolution, the number of graph nodes in the background regions slightly increases to $N_b = (50, 200, 400)$. Due to the enforcement of region connectivity during superpixel determination using SLIC [1], the resulting graph has a total of $\mathcal{N} \leq (150, 600, 1700)$ nodes for 256×256 resolution, and $\mathcal{N} \leq (150, 700, 1900)$ for 512×512 resolution. The proposed coarser-to-finer approach for inpainting a masked image is demonstrated in Figure 3. Figures 6 and 8 provide a qualitative comparison of our image inpainting method with existing approaches on both the CelebA-HQ and Places365 datasets. The first 2 rows in Figure 6, and first 3 rows in Figure 8 present results at 256×256, while the remaining rows display results at 512×512. The yellow outlined images in are generated at a higher resolution of 512×512 by upscaling the corresponding image and mask due to the lack of inference support for 256×256 resolution images. Figure 7 demonstrates the robustness of GraphFill inpainting as we progressively enlarge the masked region area in comparison with existing methods. Our experiments demonstrate that the GraphFill Network effectively fills the masked region with coarser details, enabling the Refine Network to generate visually plausible inpainting results. As presented in Table 1 and Table 3, quantitative analysis demonstrates that our proposed network achieves competitive results even with a substantially

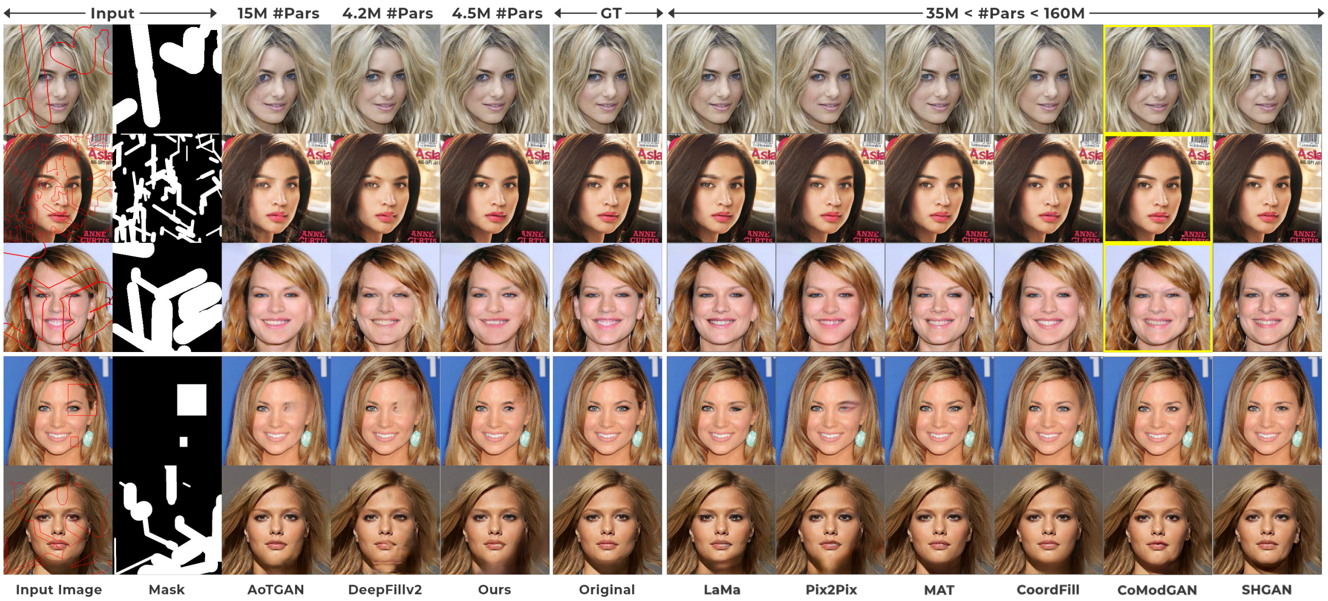

Figure 8. Qualitative comparison of our Coarser-to-Finer approach with state-of-the-art methods on CelebA-HQ[13] dataset: AOTGAN [49], DeepFillv2 [46], GraphFill (Ours), LaMa [32], Pix2Pix [37], MAT[16], CoordFill [19], CoModGAN [51], and SHGAN [41]
.

