# Deep Generative Architectures for Image Inpainting

Thesis Report Submitted to
Indian Institute of Technology Ropar
in partial fulfilment of the requirements for the
Degree of

## DOCTOR OF PHILOSOPHY

By

**Shruti Shantiling Phutke**
(Reg.No. 2018eez0019)

Under the guidance of

**Dr. Subrahmanyam Murala**

**Department of Electrical Engineering,**

**Indian Institute of Technology Ropar**

**Rupnagar-140001, Punjab, India**

**2022-23**

**December-2022**

Shruti Shantiling Phutke: *DEEP GENERATIVE ARCHITECTURES FOR IMAGE INPAINTING*

**Dedicated to My Beloved Parents**

- Who are the inspiration and power behind success of this work

# Declaration of Originality

I hereby declare that the work which is being presented in the thesis entitled **DEEP GENERATIVE ARCHITECTURES FOR IMAGE INPAINTING** has been solely authored by me. It presents the result of my own independent research conducted during the time period from JANUARY-2019 to DECEMBER-2022 under the supervision of Dr. Subrahmanyam Murala, Associate Professor, Department of Electrical Engineering. To the best of my knowledge, it is an original work, both in terms of research content and narrative, and has not been submitted or accepted elsewhere, in part or in full, for the award of any degree, diploma, fellowship, associateship, or similar title of any university or institution. Further, due credit has been attributed with appropriate citations and acknowledgments, in line with established ethical norms and practices. I also declare that any idea/data/fact/source stated in my thesis has not been fabricated/ falsified/ misrepresented. All the principles of academic honesty and integrity have been followed. I fully understand that if the thesis is found to be unoriginal, fabricated, or plagiarized, the Institute reserves the right to withdraw the thesis from its archive and revoke the associated Degree conferred. Additionally, the Institute also reserves the right to appraise all concerned sections of society of the matter for their information and necessary action (if any). If accepted, I hereby consent for my thesis to be available online in the Institute's Open Access repository, inter-library loan, and the title & abstract to be made available to outside organizations.

S.S Phutke

Signature

Name: Shruti Shantiling Phutke
Entry Number: 2018eez0019
Program: Doctor of Philosophy (Ph.D.)
Department: Electrical Engineering
Indian Institute of Technology Ropar
Rupnagar, Punjab 140001

Date: 19 December 2022

# Acknowledgement

The journey of Ph.D. has been a truly metamorphosis experience, bringing me excitement and fulfilment. I have contributed my bit through all possible efforts in the oceans of knowledge. I would like to take this opportunity to thank all the people who have been part of this wonderful journey. The support, guidance, and every skill that I learned through this journey will always be cherished by me.

No words are sufficient to express my sincere gratitude to my guide, **Dr. Subrahmanyam Murala**, for continuous encouragement, technical discussions, and invaluable guidance during my Ph.D. It has been a great honour and privilege to work under his supervision and to be his student.

I take this opportunity to convey my sincere thanks to **Prof. Rajeev Ahuja**, *Director of Indian Institute of Technology Ropar, Punjab, INDIA*, for his all round support, whole-hearted cooperation during the Ph.D journey. I am grateful to the **Head of the Electrical Engineering Department**, for his encouraging and supportive attitude. I also extend my genuine thanks to doctoral committee chair **Prof. J. S. Sahambi**, and doctoral committee members **Dr. Satyam Agarwal**, **Dr. Mukesh Saini**, **Dr. Ashish Sahani** for their insightful comments and encouragement which incented me to widen my research from various perspectives.

My special thanks and appreciation goes to **Prof. Anil Balaji Gonde**, Professor, SGGSIET, Nanded, Maharashtra and **Dr. Santosh Kumar Vipparthi**, Assistant Professor, IIT Ropar for providing valuable guidance and moral support.

A special thank to **Dr. Sachin Chaudhary**, NTU Singapore, **Dr. Akshay Dudhane**, MBZUAI Abu Dhabi, and **Dr. Prashant Patil**, Deakin University Australia for valuable guidance and support during my Ph.D.

I thank my colleagues **Dr. Praful Hambarde**, **Miss. Nancy Mehta**, **Mr. Jasdeep Singh**, **Mr. Ashutosh Kulkarni**, **Miss. Saniya Bano**, **Miss. Priyanka Mishra**, **Mr. Devendra Deshmukh**, **Miss. Shubhangi Nema**, **Mr. Chetan Gupta**, **Mr. Raqib Khan**, and **Mr. Siddharth Singh** for constant encouragement. Also, I would like to thank my friends **Samiksha Pachade**, **Manisha Patil** and **Sharda Sawant** for encouraging me throughout this research journey.

Finally, I would like to thank my backbone - my family for their trust, patience, unparalleled love, encouragement and support throughout this journey. A special thanks to my brother for always being there for me as a friend. Above all, I praise and thank the **Almighty Lord Mahadeva** and **Ganpati Bappa** for blessing me with success in this research work.

December 2022.
Place : IIT Ropar, Punjab, India. Shruti Shantiling Phutke

DEPARTMENT OF ELECTRICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY ROPAR
RUPNAGAR-140001, INDIA

# Certificate

This is to certify that the thesis entitled **DEEP GENERATIVE ARCHITECTURES FOR IMAGE INPAINTING**, submitted by **Shruti Shantiling Phutke (2018eez0019)** for the award of the degree of **Doctor of Philosophy** of Indian Institute of Technology Ropar, Punjab, INDIA, is a record of bonafide research work carried out under my guidance and supervision during 2019-22. To the best of my knowledge and belief, the work presented in this thesis is original and has not been submitted, either in part or full, for the award of any other degree, diploma, fellowship, associateship or similar title of any university or institution.

In my opinion, the thesis has reached the standard fulfilling the requirements of the regulations relating to the degree.

Signature

Dr. Subrahmanyam Murala
Associate Professor
Department of Electrical Engineering
Indian Institute of Technology Ropar
Rupnagar, Punjab 140001
Date: 19 December 2022

# Lay Summary

Inpainting is a task of completing the image which has some corrupted regions. Suppose one has an image which is taken a long-time back and it is corrupted due to some scratches. Image inpainting helps to recover this corrupted image. Also, if you are having a photograph containing some unwanted portion or object and you want to remove this unwanted object, image inpainting comes in handy. It removes the unwanted objects in an image.

The image inpainting methods complete the corrupted image in the way how a painter paints the incomplete image. Image inpainting methods generally use different ways to complete the corrupted image. Some methods first try to consider the global view of the corrupted image. Then they finely complete these globally completed image. The other methods try to directly complete the image in one take itself. In this, some paint the image from outermost part to innermost part of the corrupted region. Some methods take extra information from the users to inpaint the image. All these methods sometimes fail at painting the corrupted images effectively by generating some blurry results. Apart from this, the methods which inpaint the corrupted images effectively, they take too much time to generate the results.

In this work, we propose different methods for image inpainting. Some of our proposed methods adapt global completion followed by fine completion of corrupted image. The other methods directly complete the images. Our proposed methods give visually plausible results with less time as compared to the existing methods.

# Abstract

Image inpainting is a reconstruction method, where a corrupted image consisting of holes is filled with the most relevant contents from the valid region of same image. With the advancements in image editing applications, image inpainting is gaining more attention due to its ability to recover corrupted images efficiently. Also, it has a wide variety of applications such as reconstruction of the corrupted image, occlusion removal, reflection removal, *etc.* Existing approaches achieved superior performance with coarse-to-fine, single-stage, progressive, and recurrent architectures with a compromise of either perceptual quality (blurry, spatial inconsistencies) of results or computational complexity. Also, the performance of the existing methods degrades when images with large missing regions are considered. In order to mitigate these limitations, in this work, we propose the *deep generative architectures for image inpainting.*

Firstly, we propose the coarse-to-fine architectures for inpainting images with varying corrupted regions with improved performance as compared to state-of-the-art methods. The three proposed coarse-to-fine solutions consist of: (a) a spatial projection layer to focus on spatial consistencies in the inpainted image, (b) encoder-level feature aggregation followed by multi-scale and multi-receptive feature sharing decoder, and (c) a nested deformable multi-head attention layer to effectively merge the encoder-decoder features.

Further, to reduce the computational complexity, we proposed single-stage architectures with three solutions as: (a) a correlated multi-resolution feature fusion, (b) diverse-receptive fields based feature learning, and (c) pseudo-decoder guided reconstruction for image inpainting. The proposed architectures have less computational complexity compared to earlier one and state-of-the-art methods for image inpainting. The performance of these proposed architectures is validated in terms of qualitative, quantitative results and computational complexity in comparison with each other and existing methods for image inpainting.

Furthermore, to reduce the mask dependency of the proposed and existing approaches, we propose two novel blind image inpainting approaches consisting of (a) wavelet query multi-head attention transformer and omni-dimensional gated attention (b) high receptive fields (multi-kernel) multi-head attention and novel high-frequency offset deformable feature merging module. These proposed approaches is compared qualitatively and quantitatively with existing state-of-the-art methods for blind image inpainting. To validate the performance of the proposed architectures, the experimental analysis is done on different datasets like: CelebA-HQ, FFHQ, Paris Street View, Places2 and Imagenet.

**Keywords**: Feature Aggregation; Spatial Projections; Multi-head Attention; Diverse Receptive Fields; Image Inpainting; Blind Image Inpainting.

# Abbreviations

| | |
|---|---|
| CA | Contextual Attention |
| CNNs | Convolutional Neural Networks |
| CSA | Coherent Semantic Attention |
| DFS | Decoder Feature Sharing |
| DRB | Diverse Receptive fields Block |
| DHMA | Deformable Multi-head Attention |
| EC | Edge-connect |
| EEB | Edge Extraction Block |
| ERS | Edge Refinement Stage |
| EMLFF | Encoder Multi-level Feature Fusion |
| FAB | Feature Aggregation Block |
| FFN | Feed Forward Network |
| FFHQ | Flicker Faces High Quality |
| FID | Fréchet Inception Distance |
| FP | Feature Projection |
| GANs | Generative Adversarial Networks |
| GELU | Gated Error Linear Unit |
| GFLOPs | Giga Floating Point Operations/Second |
| GMAC | Giga Multiply-accumulate Operations |
| $H_fADM$ | High-frequency Attentive Deformable Merging |
| LPIPS | Learned Perceptual Image Patch Similarity |
| LUNA | Linear Unified Nested Attention |
| MHA | Multi-head Attention |
| MKDC | Multi-kernel Depth-wise Separable Convolution |
| MKMA | Multi-kernel Multi-head Attention |
| MKNL | Multi-kernel Non-local Attention |
| MLP | Multi Layer Perceptron |
| $M_sSCA$ | Multi-scale Spatial Channel-wise Attention |
| NDMAL | Nested Deformable Multi-head Attention Layer |
| NL | Non-local Attention |
| PSNR | Peak Signal-to-noise Ratio |
| RDRB | Residual Diverse Receptive fields Block |
| RB | Residual Block |
| ReLU | Rectified Linear Unit |
| SA | Self-attention |
| SDC | Space-depth Correlation |
| SPL | Spatial Projection Layer |
| SSIM | Structural Similarity Index Measure |
| VFF | Valid Feature Fusion |
| VRF | Varying Receptive Fields |
| w/o | Without |
| w/i | With |
| ↑ | Higher is better |
| ↓ | Lower is better |

# Contents

# List of Figures

# List of Tables

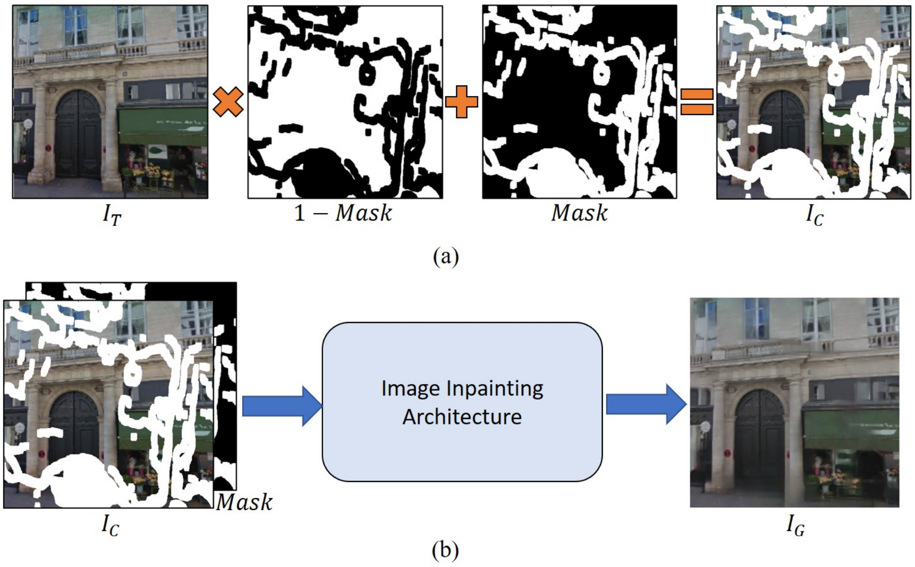
# Chapter 1

# Introduction

In this chapter, the introduction, motivation, and applications of image inpainting task are provided. Section 1.1 introduces the formation of corrupted image and the generalized flow of image inpainting. Different applications of image inpainting task are detailed in Section 1.2. Section 1.3 gives the motivation of the proposed work. The identified problems in the image inpainting task are detailed in Section 1.4. Section 1.5 defines the aims and objectives of the proposed work. The main contributions are provided in Section 1.6. Finally, Section 1.7 provides the overall thesis structure.

## 1.1 Introduction

Image inpainting is an image restoration task where the corrupted image with missing content (hole) is synthesized with the information from available (non-hole) regions. This is a one to multi-solution kind of task where the hole regions can be filled with distinct content relevant with the non-hole region. The corrupted image ($I_C$) for the task of image inpainting is generated as:



(a)

(b)

Figure 1.1: (a) Formation of corrupted image ($I_c$), (b) Image inpainting architecture.

$$I_C = I_T \cdot (1 - I_M) + I_M \tag{1.1}$$

where, $I_T$ is the target image, $I_M$ is the mask image with 1 (white region) indicating missing regions and 0 (black region) indicating available regions.

The formation of corrupted image ($I_C$) is shown in Figure 1.1. The aim is to inpaint the image with the semantically valid contents or generating visually plausible contents in the image by removing unwanted or corrupted content. Here, the inpainted/restored image ($I_I$) is obtained from generated image ($I_G$) by the inpainting method. The $I_I$ can be obtained as:

$$I_I = I_G \cdot I_M + I_C \cdot (1 - I_M) \tag{1.2}$$

Image inpainting task is categorized in two types, blind and non-blind. The non-blind means, the inpainting architectures are aware with the corrupted locations in the image by taking mask as input. Whereas, blind image inpainting methods are the ones which do not depend on any kind of information regarding the corrupted locations. Unlike normal image inpainting, blind image inpainting expects the corrupted input with the image having some noise blended with the actual image. The formation of corrupted image for the task of blind image inpainting is given as:

$$I_C = I_T \cdot (1 - I_M) + I_M \cdot Noise \tag{1.3}$$

where, $Noise$ is any noisy input like graffiti, different images, *etc.* The sample images considered for the task of blind image inpainting are shown in Figure 1.2. Also, the region of corruption may vary in size, shape and location as shown in Figure 1.2 (for blind inpainting) and Figure 1.3 (for non-blind inpainting).

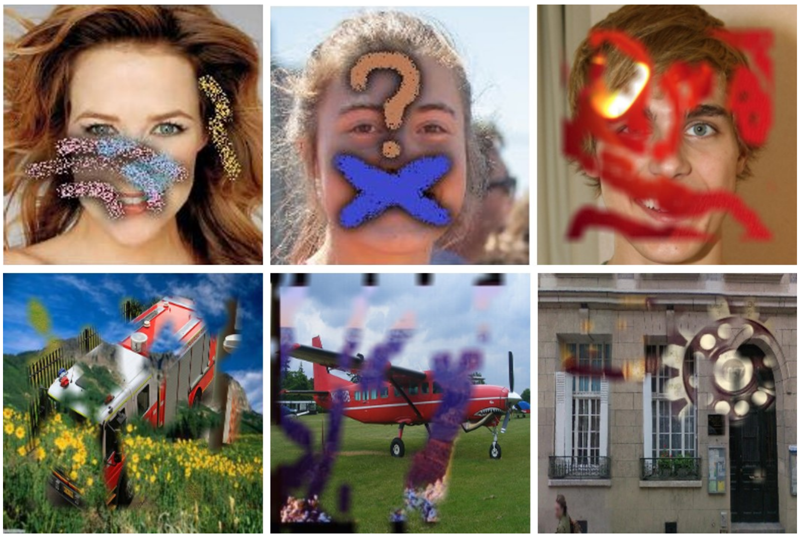


Figure 1.2: Sample corrupted inputs for blind image inpainting.

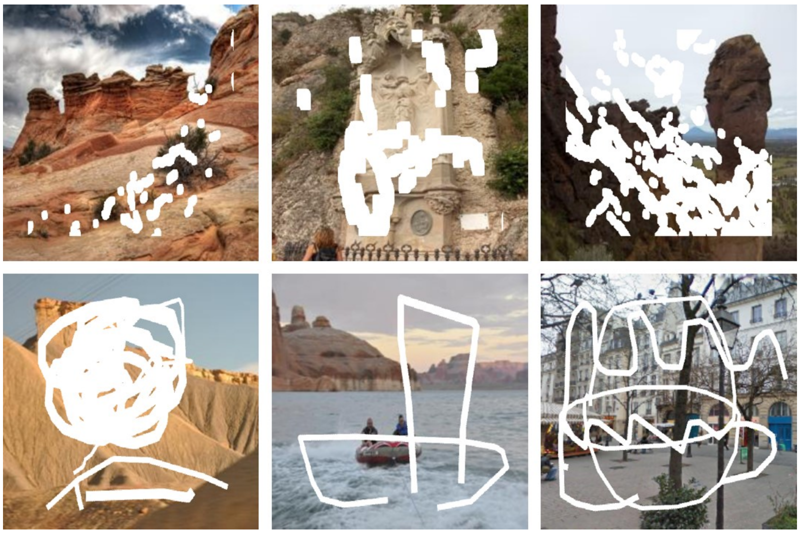


Figure 1.3: Types of corruptions in images with varying masked region.

