# DepthGen Lite: Efficient Inpainting with Dual-Gen and Depthwise Convolutions

Masa Abdallah Rosol Sharairh Tamam Alsarhan Ali Alrodan
Department of Artificial Intelligence, King Abdullah II School of
Information Techonology, The University of Jordan, Amman, Jordan
{mas0205277,rsl0203085,t_alsarhan,a.rodan}@ju.edu.jo

## Abstract

*Image inpainting modules are becoming more powerful, and with effective algorithms being released new horizons are open for limitless applications, however, these algorithms have high complexity. In this research, we propose DepthGen Lite, a lightweight module for image inpainting that can handle larger missing regions and complex textures, while producing visually consistent and realistic results. By utilizing depth-wise separable convolutions and efficient network architecture design, the proposed method achieves significant performance gains compared to state-of-the-art approaches. A Dual-Gen Depthwise Convolutions (DGDC) module is designed to capture both local and global contexts while maintaining efficient inference. Qualitative and quantitative experiments on the CelebA dataset demonstrate the superiority of the proposed method. Our code is available at* https://github.com/MasaAbdallah/DepthGen-AI.

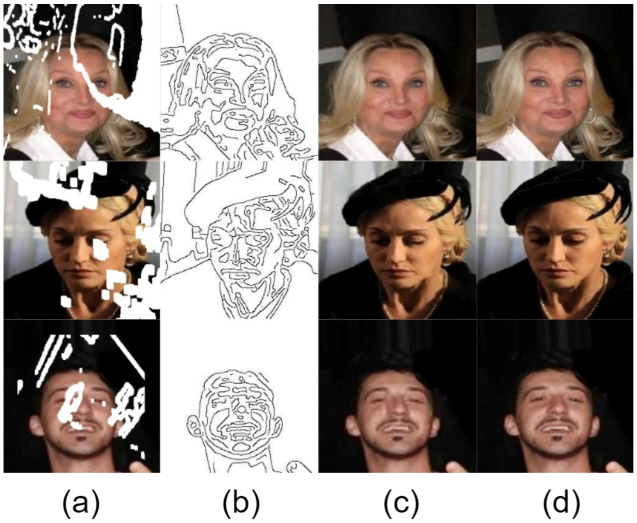

Figure 1. Our inpainting results. From left to right: (a) input corrupted images, (b) our reconstructed structures, (c) our filled results, and (d) ground-truth images.

## 1. Introduction

Image inpainting is a fundamental task in computer vision and image processing, with applications ranging from image restoration (Pillai and Khadagade 2017), image editing (Guillemot and Meur 2014), removal of the unwanted object (Lakshmanan and Gomathi 2017), image denoising, and much more [1]. Image inpainting aims to develop algorithms that effectively fill in missing or damaged regions of an image with plausible and visually consistent content [2]. The complexity of image inpainting is highly related to the need to carefully understand the image's structure, texture, and color data to be able to produce natural images. This space technique has evolved with advancements in deep learning, enabling the generation of realistic fillings for missing regions and the separation of objects not originally present in the image [3]. Traditional methods excel in reconstructing small patches but struggle with larger areas and complex images due to data limitations [4].

Recent advancements in deep learning, particularly deep Convolutional Neural Networks (CNN) and Generative Adversarial Networks (GAN) have indeed shown effectiveness in image inpainting tasks. Traditional methods often struggled to generate consistent and realistic results, whereas the use of CNNs and GANs has significantly improved the quality of inpainted images. CNN-based techniques are widely used for feature extraction and reconstruction, while GANs excel in generating visually appealing and coherent inpainted regions [5, 6]. Convolutional Neural Networks (CNNs) have improved image inpainting but often produce blurry results due to their reliance on Euclidean distance in loss functions. This blurriness occurs because the Euclidean distance minimizes loss by averaging outputs (Pathak, et al. 2016) and (Zhang, Isola and Efros 2016). To address this, Generative Adversarial Networks (GANs) are introduced, comparing real and generated images to create more realistic results (E. Denton, et al. 2015), (Radford, Metz and Chintala, Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks

2016), (Salimans, Goodfellow, et al., Improved Techniques for Training GANs 2016), and (Zhao, Mathieu and LeCun 2017).. GANs, inspired by various studies, solve the blurriness issue by using an adaptive adversarial loss function.