Model	Places365 (512x512)									CelebA-HQ (512x512)								
	Narrow Masks			Medium Masks			Wide Masks			Narrow Masks			Medium Masks			Wide Masks		
	FID ↓	LPIPS ↓	SSIM ↑	FID ↓	LPIPS ↓	SSIM ↑	FID ↓	LPIPS ↓	SSIM ↑	FID ↓	LPIPS ↓	SSIM ↑	FID ↓	LPIPS ↓	SSIM ↑	FID ↓	LPIPS ↓	SSIM ↑
GraphFill-Pix (Ours)	3.428	0.107	0.909	5.392	0.129	0.876	8.957	0.152	0.863	2.462	0.111	0.929	7.727	0.13	0.899	11.119	0.146	0.881
DeepFill-v2[46]	3.569	0.106	0.906	6.363	0.13	0.870	9.698	0.157	0.837	6.094	0.145	0.919	9.125	0.152	0.894	14.218	0.163	0.871
CRFill[50]	3.461	0.102	0.91	5.458	0.127	0.874	8.821	0.155	0.843	_	_	_	_	_	_	_	_	_
AOTGAN[49]	4.472	0.127	0.882	6.012	0.128	0.866	9.932	0.159	0.831	4.313	0.116	0.897	6.812	0.127	0.892	11.947	0.158	0.852
CoordFill[19]	3.922	0.117	0.906	5.806	0.124	0.885	7.828	0.144	0.866	3.522	0.128	0.925	3.394	0.113	0.914	3.475	0.115	0.906
CoModGAN[51]	3.302	0.113	0.898	4.67	0.127	0.869	5.7	0.148	0.843	2.02	0.124	0.917	2.44	0.127	0.893	2.676	0.131	0.881
LaMa[32]	2.486	0.091	0.915	4.056	0.112	0.887	5.587	0.136	0.866	2.113	0.106	0.929	3.095	0.12	0.905	3.673	0.129	0.894
MAT[16]	2.814	0.096	0.903	4.432	0.121	0.869	5.688	0.145	0.841	1.409	0.092	0.926	1.88	0.106	0.901	1.964	0.113	0.889
SH-GAN[41]	3.157	0.108	0.906	4.591	0.126	0.868	5.622	0.148	0.843	1.895	0.114	0.919	2.349	0.121	0.895	2.551	0.125	0.882
Model	**Places365 (256x256)**									**CelebA-HQ (256x256)**								
GraphFill-Pix (Ours)	4.782	0.102	0.919	5.061	0.101	0.899	7.217	0.143	0.857	2.802	0.088	0.918	3.372	0.093	0.905	6.775	0.125	0.875
DeepFill-v2[46]	4.996	0.104	0.901	4.931	0.104	0.891	7.778	0.145	0.843	6.522	0.126	0.901	4.15	0.103	0.901	5.853	0.12	0.872
CRFill[50]	5.348	0.104	0.904	5.286	0.104	0.894	8.34	0.145	0.845	_	_	_	_	_	_	_	_	_
AOTGAN[49]	4.853	0.117	0.892	5.502	0.116	0.890	8.932	0.167	0.832	2.504	0.091	0.921	3.495	0.118	0.904	6.337	0.138	0.865
CoordFill[19]	3.873	0.089	0.915	4.14	0.092	0.905	6.675	0.131	0.863	3.272	0.086	0.926	2.68	0.081	0.92	3.29	0.098	0.897
LaMa[32]	3.455	0.086	0.912	3.349	0.088	0.903	4.817	0.125	0.861	2.496	0.081	0.923	2.093	0.077	0.917	2.403	0.092	0.895
MAT[16]	_	_	_	_	_	_	_	_	_	1.989	0.078	0.921	1.869	0.08	0.91	2.347	0.098	0.884
SH-GAN[41]	3.712	0.100	0.917	3.789	0.101	0.883	5.344	0.140	0.841	3.036	0.091	0.910	2.768	0.089	0.903	3.706	0.107	0.874