## 1.2 Applications

With the ability of restoring the corrupted image with some missing regions, image inpainting is widely used in different applications like:

- **Corrupted image restoration** [38]: The basic application of the image inpainting task is corrupted image restoration. This helps to restore the old corrupted images with most plausible content. Figure 1.4 shows a sample corrupted image restoration application.

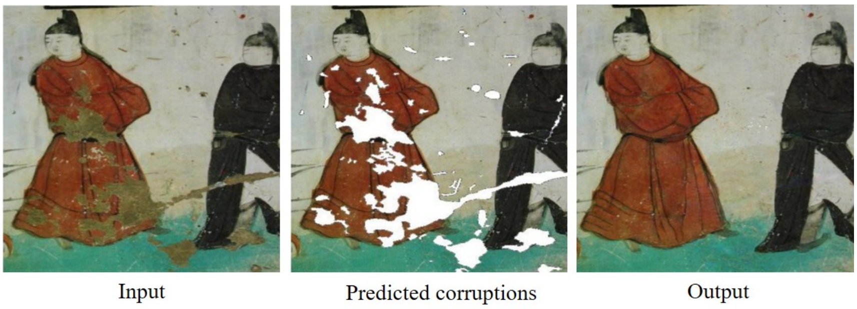


Figure 1.4: Corrupted image restoration.

- **Unwanted object removal** [22]: If someone wants to remove the unwanted objects from the image, image inpainting comes in handy. Provided with the region of object to be removed, image inpainting removes the object by replacing the masked region with relevant surrounding content. Sample application of object removal is shown in Figure 1.5.

- **Virtual try-on** [39, 40]: Nowadays, with the advancement in the technology, online shopping is gaining more attention. Image inpainting is utilized in online shopping

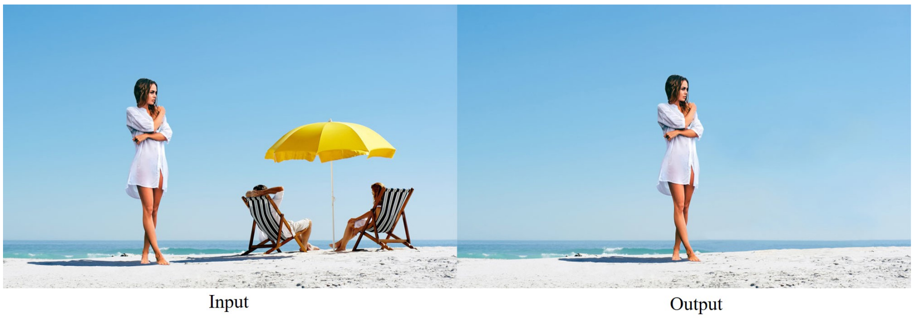


Figure 1.5: Unwanted object removal.

applications for virtual try-on. This application helps the end user to visualize the look for different clothes. Figure 1.6 shows samples of virtual try-on using image inpainting.

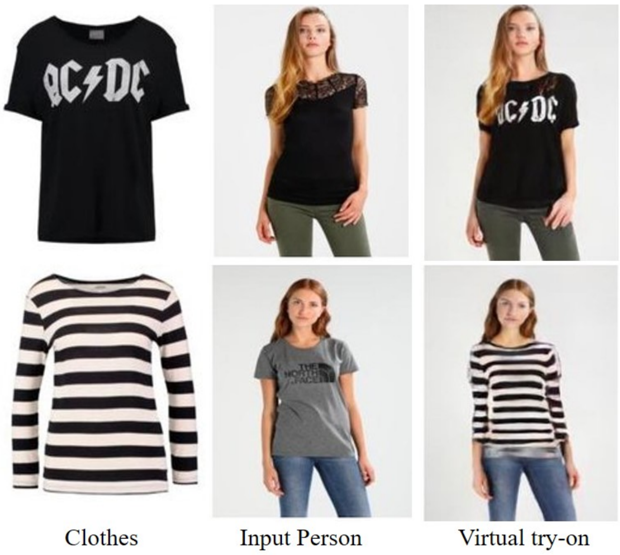


Figure 1.6: Virtual try-on.

- **Video de-captioning** [41, 42]: Considering the videos of different languages, there are captions or commercials overlapping the overall frame content. Inpainting plays an important role in removing these contents, which is termed as de-captioning as shown in Figure 1.7.

- **Single image scene synthesis**: Image inpainting is also applied to convert the arbitrary photos to life called as single image scene synthesis. For this application, the estimation beyond the input pixels is done with image inpainting task. This inpainting generates different views of input single image as shown in Figure 1.8.

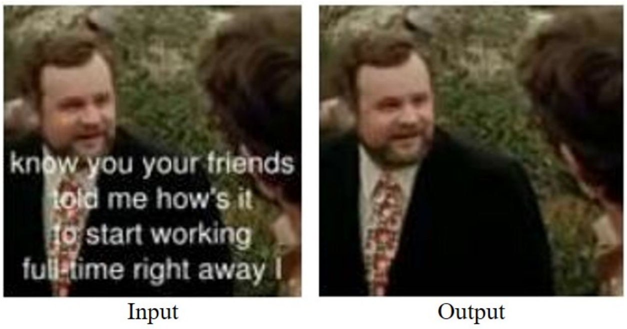


Figure 1.7: Video de-captioning.

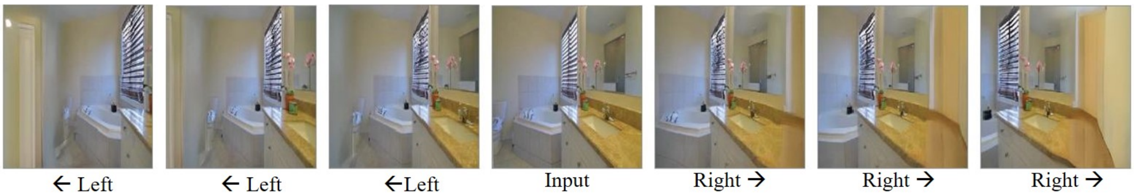


Figure 1.8: Single image scene synthesis.

## 1.3 Motivation

Advancement in image editing technology increases its demand in different applications such as image restoration, object removal, scene synthesis, dis-occlusion, *etc*. Image inpainting is one of the crucial processing techniques used for these applications. There exist numerous methods for image inpainting tasks. The conventional hand crafted methods deal with the textural statistics in the input image. They try to extract the statistical information from the available regions and utilize them to inpaint the missing regions. These methods lack in generating high-level semantics or structurally plausible results and are feasible when considering images with consistent regions like uniform background. Also, nowadays, various learning based methods are introduced for image inpainting tasks. These methods achieve great performance at inpainting the images but with the cost of high computational complexity. Some methods lack in generating structurally and visually plausible inpainted results. Further, the generative approaches nowadays are gaining more importance due to their better convergence for the tasks like image restoration, image segmentation, object detection, *etc*.

With the convincing ability of generative learning based methods towards efficient learning as compared to conventional hand-crafted methods, we are motivated to propose the generative approaches for image inpainting. In order to avoid the issue of high computational complexity, which leads to time consuming inpainting/restoration process, we aim to propose the inpainting methods with less computational complexity. Also, motivated with the reliability of blind image inpainting on various kinds of degradations, which does not depend on mask as input, we aim to propose a blind image inpainting architecture.

## 1.4 Problem Statement

The existing learning based methods efficiently inpaint the images if the masked region is of small size. The performance of these methods degrade when applied on the images with large masked regions. The existing methods have high computational complexity in terms of number of trainable parameters or run time. There are very few methods which deal with the blind image inpainting. From these observations, we define the problem statements for our work as:

- Difficulty in inpainting the image with large hole size.
- The performance of coarse-to-fine methods is dependent on each other.
- Lack of computationally efficient networks for image inpainting.
- Limited existing approaches for blind image inpainting.

## 1.5 Aim and Objectives

From the identified problems in existing image inpainting methods, we define the aim and objectives of our work as:
**Aim**: *To propose a novel approach for image inpainting with deep generative architectures.*
**Objectives**:

- To design a generative coarse-to-fine approach for inpainting images with varying hole sizes.
- To propose a single-stage lightweight approach for image inpainting with varying hole sizes.
- To propose a novel mask prediction independent approach for blind image inpainting.

## 1.6 Contributions

This study is focused on deep generative architectures for image inpainting task. The major contributions of this work are listed below:

- Inpainting the image with varying masked regions is a key requirement of any inpainting method. In this regard, we propose the novel coarse-to-fine architectures for image inpainting.
- In order to overcome the limitation of high computational complexity, we propose the novel single-stage architectures with reduced computational complexity for image inpainting.

- The availability of masks for image inpainting is another concern for non-blind methods. To solve this issue, we propose a novel mask independent single stage approach for blind image inpainting.

## 1.7 Thesis Structure

- **Chapter 1**: This chapter introduces the image inpainting task, applications of image inpainting, and motivation of the proposed work. Also, this chapter contains the preamble to the entire thesis.

- **Chapter 2**: This chapter provides the literature survey of different existing approaches in detail for image inpainting. Further, existing datasets and evaluation measures for image inpainting are explained in detail.

- **Chapter 3**: In this chapter, the exposition of proposed coarse-to-fine architectures consisting of encoder-level feature aggregation followed by feature sharing decoder, spatial projection layer, nested deformable multi-head attention, *etc.* for image inpainting is provided.

- **Chapter 4**: The proposed single-stage architectures with diverse receptive fields, multi-resolution feature fusion, and pseudo-decoder are detailed in this chapter. These single-stage architectures are computationally less complex as compared to state-of-the-art methods for image inpainting.

- **Chapter 5**: To avoid the mask dependency of inpainting architectures, the proposed blind image inpainting architecture is presented in this chapter.

- **Chapter 6**: This chapter provides the comparative study of all the proposed approaches in this work and concludes the overall thesis. It also discusses the future scope of this work to further improve the proposed work.

# Chapter 2

# Literature Survey

This chapter discusses about the existing approaches for image inpainting task. Also, the standard datasets used for experimental analysis and evaluation measures used to verify the effectiveness of the inpainting approach are discussed in this chapter.

## 2.1 Existing Approaches for Image Inpainting

Image inpainting is considered as an ill-posed task because for single image with missing regions, we can have multiple inpainted images. The image inpainting approaches are generally divided into two categories *i.e.* conventional (hand-crafted) and learning based image inpainting approaches.

### 2.1.1 Conventional Image Inpainting

The conventional inpainting methods focused mainly on texture synthesis, extracting the information from various patches of valid region in an image, and utilizing the statistical information of the textures from the valid regions. Initially, Bertalmio *et al.* [43] proposed a digital image inpainting method by analyzing how the experts will paint the corrupted image. They observed different key points like: the global view helps to determine how to fill the gap, the structure surrounding the gap can be continued to fill the gap, and different regions inside the gap can be filled with the contour lines or color followed by the textural details. Schoenemann *et al.* [44] proposed curvature based segmentation and inpainting approach. Meur *et al.* [45] proposed an exemplar based hierarchical super-resolution approach for image inpainting. In this, they inpainted the images from low-resolution to high-resolution by following two steps patch priority and texture synthesis at each resolution. Similarly, Shi and Qi [46] proposed patch selection followed by patch inpainting approach for image inpainting. Further, Hasegawa *et al.* [47] proposed signal prediction based on the non-harmonic analysis. He and Wang [48], Li and Zeng [49] proposed sparsity based image inpainting models.

Patch based approaches were proposed for image inpainting in [50, 51, 52, 53, 54, 55]. Barnes *et al.* [50] proposed a patch based method where the patch from nearest neighbour match is used for inpainting the image. Köppel *et al.* [51] proposed a fast image completion approach by finding the best texture from the spatial offsets of similar patches.

Further, Ruzic and Pizurica [52] proposed a context-aware patch-based approach by using textural descriptors to find best patch match. In order to maintain textural and structural consistency, Li *et al.* [53] used super-wavelet transform for estimating the multi-directional features of corrupted images. These features are then combined with weighted color-direction distance to find best similar patch. Similarly, to maintain structural coherence and textural consistency, Ghorai *et al.* [54] proposed a patch statistics based multiple pyramidal approach to generate multiple outcomes for single corrupted input. Further these outcomes are combined with weighted average to generate final inpainted image. Jin and Bai [55] proposed facet deduced directional derivative based patch-sparsity-based algorithm for image inpainting. Thaskani *et al.* [56] proposed a multi-view image inpainting approach with patch-based exemplar dictionary.

Casaca *et al.* [57] proposed a Laplacian coordinates segmentation approach combined with inner product-based filling order mechanism, anisotropic diffusion, and exemplar-based completion approach for interactive image inpainting. Ma *et al.* [58] proposed a group vectorized patch similarity approximation by low-rank matrix approach for image inpainting. Amrani *et al.* [59] proposed diffusion-based inpainting algorithm using partial differential equations for compressing hyper-spectral images. In order to have minimal user input, Zhang *et al.* [60] proposed super-pixel segmentation technique for image inpainting. Further, Liu *et al.* [61] proposed an architecture for exemplar-based image inpainting using a structure-guided approach. Ding *et al.* [62] proposed a Gaussian non-local texture similarity measure based patch search approach to obtain similar patches for image inpainting.

These conventional image inpainting methods generally follow exemplar, diffusion, patch-match and textural based approaches. Though the conventional methods extract statistical, structural and textural information efficiently, they either generate discontinuous texture at hole region or fail at reproducing the high-level semantic structures.

### 2.1.2 Learning Based Image Inpainting

The ability of learning approaches towards better convergence as compared to conventional approaches has been proved in different tasks like image restoration [63, 64], video segmentation [65, 66, 67], *etc.* The learning-based methods proposed for image inpainting follow different architectural pattern like coarse-to-fine (two-stage) [68, 69, 70, 12, 12] or single-stage [71, 72] architectures. In the coarse-to-fine architectures, first stage is focused on generating a globally inpainted image which is then used to generate a coarse output with finer inpainted results. The single stage architectures generally apply recurrent [14], progressive [73], and inverse generative adversarial networks (GANs) approaches for image inpainting. In the next subsections, the coarse-to-fine and single-stage architectures are explained in detail.

### Coarse-to-fine Architectures

The main motivation of the coarse-to-fine architecture lies in generating the global context first followed by generating the detailed texture in the final inpainted image. Yu *et al.* [68] proposed a coarse-to-refinement network consisting of contextual attention in order to generate finer inpainted image. With the observation of blurry outcomes of existing approaches, Liu *et al.* [69] proposed an architecture consisting of coherent semantic attention layer (CSA). The CSA layer is proposed with the motivation from human behavior to repair the corrupted picture with rough and refined steps. To inpaint the corrupted image with structural consistency, Ren *et al.* [70] proposed an architecture with structure re-constructor followed by texture generator. For structure reconstruction, they provided the corrupted image with the input structure which generate the recovered structure at first stage and then fine texture in final outcome. Yu *et al.* [12] claimed that vanilla convolutions treat all the available and missing pixel locations as the valid ones which may fail at consistent image generation. To avoid this, Yu *et al.* [12] proposed a gated convolution where a learnable mask updation mechanism is proposed unlike [18]. In order to inpaint the image with finer edges, Nazeri *et al.* [13] proposed a structure guided architecture for edge inpainting followed by image inpainting method. Similarly, Xu *et al.* [74] proposed edge-to-image generative inpainting approach. In [75], Lee *et al.* used the pre-trained model from [13] and proposed self-supervised fine-tuning algorithm for image inpainting. To inpaint ultra high resolution images, Yi *et al.* [22] proposed a residual aggregation approach in which the coarse-to-fine architecture is used to inpaint an image with normal resolution. Further, a contextual residual is aggregated onto the inpainted image to generate ultra high resolution inpainted image. Similarly, Moskalenko *et al.* [76] proposed a high-resolution inpainting approach. At first, they inpaint low resolution image and then up-sample the inpainted image. Then the four direction shifts and original image are used to inpaint final image. Liu *et al.* [77] proposed a probabilistic diverse GAN architecture which is based on vanilla convolution GAN. While inpainting the image, deep features of input random noise via coarse-to-fine architecture are modulated by previously restored image of multiple scales and the hole regions. This random noise produces pluralistic inpainted outcomes of single input image. Similarly, Peng *et al.* [78] proposed diverse structure generation approach for input corrupted image via hierarchical quantized variational auto-encoder. Wadhwa *et al.* [8] proposed the hyper-graph based approach for globally semantic inpainted image with a trainable method for hyper-graph convolutions. Further, Zeng *et al.* [79] proposed a coarse-to-fine generative approach with auxiliary contextual reconstruction loss to appropriately borrow the available regions to fill missing regions effectively. Wang *et al.* [80] proposed a conditional normalizing flow model for generating diverse structural prior followed by inpainting to achieve real-time inference to generate diverse inpainted images. GAN inversion has faithful convergence ability as the pre-trained encoder or decoder of GANs architecture is used to guide actual inpainting architecture or fixed decoder is used to train the encoder. In this regard, Wang *et al.* [81] proposed a dual-path GAN inversion approach for image inpainting. Similarly, Yoon *et al.* [82] proposed styleGAN inversion approach for image inpainting. Cao *et al.* [83] proposed a sketch-tensor space based approach for image inpainting.