For example, in CTSDG [7], the authors propose a novel approach for image inpainting by integrating texture and structure features through a dual generation process. The proposed method aims to fill in missing parts of an incomplete image by generating texture and structure features that are refined and tailored to be both texture- and structure-aware. Pathak et al. [8] proposed a context encoder-decoder Generative Adversarial Network (GAN) that synthesizes more visually realistic results compared to traditional methods for image inpainting. To enhance the inpainting ability further, subsequent methods ( [9], [10], [11]) introduced the U-shaped encoder-decoder GAN. This architecture has superior generation capacity for complex structured images, enabling it to produce high-quality results in image inpainting tasks. Moreover, Zhang et al. [12] combined the self-attention with GAN, and improved the ability of network for image synthesis. The previously mentioned coarse-to-fine architectures are resource-demanding techniques, requiring e numerous resources which lead to high computational costs of the network (G. Wadhwa, A. Dhall, S. Murala, and U. Tariq, "Hyperrealistic image inpainting with hypergraphs," in Proc. IE)]. The existing methods succeeded in achieving effective outcomes. However, their computational complexity is high as shown in TABLE I. Considering these limitations of existing methods, we have proposed a light-weight architecture for image inpainting. To reduce the computational complexity, our proposed model is built by replacing the standard convolutions with depthwise separable convolutions [13]. A depthwise separable convolution model is a departure from earlier VGG-style architecture in 2014 [14] which were stacks of simple convolution layers. While depthwise separable convolution modules are conceptually similar to convolutions, they empirically appear to be capable of learning richer representations with fewer parameters.

## 2. Related Works

Our work relies essentially on prior efforts in the following areas:

- Convolutional neural networks [15–17], especially the VGG-16 architecture [14], which is used in CTSDG [7].

- The Inception architecture family of convolutional neural networks [18–21], which first showed the advantages of factoring convolutions into multiple branches operating successively on channels and then on space.

- Depthwise separable convolutions, which our proposed architecture is entirely based upon. While the use of spatially separable convolutions in neural networks has a long history, going back to at least 2012 [22]. Laurent Sifre developed depthwise separable convolutions during an internship at Google Brain in 2013, and used them in AlexNet to obtain small gains in accuracy and large gains in convergence speed, as well as a significant reduction in model size. An overview of his work was first made public in a presentation at ICLR 2014 [23]. Detailed experimental results are reported in Sifre's thesis, section 6.2 [24]. This initial work on depthwise separable convolutions was inspired by prior research from Sifre and Mallat on transformation-invariant scattering [24, 25]. Later, a depthwise separable convolution was used as the first layer of Inception V1 and Inception V2 [18, 19]. Within Google, Andrew Howard [26] has introduced efficient mobile models called MobileNets using depthwise separable convolutions. Jin et al. in 2014 [27] and Wang et al. in 2016 [28] also did related work aiming at reducing the size and computational cost of convolutional neural networks using separable convolutions. Additionally, our work is only possible due to the inclusion of an efficient implementation of depthwise separable convolutions in the TensorFlow framework [13]

## 3. Methodology

As illustrated in Figure 2. The suggested technique is implemented as a generative adversarial network, in which the two-stream generator simultaneously creates textures and structures, and the discriminator assesses the consistency and quality of them. We go into great detail about the discriminator and the generator in this section.

### 3.1. Generator

There are two streams in the generator's architecture, modeled by a U-Net variant, as shown in Figure 2 (a). The corrupted image and the associated edge map are separately projected into the latent space during the encoding stage, with the left branch concentrating on texture features and the right branch on structure features. During the decoding phase, the texture decoder generates structure-constrained textures by utilizing structure features obtained from the structure encoder. Simultaneously, the structure decoder reconstructs texture-guided structures by utilizing texture features obtained from the texture encoder. The combination of a dual-generation architecture results in a harmonious blend of structures and textures, ultimately enhancing the overall outcome.

In this encoder-decoder based backbone, partial convolution layers are used, to better capture information from