Table 3. Quantitative comparison of our proposed method with state-of-the-art Image Inpainting methods using Frechet inception distance (FID) metrics, Learned perceptual image patch similarity (LPIPS), and Structural Similarity (SSIM) metrics. Symbol ↓ denotes lower values are better, and ↑ denotes larger values are better. Symbol '_' is filled if the corresponding trained model is not publicly available or the model does not support the evaluation of the respective resolution. Note that, as illustrated in Table 1, our proposed model has substantially fewer parameters and performs competitively compared to other existing methods.

lower number of learnable parameters than heavy-weight existing methods and deep baseline networks. We train the GraphFill Network for an initial 5 epochs, aiming to grasp a coarser representation. Subsequently, we combine the Refine Network and proceed with an end-to-end training approach. On the Places365 Dataset [53], our training spans 10 epochs, while for the CelebA-HQ Dataset [13], we train for 25 epochs. All experiments are conducted on a machine with a 20-core CPU and an NVIDIA Tesla V100 GPU. To demonstrate the effectiveness of the proposed method for mobile deployment, we convert the model to TFLite format with $INT8$ quantization. The size of the resulting TFLite model is 4.6MB. The model takes about 13 *ms* to load, and the entire inference process, including data preprocessing and model runtime, takes about 105 *ms*. The experiment is evaluated on the SAMSUNG GALAXY S23 smartphone.

Model	Places365 (512x512)									CelebA-HQ (512x512)								
	Narrow Masks			Medium Masks			Wide Masks			Narrow Masks			Medium Masks			Wide Masks		
	FID ↓	LPIPS ↓	SSIM ↑	FID ↓	LPIPS ↓	SSIM ↑	FID ↓	LPIPS ↓	SSIM ↑	FID ↓	LPIPS ↓	SSIM ↑	FID ↓	LPIPS ↓	SSIM ↑	FID ↓	LPIPS ↓	SSIM ↑
GraphFill-Pix (Iterative)	3.81	0.111	0.908	7.792	0.133	0.876	9.067	0.158	0.86	2.462	0.111	0.929	7.727	0.13	0.899	11.119	0.146	0.881
GraphFill-Pix (Non-Iterative)	4.285	0.113	0.907	8.833	0.137	0.876	10.47	0.16	0.841	3.207	0.115	0.922	9.019	0.135	0.892	15.356	0.15	0.876
GraphFill-FFC (Iterative)	4.987	0.132	0.91	8.787	0.134	0.877	11.095	0.162	0.841	4.934	0.133	0.919	8.571	0.143	0.895	14.832	0.156	0.883
GraphFill-FFC (Non-Iterative)	5.804	0.135	0.892	9.37	0.14	0.872	11.559	0.169	0.848	5.07	0.145	0.919	9.254	0.148	0.89	15.645	0.161	0.879
Pix2Pix (Shallow)	5.068	0.114	0.909	9.194	0.139	0.875	13.652	0.17	0.849	4.075	0.136	0.911	9.395	0.144	0.897	18.52	0.169	0.882
Pix2Pix (Deep)	3.288	0.108	0.91	5.816	0.127	0.881	8.927	0.152	0.857	3.602	0.123	0.927	7.328	0.137	0.904	11.763	0.153	0.887
FFCResNet (Shallow)	5.977	0.142	0.881	9.151	0.145	0.862	11.807	0.178	0.86	5.287	0.14	0.918	9.919	0.146	0.896	17.379	0.162	0.88
FFCResNet (Deep)	2.976	0.105	0.912	5.297	0.125	0.882	7.919	0.149	0.858	3.947	0.124	0.932	6.614	0.134	0.907	8.772	0.142	0.893

Table 4. Ablation studies on the effect of GraphFill integration on Shallow Baselines: Notable performance improvements and competitive performance compared to deep counterparts.