Apart from coarse-to-fine architectures, researchers have proposed multi stage architectures. Xiong *et al.* [84] proposed a foreground aware architecture for image inpainting. In this, they proposed a contour detection followed by contour completion and then image completion module. Failure in contour detection will lead to failure in final image inpainting results. Hedjazi *et al.* [85] proposed a multi-scale texture-aware GANs architecture for image inpainting. Similarly, Qu *et al.* [86] proposed a multi-scale architecture in inpaint the structure first and then the texture with pyramidal generators. Kim *et al.* [87] proposed a super-resolution based approach for image inpainting. In this, they applied the super-resolution on the coarse inpainted output followed by refinement. The refined output is then downscaled in order to get original dimensions. In [88], Dong *et al.* applied the structure restoration first by considering corrupted image and corrupted edges as input followed by the structure feature encoder and then final Fourier convolution texture restoration. In order to inpaint the image with local and global consistency, Quan *et al.* [89] applied a three-stage architecture for image inpainting consisting of first stage for global inpainting followed by two stages for global and local refinements respectively. Cai *et al.* [90] proposed a multi-stage coarse-to-fine architecture for inpainting the image with multiple stages representing respective scales. Li *et al.* [91] proposed a multi-level approach with Siamese filtering consisting of kernel prediction branch followed by image filtering branch for image inpainting. Yamashita *et al.* [92] proposed the depth and edge inpainting followed by image inpainting approach for depth-aware image inpainting. Also, Wang *et al.* [93] proposed a novel approach with monochromatic reconstruction and the multi-stage internal color restoration approach. These coarse-to-fine architectures produce visually better inpainted images but with the compromise in the computational complexity.

### Single-stage Architectures

Xie *et al.* [71] proposed the very first learning based approach for image inpainting containing the combination of sparse coding and pre-trained denoising auto encoder. Further, Köhler *et al.* [72] proposed an approach to learn mapping from image patches with the help of deep neural network. Later on, with the emergence of GANs, Pathak *et al.* [94] introduced the context encoder based adversarial learning approach for image inpainting. Yeh *et al.* [95] used the trained generative model for searching the closest encoding of the input corrupted image by utlizing the context and prior losses for image inpainting. With the claim on normal convolutional neural networks (CNNs) that they process the valid and missing content with same priority, Liu *et al.* [18] proposed a partial convolution layer for image inpainting. In partial convolution, the mask information is utilized to determine the valid and invalid content to process the features at each encoder layer [12]. The feature normalization and mask updation in this method is rule based which is then made learnable by [12]. Xie *et al.* [96] proposed the learnable bi-directional attention maps for mask updation. Su *et al.* [97] proposed the dense connections based U-Net architecture [98] consisting of partial convolutions for image inpainting. Introduction of contextual attention mechanism paved a way towards efficient image inpainting in various research works. Li *et al.* [99] proposed context-aware semantic approach for image inpainting in order to maintain spatial information accurately. Zeng *et al.* [21] proposed a pyramidal context-encoder with multi-scale decoder approach for image inpainting. Similar to contextual attention Li *et al.* [14] proposed a recurrent approach consisting of knowledge consistent attention for image inpainting. Wang *et al.* [100] proposd a contextual attention layer based network with parallel streams for damaged image and mask processing.

With the observation of predicting the missing content by propagating the surrounding features via convolution layer in context encoder, Yan *et al.* [10] proposed shift-connection layer approach for image inpainting. In the shift-connection layer, the CNN features are shifted to form an estimation of missing features. In this, the decoding features are ignored which is then modified with the introduction of bishift layer [101] which captures information from both the encoder and decoder. Wang *et al.* [9] proposed a single stage architecture which synthesizes the multiple image components in parallel with a confidence driven reconstruction loss for detail enhancement. Similarly, Sagong *et al.* [102, 103] proposed parallel extended decoder approach for image inpainting with less computational complexity. Zhu *et al.* [104] proposed the recovery and refinement decoders to inpaint arbitrary missing regions. Ma *et al.* [105] proposed a contrastive attention based network with two parallel encoders. Wang *et al.* [106] proposed a parallel multi-resolution fusion approach for image inpainting.

Yang *et al.* [107] proposed a two stage architecture consisting of a content and texture network. The content network is mainly proposed for content generation in the missing regions whereas the texture network generates the fine texture in the inpainted image. Zheng *et al.* [11] proposed probabilistically principled framework for pluralistic image inpainting task. To inpaint the images with large missing regions, Li *et al.* [73] proposed a progressive image inpainting approach which progressively inpaints the structure of corrupted image. Saad *et al.* [108] proposed a novel discriminator approach to determine patch-wise real or fake output which helps to generate visually realistic results. Li *et al.*

[109] proposed the progressive decomposition of features into two different streams followed by the fusion of them in order to get sharp textures in the inpainted image. Lahiri *et al.* [110] proposed a two stage training approach with GANs and noise prior prediction training to infer intermediate noise for input corrupted image. Zhao *et al.* [111] proposed a cross semantic attention layer with primary branch for manifold projection to map instance image space to image completion space and secondary branch with conditional encoder to generate the label. Liao *et al.* [112] proposed a semantic guidance and estimation based network to iteratively evaluate the uncertainty of the inpainted image using the pixel-wise semantic segmentation followed by the alternate optimization of structural prior and inpainted contents. Liao *et al.* [113] also proposed an architecture guided by coherence prior of textures and semantics. Similarly, Chen and Liu [114] proposed dual encoder branches: one to process corrupted image and the other for edges which are then merged in the dual attention layer. Han and Wang [115] proposed the evolutionary GANs approach for facial image inpainting. Suvorov *et al.* [116] proposed Fourier convolutions for inpainting the images with large masks. Further, Lu *et al.* [117] proposed the Fourier convolutions based architecture with spatial and frequency loss for image inpainting. Jam *et al.* [118] proposed a reverse masking network to blend the valid to missing regions efficiently. Guo *et al.* [16] proposed an architecture for conditional structure in terms of edges and texture generation. Yu *et al.* [119] proposed a discrete wavelet transform (DWT) based approach for image inpainting. Suin *et al.* [120] proposed a knowledge distillation approach to provide feature level supervision while training the network for image inpainting. Zhao *et al.* [121] proposed a co-modulated GANs architecture for image inpainting. Likewise, Zheng *et al.* [122] proposed cascaded modulation GANs for image inpainting.

The ability of extracting long-range dependencies of the transformers is utilized in different image inpainting approaches [123, 124, 17, 125, 126]. Fan *et al.* [124] proposed a spatial attention transformer network for image inpainting. Also, Li *et al.* [17] proposed a transformer based approach where the long-range dependencies are modelled by the valid tokens from the mask. Zhou *et al.* [125] proposed a color-spatial transformer to adjust the color and spatial misalignments of multiple global homographies. All the homographies are merged in order to generate the final inpainted image. Wan *et al.* [126] proposed a pluralistic image generation architecture with two networks: bidirectional transformer to generate the probability distributions for missing regions and a CNN network for appearance generation. Due to down-sampling of the input into much lower resolution in the transformer approaches, there is a chance of information loss which is reduced in [123] by introducing patch based auto-encoder and un-quantized transformer for image inpainting. With the evolution of denoising diffusion models, Lugmayr *et al.* [127] proposed the denoising diffusion probabilistic models based approach for image inpainting. Most of the above discussed architectures generally have high complexity or they lack in producing the plausible results when images with large missing regions are considered.

## 2.2 Existing Approaches for Blind Inpainting

The emergence of blind image inpainting with learning based methods appeared with the CNN in [128]. Liu *et al.* [129] proposed a residual learning based approach with the horizontal and vertical gradients to generate the detailed clear image. Prior to these works, Xie *et al.* [71] utilized the sparse auto-encoder for image denoising and blind image inpainting. Similarly, in [130], Ren *et al.* proposed Shepard convolutional network for image denoising and blind image inpainting. These approaches consider simple contaminations like text imposed on images or images with some part appended from other images of masks with thin size.

In order to consider the complex contaminations, Wang *et al.* [23] proposed coarse-to-fine architecture for blind image inpainting. The network proposed by [23] is a visual consistency network which first estimates where to inpaint by predicting the masks and then utilize the image inpainting network. Further, Wang *et al.* [131] include the contextual coherence and additional frequency modality input for mask prediction task. This is followed by the landmark prediction and then final inpainting of facial images. Both [23] and [131] use coarse-to-fine architectures containing mask prediction followed by inpainting. In this kind of inpainting, there may be a chance of unavoidable mask prediction error leading to undesired image inpainting results. In this regard, Zhao *et al.* [24] observed that, in blind image inpainting, the differentiation of contaminated and valid regions and mask prediction are heavily correlated. With this assumption, Zhao *et al.* [24] proposed a single stage hybrid encoder-decoder network for blind image inpainting. In order to capture global context, the transformer encoder is used [24] and CNN decoder is used to revamp the contaminations.

## 2.3 Existing Experimental Datasets

For experimental analysis, we have used five publicly available image datasets corrupted with different mask ratios.

### 2.3.1 Image Datasets

The different image datasets are:

**CelebA-HQ [1]:** This dataset contains a high quality images of celebrity faces. This dataset contains a total of 30000 images with 28000 for training and 2000 for validation.

**Places2 [3]:** It contains approximately 1.8M images from 365 different places categories.

For our experiment, we have considered 20 different places categories with respective images from validation splits.

**Paris street view [4]:** This dataset contains 14900 training and 100 validation images from the street views of Paris.

**Flickr-Faces-HQ (FFHQ) [2]:** This dataset consists of 70,000 high-quality face images with different variations in terms of age, ethnicity and image background.

**ImageNet [5]:** This dataset contains 14 million images, a little more than 21 thousand groups or classes. Figure 2.1 shows the sample images from all the considered datasets.

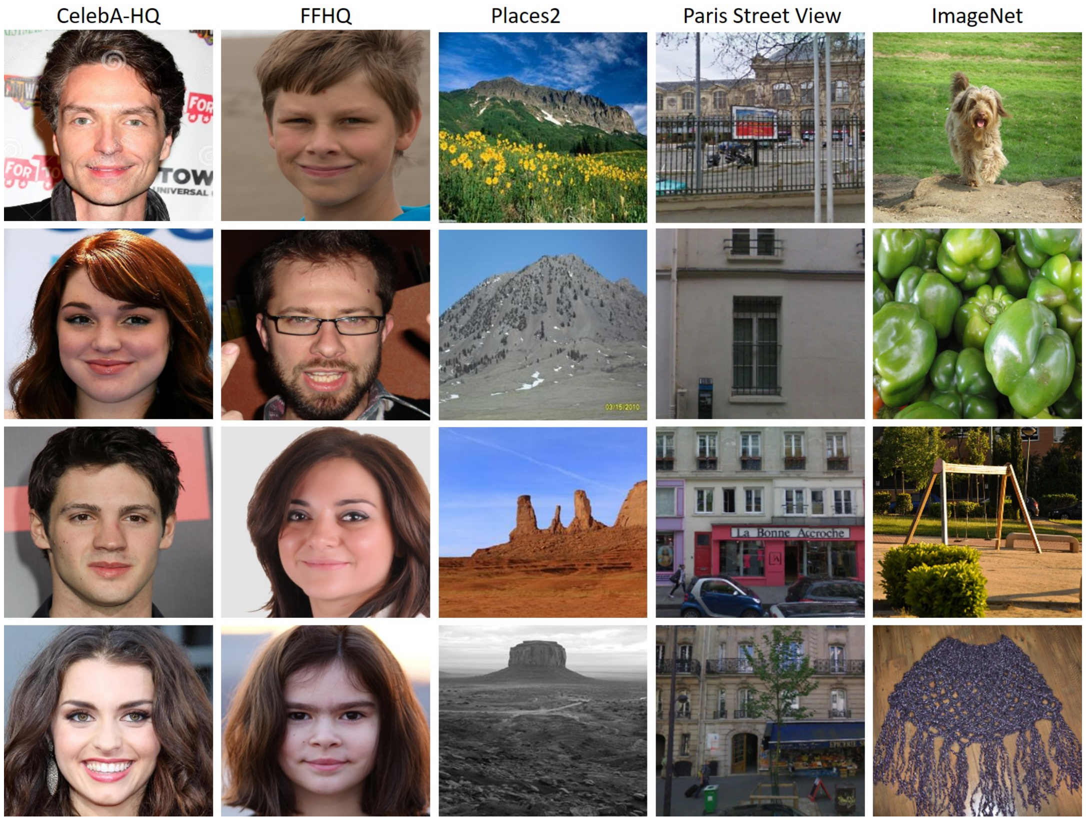


Figure 2.1: Sample images from CelebA-HQ [1], FFHQ [2], Places2 [3], Paris Street View [4] and ImageNet [5] datasets.

### 2.3.2 Mask Datasets

Three different types of masks are used to corrupt the images. The three mask datasets are: NVIDIA mask dataset [6], quick draw irregular mask dataset (QD-IMD) [7] and synthetically generated masks [8].

**NVIDIA Mask Dataset [6]:** This dataset contains around 54000 masks for training and 12000 masks for testing. To generate the training masks, we have to threshold the images and then randomly dilate the binarized masks in order to get diversity in the masks for training. Further, testing set of NVIDIA mask dataset covers different hole-to-image area *i.e.,* mask ratios in the range $(0.01, 0.6]$. In total, there are 12k masks available which are divided into six sets with $(0.01, 0.1]$, $(0.1, 0.2]$, $(0.2, 0.3]$, $(0.3, 0.4]$, $(0.4, 0.5]$, and $(0.5, 0.6]$ mask ratio.

**Quick Draw Irregular Mask Dataset (QD-IMD) [7]:** This is a mask dataset with strokes drawn by human hand called as quick draw irregular mask dataset (QD-IMD) [7]. The two mask datasets differ from each other where, the NVIDIA mask dataset is based on occlusion/dis-occlusion mask estimation between two consecutive frames which has sharp edges due of rough crops near to borders and the QD-IMD consists of irregularly drawn strokes without sharp edges.

**Synthetic masks [8]:** In synthetic mask dataset [8], we can generate five mask sets with mask ratios $0.1-0.2$, $0.2-0.3$, $0.3-0.4$, $0.4-0.5$, $0.5-0.6$ for training and testing. The synthetic mask dataset generates random holes (*with ones at holes and zeros at non-hole region*) by simulating spots, scratches *etc.* The sample masks are given in 2.2.

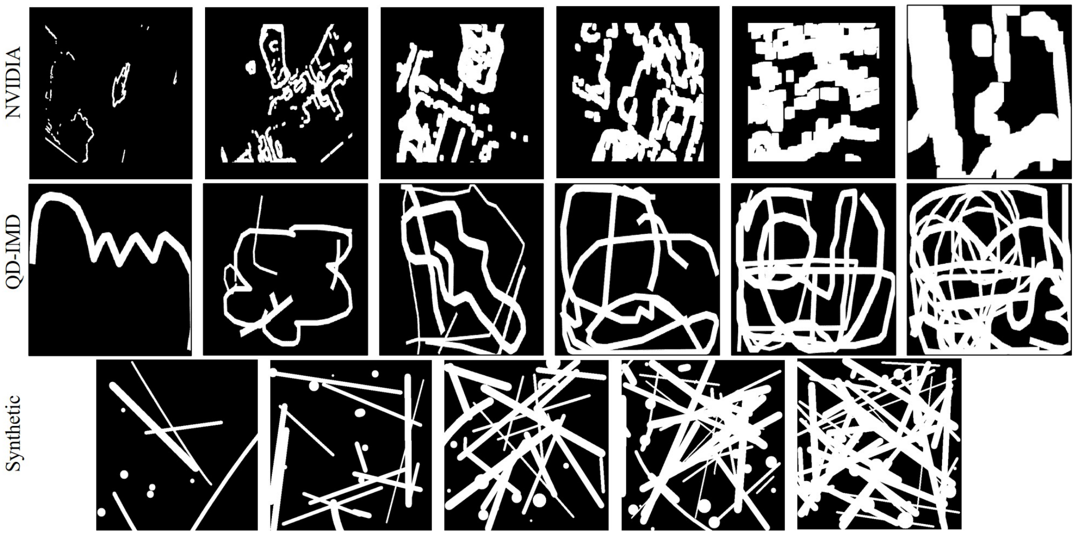


Figure 2.2: Sample masks from NVIDIA [6], QD-IMD [7] and synthetic [8] datasets

## 2.4 Training Losses

Given the image with holes ($I_C$) and the mask ($I_M$) with ones at holes and zeros at the non-hole region, it is required to generate the inpainted image ($I_I$) similar to the target image ($I_T$). While training, instead of calculating the loss on the overall image which will create the disturbances in the hole and non-hole region, we have used the separate loss function for the hole and non-hole region which is as given in Eq. (2.1) and (2.2), respectively.

$$L_1^{Holes} = \|I_M \circ (I_I - I_T)\| \tag{2.1}$$

$$L_1^{Non-holes} = \|(1 - I_M) \circ (I_I - I_T)\| \tag{2.2}$$

where, $\circ$ stands for the Hadamard product. For the generation of the globally and locally consistent realistic image, the adversarial loss plays an important role [132], [133]. The adversarial loss is the min-max problem between generator and discriminator. It can be explained with the Eq. (2.3).

$$L_{Adv} = \underset{\mathbf{D}}{max}\, \underset{\mathbf{G}}{min}\, \mathbb{E}[log(D(I_C, I_T))] + \mathbb{E}[log(1 - D(I_C, G(I_C, I_M)))] \tag{2.3}$$

where, $D$ is the discriminator and $G$ is the generator. To guide the network for textural and structural information, the perceptual loss is calculated between the deep feature maps of the ground-truth and inpainted images by passing them through the pre-trained VGG19 model [134]. The perceptual loss is given as:

$$L_{Per} = \sum_{s=1}^{S} \left( \sum_{i=1}^{M} \sum_{j=1}^{N} \sum_{k=1}^{K} \frac{1}{MNK} \left\| \phi_s(I_T)_{i,j,k} - \phi_s(I_I)_{i,j,k} \right\|_1 \right) \tag{2.4}$$

where, $\phi$ are the feature maps ($s \in (1, S)$) of the VGG19 model. $M$, $N$, and $K$ are the dimensions of the feature maps. Along with these losses, the edge loss is also considered to focus on the edge enhancement while training. The edge loss is formulated as:

$$L_{edge} = \left\| \mathbb{S}(I_T) - \mathbb{S}(I_I) \right\|_1 \tag{2.5}$$

where, $\mathbb{S}$ is the sobel operator. Along with all these losses, a structural similarity loss $L_{SSIM}$ is optimized to minimize the per pixel difference in the output image.