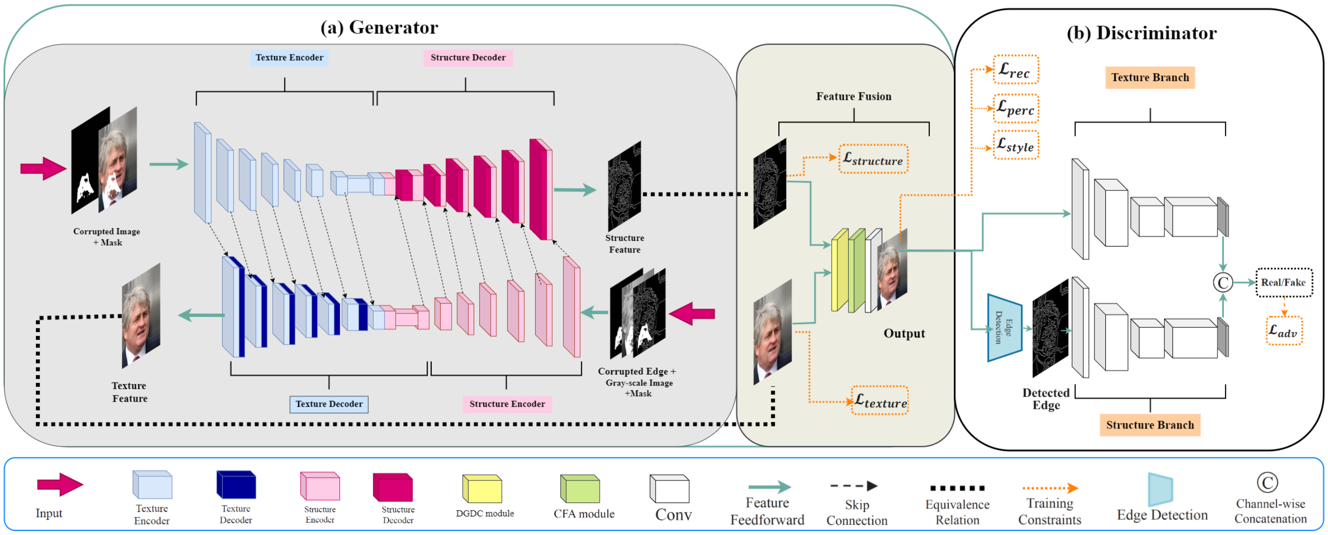

Figure 2. Overview of the proposed method. Generator: Image inpainting is divided into two parts, i.e., structure-restricted texture generation (left) and texture-driven structural reconstruction (right), additionally, the two parallel-coupled streams exchange encoded deep features. The Dual-Gen and Depthwise Convolutions (DGDC) module and Contextual Feature Aggregation (CFA) module are piled at the generator's end to further fine-tune the output. Discriminator: The image branch estimates the generated texture, while the edge branch guides structure reconstruction.

irregular boundaries, since partial convolutions are conditioned only on uncorrupted pixels. Additionally, skip connections are employed to generate more advanced predictions through the fusion of low-level and high-level features across various scales. In order to improve the uniformity of the reconstructed structures and textures, the feature maps produced by both branches are combined using a specially developed DGDC module and then processed through a CFA module to generate the ultimate outcome.

**Dual-Gen Depthwise Convolutions (DGDC).** This module aims to delicately and lightly integrate the decoded texture and structure features to a greater extent. Exchanging between the two types of data is facilitated through the use of soft gating to regulate the flow of information. As a result of this integration process, the features undergo enhancement while being aware of both texture and structure. Figure 3 illustrates the DGDC module.

Specifically, the texture feature map output by the decoder is denoted as Ft and the structure feature map is denoted as Fs. To build texture-aware structure features, a soft gating Gt, which controls to what extent the texture information is integrated, is formulated as:

$$G_t = \sigma(g(Concat(F_t, F_s))) \tag{1}$$

where Concat(·) is channel-wise concatenation, g(·) is the mapping function implemented by a depthwise separable convolution layer, illustrated in Figure 4, with a kernel size of 3, and $\sigma(.)$ is Sigmoid activation. With $G_t$, we adaptively merge $F_t$ into $F_s$ as:

$$F'_s = \alpha(G_t \odot F_t) \oplus F_s \tag{2}$$

where $\alpha$ is a training parameter initialized to zero, and $\odot$ and $\oplus$ denote element-wise multiplication and element-wise addition, respectively.

Symmetrically, we calculate the structure-aware texture feature $F'_t$ as follows:

$$G_s = \alpha(h(Concat(F_t, F_s))) \tag{3}$$

$$F'_t = \beta(G_s \odot F_s) \oplus F_t \tag{4}$$

where h follows the same pattern as g and $\beta$ is a training parameter initialized to zero as $\alpha$. Finally, we fuse $F'_s$ and $F'_t$ to obtain the integrated feature map $F_b$ by channel-wise concatenation:

$$F_b = Concat(F'_s, F'_t) \tag{5}$$

**Contextual Feature Aggregation (CFA).** To better learn which existing regions contribute to filling holes, this module is designed, to enhance the correlation between local features of an image and maintains the overall image consistency. It is inspired by [29], but unlike its fixed-scale patch matching scheme, in this study, multi-scale feature aggregation is adopted to encode rich semantic features at multiple scales so that it well balances the accuracy and complexity to handle more challenging cases, in particular, scale changes. The detailed process is depicted in Figure 5.

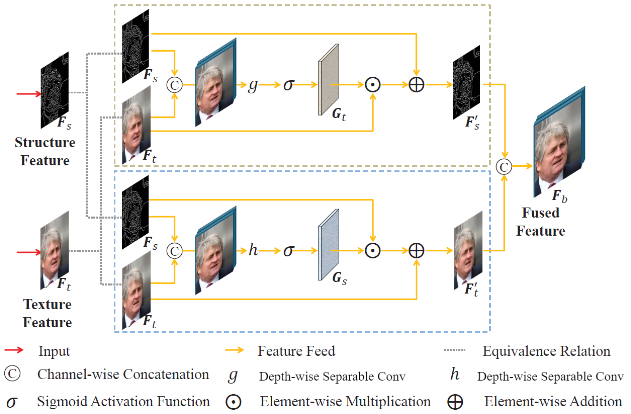

Figure 3. Illustration of the Dual-Gen Depthwise Convolutions (DGDC) module, which entangles the decoded structure and texture features to refine the results.