Ablation Studies. We conducted several ablation studies to evaluate the performance of GraphFill and its integration with two shallow variants of Refine Networks: Pix2Pix [37], and FFCResNet proposed by [32]. We also tested iterative graph-filling (as discussed in section 3.2) and non-iterative graph-filling schemes. In the non-iterative scheme, we directly input the full-graph $\mathcal{G}'$ to the GraphFill Network with the adjacency matrix $\mathcal{A}$ calculated from the connectivity information in $\mathcal{E}$. The non-iterative graph-filling scheme does not involve the merger operation at every successive pyramid level. Instead, the output at every pyramid level sub-graph is converted to coarser images using the G2I layer and averaged for coarse to masked union operation needed before Refine Network. We quantitatively compare our coarse-to-finer inpainting variants in Table 4. The GraphFill neural network is trained for 10 epochs to learn the coarser representation. This pre-trained GraphFill Network is combined with shallow Refine Networks, and the entire model is trained end-to-end for refinement. All variants listed in Table 4 are trained for 5 epochs on the Places365 dataset [53] and 25 epochs on the CelebA-HQ dataset [13]. As evident in Table 4, shallow networks integrated with the GraphFill module generate competitive inpainting results despite having lower learning parameters than their deep counterparts. Also, the results indicate that the variant with Iterative GraphFill integrated with shallow Pix2Pix architecture performs the best on average on both the validation split of the Places365 and CelebA-HQ datasets. We evaluate the performance of the RRPG by creating a validation split where the masked region is constrained to a 256×256 square patch within a 512×512 resolution image. Table 2 compares the GraphFill inpainting method with and without the Resolution-Robust Pyramidal Graph (RRPG) approach. Visual comparison between the two graph filling inpainting techniques is depicted in Figure 5. The RRPG is designed to reduce computational complexity while maintaining inpainting performance, allowing efficient processing of high-resolution images. Additional qualitative results on the RRPG approach and Non-Iterative GraphFill can be found in the *suppl. material.*

5. Conclusion

This work introduces a novel framework for image inpainting based on deep graph learning and pyramidal graph construction. Our approach outperforms existing methods having a similar number of learnable parameters and obtains competitive performance compared to existing heavy-weight models. Our method effectively captures long-range, non-local contextual information. Through extensive ablation studies, we demonstrate that the integration of GraphFill architecture significantly improves the performance of shallow baselines. Our results indicate that the merger operation in iterative graph-filling enables better passage of global context from coarser to finer pyramid levels compared to non-iterative graph-filling variants. We also propose a Resolution-Robust Pyramidal Graph construction method for high-resolution image inpainting, which reduces computational complexity with minimal deterioration in performance. Finally, due to the lightweight nature of our model, it can be easily deployed on mobile devices with computational limitations. Our approach provides a promising solution for image inpainting with practical implications in real-world scenarios.

6. Acknowledgements

This work is supported by the Jibaben Patel Chair in Artificial Intelligence. The authors extend their gratitude to Samsung R&D Institute for their support, resources, and invaluable insights.

References

[1] Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, and Sabine Süsstrunk. Slic superpixels compared to state-of-the-art superpixel methods. *IEEE transactions on pattern analysis and machine intelligence*, 34(11):2274–2282, 2012.

[2] Marcelo Bertalmio, Guillermo Sapiro, Vincent Caselles, and Coloma Ballester. Image inpainting. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 417–424, 2000.

[3] Marcelo Bertalmio, Luminita Vese, Guillermo Sapiro, and Stanley Osher. Simultaneous structure and texture image inpainting. *IEEE transactions on image processing*, 12(8):882–889, 2003.

[4] Antonio Criminisi, Patrick Pérez, and Kentaro Toyama. Region filling and object removal by exemplar-based image inpainting. *IEEE Transactions on image processing*, 13(9):1200–1212, 2004.