# Chapter 3

# Coarse-to-fine Image Inpainting

In this chapter, we discuss the proposed two-stage architectures for image inpainting. Some of the existing approaches for image inpainting depend on the prior information [13, 74] to inpaint the input corrupted image. Also, different methods utilized more than two cascaded models to inpaint the images [84, 85, 86]. These methods sometimes generate inconsistent results or they have high computational complexity. In order to balance these issues of quality and complexity, we have proposed three different solutions with coarse-to-fine architectures to inpaint the images with varying missing regions. The three contributions with coarse-to-fine architectures are:

1. Image Inpainting via Spatial Projections.
2. Nested Deformable Multi-head Attention for Facial Image Inpainting.
3. FASNet: Feature Aggregation and Sharing Network for Image Inpainting.

These solutions are explained in the next sections.

## 3.1 Image inpainting via spatial projections

Following the existing works [79, 22, 69, 70, 12, 8, 13], we use coarse-to-fine architecture for image inpainting. This will help the network to progressively inpaint the image without any disturbances in the image. Further, to effectively inpaint the image it is necessary to focus on the edges in an image for effective inpainting. So, while training the network for inpainting task we have provided a novel canny edge based loss for optimization. The existing state-of-the-art method use the edge loss with the Sobel operator [8]. The Sobel operator works well if there is less noise in the input. With more noisy input, the Sobel operator will generate false edges. This will lead to false training as the network will try to optimize the generated output with the false edges of ground truth. In [135], the authors proposed the region based loss for image super-resolution task, where the considered region is the edges calculated from target by canny edge detector. Hence, in order to avoid the chance of false edge guided optimization of the network while training, we propose the use of Canny edge operator, after its success in image super-resolution task [135] and style transfer task [136]. The Canny edge operator is less sensitive to noise due to which there is no effect of noise in the generated edges. This will help the network for better optimization while training. Our main contributions are:

- A novel architecture is proposed for image inpainting without any self-attention mechanism.
- A novel spatial projection layer is proposed to project the spatial information from non-hole regions to the hole regions for introducing efficient spatial consistencies in the inpainted image.
- Unlike existing state-of-the-art architectures for image inpainting, we introduced the use of edge loss with Canny edge operator for better optimization of the network.

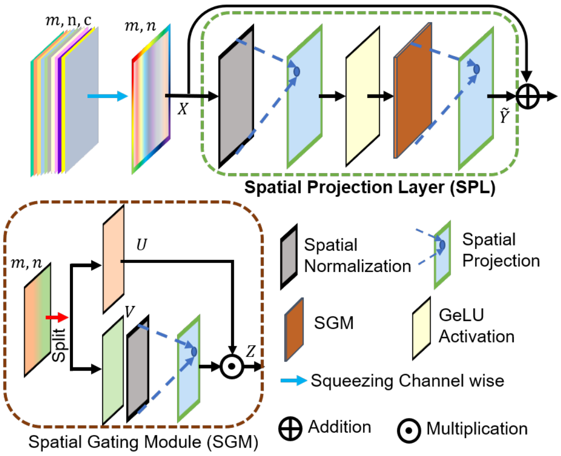


Figure 3.1: Proposed spatial projection layer to introduce efficient spatial consistencies.

### 3.1.1 Proposed Framework

Here, first we give the exposition to multi-layer perceptron (MLP) based gMLP [35]. Then we give an overview of our proposed spatial projection layer to introduce efficient spatial consistencies in the inpainted image. Further, we elaborate on the proposed architecture for image inpainting.

#### Gated MLP Overview

Generally, the self-attention blocks try to combine the spatial information from all the representations. The attention mechanism gives the bias that the spatial interactions should be dynamically parametrized based on the input representations. But, the MLPs in the self-attention represent the static parametrization for arbitrary functions [35]. Because of this, it is difficult to give remarkable effectiveness with the attention mechanisms [35]. So, in [35], the authors introduced the MLP-based alternative without self-attentions, consisting of channel projections and spatial projections with static parametrization for

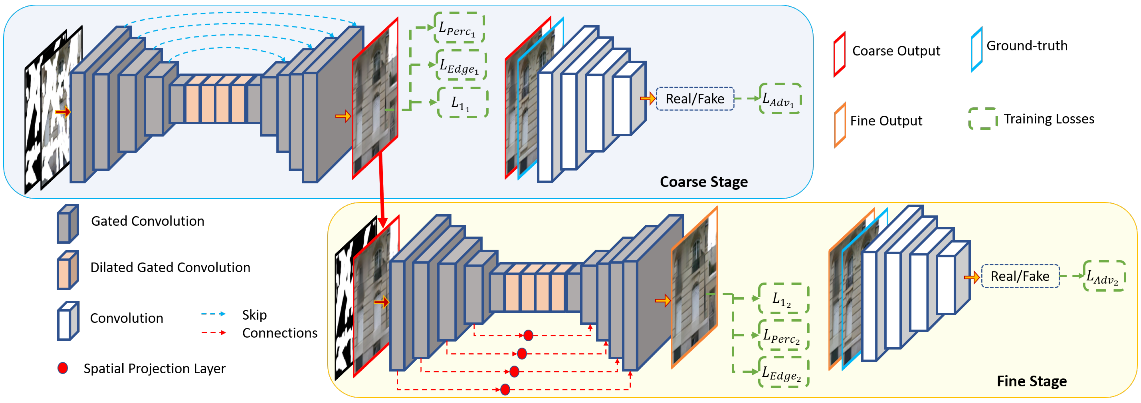


Figure 3.2: Architectural details of the proposed framework for image inpainting. Architecture consists of two stages: coarse and fine. In fine stage, a spatial projection layer is proposed to project the spatial information from non-hole regions to hole regions in an image (Better viewed in color).

image classification task. In their work, the projections are utilized with the linear and multiplicative gating mechanism and named as gMLP since it is built with basic MLP layers with gating mechanism. The gMLP block is defined as:

$$\tilde{Y} = \tilde{Z} \odot V; \quad \tilde{Z} = s(Z); \quad Z = \sigma(X \odot U) \tag{3.1}$$

where, $\tilde{Y}$ is the output of gMLP, $X$ is input, $\sigma$ is GeLU activation function, $U$, $V$ are linear projections along channel dimensions, $s(.)$ is a spatial interaction (spatial gating unit) and $\odot$ is element-wise multiplication.

To introduce spatial interactions, the spatial gating unit contains the contraction operation over the spatial dimension. Here, the linear projection *i.e.,* spatially linear mapping is considered as spatial projection. The $s(.)$, a spatial interaction output or the output of linear gating mechanism is given as:

$$s(Z) = Z \odot f_{w,b}(Z) \tag{3.2}$$

where, $\odot$ is element-wise multiplication operation. This gMLP layer with repetitive layers, first introduced for image classification task [35] with superior performance than VIT [137].

Table 3.1: Differences between gMLP [35] and the proposed SPL

| Method → | gMLP [35] | Proposed SPL |
|---|---|---|
| Input | Input Embeddings | Channel-wise averaged Input features |
| Projections | Channel | Spatial |
| Normalizations | Channel | Spatial |

### Proposed Spatial Projection Layer

Inspired by the success of gMLP [35] for image classification task, in this work, we propose the spatial projection layer (SPL) to focus on the spatial features relevant for image inpainting task. The overview of the spatial projection layer is shown in Figure 3.1. The SPL is designed to project the spatial information from various non-hole locations to the hole region in that specific feature map. This projected spatial information is then added to the input feature maps to fill the hole regions effectively.

The input feature maps of size $m \times n \times c$ are first squeezed channel-wise to the size $m \times n$ using global average pooling. These squeezed feature maps contain the global information from all the channels of the input feature maps. The squeezed feature map is then fed to the spatial normalization layer. The spatial normalization with $X$ as input feature maps can be equated as:

$$norm(X) = \frac{f_{in} - \mathbb{E}(X)}{\sqrt{Var(X) + \epsilon}} \tag{3.3}$$

where, $\mathbb{E}$ is mean, $Var$ is variance. This normalization helps to normalize all the feature values in the desired range, which in turn reduces the inconsistencies in the hole and non-hole regions by filling of hole region with neighbouring features in the feature map. This normalized feature map is then fed to the spatial projection layer which is a linear projection along spatial dimension. The spatial projections are given as:

$$f_{w,b}(X) = wX + b \tag{3.4}$$

where, $w$ is a matrix with the dimension same as that of sequence length, $b$ is the respective bias. Unlike the self-attentions, the spatial projection matrix, $w$, is independent of input representations. Output of SPL with $X$ as input is given as:

$$Out_{SPL} = \tilde{Y} + X \tag{3.5}$$

$$\tilde{Y} = f_{w,b}(S(\sigma(f_{w,b}(norm(X))))) \tag{3.6}$$

where, $X$ is the input feature map, $f_{w,b}$ is the spatial projection, $\sigma$ is GeLU activation layer, $S(.)$ is the spatial gating module, and $Z$ is formulated as:

$$Z = U \odot f_{w,b}(V'); \qquad V' = norm(V) \tag{3.7}$$

where, $\odot$ is element-wise multiplication, $U$ and $V$ are the linear projections along `spatial` dimension. Unlike gMLP [35], where the main focus is on extracting the information from `channel` dimension for channel-wise information projection. Here in proposed SPL, the information is projected in the `spatial` dimensions. This `spatial` projection of information helps the network to extract the relevant information from the non-hole feature space to fill the hole region effectively (*See Table 3.1 for the difference between gatedMLPs [35] and proposed SPL*).

#### Inpainting Framework

The proposed inpainting framework makes use of cascaded two-stage GANs model for image inpainting task (*see Figure 3.2*). The purpose behind the use of coarse-to-fine architecture is that, using the spatial projection layer with only one stage may produce disturbances in the output. SPL is designed to mainly focus on the non-hole information which is further projected in the hole regions for effective inpainting. It was observed through experiments, utilizing this SPL in coarse stage generates discontinuity in the feature maps, as they contain the hole region which denormalize the features (*the effect of applying SPL in coarse and fine stage is analysed in Section 3.1.3*). Unlike in coarse stage, the fine stage has some approximate content at the hole locations which may be easily utilized for overall feature map normalization. This paves a way towards efficient projection of information.

The generator model in coarse stage consists of gated convolution layers followed by the dilated gated convolution layers and decoder layers with skip connections from the respective encoder layers. The dilated convolution layers are utilized to focus on maximum receptive field. The first generator generates the coarsely inpainted outcome which is then fed to the fine stage. The second generator *i.e.,* fine stage consists of the gated convolution layers followed by the dilated gated convolution layers and decoder layers with *skip connections processed through proposed spatial projection layers.* The SPL provides the spatially projected information from the neighbouring locations in a feature map to the hole locations. These processed feature maps guide the decoder with spatially correlated information for reconstruction of the image.

We have trained both the stages in adversarial learning manner. The output of both the generators is then fed to respective discriminator to discriminate it as real or fake. Learning both the stages in adversarial manner helps the network to converge efficiently. The discriminator used for this task is similar to the PatchGAN [133]. The coarse output and ground-truth pair is used for first discriminator loss calculation and fine output with ground-truth is given for second stage discriminator loss calculation. The discriminator is used in parallel combination where total discriminator loss is average of first stage and second stage loss.

### 3.1.2 Training Details of Proposed Network

**Implementation Details:** We implemented the proposed network architecture in PyTorch. Network is trained on images of $256 \times 256 \times 3$ size with a batch size of 1. Network parameters are optimized using the Adam optimizer with a learning rate $= 0.0002$, $\beta_1 = 0.5$

Table 3.2: Comparison of the proposed method with state-of-the-art methods for image inpainting on CelebA-HQ dataset with NVIDIA masks from [6].

| Mask Ratio | Method | Publication | PSNR↑ | SSIM↑ | $L_1$ ↓ | LPIPS↓ | FID↓ |
|---|---|---|---|---|---|---|---|
| 0.1-0.2 | SN [10] | ECCV-18 | 28.70 | 0.949 | 3.439 | 0.0584 | 4.342 |
| | GMCNN [9] | NIPS-18 | 28.03 | 0.943 | 3.070 | 0.074 | 11.259 |
| | PIC [11] | CVPR-19 | 28.39 | 0.953 | 2.054 | 0.062 | 5.420 |
| | Gconv [12] | ICCV-19 | 27.56 | 0.947 | 2.216 | 0.0612 | 5.563 |
| | EC [13] | CVPRW-19 | 29.20 | 0.962 | 1.952 | 0.0637 | 4.638 |
| | RFR [14] | CVPR-20 | 29.38 | 0.946 | 1.900 | 0.029 | 3.894 |
| | MANET [15] | PR-20 | 32.42 | 0.952 | 0.630 | 0.0248 | 3.254 |
| | HR [8] | WACV-21 | 30.95 | 0.969 | 2.544 | 0.0229 | 3.224 |
| | CTSDG [16] | ICCV-21 | 32.11 | 0.971 | 0.859 | 0.025 | 3.326 |
| | **Ours** | PR-22 | **34.27** | **0.983** | **0.540** | **0.023** | **1.870** |
| 0.3-0.4 | SN [10] | ECCV-18 | 22.80 | 0.891 | 3.354 | 0.2358 | 31.899 |
| | GMCNN [9] | NIPS-18 | 23.36 | 0.843 | 4.157 | 0.1943 | 33.877 |
| | PIC [11] | CVPR-19 | 22.99 | 0.854 | 3.893 | 0.172 | 25.971 |
| | Gconv [12] | ICCV-19 | 23.59 | 0.854 | 3.893 | 0.152 | 12.429 |
| | EC [13] | CVPRW-19 | 24.97 | 0.904 | 3.167 | 0.1445 | 12.084 |
| | RFR [14] | CVPR-20 | 25.06 | 0.901 | 3.116 | 0.1586 | 17.056 |
| | MANET [15] | PR-20 | 26.67 | 0.874 | 1.743 | 0.1248 | 11.522 |
| | HR [8] | WACV-21 | 25.52 | 0.927 | 3.486 | 0.1529 | 12.490 |
| | CTSDG [16] | ICCV-21 | 26.81 | 0.929 | 2.970 | 0.1055 | 11.299 |
| | **Ours** | PR-22 | **28.86** | **0.944** | **1.240** | **0.0732** | **6.570** |
| 0.4-0.5 | SN [10] | ECCV-18 | 21.21 | 0.787 | 5.820 | 0.327 | 34.530 |
| | GMCNN [9] | NIPS-18 | 19.66 | 0.785 | 5.652 | 0.3403 | 48.739 |
| | PIC [11] | CVPR-19 | 20.85 | 0.776 | 5.016 | 0.2932 | 42.440 |
| | Gconv [12] | ICCV-19 | 21.25 | 0.852 | 5.421 | 0.245 | 42.220 |
| | EC [13] | CVPRW-19 | 22.46 | 0.866 | 5.597 | 0.2345 | 41.453 |
| | RFR [14] | CVPR-20 | 23.77 | 0.839 | 4.584 | 0.2045 | 29.827 |
| | MANET [15] | PR-20 | 24.51 | 0.892 | 2.012 | 0.181 | 11.085 |
| | HR [8] | WACV-21 | 23.28 | 0.889 | 4.265 | 0.1974 | 26.455 |
| | CTSDG [16] | ICCV-21 | 23.96 | 0.895 | 3.975 | 0.185 | 10.872 |
| | **Ours** | PR-22 | **26.37** | **0.911** | **1.970** | **0.1081** | **8.650** |

and $\beta_2 = 0.999$ for both the generator and discriminator. The total loss is a combination of $L_1$, adversarial, perceptual and proposed edge loss (Eq. 3.8) for both the stages as given in Eq. 3.9 is used for the optimization of the network. The training of the proposed network is done on NVIDIA DGX station with 2.2 GHz processor, Intel Xeon E5-2698, NVIDIA Tesla V100 16 GB GPU and tested on CPU.