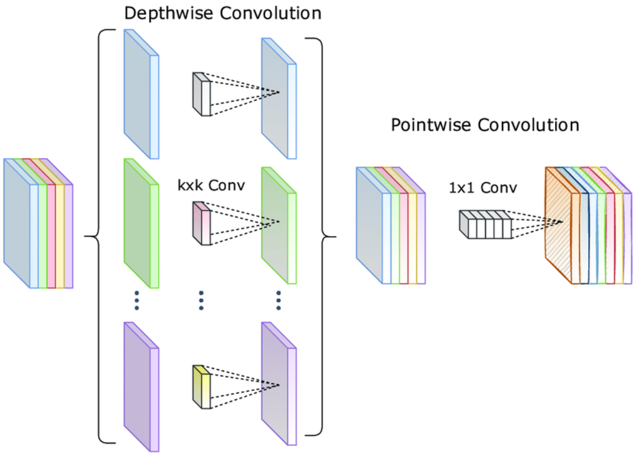

Figure 4. Illustration of Depth-wise separable convolutions ) module.

To be specific, given a feature map $F$, we first extract the patches of 3×3 pixels and calculate their cosine similarities as:

$$S^{i,j}_{contextual} = \left\langle \frac{f_i}{\|f_i\|_2} \cdot \frac{f_j}{\|f_j\|_2} \right\rangle \tag{6}$$

where $f_i$ and $f_j$ correspond to the $i$-th and $j$-th patch of the feature map, respectively.

We then apply softmax to the similarities to obtain the attention score of each patch:

$$S^{i,j}_{contextual} = \frac{exp\left(S^{i,j}_{contextual}\right)}{\sum_{j=1}^{N} exp\left(S^{i,j}_{contextual}\right)} \tag{7}$$

Next, the extracted patches are reused to reconstruct the feature map based on the attention map:

$$\tilde{f}_i = \sum_{j=1}^{N} f_j \cdot \hat{S}^{i,j}_{contextual} \tag{8}$$

where $\tilde{f}_i$ is the $i$-th patch of the reconstructed feature map Frec. The operations above are implemented as convolution, channel-wise softmax, and deconvolution, respectively.

When the feature map is reconstructed, four sets of dilated convolution layers with different dilation rates are used to capture multi-scale semantic features:

$$F^k_{rec} = Conv_k(F_{rec}) \tag{9}$$

where $Conv_k(\cdot)$ denotes dilated convolution layers with dilation rate of k, k $\in \{1, 2, 4, 8\}$

To better aggregate the multi-scale semantic features, we further design a pixel-level weight map generator Gw, which aims to predict the pixel-wise weight maps. In our implementation, $G_w$ consists of two convolution layers with the kernel size of 3 and 1, respectively, each of which is followed by ReLU non-linear activation, and the number of the output channels for $G_w$ is set to 4. The pixel-wise weight maps are calculated as:

$$W = \text{Softmax}(G_w(F_{rec})) \tag{10}$$

$$W^1, W^2, W^4, W^8 = \text{Slic}(W) \tag{11}$$

where Softmax($\cdot$) is channel-wise softmax and Slice($\cdot$) is channel-wise slice. Finally, the multi-scale semantic features are aggregated to produce the refined feature map Fc by element-wise weighted sum:

$$F_c = (F^1_{rec} \odot W^1) \oplus (F^2_{rec} \odot W^2) \oplus (F^4_{rec} \odot W^4) \oplus (F^8_{rec} \odot W^8) \tag{12}$$

Note, that the mask update mechanism of partial convolution layers are exploited, there is no need to distinguish the foreground and background pixels of the image as [29] does. Skip connection [30] is adopted to prevent semantic damage caused by patch-shift operations and a pair of convolution and deconvolution layers are seamlessly embedded into our architecture to improve computational efficiency.

## 3.2. Discriminator

Motivated by global and local GANs [31], Gated Convolution [10] and Markovian GANs [32], we develop a two-stream discriminator to distinguish genuine images from the generated ones by estimating the feature statistics of both texture and structure. The discriminator is shown in Figure 2 (b). The texture branch includes three convolution layers with a kernel size of 4 and stride of 2, tailed by two convolution layers with a kernel size of 4 and stride of 1. We use

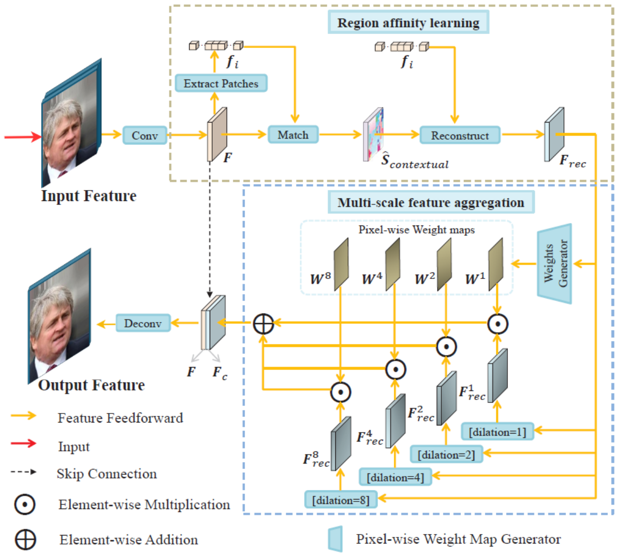

Figure 5. Illustration of the Contextual Feature Aggregation (CFA) module, which models long-term spatial dependency by capturing features at diverse semantic levels.

the Sigmoid non-linear activation function at the last layer and the Leaky ReLU with a slope of 0.2 for other layers. The structure branch shares the same pattern as the upper stream, where the input edge map is detected by a residual block [33] followed by a convolution layer with a kernel size of 1. Finally, the outputs of the two branches are concatenated in the channel dimension, based on which we calculate the adversarial loss.