[5] Qiaole Dong, Chenjie Cao, and Yanwei Fu. Incremental transformer structure enhanced image inpainting with masking positional encoding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

[6] Aalok Gangopadhyay, Shashikant Verma, and Shanmuganathan Raman. Dmd-net: deep mesh denoising network. In *2022 26th International Conference on Pattern Recognition (ICPR)*, pages 3168–3175. IEEE, 2022.

[7] I Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets (advances in neural information processing systems)(pp. 2672–2680). *Red Hook, NY Curran*, 2014.

[8] 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.

[9] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng Wang. Lightgcn: Simplifying and powering graph convolution network for recommendation. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, pages 639–648, 2020.

[10] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.

[11] Seunghoon Hong, Xinchen Yan, Thomas S Huang, and Honglak Lee. Learning hierarchical semantic image manipulation through structured representations. *Advances in Neural Information Processing Systems*, 31, 2018.

[12] Satoshi Iizuka, Edgar Simo-Serra, and Hiroshi Ishikawa. Globally and locally consistent image completion. *ACM Transactions on Graphics (ToG)*, 36(4):1–14, 2017.

[13] 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.

[14] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[15] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[16] Wenbo Li, Zhe Lin, Kun Zhou, Lu Qi, Yi Wang, and Jiaya Jia. Mat: Mask-aware transformer for large hole image inpainting. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10758–10768, 2022.

[17] Guilin Liu, Fitsum A Reda, Kevin J Shih, Ting-Chun Wang, Andrew Tao, and Bryan Catanzaro. Image inpainting for irregular holes using partial convolutions. In *Proceedings of the European conference on computer vision (ECCV)*, pages 85–100, 2018.

[18] Hongyu Liu, Bin Jiang, Yibing Song, Wei Huang, and Chao Yang. Rethinking image inpainting via a mutual encoder-decoder with feature equalizations. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, pages 725–741. Springer, 2020.

[19] Weihuang Liu, Xiaodong Cun, Chi-Man Pun, Menghan Xia, Yong Zhang, and Jue Wang. Coordfill: Efficient high-resolution image inpainting via parameterized coordinate querying. *arXiv preprint arXiv:2303.08524*, 2023.

[20] Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. Repaint: Inpainting using denoising diffusion probabilistic models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11461–11471, 2022.

[21] Kamyar Nazeri, Eric Ng, Tony Joseph, Faisal Z Qureshi, and Mehran Ebrahimi. Edgeconnect: Generative image inpainting with adversarial edge learning. *arXiv preprint arXiv:1901.00212*, 2019.

[22] Evangelos Ntavelis, Andrés Romero, Iason Kastanis, Luc Van Gool, and Radu Timofte. Sesame: Semantic editing of scenes by adding, manipulating or erasing objects. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXII 16*, pages 394–411. Springer, 2020.

[23] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2536–2544, 2016.

[24] Jialun Peng, Dong Liu, Songcen Xu, and Houqiang Li. Generating diverse structure for image inpainting with hierarchical vq-vae. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10775–10784, 2021.

[25] Federico Perazzi, Jordi Pont-Tuset, Brian McWilliams, Luc Van Gool, Markus Gross, and Alexander Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 724–732, 2016.

[26] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017.

[27] Elad Richardson, Yuval Alaluf, Or Patashnik, Yotam Nitzan, Yaniv Azar, Stav Shapiro, and Daniel Cohen-Or. Encoding in style: a stylegan encoder for image-to-image translation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2287–2296, 2021.

[28] Weijing Shi and Raj Rajkumar. Point-gnn: Graph neural network for 3d object detection in a point cloud. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1711–1719, 2020.

[29] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, pages 2256–2265. PMLR, 2015.

[30] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

[31] Yuhang Song, Chao Yang, Zhe Lin, Xiaofeng Liu, Qin Huang, Hao Li, and C-C Jay Kuo. Contextual-based image inpainting: Infer, match, and translate. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 3–19, 2018.

[32] 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.

