GraphFill: Deep Image Inpainting using Graphs

Shashikant Verma¹, Aman Sharma², Roopa Sheshadri², Shanmuganathan Raman¹
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-
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][43][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. [32][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 $\hat{\mathcal{G}}$ from $\mathcal{G}'$. 


The complete set of all superpixel maps in the pyramid is 

\[ S_i \] superpixel map of superpixels for the foreground region where 

\[ M \]

and the background region where 

\[ I \]

mid. Assuming 

\[ \text{SLIC} \]

Simple Linear Iterative Clustering (SLIC) technique proposed by [1] to create superpixels at each level of the pyramid graph for an image 

\[ \mathcal{I} \]

To construct the pyramidal graph, 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 \( M(x, y) = 1 \) and the background region where \( M(x, y) = 0 \), respectively. The superpixel map \( \mathcal{S}_i = \{ S^p_i \}_{i=0}^{n_i} \) is defined as the collection of all superpixels, where \( n_i \in \mathbb{N} \) denotes the total number of superpixels and \( S^k_i \in \mathcal{S}_i \) 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 

\[ S = \{ S^p_i \}_{i=1}^p \].

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 \( \mathcal{S}_i = \{ S^f_i \} \cup \{ S^b_i \} \), where \( S^f_i \) and \( S^b_i \) are superpixels corresponding to foreground and background regions, 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 superpixels 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 

\[ \mathcal{G} = \{ G^p_i \}_{i=1}^p = (\mathcal{V}, \mathcal{E}, \mathcal{F}) \].

The set \( \mathcal{V} \) contains nodes represented by superpixels \( S^k_i \in \mathcal{S}_i \), \( \forall S_i \in \mathcal{S} \) and total number of nodes is 

\[ |\mathcal{V}| = \sum n_i, \forall n_i \in \mathcal{N} \].

The node features are represented by 

\[ \mathcal{F} \in \mathbb{R}^{|\mathcal{V}|} \].

We can obtain graph \( \mathcal{G}' \) from graph \( \mathcal{G} \) by setting node feature \( S^k_i = 0 \), \( \forall S^k_i \in \mathcal{S}_i \) 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 

\[ G^m_i = (\mathcal{V}_i, \mathcal{E}_i, \mathcal{F}'_i) \].

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-
ing pairs for our network GraphFill consist of both graphs $G$ and $G'$, 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 $G'$ constructed from the masked image $I_m$, to obtain a coarser inpainting of the missing regions. The sub-graph of $G$ formed from the superpixel map $S_i$ containing the minimum number of superpixels (i.e., min($N$)) will be referred to as the coarsest sub-graph, and the one with the maximum number of superpixels (i.e., max($N$)), 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 $S^k_i \in S_i$ corresponds to a node in the graph $G_i = (V_i, E_i, F_i)$, with $V_i$ being the set of nodes, $E_i$ the set of edges and $F_i \in \mathbb{R}^{[S_i] \times 3}$ being the feature matrix for each node. The node features are defined as the mean values of the pixels in $P^k_i$, which is the set of all pixels in the image $I$ that superpixel $S^k_i$ contains. An edge $e^m_{i}$ is added between nodes $S^m_{i}$ and $S^n_{i}$ 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 $S_i \in S$ 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(f-i) 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 $G'$ iteratively, taking a high dimensional feature vector $X_i \in \mathbb{R}^{[S_i] \times k}$ extracted at a certain depth from the input feature matrix $P_i$, 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 $g(X_i, A_i) = \sigma_r(D_i^{-\frac{1}{2}} \hat{A}_i D_i^{-\frac{1}{2}} X_i W_f) \odot \sigma_g(D_i^{-\frac{1}{2}} \hat{A}_i D_i^{-\frac{1}{2}} X_i W_g)$, where $W_f$ and $W_g$ are learnable weight matrices, $\hat{A}_i = A_i + I$ ($I$ being the identity matrix), $D_i$ is the diagonal node degree matrix of $A_i$, and $\sigma_r$ and $\sigma_g$ are ReLU and sigmoid activation functions, respectively. The aggregation operation is shown in Figure 2(d), where we use shared weights $W_f$ and $W_g$ across all iterations for all sub-graphs $G_i$ in $G$.

**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 $S_{i-1}$ results in a coarse image denoted by $\hat{C}_{i-1}$. Subsequently, the I2G-layer transforms $\hat{C}_{i-1}$ to a finer sub-graph $\hat{G}^1_{i-1} = (V_i, E_i, F_i^1)$ corresponding to the
Figure 5. Visual comparison of results using the Full Pyramidal and RRPG-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: AOTGAN , DeepFillv2, CRFill, GraphFill (Ours), LaMa, Pix2Pix, MAT, CoordFill, CoModGAN, and SHGAN.

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

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 . 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 $I_m$ using the Refine Network. We perform the combination of these two inputs through a masked update, which involves $(I_m \odot M) + (\hat{C}_p \odot (1 - M))$, represented as Coarse to Masked Union in Figure 1.