**Loss Functions:** Generally, to calculate the edge loss, a Sobel operator is used [8]. The limitation of using the Sobel operator is its sensitivity to noise. With the increase in noise, the gradient magnitude of the edges also degrades which produces inaccurate edges. In this context, inspired from style transfer task [136], unlike existing state-of-the-art methods for image inpainting, *we propose the use of the edge loss with the* ***Canny edge detection*** *operator*. The edge loss $L_{Edge_s}$ for stage $s$ is calculated as:

$$L_{Edge_s} = \|\mathbb{C}(I_{Gen_s}) - \mathbb{C}(I_{Gt})\| \tag{3.8}$$

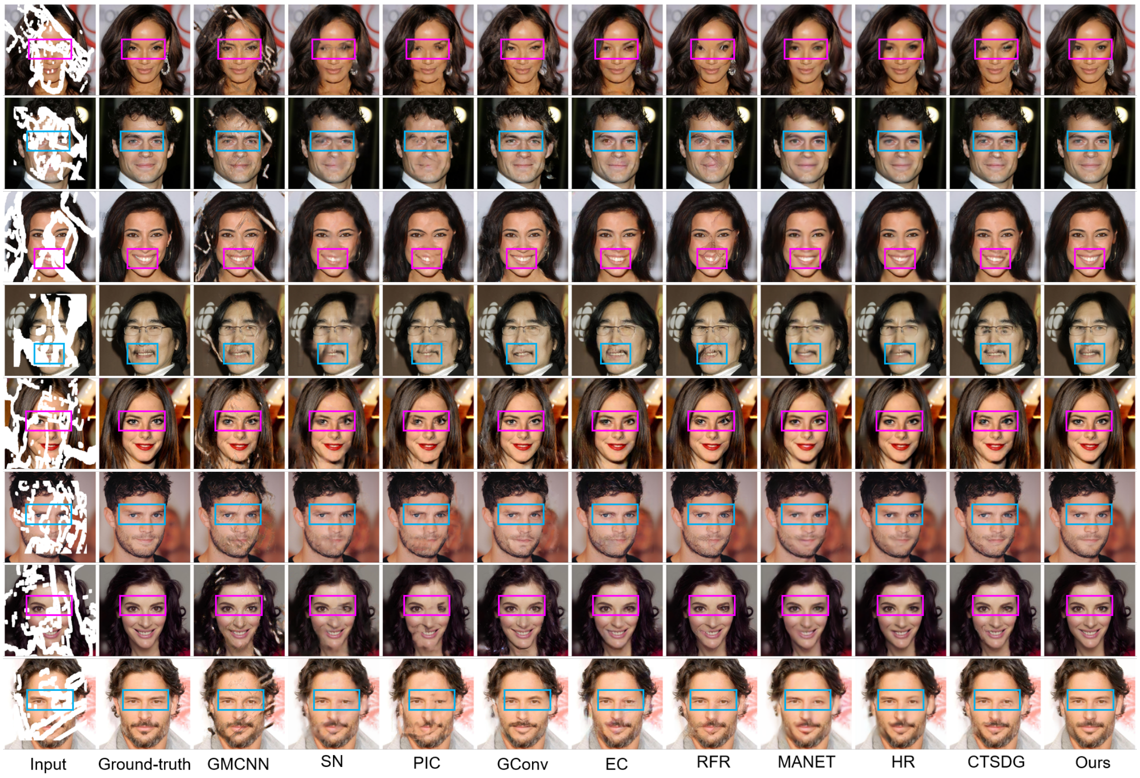


Figure 3.3: Qualitative results comparison of the proposed and existing methods (GMCNN [9], SN [10], PIC [11], GConv [12], EC [13], RFR [14], MANET [15], HR [8], CTSDG [16]) for image inpainting on CelebA-HQ dataset.

where, $\mathbb{C}$ is the Canny edge detection operator. *The effect of utilizing the Canny edge detection operator over Sobel operator for edge loss calculation is analysed in Section 3.1.3.* Along with the above mentioned losses, the $L_1$ loss is used for both the coarse and fine stages. For calculating the end-to-end loss from coarse and fine stages, we have taken the average of losses from both the stages. So, the total loss used for training the proposed network is weighted sum of all losses as given in:

$$L_{Total_s} = \lambda_1 L_{1_s} + \lambda_{Adv} L_{Adv_s} + \lambda_{Per} L_{Per_s} + \lambda_{Edge} L_{Edge_s} \tag{3.9}$$

We use the weights $\lambda_1 = 1$, $\lambda_{Adv} = 0.01$, $\lambda_{Per} = 0.4$, $\lambda_{Edge} = 0.25$ for training the network.

### 3.1.3 Experimental Analysis

Extensive experimental analysis is carried out on the proposed architecture over existing state-of-the-art methods with different datasets and different ratios of masks. The ablation study is also conducted to show the effectiveness of the proposed layer and the use of loss while training. A user study is carried out to show the superiority of inpainted images from proposed architecture over existing state-of-the-art methods (*the quantitative and qualitative results are taken from the available source codes and pre-trained weights*

Table 3.3: Comparison of the proposed method with state-of-the-art methods for image inpainting on Places2 dataset with NVIDIA masks from [6].

| Mask Ratio | Method | Publication | PSNR↑ | SSIM↑ | $L_1$ ↓ | LPIPS↓ | FID↓ |
|---|---|---|---|---|---|---|---|
| 0.1-0.2 | SN [10] | ECCV-18 | 25.69 | 0.831 | 3.150 | 0.180 | 15.492 |
| | GMCNN [9] | NIPS-18 | 25.74 | 0.861 | 3.895 | 0.140 | 15.082 |
| | PIC [11] | CVPR-19 | 26.32 | 0.937 | 1.096 | 0.183 | 9.588 |
| | Gconv [12] | ICCV-19 | 26.05 | 0.894 | 2.985 | 0.172 | 9.867 |
| | EC [13] | CVPRW-19 | 27.28 | 0.943 | 1.060 | 0.185 | 6.094 |
| | RFR [14] | CVPR-20 | 28.28 | 0.954 | 1.033 | 0.128 | 5.149 |
| | MANET [15] | PR-20 | 28.84 | 0.910 | 1.245 | 0.078 | 5.216 |
| | HR [8] | WACV-21 | 28.79 | 0.920 | 2.017 | 0.132 | 5.235 |
| | CTSDG [16] | ICCV-21 | 29.69 | 0.957 | 0.924 | 0.089 | 5.129 |
| | **Ours** | **PR-22** | **30.82** | **0.968** | **0.912** | **0.057** | **5.020** |
| 0.3-0.4 | SN [10] | ECCV-18 | 22.42 | 0.754 | 5.307 | 0.261 | 28.511 |
| | GMCNN [9] | NIPS-18 | 22.65 | 0.796 | 5.471 | 0.246 | 25.994 |
| | PIC [11] | CVPR-19 | 20.77 | 0.771 | 3.447 | 0.237 | 34.240 |
| | Gconv [12] | ICCV-19 | 23.45 | 0.850 | 4.895 | 0.225 | 21.453 |
| | EC [13] | CVPRW-19 | 22.27 | 0.879 | 2.506 | 0.227 | 18.935 |
| | RFR [14] | CVPR-20 | 23.28 | 0.875 | 2.534 | 0.219 | 15.540 |
| | MANET [15] | PR-20 | 23.96 | 0.885 | 2.863 | 0.199 | 15.974 |
| | HR [8] | WACV-21 | 24.02 | 0.877 | 4.169 | 0.215 | 18.184 |
| | CTSDG [16] | ICCV-21 | 23.50 | 0.876 | 2.503 | 0.209 | 16.879 |
| | **Ours** | **PR-22** | **24.71** | **0.899** | **2.410** | **0.161** | **15.220** |
| 0.4-0.5 | SN [10] | ECCV-18 | 19.66 | 0.682 | 6.719 | 0.373 | 59.790 |
| | GMCNN [9] | NIPS-18 | 19.31 | 0.714 | 6.537 | 0.325 | 58.571 |
| | PIC [11] | CVPR-19 | 20.64 | 0.728 | 5.133 | 0.324 | 56.870 |
| | Gconv [12] | ICCV-19 | 20.98 | 0.762 | 5.892 | 0.301 | 50.450 |
| | EC [13] | CVPRW-19 | 21.00 | 0.785 | 5.331 | 0.299 | 49.650 |
| | RFR [14] | CVPR-20 | 21.53 | 0.768 | 5.846 | 0.286 | 48.250 |
| | MANET [15] | PR-20 | 21.98 | 0.792 | 4.622 | 0.270 | 39.876 |
| | HR [8] | WACV-21 | 21.86 | 0.778 | 5.230 | 0.275 | 45.127 |
| | CTSDG [16] | ICCV-21 | 22.01 | 0.830 | 4.952 | 0.273 | 39.110 |
| | **Ours** | **PR-22** | **23.12** | **0.848** | **3.480** | **0.222** | **34.071** |

*provided by respective authors*).

### Quantitative Result Analysis

The evaluation of the proposed method is performed on three publicly available datasets CelebA-HQ, Places2 and Paris_SV for image inpainting with the masks from [6]. Table 3.2, 3.3 and 3.4 show the comparison of proposed method with state-of-the-art methods: multi-column image inpainting (GMCNN) [9], Shift-Net (SN) [10], pluralistic image inpainting (PIC) [11], gated convolutions (GConv) [12], EdgeConnect (EC) [13], recurrent feature reasoning (RFR) [14], hyper-realistic image inpainting with hyper-graphs (HR) [8], MANET [15] and CTSDG [16] in terms of peak signal-to-noise ratio (PSNR), structural similarity index (SSIM), mean $L_1$ error, Fréchet inception distance (FID) and Learned Perceptual Image Patch Similarity (LPIPS)[1]. The comparison is carried out for irregular

[1] The quantitative values are calculated from the results obtained with the source code with pre-trained weights provided by respective authors. For MANET, due to unavailability of pre-trained weights, the

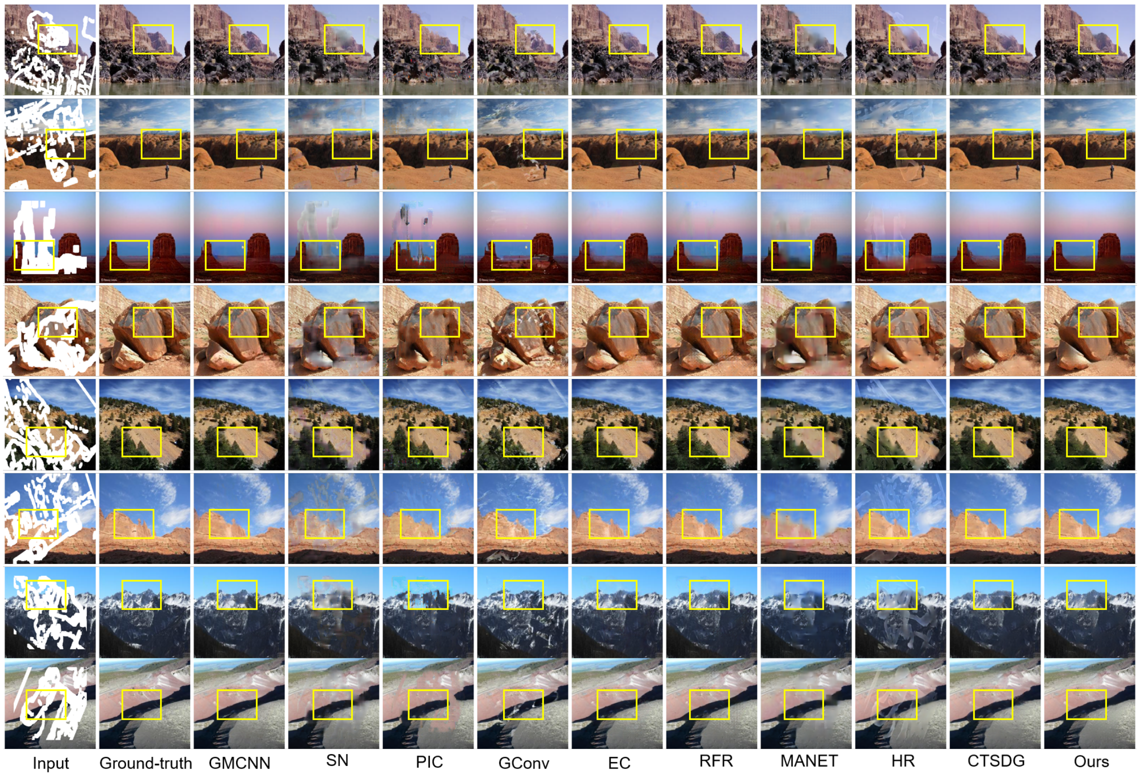


Figure 3.4: Qualitative results comparison of the proposed and existing methods (GMCNN [9], SN [10], PIC [11], GConv [12], EC [13], RFR [14], MANET [15], HR [8], CTSDG [16]) for image inpainting on Places2 dataset.

masks with three different mask ratios *i.e.,* $0.1 - 0.2$, $0.3 - 0.4$, $0.4 - 0.5$. As shown in Table 3.2, 3.3 and 3.4, it can be concluded that our proposed method easily compete the existing-state-of-the-art methods on all types of masks and ratios for image inpainting task. We give this credit to our proposed SPL.

### Qualitative Result Analysis

For visual comparison of proposed method with state-of-the-art methods we have used the source codes from the respective methods. We compare our results with GMCNN [9], SN [10], PIC [11], GConv [12], EC [13], RFR [14], HR [8] and CTSDG [16] *etc.* In Figure 3.3, we can clearly see that the proposed method generate efficient structural information of corrupted region of a face as compared to existing methods (*see the bounding boxes highlighting the consistent facial inpainting ability of proposed method*). Also, the Figure 3.4 depicts the superiority of the proposed method by reflecting spatially consistent inpainted regions in the images (*see the structure and edges of mountains in the bounding box*). Similarly, Figure 3.5 revels the ability of proposed method to generate spatially consistent structural information in the inpainted image. Here, the bounding boxes are used to highlight the effectiveness of proposed method over existing methods for inpainting

network is retrained with the code provided by respective author.

Table 3.4: Comparison of the proposed method with state-of-the-art methods for image inpainting on Paris_SV dataset with NVIDIA masks from [6].

| Mask Ratio | Method | Publication | PSNR↑ | SSIM↑ | $L_1$ ↓ | LPIPS↓ | FID↓ |
|---|---|---|---|---|---|---|---|
| 0.1-0.2 | SN [10] | ECCV-18 | 30.6 | 0.958 | 1.407 | 0.0865 | 22.038 |
| | GMCNN [9] | NIPS-18 | 30.63 | 0.968 | 1.027 | 0.0829 | 21.953 |
| | PIC [11] | CVPR-19 | 30.71 | 0.941 | 2.116 | 0.0727 | 21.321 |
| | Gconv [12] | ICCV-19 | 26.49 | 0.891 | 2.355 | 0.0625 | 19.992 |
| | EC [13] | CVPRW-19 | 30.87 | 0.944 | 2.796 | 0.0691 | 14.824 |
| | RFR [14] | CVPR-20 | 31.64 | 0.946 | 1.105 | 0.0623 | 16.620 |
| | MANET [15] | PR-20 | 32.38 | 0.957 | 0.890 | 0.0582 | 10.528 |
| | HR [8] | WACV-21 | 32.25 | 0.966 | 1.734 | 0.0652 | 16.171 |
| | CTSDG [16] | ICCV-21 | 32.50 | 0.956 | 0.912 | 0.0465 | 9.195 |
| | **Ours** | **PR-22** | **33.13** | **0.972** | **0.686** | **0.0463** | **8.516** |
| 0.3-0.4 | SN [10] | ECCV-18 | 22.49 | 0.779 | 3.109 | 0.2245 | 64.818 |
| | GMCNN [9] | NIPS-18 | 22.03 | 0.716 | 3.686 | 0.2108 | 63.902 |
| | PIC [11] | CVPR-19 | 24.86 | 0.741 | 4.346 | 0.2051 | 61.277 |
| | Gconv [12] | ICCV-19 | 22.15 | 0.757 | 4.808 | 0.1998 | 93.584 |
| | EC [13] | CVPRW-19 | 25.66 | 0.706 | 3.35 | 0.1994 | 45.482 |
| | RFR [14] | CVPR-20 | 26.19 | 0.799 | 2.767 | 0.01521 | 40.170 |
| | MANET [15] | PR-20 | 26.95 | 0.873 | 2.212 | 0.1448 | 51.946 |
| | HR [8] | WACV-21 | 27.95 | 0.859 | 2.874 | 0.1454 | 52.031 |
| | CTSDG [16] | ICCV-21 | 27.02 | 0.858 | 2.651 | 0.1455 | 32.348 |
| | **Ours** | **PR-22** | **27.80** | **0.907** | **2.63** | **0.1345** | **31.751** |
| 0.4-0.5 | SN [10] | ECCV-18 | 21.74 | 0.749 | 6.124 | 0.2801 | 91.685 |
| | GMCNN [9] | NIPS-18 | 22.25 | 0.738 | 5.041 | 0.2434 | 89.001 |
| | PIC [11] | CVPR-19 | 22.02 | 0.731 | 5.296 | 0.2321 | 82.492 |
| | Gconv [12] | ICCV-19 | 23.58 | 0.799 | 4.951 | 0.2215 | 79.525 |
| | EC [13] | CVPRW-19 | 23.86 | 0.812 | 3.563 | 0.2201 | 68.402 |
| | RFR [14] | CVPR-20 | 23.95 | 0.842 | 3.552 | 0.2123 | 65.152 |
| | MANET [15] | PR-20 | 24.980 | 0.843 | 3.524 | 0.2036 | 66.172 |
| | HR [8] | WACV-21 | 24.01 | 0.843 | 3.781 | 0.2036 | 65.318 |
| | CTSDG [16] | ICCV-21 | 24.17 | 0.843 | 3.108 | 0.1944 | 55.172 |
| | **Ours** | **PR-22** | **25.69** | **0.851** | **3.051** | **0.1885** | **52.140** |

the structures and fine edges.

From the above discussion, it is clear that the proposed method produce superior visual results as compared to existing state-of-the-art methods. We give this credit of generating efficient structural and spatially correlated effective information to our proposed spatial projection layer. Similarly, we give the credit of generating refined edge information in the inpainted images to the proposed use of Canny edge operator.