Different from the case in the texture branch, it is intractable to optimize the adversarial loss of the structure branch only with the detected edge map, mainly due to the sparse nature of the edge. We therefore adopt the gray scale image as an additional condition and feed the paired data as the input in the structure branch, as several previous studies do [34,35]. As such, the structure discriminator not only estimates the authenticity of the generated structure but also guarantees its consistency with the ground-truth image. Besides, spectral normalization [36] is used, as it proves effective in solving the well-known training instability problem of generative adversarial networks.

# 4. Experiments

We conducted extensive experiments on the following datasets to evaluate the performance of our proposed method.

## 4.1. Datasets

### 4.1.1 Image Dataset

We evaluate the proposed method on the CelebA dataset [37], which is widely adopted in the literature for various facial recognition and image processing tasks. The CelebA dataset consists of more than 200,000 celebrity images. This dataset is particularly valuable for image inpainting due to its diversity and large quantity of high-resolution images

To ensure consistency and comparability with previous works, we follow the original training, testing, and validation splits of the CelebA dataset.

### 4.1.2 Mask Dataset

For generating irregular masks, we used the Irregular Masks dataset, as referenced in [38]. These masks are crucial for simulating realistic image inpainting scenarios where regions of varying sizes and shapes are missing.

### 4.1.3 Data Preparation

To prepare the images and masks for our inpainting task, the following preprocessing steps were carried out:

- **Image and Mask Resizing**: All images and their corresponding masks were resized to 256 × 256 pixels. This standardization ensures uniform input sizes for the model, facilitating efficient training and evaluation.
- **Mask Classification**: The irregular masks were categorized based on the proportion of the image they occlude, with hole sizes increasing in 10% increments. This classification helps in assessing the model's performance under varying degrees of image corruption.

By adhering to the original training, testing, and validation splits of the CelebA dataset and utilizing the Irregular Masks dataset, we ensured a robust and comprehensive evaluation of our proposed image inpainting method. This approach enables our model to effectively handle various levels of occlusion, preserving fine details and textures in the inpainted regions.

## 4.2. Experimental Settings

### 4.2.1 Hyperparameters

The model was implemented in PyTorch 2.2.2+cu121 and optimized using the Adam optimizer.. Our experiments utilized the following hyperparameters: 4 workers, a batch size of 1, and a load size of 256 × 256 pixels. Additional parameters included a sigma value of 2.0, a generator learning rate of 0.0002, and a discriminator to generator learning rate ratio of 0.1. The weights for various loss components were set as follows: validation loss at 10.0, hole loss at 60.0, perceptual loss at 0.1, style loss at 250.0, adversarial loss at 0.1, and intermediate loss at 1.0. The local rank was set to 0.

#### 4.2.2 Hardware and Software Environment

The hardware environment for our experiments included an NVIDIA GeForce RTX 3050Ti Laptop GPU, an 11th Gen Intel(R) Core(TM) i7-11370H CPU running at 3.30GHz, 16GB of RAM, and the system operated on Microsoft Windows 11 Pro (x64-based PC).

## 5. Results and Discussion

In this section, we present a detailed analysis of the performance of our inpainting model.

### 5.1. Quantitative Comparison

To comprehensively evaluate the performance of our inpainting model, we employed a combination of quantitative metrics including Peak Signal-to-Noise Ratio (PSNR) [39], Structural Similarity Index (SSIM) [40], and Learned Perceptual Image Patch Similarity (LPIPS) [41].

PSNR measures the quality of the reconstructed image compared to the original image, with higher values indicating better quality and less distortion. SSIM assesses the structural similarity between the inpainted and original images, where values closer to 1 indicate higher similarity and better preservation of structural details. LPIPS evaluates the perceptual similarity between the original and inpainted images using deep learning-based features, with lower values indicating better perceptual similarity and visual fidelity.

Table 1 shows the quantitative comparison of our model with state-of-the-art models on the CelebA dataset. Our model achieves outstanding results in terms of PSNR, SSIM, and LPIPS, indicating high-quality inpainting performance that preserves both the structural and perceptual fidelity of the images. Notably, our model outperforms all other models in terms of SSIM, demonstrating superior structural similarity and higher quality reconstructions.

### 5.2. Performance Comparison

To measure the efficiency and scalability of our model, we used the following performance metrics: GPU Memory Usage per Batch and FLOPs (Floating Point Operations per Second).

GPU Memory Usage per Batch measures the amount of GPU memory utilized during the processing of a batch, where lower memory usage indicates higher efficiency, allowing the model to run on hardware with limited resources. FLOPs indicate the computational complexity of the model, with lower FLOPs suggesting a more efficient model in terms of computation, translating to faster processing and lower energy consumption.