[33] Qiaoyu Tan, Ninghao Liu, and Xia Hu. Deep representation learning for social network analysis. *Frontiers in big Data*, 2:2, 2019.

[34] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Deep image prior. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9446–9454, 2018.

[35] Shashikant Verma, Rajendra Nagar, and Shanmuganathan Raman. Fast semantic feature extraction using superpixels for soft segmentation. In *Computer Vision and Image Processing: 4th International Conference, CVIP 2019, Jaipur, India, September 27–29, 2019, Revised Selected Papers, Part I 4*, pages 61–72. Springer, 2020.

[36] 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*, pages 8798–8807, 2018.

[37] 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.

[38] Guo-Sen Xie, Jie Liu, Huan Xiong, and Ling Shao. Scale-aware graph neural network for few-shot semantic segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5475–5484, 2021.

[39] Wei Xiong, Jiahui Yu, Zhe Lin, Jimei Yang, Xin Lu, Connelly Barnes, and Jiebo Luo. Foreground-aware image inpainting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5840–5848, 2019.

[40] Shunxin Xu, Dong Liu, and Zhiwei Xiong. E2i: Generative inpainting from edge to image. *IEEE Transactions on Circuits and Systems for Video Technology*, 31(4):1308–1322, 2020.

[41] Xingqian Xu, Shant Navasardyan, Vahram Tadevosyan, Andranik Sargsyan, Yadong Mu, and Humphrey Shi. Image completion with heterogeneously filtered spectral hints. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 4591–4601, 2023.

[42] Zhaoyi Yan, Xiaoming Li, Mu Li, Wangmeng Zuo, and Shiguang Shan. Shift-net: Image inpainting via deep feature rearrangement. In *Proceedings of the European conference on computer vision (ECCV)*, pages 1–17, 2018.

[43] Lumin Yang, Jiajie Zhuang, Hongbo Fu, Xiangzhi Wei, Kun Zhou, and Youyi Zheng. Sketchgnn: Semantic sketch segmentation with graph neural networks. *ACM Transactions on Graphics (TOG)*, 40(3):1–13, 2021.

[44] Zili Yi, Qiang Tang, Shekoofeh Azizi, Daesik Jang, and Zhan Xu. Contextual residual aggregation for ultra high-resolution image inpainting. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7508–7517, 2020.

[45] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Generative image inpainting with contextual attention. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5505–5514, 2018.

[46] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Free-form image inpainting with gated convolution. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4471–4480, 2019.

[47] Yingchen Yu, Fangneng Zhan, Rongliang Wu, Jianxiong Pan, Kaiwen Cui, Shijian Lu, Feiying Ma, Xuansong Xie, and Chunyan Miao. Diverse image inpainting with bidirectional and autoregressive transformers. In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 69–78, 2021.

[48] Yanhong Zeng, Jianlong Fu, Hongyang Chao, and Baining Guo. Learning pyramid-context encoder network for high-quality image inpainting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1486–1494, 2019.

[49] Yanhong Zeng, Jianlong Fu, Hongyang Chao, and Baining Guo. Aggregated contextual transformations for high-resolution image inpainting. *IEEE Transactions on Visualization and Computer Graphics*, 2022.

[50] Yu Zeng, Zhe Lin, Huchuan Lu, and Vishal M. Patel. Cr-fill: Generative image inpainting with auxiliary contextual reconstruction. In *Proceedings of the IEEE International Conference on Computer Vision*, 2021.

[51] Shengyu Zhao, Jonathan Cui, Yilun Sheng, Yue Dong, Xiao Liang, Eric I Chang, and Yan Xu. Large scale image comple-

tion via co-modulated generative adversarial networks. In *International Conference on Learning Representations (ICLR)*, 2021.

[52] Shengyu Zhao, Jonathan Cui, Yilun Sheng, Yue Dong, Xiao Liang, Eric I Chang, and Yan Xu. Large scale image completion via co-modulated generative adversarial networks. *arXiv preprint arXiv:2103.10428*, 2021.

[53] 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.

[54] 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.