### Object Removal Application

Also, to verify the applicability of our proposed method, we have conducted an experiment with the real world object removal scenarios. For this analysis, we have considered the DAVIS-2016 [138] dataset. The qualitative results for object removal task are as shown in Figure 3.6. From this, it is clear that our proposed method gives comparative results with existing methods for object removal task.

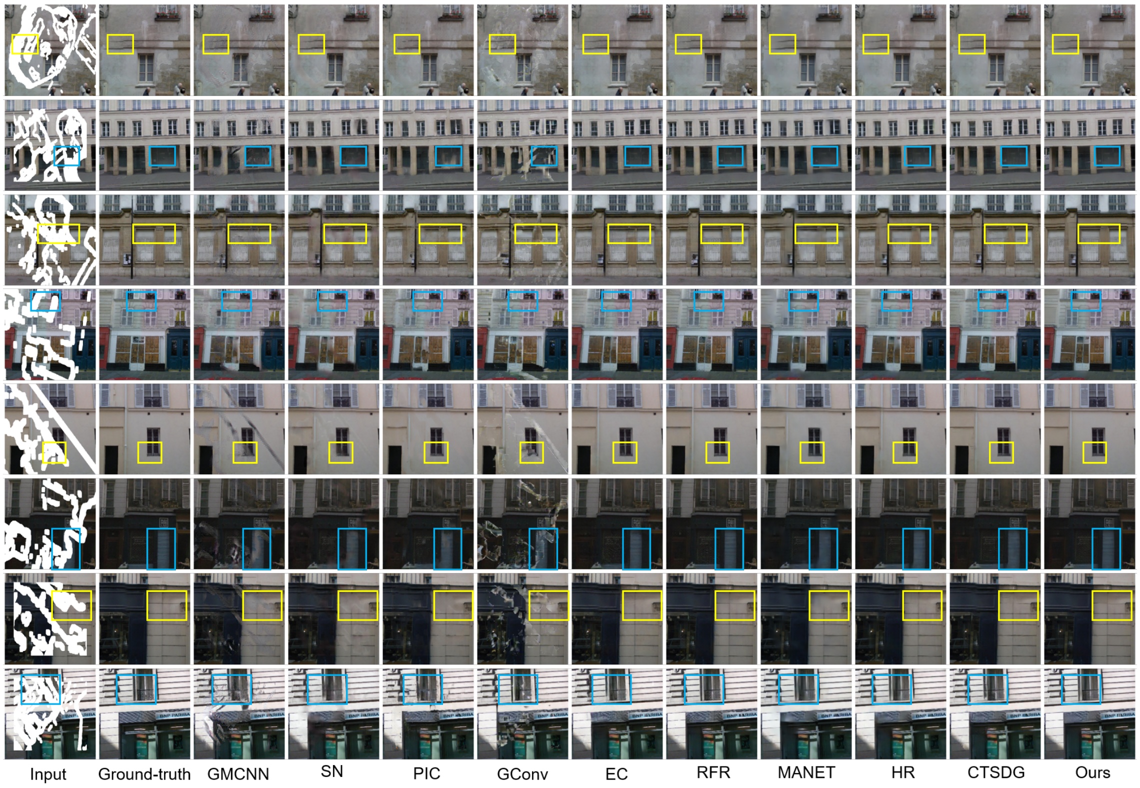


Figure 3.5: Qualitative results comparison of the proposed and existing methods (GMCNN [9], SN [10], PIC [11], GConv [12], EC [13], RFR [14], MANET [15], HR [8], CTSDG [16]) for image inpainting on Paris_SV dataset.

Table 3.5: Quantitative results of ablation study on CelebA-HQ dataset for $0.4 - 0.5$ mask ratio. *Note- (a) w/o SPL and w/o canny edge loss, (b) w/o SPL and w/i canny edge loss, (c) w/i SPL and w/o canny edge loss, and (d) the proposed w/i SPL and w/i canny edge loss.*

| Mask Ratio | Metric | a | b | c | d |
|---|---|---|---|---|---|
| 0.1-0.2 | PSNR↑ | 32.42 | 33.02 | 33.89 | **34.27** |
| | SSIM↑ | 0.921 | 0.942 | 0.956 | **0.983** |
| | $L_1$ ↓ | 1.180 | 0.950 | 0.790 | **0.540** |
| | LPIPS↓ | 0.095 | 0.078 | 0.058 | **0.023** |
| | FID↓ | 4.740 | 3.570 | 2.890 | **1.870** |
| 0.3-0.4 | PSNR↑ | 26.65 | 27.06 | 27.98 | **28.86** |
| | SSIM↑ | 2.28 | 1.95 | 1.08 | **0.944** |
| | $L_1$ ↓ | 2.970 | 2.560 | 2.040 | **1.240** |
| | LPIPS↓ | 0.1982 | 0.158 | 0.105 | **0.0732** |
| | FID↓ | 9.790 | 8.250 | 7.620 | **6.570** |
| 0.4-0.5 | PSNR↑ | 24.48 | 25.03 | 25.86 | **26.37** |
| | SSIM↑ | 0.878 | 0.883 | 0.895 | **0.911** |
| | $L_1$ ↓ | 3.158 | 3.159 | 2.047 | **1.970** |
| | LPIPS↓ | 0.2185 | 0.1958 | 0.1484 | **0.1081** |
| | FID↓ | 10.614 | 10.863 | 9.234 | **8.650** |

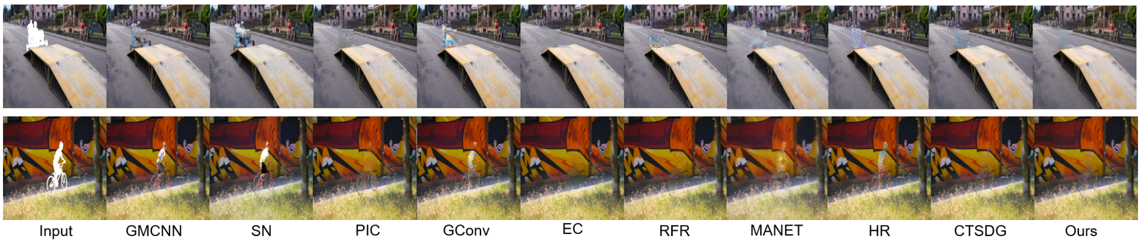


Figure 3.6: Qualitative results comparison of the proposed and existing methods (GMCNN [9], SN [10], PIC [11], GConv [12], EC [13], RFR [14], MANET [15], HR [8], CTSDG [16]) for object removal task.

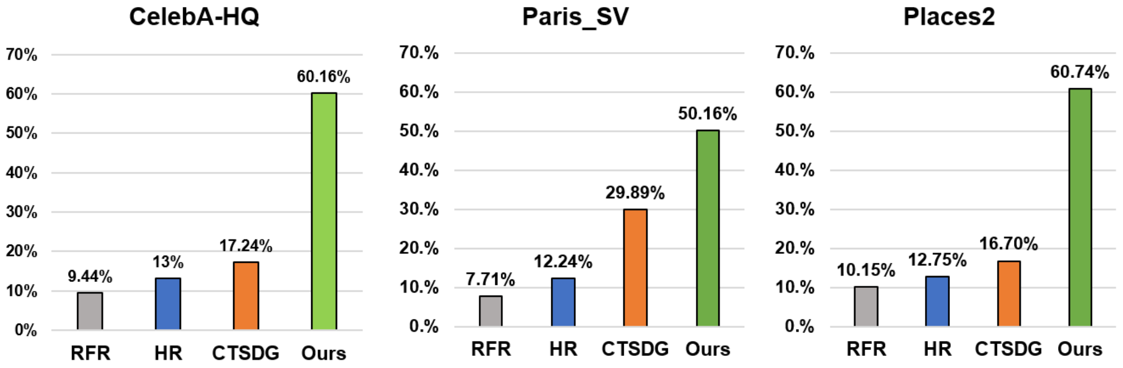


Figure 3.7: Inpainting analysis via user study (*Note- RFR [14], HR [8], CTSDG [16]*).

### User Study

To verify the effectiveness of our proposed method, we have performed an experiment of user study. For this study, we have used all the three datasets CelebA-HQ, Places2 and Paris_SV with random masks. We consider 20 images from each dataset. This creates total 60 different questions for user study. For this analysis, we have considered existing state-of-the-art RFR [14], HR [8], CTSDG [16] methods. We have shared this user study questionnaire to 30 users with and without technical background and asked to vote more realistic inpainting for each question. The analysis of the user study is depicted in Figure 3.7. The user study analysis in Figure 3.7 proves realistic inpainting ability of the proposed method over existing state-of-the-art methods.

### Ablation study

*Effect of SPL and Canny Edge Loss:* To examine effectiveness of the proposed spatial projection layer (SPL) and use of edge loss with Canny operator, we conducted the experiments as: (a) without (w/o) SPL and w/o Canny edge loss, (b) w/o SPL and with (w/i) Canny edge loss, (c) w/i SPL and w/o Canny edge loss, and the proposed (d) w/i SPL and w/i Canny edge loss. First, we examine **Does the SPL contribute toward spatially correlated results?** This is evaluated with the experiment (a), (b) and (c). From Figure 3.8 (a), (b) and (c), we can see that the inpainted results with SPL

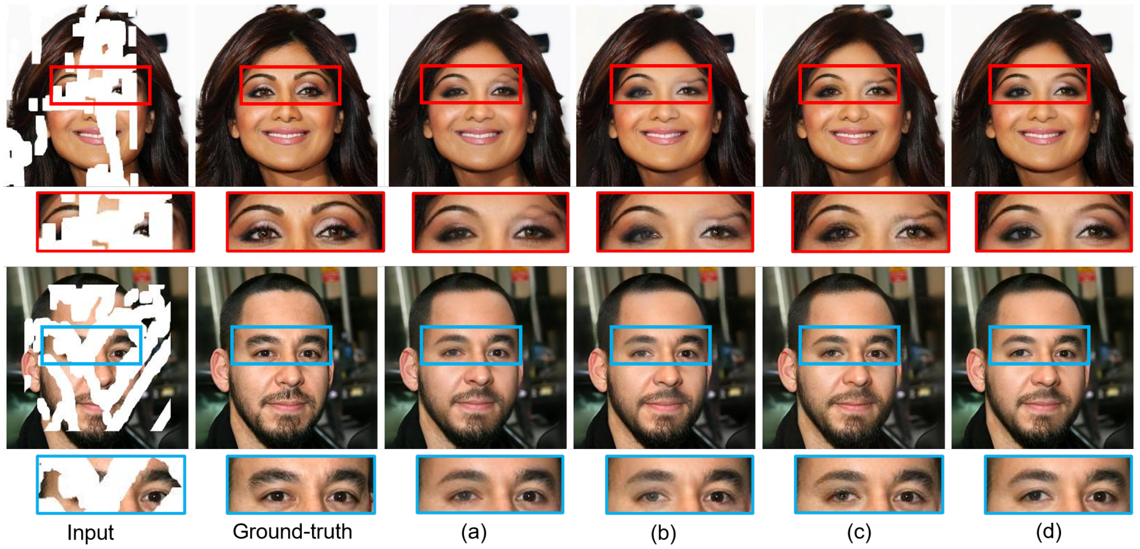


Figure 3.8: Qualitative results comparison of ablation study: (a) w/o SPL and w/o Canny edge loss, (b) w/o SPL and w/i Canny edge loss, (c) w/i SPL and w/o Canny edge loss, and the proposed, and (d) w/i SPL and w/i Canny edge loss.

Table 3.6: Quantitative analysis of placing the proposed SPL in coarse (Stage 1) and fine stage (Stage 2) in the proposed architecture for image inpainting on CelebA-HQ dataset.

| Mask Ratio | 0.1-0.2 | | 0.3-0.4 | | 0.4-0.5 | |
|---|---|---|---|---|---|---|
| Metric | Stage 1 | **Stage 2** | Stage 1 | **Stage 2** | Stage 1 | **Stage 2** |
| PSNR ↑ | 32.58 | **34.27** | 27.65 | **28.86** | 25.04 | **26.37** |
| SSIM ↑ | 0.961 | **0.983** | 0.937 | **0.944** | 0.901 | **0.911** |
| L1 ↓ | 1.18 | **0.54** | 1.97 | **1.24** | 2.52 | **1.97** |
| LPIPS ↓ | 0.085 | **0.023** | 0.0982 | **0.0732** | 0.1526 | **0.1081** |
| FID ↓ | 2.74 | **1.87** | 8.79 | **6.57** | 10.18 | **8.65** |

generate spatially correlated information. Also, the quantitative evaluation is provided in Table 3.5. The results of Table 3.5 and Figure 3.8 show the efficiency of the SPL at producing the spatially consistent information in the hole region.

Next, we evaluate, **Whether the use of Canny edge operator contribute to the true edges in the inpainted image?** For this evaluation, we considered the experiment with using Sobel operator for edge loss calculation. We can easily see in Figure 3.8 (d), the use of Canny loss generated the true edges (with Canny operator) unlike Figure 3.8 (c) where some false edges are introduced (with Sobel operator).

*Effect of the Proposed SPL in Stage 1 and Stage 2:* The effect of utilizing the proposed spatial projection layer (SPL) in stage 1 and stage 2 is analysed in this study. The quantitative comparison is given in Table 3.6 and the qualitative comparison is given in Figure 3.9. Stage I: In this case, the SPL is used in the first stage. Stage II (proposed method): In this case, the SPL is used in the second stage.

*Analysis of the Proposed SPL and Existing Self-attention:* Also, we have verified the

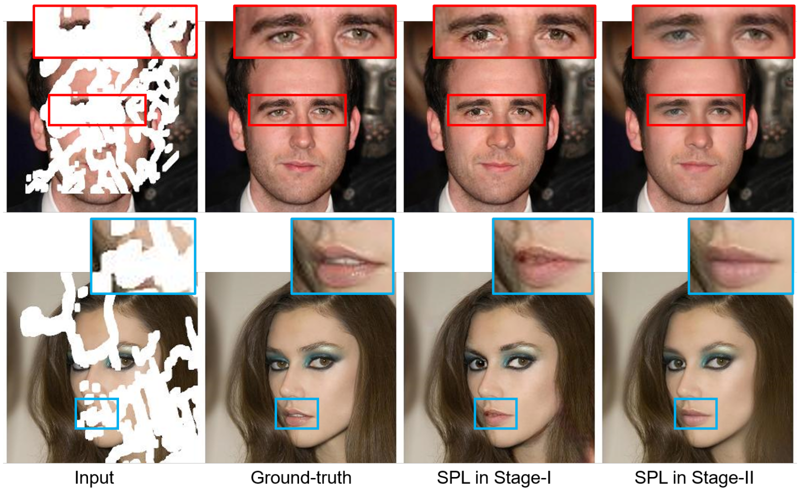


Figure 3.9: Qualitative results comparison of the proposed SPL placed in coarse (Stage I) and fine stage (Stage II: Proposed Method) of the architecture for image inpainting.

Table 3.7: Analysis on placing existing self-attention (SA) instead of proposed SPL for image inpainting on CelebA-HQ dataset.

| Mask Ratio | 0.1-0.2 | | 0.3-0.4 | | 0.4-0.5 | |
|---|---|---|---|---|---|---|
| Metric | SA | **SPL** | SA | **SPL** | SA | **SPL** |
| PSNR ↑ | 33.08 | **34.27** | 27.45 | **28.86** | 25.86 | **26.37** |
| SSIM ↑ | 0.972 | **0.983** | 0.934 | **0.944** | 0.852 | **0.911** |
| L1 ↓ | 0.86 | **0.54** | 1.98 | **1.24** | 3.158 | **1.97** |
| LPIPS ↓ | 0.048 | **0.023** | 0.0985 | **0.0732** | 0.1245 | **0.1081** |
| FID ↓ | 2.06 | **1.87** | 7.85 | **6.57** | 9.824 | **8.65** |

Table 3.8: Analysis on different losses for training of the proposed network for image inpainting on CelebA-HQ dataset. *Note: Different combinations ($L_a$, $L_b$, $L_c$, $L_T$) of total loss $L_{Total}$ are analysed where,* $L_a \rightarrow \lambda_1 L_1$; $L_b \rightarrow L_a + \lambda_{Adv} L_{Adv}$; $L_c \rightarrow L_b + \lambda_{Per} L_{Per}$; $L_T \rightarrow L_c + \lambda_{Edge} L_{Edge}$

| Mask Ratio | Metric | $L_a$ | $L_b$ | $L_c$ | $L_T$ |
|---|---|---|---|---|---|
| 0.1-0.2 | PSNR↑ | 32.86 | 33.27 | 33.85 | **34.27** |
| | SSIM↑ | 0.961 | 0.964 | 0.975 | **0.983** |
| | $L_1$ ↓ | 1.570 | 1.060 | 0.920 | **0.540** |
| | LPIPS↓ | 0.097 | 0.085 | 0.052 | **0.023** |
| | FID↓ | 3.010 | 2.560 | 2.110 | **1.870** |
| 0.3-0.4 | PSNR↑ | 26.61 | 27.12 | 27.98 | **28.86** |
| | SSIM↑ | 0.921 | 0.929 | 0.935 | **0.944** |
| | $L_1$ ↓ | 2.860 | 2.080 | 1.940 | **1.240** |
| | LPIPS↓ | 0.1125 | 0.1054 | 0.0915 | **0.0732** |
| | FID↓ | 8.260 | 7.620 | 7.080 | **6.570** |
| 0.4-0.5 | PSNR↑ | 24.86 | 25.94 | 26.02 | **26.37** |
| | SSIM↑ | 0.875 | 0.886 | 0.892 | **0.911** |
| | $L_1$ ↓ | 3.010 | 2.950 | 2.580 | **1.970** |
| | LPIPS↓ | 0.1821 | 0.1614 | 0.1572 | **0.1081** |
| | FID↓ | 10.050 | 9.720 | 9.140 | **8.650** |

**effect of proposed SPL compared to existing self-attention**. Table 3.7 shows the quantitative comparison of proposed SPL versus existing self-attention in the proposed architecture for image inpainting. From Table 3.7, we can clearly observe that the proposed SPL gives superior performance over existing self-attention for image inpainting.