Table 2 presents the complexity comparison between our model and the CTSDG [7] model. Our model demonstrates superior efficiency, evidenced by lower GPU memory usage and fewer FLOPs. These metrics underscore the practicality and scalability of our approach.

### 5.3. Performance Highlights

Our model excels in several key areas, showcasing its remarkable efficiency and effectiveness:

- **Inference Time per Batch**: Our model reduces the inference time by approximately 4.36% compared to the CTSDG [7] model. This substantial reduction allows for quicker processing of images, making our model highly suitable for real-time applications where speed is crucial.

- **GPU Memory Usage per Batch**: Our model uses slightly less GPU memory per batch compared to the CTSDG [7] model, enhancing its ability to run effectively on hardware with limited resources. This efficiency makes our model accessible for a wider range of deployment scenarios, including edge devices and consumer-grade hardware.

- **FLOPs**: Our model achieves a significantly lower computational complexity (83.4781 GFLOPs) compared to the CTSDG [7] model (91.9171 GFLOPs). This remarkable reduction in complexity translates to faster computations and lower energy consumption, demonstrating the superior efficiency of our approach.

### 5.4. Qualitative Comparison

In Figure 6, we provide a qualitative comparison between the inpainting results of the CTSDG model and our proposed model on the CelebA dataset. Column 1 shows the ground truth images, Column 2 shows the masked images, Column 3 displays the edge maps used for guiding the inpainting process, Column 4 presents the results from the CTSDG model, and Column 5 shows the results from our proposed model. The edge maps were generated using the same method for both models, ensuring a fair comparison. As evident from the figure, our model preserves finer details and generates more visually consistent inpainting results compared to the CTSDG model. The visuals from both models are very close and nearly identical in quality; however, our model achieves these results with a less complex architecture, highlighting the efficiency and effectiveness of our approach.

Overall, these improvements highlight the robustness and practicality of our model. It provides high-quality inpainting results while maintaining exceptional efficiency, striking a perfect balance between performance and resource usage. Our approach is not only competitive with

| Metrics | LPIPS† | | | PSNR¶ | | | SSIM¶ | | |
|---|---|---|---|---|---|---|---|---|---|
| Mask Ratio | 0-20% | 20-40% | 40-60% | 0-20% | 20-40% | 40-60% | 0-20% | 20-40% | 40-60% |
| PatchMatch [42] | 0.059 | 0.202 | 0.371 | 29.81 | 23.49 | 18.77 | 0.878 | 0.704 | 0.516 |
| PConv [38] | 0.046 | 0.122 | 0.221 | 31.89 | 26.48 | 21.32 | 0.899 | 0.750 | 0.558 |
| DeepFillv2 [10] | 0.040 | 0.107 | 0.214 | 32.48 | 26.93 | 21.70 | 0.906 | 0.757 | 0.569 |
| RFR [43] | 0.031 | 0.090 | 0.185 | 33.50 | 27.63 | 22.69 | 0.916 | 0.780 | 0.603 |
| EdgeConnect [35] | 0.042 | 0.117 | 0.215 | 32.12 | 26.79 | 21.66 | 0.904 | 0.758 | 0.566 |
| PRVS [44] | 0.039 | 0.112 | 0.209 | 32.34 | 26.89 | 21.78 | 0.908 | 0.762 | 0.573 |
| MED [45] | 0.037 | 0.106 | 0.203 | 32.68 | 27.01 | 21.86 | 0.907 | 0.763 | 0.575 |
| CTSDG [7] | **0.028** | **0.081** | **0.179** | **33.91** | **27.73** | **22.70** | 0.920 | 0.788 | 0.609 |
| Ours | 0.032 | 0.093 | 0.1888 | 31.947 | 26.463 | 21.91 | **0.936** | **0.862** | **0.743** |

Table 1. Objective quantitative comparison on CelebA (†Lower is better; ¶Higher is better).

| Model | GPU Memory Usage per Batch (MB)† | FLOPs (GFLOPs)† |
|---|---|---|
| CTSDG [7] | 871.6295 | 91.9171 |
| Ours | **871.1383** | **83.4781** |

Table 2. Complexity comparison of different models on CelebA (†Lower is better).

state-of-the-art models but also sets a new standard for computational efficiency and practical applicability in real-world scenarios.

# 6. Limitations

While our proposed model demonstrates significant advancements in image inpainting, there are several limitations that need to be acknowledged.

## 6.1. Scope and Generalizability

Our study primarily evaluates the proposed method on the CelebA dataset, which is widely adopted in the literature. However, the generalizability of our results to other datasets or real-world applications remains to be fully validated. Future work should explore the performance of our model on a broader range of datasets, including those with more diverse image types and complexities.

## 6.2. Time and Resource Constraints

We had approximately five months to complete this project and were limited by the available computational resources. These constraints restricted the extent of our experimentation and optimization, potentially limiting the overall performance of the model. More extensive experimentation with different architectures and hyperparameters might yield further improvements. The limited computational resources also impacted our ability to conduct large-scale tests and fine-tune the model extensively.