Loss Functions. At each level $i$ of the pyramid, the CoarseNet estimates the sub-graph $\hat{C}_i$ 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 $I_m$. 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 $I$. 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 .

More details included in the supplementary material
Table 1. Total Number of learnable parameters in GraphFill, Re- 
Fine Network baselines, and other existing methods.

<table>
<thead>
<tr>
<th>Model</th>
<th>Pars</th>
<th>Model</th>
<th>Pars</th>
</tr>
</thead>
<tbody>
<tr>
<td>GraphFill (Ours)</td>
<td>175K</td>
<td>FFCResNet (Non-Iterative) without RRPG</td>
<td>4.4M</td>
</tr>
<tr>
<td>GraphFill-Pix (Ours)</td>
<td>175K + 4.4M</td>
<td>FFCResNet (Deep)</td>
<td>27M</td>
</tr>
<tr>
<td>Pix2Pix (Shallow)</td>
<td>4.4M</td>
<td>GraphFill-Pix (Ours)</td>
<td>15.2M</td>
</tr>
<tr>
<td>Pix2Pix (Deep)</td>
<td>45.6M</td>
<td>GraphFill-Pix (Ours)</td>
<td>15.2M</td>
</tr>
<tr>
<td>CRFill[50]</td>
<td>0.0301</td>
<td>Big LaMa[9]</td>
<td>35M</td>
</tr>
<tr>
<td>CoModGAN[51]</td>
<td>0.980</td>
<td>DeepFillv2[46]</td>
<td>0.0307</td>
</tr>
<tr>
<td>LaMa[32]</td>
<td>159.6M</td>
<td>GraphFill-Pix (Ours)</td>
<td>175K + 4.4M</td>
</tr>
<tr>
<td>Pix2Pix[37] (Deep)</td>
<td>0.0308</td>
<td>GraphFill-Pix (Ours)</td>
<td>175K + 4.4M</td>
</tr>
<tr>
<td>CRFill[50]</td>
<td>0.0307</td>
<td>GraphFill-Pix (Ours)</td>
<td>175K + 4.4M</td>
</tr>
<tr>
<td>CoModGAN[51]</td>
<td>0.980</td>
<td>GraphFill-Pix (Ours)</td>
<td>175K + 4.4M</td>
</tr>
<tr>
<td>FFCResNet (Shallow)</td>
<td>3.8M</td>
<td>GraphFill-Pix (Ours)</td>
<td>175K + 4.4M</td>
</tr>
<tr>
<td>FFCResNet (Deep)</td>
<td>27M</td>
<td>GraphFill-Pix (Ours)</td>
<td>175K + 4.4M</td>
</tr>
<tr>
<td>ADGAN[9]</td>
<td>15.2M</td>
<td>GraphFill-Pix (Ours)</td>
<td>175K + 4.4M</td>
</tr>
</tbody>
</table>

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 [52] dataset.

<table>
<thead>
<tr>
<th>Model</th>
<th>FID ↑</th>
<th>LPIPS ↓</th>
<th>SSIM ↓</th>
</tr>
</thead>
<tbody>
<tr>
<td>GraphFill-Pix (Iterative) with RRPG</td>
<td>1.509</td>
<td>0.0301</td>
<td>0.981</td>
</tr>
<tr>
<td>GraphFill-Pix (Iterative) without RRPG</td>
<td>1.505</td>
<td>0.0298</td>
<td>0.980</td>
</tr>
<tr>
<td>GraphFill-Pix (Non-Iterative) with RRPG</td>
<td>1.719</td>
<td>0.0308</td>
<td>0.976</td>
</tr>
<tr>
<td>GraphFill-Pix (Non-Iterative) without RRPG</td>
<td>1.704</td>
<td>0.0307</td>
<td>0.979</td>
</tr>
</tbody>
</table>

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 \( n \in \mathbb{N} \) 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 \( \mathcal{C}_p \) 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 \( \mathcal{C}_{i−1} \) is obtained by applying the G2I layer on \( \mathcal{G}_{i−1} \) with the superpixel map \( S_{i−1} \). This coarse image is then cropped using the cropping parameters at the \( i \)-th level of the pyramid, resulting in \( \mathcal{C}^\cap_{i−1} \). The 12G-layer then transforms \( \mathcal{C}^\cap_{i−1} \) instead of \( \mathcal{C}_{i−1} \) to obtain a finer sub-graph \( \mathcal{G}^\cap_{i−1} \) corresponding to the \( i \)-th pyramid level, using the superpixel map \( S_i \). 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 \( \mathcal{C}^\cap_{i−1} \) at \( i \)-th level of pyramid. Figure 4(h) shows the averaged output of all \( \mathcal{C}^\cap_{i−1} \)'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 \( 256 \times 256 \) 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 \( 512 \times 512 \) 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 [11], the resulting graph has a total of \( N \leq (150, 600, 1700) \) nodes for \( 256 \times 256 \) resolution, and \( N \leq (150, 700, 1900) \) for \( 512 \times 512 \) resolution. The proposed coarser-to-finer approach for inpainting a masked image is demonstrated in Figure 5. Figures 5(a) and 5(b) 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 5(a) and first 3 rows in Figure 5(a) present results at \( 256 \times 256 \), while the remaining rows display results at \( 512 \times 512 \). The yellow outlined images in are generated at a higher resolution of \( 512 \times 512 \) by upscaling the corresponding image and mask due to the lack of inference support for \( 256 \times 256 \) 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 2, 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], DeepFill-v2 [46], GraphFill (Ours), LaMa [32], Pix2Pix [37], MAT [16], CoordFill [19], CoModGAN [51], and SHGAN [41].