*Effect of different training losses:* In this study, we analyse the effect of different losses used to train the proposed network for image inpainting. The study is carried out on CelebA-HQ dataset. Various combinations of the losses are considered to train the network. The quantitative comparison of proposed network trained with different combinations is provided in Table 3.8. From Table 3.8 it is clear that, the combination of $L_1$, adversarial ($L_{Adv}$), perceptual ($L_{Per}$) and edge ($L_{Edge}$) loss leads to the better outcome for image inpainting task.

## 3.2 Nested Deformable Multi-head Attention for Facial Image Inpainting

The attention mechanism in general is an effective approach for image inpainting task. Since it helps the network to effectively extract the features from valid locations for inpainting the holes. Also, to fill the holes of large size, it is necessary to have a varying receptive field in consideration while extracting the features. The multi-head attention urges to weigh the feature maps with the valid features. Considering these points, in this work, we propose a nested deformable multi-head attention layer (NDMAL) to transfer the encoder features for effective reconstruction while considering diverse receptive fields. Inspired by the success of linear unified nested attention [139] for a sequence modelling task, we propose a nested deformable multi-head attention layer for image inpainting task. Unlike [139], we consider encoder and decoder features as packed and unpacked inputs. Though, encoder and decoder features are inputs to the multi-head attention, our proposed layer has linear complexity. Since we utilize the channel-wise attention instead of spatial attention. The proposed NDMAL helps the network to effectively extract the features from the valid region (background) to fill the holes. Further, we propose deformable multi-head attention (DMHA) for extracting decoder features from diverse fields which are then merged with the skip features from the encoder. Also, a gated feed-forward layer is utilized to again pass the weighted features for reconstruction. Resembling the encoder skip features as a query sequence, packed attention is calculated, called packed context. This packed context is again processed through DMHA with query sequence as decoder features and the unpacked context is generated. Both of these packed and unpacked contexts are merged and then forwarded to the next layer. These packed and unpacked context features assist in the effective reconstruction of the inpainted image. The main contributions of our work are:

- Formulating a lightweight architecture consisting of novel transformer layer for facial image inpainting.
- We propose a nested deformable multi-head attention transformer layer (NDMAL) to effectively fuse the encoder and decoder features. The use of NDMAL allows the network to effectively capture long term dependencies and to extract the valid features from maximum receptive fields.
- We propose the analysis of inpainting methods on seen and unseen types of masks.

The ablation study is carried out to verify the efficiency of the proposed NDMAL. Comparative analysis of the proposed approach on CelebA-HQ dataset corrupted with masks from two different datasets and Places2 dataset proves its efficiency for image inpainting task.

### 3.2.1 Proposed Framework

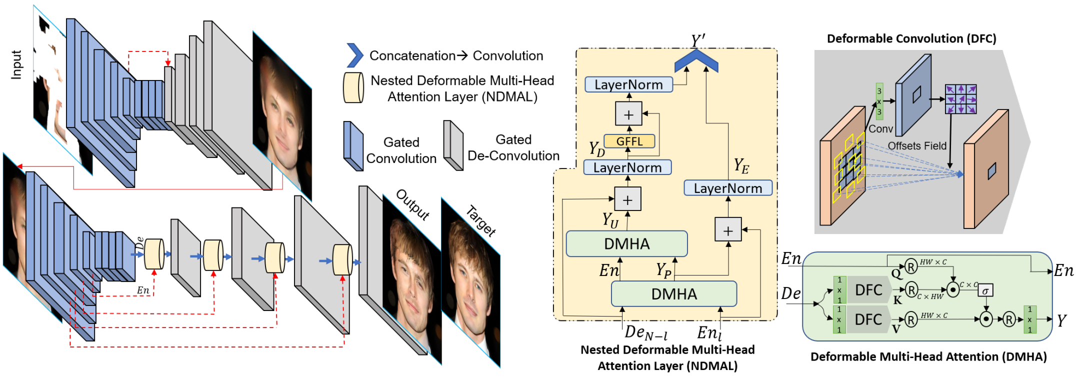

Input
En
Output
Target
Concatenation→ Convolution
Nested Deformable Multi-Head Attention Layer (NDMAL)
Gated Convolution
Gated De-Convolution
Y′
LayerNorm
GFFL
$Y_D$
LayerNorm
$Y_E$
LayerNorm
$Y_U$
DMHA
En
$Y_P$
DMHA
$De_{N-l}$
$En_l$
Nested Deformable Multi-Head Attention Layer (NDMAL)
Deformable Convolution (DFC)
3x3
Conv
Offsets Field
En
De
Q
K
V
DFC
DFC
HW × C
C × HW
C × C
σ
1x1
En
Y
Deformable Multi-Head Attention (DMHA)

Figure 3.10: Proposed architecture for image inpainting. We propose a nested deformable multi-head attention transformer layer (NDMAL) to focus on large receptive fields with long term dependencies. The proposed layer consists of single layer in turn reducing the computational complexity of the network.

In this section we first introduce in general multi-head attention used in the transformer [140], the linear unified nested attention [139] and then we put a light on the proposed nested deformable multi-head attention layer (NDMAL) used for image inpainting task.

#### Transformer with Self Attention

The multi-head attention [140] mapping $A \in \mathbb{R}^{n\times p} \times B \in \mathbb{R}^{m\times p} \rightarrow Y \in \mathbb{R}^{n\times p}$ is generally formulated as:

$$Y = Attn(A, B) = \sigma(\frac{A\phi_q(B\phi_k)^T}{\sqrt{d_k}})B\phi_v \tag{3.10}$$

where, $A$ and $B$ are the query and context sequences with length $n$ and $m$ respectively, $\sigma$ is the Softmax activation, $p$ is the embedding dimension, $\phi_q$, $\phi_k$ and $\phi_v$ are the trainable parameters used to project the input into query, key and values, $d_k$ is dimension of key. In [140] for multi-head attention $A = B$ is considered, called as *self-attention*. The output of this multi-head attention *i.e.,* self-attention is fed to position wise feed-forward layer followed by layer normalization. The final output of the transformer ($Y'$) is given as:

$$Y' = \eta(FFN(Y_A) + Y_A) \tag{3.11}$$

where, $\eta$ is `LayerNormalization`, $Y_A = \eta(Y + A)$. These transformer layers are sequentially utilized $l$ times in each block. The feed-forward network (FFN) is independently applied on each position and layer normalization controls the gradient scales [140]. The SA generally has quadratic complexity. The computational load of the SA is reduced with applying the SA on small spatial *window size*, $ws = 8 \times 8$ [141, 142] instead of global attention.

### Linear Unified Nested Attention

The linear unified nested attention [139] (LUNA) deals with the quadratic memory and computational complexity of transformers ($\mathcal{O}(mn)$) (Section 3.2.1) by introducing an extra input sequence of fixed length in order to have two outputs. This in turn gives linear complexity to the transformer layer. The pack ($Y_P$) and unpack ($Y_U$) attentions are introduced as:

$$Y_P = Attn(C, B); \qquad Y_U = Attn(A, Y_P) \tag{3.12}$$

where, $C \in \mathbb{R}^{l \times p}$ is an extra input sequence with fixed length $l$. The packed and unpacked attentions have the complexity of $\mathcal{O}(lm)$ and $\mathcal{O}(ln)$. So, the LUNA takes three inputs in general ($A$, $B$ and $C$) and produces a packed and unpacked attention as output. The LUNA layers take these attentions to further process via FFN and `LayerNormalization` as:

$$\begin{aligned} Y_P, Y_U &= LunaAttn(A, C, B) \\ Y_A, C_A &= \eta(Y_P + A), \eta(Y_U + C) \\ Y', C' &= \eta(FFN(Y_A) + Y_A), C_A \end{aligned} \tag{3.13}$$

where, $Y'$ and $C'$ are the outputs of the LUNA Layer.

### Proposed Nested Deformable Multi-head Attention

In combination to both of the multi-head attention (Section 3.2.1) and LUNA attention (Section 3.2.1), we propose a nested deformable multi-head attention for the task of image inpainting (Figure 3.10). The LUNA attention provides an extra input with actual inputs to have linear complexity. Applying the self-attention to the image inpainting task may provide relative contextual information either from the encoded features or from decoder features. Whereas, in our proposed approach, we provide the decoder ($De$) and skip connection features from the encoder ($En$) as input. Considering both the features from the encoder and decoder may allow delving into the valid feature space efficiently. Also, in order to extract maximum receptive field from the decoder processed features, we leverage the deformable convolution layer [143] unlike [140] and [139]. Here, we consider the encoder features to be the context information provided to the decoder for effective reconstruction. So, the proposed deformable multi-head attention (DMHA) is formulated as:

$$Y = DMHA(De_{N-l}, En_l) = \sigma\left(En_l \phi_q (De_{N-l} \phi_k^{df})^T\right) De_{N-l} \phi_v^{df} \tag{3.14}$$

where, $\phi^{df}$ shows the `deformable convolution` applied to the decoder features to delve into the maximum receptive fields, $l \in (1, 4)$ is the number of layers and $N = 5$ (*see DMHA in Figure 3.10*). In deformable convolution, the normal grid $O = \{(-1,-1), (-1,0), ..., (0,1), (1,1)\}$ is augmented with the offsets $\{\triangle p_n | n = 1, ......, P\}$,

$P = |O|$. So, for each location $p_0$ in the output feature map $\phi^{df}$,

$$\phi^{df}(p_0) = \sum_{p_n \in O} w(p_n).x(p_0 + p_n + \triangle p_n) \tag{3.15}$$

Further, we introduce the nested deformable attention mechanism to increase the required receptive field and to focus on long-term dependencies. Also, nesting of DMHA makes sense that, it can capture sufficient contextual information. The packed $(Y_P)$ and unpacked $(Y_U)$ outcomes of the nested deformable attention are given as:

$$\begin{aligned} Y_P &= DMHA(De_{N-l}, En_l) \\ Y_U &= DMHA(Y_P, De_{N-l}) \end{aligned} \tag{3.16}$$

Since we consider the encoder layer features with the input sequence, it will be able to pack the global context of the input efficiently. The packed and unpacked outputs are then forwarded to layer normalization and gated feed-forward layer (GFFL). The output $(Y')$ of proposed NDMAL is given as:

$$\begin{aligned} Y_E, Y_D &= \eta(Y_P + En_l), \eta(Y_U + De_{N-l}) \\ Y' &=< \eta(GFFL(Y_D) + Y_D), Y_E > \end{aligned} \tag{3.17}$$

where, $< . >$ indicates concatenation operation. The GFFL is the gated feed forward layer which is used to suppress any undesired features if present. The GFFL is represented as:

$$GFFL(f_{in}) = \phi(f_{in}) + \mathbb{G}(\psi(f_{in})) \tag{3.18}$$

where, $\mathbb{G}$ is `GELU` activation function, $\phi$ and $\psi$ are learn-able functions (i.e., convolution layers).

#### Overall Architecture

The overall architecture of the proposed approach is visualized in Figure 3.10. We follow a coarse-to-fine architecture. The purpose behind the coarse-to-fine architecture is to forward the coarse output features through the proposed NDMAL as a query to provide sufficient contextual information. So that the network will be able to capture long-term dependencies effectively. The proposed NDMAL is utilized in the fine stage which takes input from the encoder layer and considers it as a query to the respective decoder feature key and values. Also, the packed attention in the NDMAL is calculated with respect to the encoder skip inputs which is then concatenated with the processed unpacked attention. The concatenation of both allows to preserve the valid content efficiently.

The encoder and decoder layers of both the coarse and fine stages are designed with the `gated convolution` layer followed by a `LeakyReLu` activation. The successive encoder

| Configuration (Parameters) | PSNR | SSIM | $L_1$ | LPIPS | FID |
|---|---|---|---|---|---|
| SA on En Feat (3.61M) | 24.25 | 0.842 | 4.259 | 0.162 | 9.482 |
| SA on De Feat (3.61M) | 24.98 | 0.857 | 4.008 | 0.151 | 9.106 |
| +Nested -Deformable (3.62M) | 27.68 | 0.915 | 3.007 | 0.104 | 7.864 |
| -Nested +Deformable (3.85M) | 26.28 | 0.897 | 3.856 | 0.122 | 8.567 |
| **Proposed Network (4.12M)** | **28.19** | **0.931** | **2.575** | **0.082** | **6.844** |

Table 3.9: Quantitative comparison for different configurations of the proposed network for image inpainting on 0.01 − 0.6 mask ratio on CelebA-HQ dataset (*Note: + indicates inclusion and - indicates exclusion of particular block, SA is self-attention, En Feat and De Feat are encoder and decoder features respectively*).

layers at the bottleneck of the coarse stage allow focusing on the different receptive fields which produce an approximate output. This coarse output is then fed to the fine stage which includes the proposed NDMAL. The overall architecture with effective usage of NDMAL generates faithful inpainted results. As we are considering the deformable multi-head attention, it may help the network to extract information from maximum receptive fields. Also, the nested multi-head attention applied to the encoded and decoded features may help to capture the long-term dependencies. So, unlike existing transformer architectures, our proposed NDMAL consists of only one block with $ws = 8 \times 8$. This helps to reduce the computational cost of our proposed inpainting network. Though the two inputs to the proposed NDMAL are having the length of $n, m$, it preserves the linear complexity. This is because we apply the attention channel-wise instead of spatially [32]. So, the attention will effectively encode the global context by computing the cross-covariance across the channels. This also reduces the necessity of an extra input with constant length ($l$) like [139].

### 3.2.2 Training of the Proposed Network

The proposed architecture is trained with the corrupted image and its mask as input and generates an inpainted image as output. The discriminator network is the same as that of [133]. While training, the image values are linearly scaled between the range $[0, 1]$. Weight parameters of the network are updated on NVIDIA DGX station having Tesla V100 $1 \times 16$ GB GPU with the batch size of 1 for 200 epochs (38 GPU Hours). The ADAM optimizer [144] with the learning rate of $2 \times 10^{-4}$, $\beta_1 = 0.5$ and $\beta_2 = 0.99$ is used. The total loss used for training the proposed network is weighted sum of all losses as given in:

$$L_{Total_s} = \lambda_1 L_{1_s} + \lambda_{Adv} L_{Adv_s} + \lambda_{Per} L_{Per_s} + \lambda_{Edge} L_{Edge_s} \tag{3.19}$$

We use the weights $\lambda_1 = 10$, $\lambda_{Adv} = 0.01$, $\lambda_{Per} = 0.5$, $\lambda_{Edge} = 0.3$ for training the network.

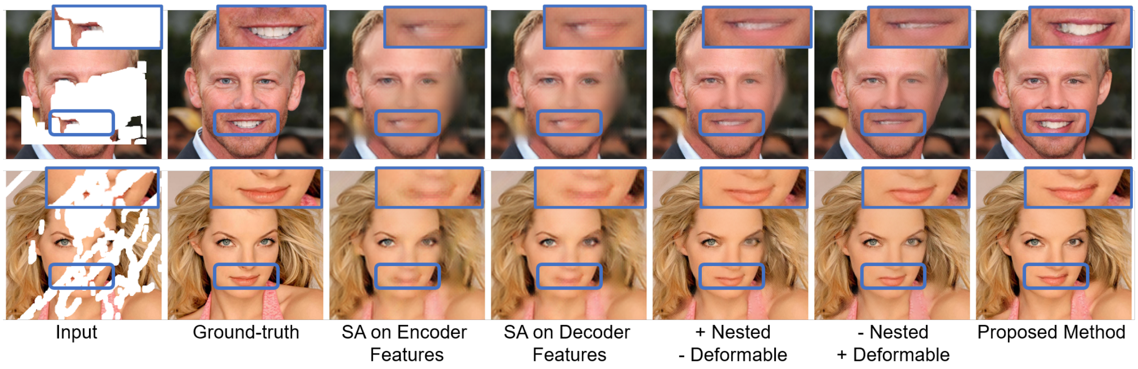


Figure 3.11: Analysis on different configurations of proposed method.