# 7. Future Work

To address these limitations, future research could focus on:

## 7.1. Extending Evaluation and Enhancing Robustness

Extending the evaluation to a wider range of datasets and enhancing the model's ability to handle more complex and diverse image types, ensuring robust performance across various application scenarios.

## 7.2. Reducing Computational Requirements and Extensive Experimentation

Developing techniques to reduce computational requirements and conducting more extensive experimentation and optimization, enabling deployment on more resource-constrained devices and improving model performance.

By acknowledging these limitations, we aim to provide a comprehensive and balanced perspective on our research, laying the groundwork for future improvements and investigations.

# 8. Conclusion

In this study, we presented a novel image inpainting method that leverages both channel and pixel attention mechanisms to enhance the quality of inpainted images. Our primary contribution lies in achieving a remarkable balance between computational efficiency and high accuracy.

Our model was rigorously evaluated on the CelebA dataset, demonstrating outstanding performance. Quantita-

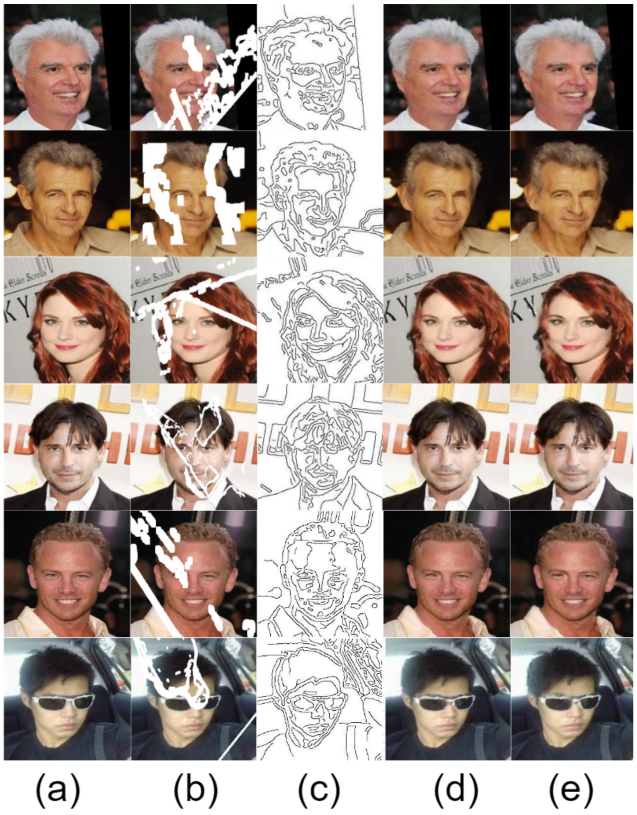

Figure 6. Qualitative comparison of inpainting results. (From left to right) (a) Ground truth images, (b) masked images, (c) edge maps, (d) results from the CTSDG model, and (e) results from our proposed model. The edge maps used to guide the inpainting process were generated similarly for both models. Despite the close visual similarity between the results of the two models, our proposed model achieves these high-quality results with significantly lower computational complexity.

tively, our method achieved competitive results across multiple metrics, particularly excelling in Structural Similarity Index (SSIM), which underscores the model's superior ability to preserve structural consistency and details in the inpainted regions.

Moreover, one of the significant highlights of our work is the reduction in computational complexity. Our model requires fewer floating point operations (FLOPs) and uses less GPU memory per batch compared to the CTSDG [7] model. Specifically, our model reduces the inference time by approximately 4.36%, indicating its suitability for real-time applications and deployment in resource-constrained environments.

Despite these advancements, we acknowledge certain limitations, such as the scope of dataset evaluation and constraints due to limited computational resources and time. Future work will focus on extending the evaluation to a broader range of datasets, reducing computational requirements further, and conducting more extensive experimentation to enhance model performance and robustness.

In conclusion, our proposed method represents a significant advancement in the field of image inpainting, providing a model that is not only computationally efficient but also highly effective in preserving image quality. These contributions pave the way for future innovations and improvements in image restoration technologies, making our approach a valuable addition to the existing body of research.

## References

[1] Nermin M Salem. A survey on various image inpainting techniques. *Future Engineering Journal*, 2(2), 2021. 1

[2] Marcelo Bertalmío, Guillermo Sapiro, Vicent Caselles, and Coloma Ballester. Image inpainting. In Judith R. Brown and Kurt Akeley, editors, *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 2000, New Orleans, LA, USA, July 23-28, 2000*, pages 417–424. ACM, 2000. 1

[3] Zhen Qin, Qingliang Zeng, Yixin Zong, and Fan Xu. Image inpainting based on deep learning: A review. *Displays*, 69:102028, 2021. 1

[4] Soureesh Patil, Amit Joshi, and Suraj Sawant. Recovering images using image inpainting techniques. In Hariharan Muthusamy, János Botzheim, and Richi Nayak, editors, *Robotics, Control and Computer Vision*, pages 27–38, Singapore, 2023. Springer Nature Singapore. 1

[5] Yi Jiang, Jiajie Xu, Baoqing Yang, Jing Xu, and Junwu Zhu. Image inpainting based on generative adversarial networks. *IEEE Access*, 8:22884–22892, 2020. 1