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.

<table>
<thead>
<tr>
<th>Model</th>
<th>Places365 (512x512)</th>
<th>CelebA-HQ (512x512)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Narrow Masks</td>
<td>Medium Masks</td>
</tr>
<tr>
<td></td>
<td>FID ↓ LPIPS ↑ SSIM ↓</td>
<td>FID ↓ LPIPS ↑ SSIM ↓</td>
</tr>
<tr>
<td>GraphFill-Pix (Ours)</td>
<td>3.428 0.107 0.909</td>
<td>5.392 0.129 0.876</td>
</tr>
<tr>
<td>DeepFill-v2 [46]</td>
<td>3.569 0.106 0.906</td>
<td>6.636 0.113 0.870</td>
</tr>
<tr>
<td>CRFill [10]</td>
<td>3.461 0.102 0.911</td>
<td>5.458 0.127 0.874</td>
</tr>
<tr>
<td>AOTGAN [49]</td>
<td>4.472 0.127 0.882</td>
<td>6.013 0.128 0.866</td>
</tr>
<tr>
<td>CoordFill [19]</td>
<td>3.922 0.117 0.906</td>
<td>5.806 0.124 0.885</td>
</tr>
<tr>
<td>CoModGAN [51]</td>
<td>3.302 0.113 0.898</td>
<td>4.670 0.127 0.809</td>
</tr>
<tr>
<td>MAT [16]</td>
<td>2.486 0.091 0.915</td>
<td>4.056 0.112 0.887</td>
</tr>
<tr>
<td>SHGAN [41]</td>
<td>3.157 0.108 0.906</td>
<td>4.591 0.126 0.868</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Narrow Masks</td>
<td>Medium Masks</td>
</tr>
<tr>
<td></td>
<td>FID ↓ LPIPS ↑ SSIM ↓</td>
<td>FID ↓ LPIPS ↑ SSIM ↓</td>
</tr>
<tr>
<td>GraphFill-Pix (Ours)</td>
<td>4.782 0.102 0.919</td>
<td>5.061 0.101 0.899</td>
</tr>
<tr>
<td>DeepFill-v2 [46]</td>
<td>4.996 0.104 0.901</td>
<td>4.931 0.104 0.891</td>
</tr>
<tr>
<td>CRFill [10]</td>
<td>5.348 0.104 0.904</td>
<td>5.286 0.104 0.894</td>
</tr>
<tr>
<td>AOTGAN [49]</td>
<td>4.835 0.117 0.892</td>
<td>5.502 0.116 0.890</td>
</tr>
<tr>
<td>CoordFill [19]</td>
<td>3.873 0.089 0.915</td>
<td>4.14 0.092 0.905</td>
</tr>
<tr>
<td>LaMa [32]</td>
<td>3.455 0.086 0.912</td>
<td>3.349 0.088 0.903</td>
</tr>
<tr>
<td>MAT [16]</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>SHGAN [41]</td>
<td>3.712 0.100 0.917</td>
<td>3.789 0.101 0.883</td>
</tr>
</tbody>
</table>

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 ‘wo Masks’ 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.
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 $G'$ to the GraphFill Network with the adjacency matrix $A$ calculated from the connectivity information in $E$. The non-iterative graph-filling scheme does not involve the merger operation at every successive pyramid level. Instead, the output at every pyramid level is computed and averaged for the 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 on the CelebA-HQ dataset for the GraphFill Network with the adjacency matrix $A$ calculated from the connectivity information in $E$. The non-iterative graph-filling scheme does not involve the merger operation at every successive pyramid level. Instead, the output at every pyramid level is computed and averaged for the 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 on the CelebA-HQ dataset for the GraphFill Network with the adjacency matrix $A$ calculated from the connectivity information in $E$. The non-iterative graph-filling scheme does not involve the merger operation at every successive pyramid level. Instead, the output at every pyramid level is computed and averaged for the coarse to masked union operation needed before Refine Network. We quantitatively compare our coarse-to-finer inpainting variants in Table 4.

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

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 pyramid graph construction. Our approach outperforms existing methods having a similar number of learnable parameters and obtains competitive performance compared to existing heavyweight 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 Jiben Patel Chair in Artificial Intelligence. The authors extend their gratitude to Samsung R&D Institute for their support, resources, and invaluable insights.
References


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