[6] Xiaobo Zhang, Donghai Zhai, Tianrui Li, Yuxin Zhou, and Yang Lin. Image inpainting based on deep learning: A review. *Information Fusion*, 90:74–94, 2023. 1

[7] Xiefan Guo, Hongyu Yang, and Di Huang. Image inpainting via conditional texture and structure dual generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14134–14143, 2021. 2, 6, 7, 8

[8] 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. 2

[9] Jireh Jam, Connah Kendrick, Vincent Drouard, Kevin Walker, Gee-Sern Hsu, and Moi Hoon Yap. R-mnet: A perceptual adversarial network for image inpainting. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2714–2723, 2021. 2

[10] 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 (ICCV)*, pages 4471–4480, 2019. 2, 4, 7

[11] Tao Yu, Zongyu Guo, Xin Jin, Shilin Wu, Zhibo Chen, Weiping Li, Zhizheng Zhang, and Sen Liu. Region normalization for image inpainting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 12733–12740, 2020. 2

[12] Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. Self-attention generative adversarial networks. In *International conference on machine learning*, pages 7354–7363. PMLR, 2019. 2

[13] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, et al. Tensorflow: Large-scale machine learning on heterogeneous systems, 2015. 2

[14] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 2

[15] Yann LeCun, Lawrence D Jackel, Léon Bottou, Corinna Cortes, John S Denker, Harris Drucker, Isabelle Guyon, Urs A Muller, Eduard Sackinger, Patrice Simard, et al. Learning algorithms for classification: A comparison on handwritten digit recognition. *Neural networks: the statistical mechanics perspective*, 261(276):2, 1995. 2

[16] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012. 2

[17] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I 13*, pages 818–833. Springer, 2014. 2

[18] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015. 2

[19] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. pmlr, 2015. 2

[20] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016. 2

[21] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31, 2017. 2

[22] Alessandro Villa, Włodzisław Duch, Péter Érdi, Francesco Masulli, and Günther Palm. *Artificial Neural Networks and Machine Learning–ICANN 2012: 22nd International Conference on Artificial Neural Networks, Lausanne, Switzerland, September 11-14, 2012, Proceedings, Part II*, volume 7553. Springer, 2012. 2

[23] Vincent Vanhoucke. Learning visual representations at scale. *ICLR invited talk*, 1(2), 2014. 2

[24] Sifre Laurent. Rigid-motion scattering for image classification. *Ph. D. thesis section*, 6(2), 2014. 2

[25] Laurent Sifre and Stéphane Mallat. Rotation, scaling and deformation invariant scattering for texture discrimination. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1233–1240, 2013. 2

[26] G Andrew and Z Menglong. Efficient convolutional neural networks for mobile vision applications, mobilenets. *arXiv preprint arXiv:1704.04861*, 2017. 2

[27] Jonghoon Jin, Aysegul Dundar, and Eugenio Culurciello. Flattened convolutional neural networks for feedforward acceleration. *arXiv preprint arXiv:1412.5474*, 2014. 2

[28] Min Wang, Baoyuan Liu, and Hassan Foroosh. Design of efficient convolutional layers using single intra-channel convolution, topological subdivisioning and spatial" bottleneck" structure. *arXiv preprint arXiv:1608.04337*, 2016. 2

[29] 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. 3, 4

[30] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, pages 234–241. Springer, 2015. 4

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

[32] Chuan Li and Michael Wand. Precomputed real-time texture synthesis with markovian generative adversarial networks. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part III 14*, pages 702–716. Springer, 2016. 4

[33] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 5

[34] 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. 5

[35] Kamyar Nazeri, Eric Ng, Tony Joseph, Faisal Qureshi, and Mehran Ebrahimi. Edgeconnect: Structure guided image

inpainting using edge prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, pages 0–0, 2019. 5, 7

[36] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. *arXiv preprint arXiv:1802.05957*, 2018. 5

[37] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 3730–3738, 2015. 5

[38] 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. 5, 7

[39] Que Huynh-Thu and Mohamed Ghanbari. Scope of validity of psnr in image/video quality assessment. *Electronics letters*, 44(13):800–801, 2008. 6

[40] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004. 6

[41] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 586–595, 2018. 6

[42] Connelly Barnes, Eli Shechtman, Adam Finkelstein, and Dan B Goldman. Patchmatch: A randomized correspondence algorithm for structural image editing. *ACM Transactions on Graphics (TOG)*, 28(3):24, 2009. 7

[43] Jingyuan Li, Ning Wang, Lefei Zhang, Bo Du, and Dacheng Tao. Recurrent feature reasoning for image inpainting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7760–7768, 2020. 7

[44] Jingyuan Li, Fengxiang He, Lefei Zhang, Bo Du, and Dacheng Tao. Progressive reconstruction of visual structure for image inpainting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 7760–7768, 2019. 7

[45] Hongyu Liu, Bin Jiang, Yibing Song, Wei Huang, and Chao Yang. Rethinking image inpainting via a mutual encoder-decoder with feature equalizations. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 725–742, 2020. 7