Table 3.10: Quantitative comparison of the proposed method (Ours) with the state-of-the-art methods on NVIDIA [18] masks for image inpainting on CelebA-HQ dataset.

| Mask Ratio | Method | Publication | PSNR↑ | SSIM↑ | $L_1$ ↓ | LPIPS↓ | FID↓ |
|---|---|---|---|---|---|---|---|
| 0.01-0.2 | SN [10] | ECCV-18 | 30.84 | 0.961 | 2.827 | 0.0601 | 4.134 |
| | GMCNN [9] | NIPS-18 | 30.54 | 0.957 | 2.867 | 0.0572 | 7.537 |
| | PIC [11] | CVPR-19 | 32.08 | 0.967 | 2.689 | 0.0382 | 4.042 |
| | Gconv [12] | ICCV-19 | 32.06 | 0.960 | 2.681 | 0.0384 | 4.309 |
| | EC [13] | CVPRW-19 | 32.04 | 0.973 | 3.108 | 0.0387 | 4.042 |
| | RFR [14] | CVPR-20 | 33.45 | 0.973 | 1.824 | 0.0453 | 2.516 |
| | HR [8] | WACV-21 | 33.28 | 0.976 | 1.925 | 0.0418 | 2.257 |
| | CTSDG [16] | ICCV-21 | 33.57 | 0.979 | 1.329 | 0.0280 | 2.105 |
| | MAT [17] | CVPR-22 | 33.56 | 0.977 | 1.147 | 0.1860 | 2.032 |
| | **Ours** | **WACV-23** | **33.99** | **0.982** | **1.017** | **0.0229** | **1.775** |
| 0.2-0.4 | SN [10] | ECCV-18 | 25.77 | 0.896 | 4.246 | 0.2091 | 10.643 |
| | GMCNN [9] | NIPS-18 | 24.49 | 0.894 | 4.120 | 0.1711 | 28.170 |
| | PIC [11] | CVPR-19 | 25.30 | 0.891 | 3.691 | 0.1772 | 14.376 |
| | Gconv [12] | ICCV-19 | 25.48 | 0.904 | 4.147 | 0.1668 | 11.010 |
| | EC [13] | CVPRW-19 | 26.30 | 0.901 | 3.194 | 0.1630 | 7.338 |
| | RFR [14] | CVPR-20 | 26.44 | 0.917 | 3.022 | 0.1414 | 11.767 |
| | HR [8] | WACV-21 | 26.76 | 0.935 | 3.213 | 0.1341 | 10.330 |
| | CTSDG [16] | ICCV-21 | 27.02 | 0.936 | 2.466 | 0.1020 | 7.516 |
| | MAT [17] | CVPR-22 | 27.13 | 0.931 | 2.466 | 0.0944 | 6.620 |
| | **Ours** | **WACV-23** | **27.43** | **0.948** | **2.382** | **0.0740** | **5.862** |
| 0.4-0.6 | SN [10] | ECCV-18 | 18.65 | 0.657 | 8.852 | 0.3690 | 61.160 |
| | GMCNN [9] | NIPS-18 | 18.74 | 0.744 | 6.747 | 0.4060 | 50.981 |
| | PIC [11] | CVPR-19 | 19.01 | 0.679 | 7.011 | 0.3451 | 49.120 |
| | Gconv [12] | ICCV-19 | 19.70 | 0.860 | 5.695 | 0.3017 | 34.940 |
| | EC [13] | CVPRW-19 | 21.33 | 0.809 | 5.828 | 0.2755 | 33.011 |
| | RFR [14] | CVPR-20 | 21.23 | 0.755 | 6.354 | 0.2551 | 30.650 |
| | HR [8] | WACV-21 | 22.04 | 0.831 | 5.345 | 0.2429 | 28.498 |
| | CTSDG [16] | ICCV-21 | 22.24 | 0.845 | 4.451 | 0.1910 | 14.371 |
| | MAT [17] | CVPR-22 | 22.55 | 0.847 | 4.402 | 0.1811 | 13.121 |
| | **Ours** | **WACV-23** | **23.14** | **0.858** | **4.326** | **0.1479** | **12.897** |

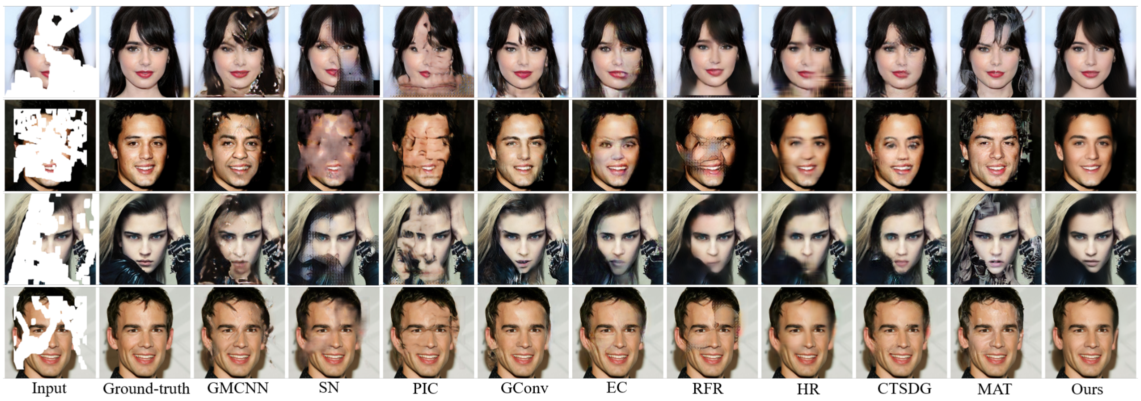


Figure 3.12: Qualitative comparison of the proposed method (Ours) with existing methods (GMCNN [9], SN [10], PIC [11], GConv [12], EC [13], RFR [14], HR [8], CTSDG [16], MAT [17]) on CelebA_HQ dataset for NVIDIA [18] mask.

### 3.2.3 Experimental Analysis

Here, we provide details of baselines, the ablation study on different configurations of proposed architecture, comparative and computational complexity analysis.

#### Baselines

To examine the efficiency, we consider the comparison of our proposed method with existing state-of-the-art methods for image inpainting : Shift-net (SN) [10], GMCNN [9], pluristic-image completion (PIC) [11], gated-convolutions (GConv) [12], edge-connect (EC) [13], recurrent feature reasoning (RFR) [14], hyper-graphs (HR) [8], contextual texture-structure dual generation (CTSDG) [16], and mask aware transformers (MAT) [17].

#### Ablation Study

In order to come up with an optimum architecture for image inpainting task, we carried out meticulous experiments with different combinations of our network. These experiments include, (a) considering the self attention (Section 3.2.1) applied on the encoder features and merged with decoder features (*SA on encoder features*) , (b) self attention (Section 3.2.1) applied on the decoder features and merged with encoder features (*SA on decoder features*), (c) applying the nested attention without deformable layer (similar to LUNA Section 3.2.1) (*+Nested -Deformable*), (d) applying the deformable multi-head attention without nested attention layers (*-Nested +Deformable*), (e) finally, applying the nested deformable multi-head attention layer (*+Nested +Deformable i.e., Proposed Network*) (*see Table 3.9*).

Purpose of this study is to compare quantitative and qualitative differences between different configurations of the proposed network. **We examine whether the**

Table 3.11: Quantitative comparison of the proposed method (Ours) with state-of-the-art methods on QD-IMD [7] masks for image inpainting on CelebA-HQ dataset.

| Mask Ratio | Method | PSNR↑ | SSIM↑ | $L_1$ ↓ | LPIPS↓ | FID↓ |
|---|---|---|---|---|---|---|
| 0.01-0.2 | EC [13] | 33.19 | 0.972 | 1.340 | 0.0404 | 2.929 |
| | RFR [14] | 33.45 | 0.973 | 1.824 | 0.0291 | 2.516 |
| | HR [8] | 33.28 | 0.974 | 1.143 | 0.0259 | 2.051 |
| | CTSDG[16] | 34.55 | 0.981 | 0.984 | 0.0186 | 1.913 |
| | MAT [17] | 34.66 | 0.982 | 0.945 | 0.0201 | 1.627 |
| | **Ours** | **35.05** | **0.989** | **0.818** | **0.0172** | **1.567** |
| 0.2-0.4 | EC [13] | 25.85 | 0.933 | 2.719 | 0.1319 | 7.561 |
| | RFR [14] | 26.92 | 0.939 | 2.513 | 0.1182 | 7.267 |
| | HR [8] | 27.68 | 0.948 | 2.443 | 0.1082 | 6.652 |
| | CTSDG[16] | 28.48 | 0.956 | 2.089 | 0.0540 | 6.262 |
| | MAT [17] | 28.62 | 0.957 | 1.930 | 0.0535 | 6.016 |
| | **Ours** | **28.94** | **0.961** | **1.807** | **0.0533** | **5.181** |
| 0.4-0.6 | EC [13] | 22.43 | 0.856 | 5.007 | 0.2136 | 19.543 |
| | RFR [14] | 22.93 | 0.868 | 4.754 | 0.1801 | 18.650 |
| | HR [8] | 23.37 | 0.871 | 4.039 | 0.1734 | 17.685 |
| | CTSDG[16] | 23.80 | 0.880 | 3.707 | 0.1308 | 16.111 |
| | MAT [17] | 24.03 | 0.887 | 3.637 | 0.1229 | 15.921 |
| | **Ours** | **24.56** | **0.895** | **3.508** | **0.1186** | **15.493** |

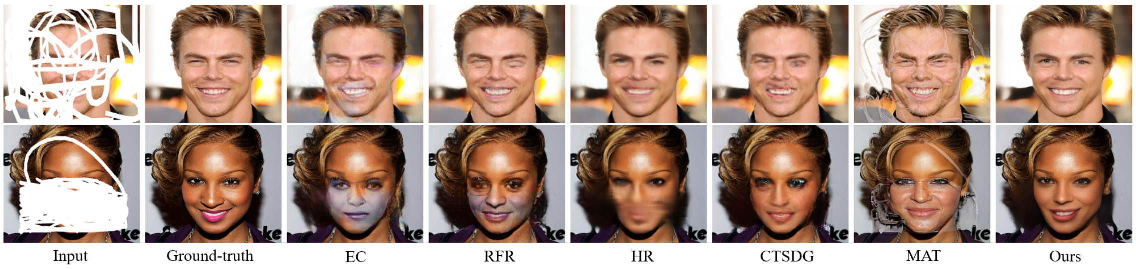


Figure 3.13: Qualitative comparison of the proposed method (Ours) with existing methods (EC [13], RFR [14], HR [8], CTSDG[16], MAT [17]) on CelebA_HQ dataset for unknown mask dataset QD-IMD [7].

**self-attention applied on either encoder or decoder features works better.** The existing self attention tries to extract the long term dependencies from the input feature maps. Applying it on the encoder or decoder features affects differently while reconstructing the image. Row 2 and 3 in Table 3.9 show the results for the configuration where the self attention is applied on encoder and decoder features respectively. From Table 3.9 and Figure 3.11, it is clear that, the self attention when applied with encoder (*row 2 of Table 3.9*) or decoder (*row 3 of Table 3.9*) feature map as input fails to produce efficient outcome in terms numeric and visual results. Inspired with the LUNA attention, **we ought to include the LUNA layer in the inapinting architecture to verify its ability to delve into the valid features**. Contrary to self attention, the results are improved quantitatively and also generate better structural information visually (*see row 4 in Table 3.9* and column 5 in Figure 3.11). The reason behind this might be, here

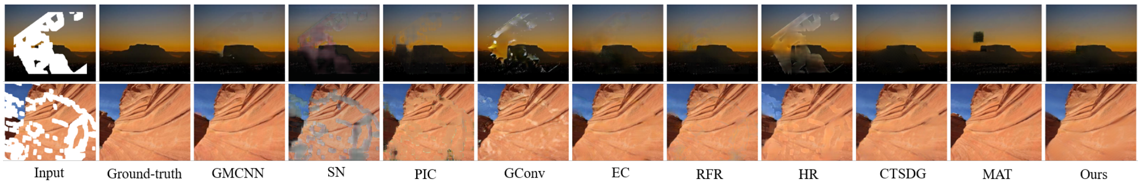


Figure 3.14: Qualitative comparison of the proposed method (Ours) with existing methods (GMCNN [9], SN [10], PIC [11], GConv [12], EC [13], RFR [14], HR [8], CTSDG [16], MAT [17]) on Places2 dataset for NVIDIA [18] mask.

we consider both the information from encoder features and decoder features in order to get better contextual information as compared to considering either of them. Further, we pondered that, if we try to consider maximum receptive field, it will further help the network towards better outcome. **A study is carried out to determine whether addition of deformable convolution works well to extract maximum receptive field.** In light of that, we considered a deformable multi-head attention (*row 5 of Table 3.9*) for extracting the contextual information from the input feature maps which resulted into better convergence of structural information. So, in combination to *+Nested* and *+Deformable* (*see Proposed Network in Table 3.9 and Figure 3.11*), we come up with our proposed network, nested deformable multi-head attention layer (NDMAL) for image inpainting. This proposed NDMAL gives inpainted output akin to ground-truth (*see Proposed Network in Table 3.9 and Figure 3.11*).

#### Comparative Analysis

We train our network on CelebA-HQ image dataset corrupted with NVIDIA mask training dataset similar to baselines (Section 3.2.3). For comparative analysis, we considered two types of masks NVIDIA [6] and QD-IMD [7]. For both mask datasets, we considered $0.01-0.2$, $0.2-0.4$ and $0.4-0.6$ mask ratios. Quantitative comparison of the proposed method with existing baselines in terms of PSNR, SSIM, $L_1$ norm, LPIPS and FID is given in Table 3.10. From Table 3.10, we can clearly mention that the proposed method effectively outperforms the baselines for all mask ratios. Along with the numeric superiority, we assess visual comparison of proposed method with existing baselines. Visual comparison is depicted in Figure 3.12. With the comparison, we come up with some observations: our proposed method does not generate ghosting outcomes, it does not create stitching effects, it does not produce over sharp results, *etc.* Furthermore, our outputs are more accurate when compared with baselines because their resemblance to ground truth is greater.

Along with this comparison on NVIDIA dataset masks, we urge to verify reliability of our method with other mask datasets. For this experiment, we consider the CelebA-HQ images corrupted with QD-IMD [7] dataset. ***Similar to existing baselines, our model is also not trained for these type of masks.*** It means, we are comparing all the methods (including ours) with unknown types of masks. *In order to make it simple, we*

Table 3.12: Quantitative comparison of the proposed method (Ours) with the state-of-the-art methods on NVIDIA [18] masks for image inpainting on Places2 dataset.

| Mask Ratio | Method | PSNR↑ | SSIM↑ | $L_1$ ↓ | LPIPS↓ | FID↓ |
|---|---|---|---|---|---|---|
| 0.01-0.2 | SN [10] | 27.88 | 0.876 | 3.371 | 0.1340 | 10.763 |
| | GMCNN [9] | 28.22 | 0.894 | 3.637 | 0.1125 | 10.543 |
| | PIC [11] | 29.52 | 0.917 | 2.796 | 0.1362 | 8.447 |
| | Gconv [12] | 29.50 | 0.921 | 2.698 | 0.1269 | 7.718 |
| | EC [13] | 29.69 | 0.915 | 2.585 | 0.1322 | 7.499 |
| | RFR [14] | 30.64 | 0.928 | 1.181 | 0.1020 | 6.104 |
| | HR [8] | 30.12 | 0.936 | 1.661 | 0.0975 | 6.148 |
| | CTSDG [16] | 30.61 | 0.953 | 1.490 | 0.0660 | 4.459 |
| | MAT [17] | 30.68 | 0.954 | 1.411 | 0.0442 | 3.696 |
| | **Ours** | **32.51** | **0.968** | **1.104** | **0.0622** | **3.639** |
| 0.2-0.4 | SN [10] | 22.67 | 0.816 | 5.173 | 0.2394 | 29.126 |
| | GMCNN [9] | 22.82 | 0.858 | 5.532 | 0.2232 | 27.398 |
| | PIC [11] | 23.46 | 0.842 | 4.410 | 0.2180 | 25.799 |
| | Gconv [12] | 22.80 | 0.872 | 4.393 | 0.2050 | 22.007 |
| | EC [13] | 23.70 | 0.877 | 4.081 | 0.2027 | 21.018 |
| | RFR [14] | 24.22 | 0.850 | 3.828 | 0.1935 | 20.218 |
| | HR [8] | 24.18 | 0.856 | 3.638 | 0.1837 | 19.326 |
| | CTSDG [16] | 25.10 | 0.877 | 3.327 | 0.1833 | 18.427 |
| | MAT [17] | 25.23 | 0.884 | 3.067 | 0.1660 | 14.839 |
| | **Ours** | **26.22** | **0.893** | **2.661** | **0.1739** | **14.254** |
| 0.4-0.6 | SN [10] | 18.19 | 0.621 | 9.330 | 0.4468 | 74.150 |
| | GMCNN [9] | 18.19 | 0.660 | 7.499 | 0.3997 | 73.696 |
| | PIC [11] | 18.82 | 0.692 | 7.111 | 0.3713 | 73.408 |
| | Gconv [12] | 19.48 | 0.724 | 6.657 | 0.3567 | 68.005 |
| | EC [13] | 19.52 | 0.719 | 6.361 | 0.3598 | 54.341 |
| | RFR [14] | 20.76 | 0.726 | 6.486 | 0.3426 | 49.204 |
| | HR [8] | 20.83 | 0.745 | 5.999 | 0.3351 | 55.461 |
| | CTSDG [16] | 21.03 | 0.770 | 5.763 | 0.3285 | 40.266 |
| | MAT [17] | 21.18 | 0.676 | 5.333 | 0.2480 | 35.810 |
| | **Ours** | **21.89** | **0.776** | **5.037** | **0.3117** | **37.887** |

*compare our method with only best five baselines.* The quantitative and qualitative results' comparison is provided in Table 3.11 and Figure 3.13 respectively. Our proposed approach gives quantitatively improved results as compared with the existing baselines. In Figure 3.13, comparing our results with existing best methods, we find that our method is able to generate more plausible results. We give this credit of faithful image inpainting to our proposed nested deformable multi-head attention, since it is able to easily extract the contextual information from both the encoded features and decoded features.

To show the generalizability of our proposed method, we have considered a Places2 natural images dataset [3]. The quantitative and qualitative comparison on Places2 dataset is given in Table 3.12 and Figure 3.14 respectively. This comparison shows that our proposed method performs well for the non-face/natural image inpainting. *Though our proposed* *method has very less number of parameters (4.1M) i.e., $\frac{1}{15}^{th}$ of the baseline [17] (60M), it performs well for face and non-face image inpainting task.*

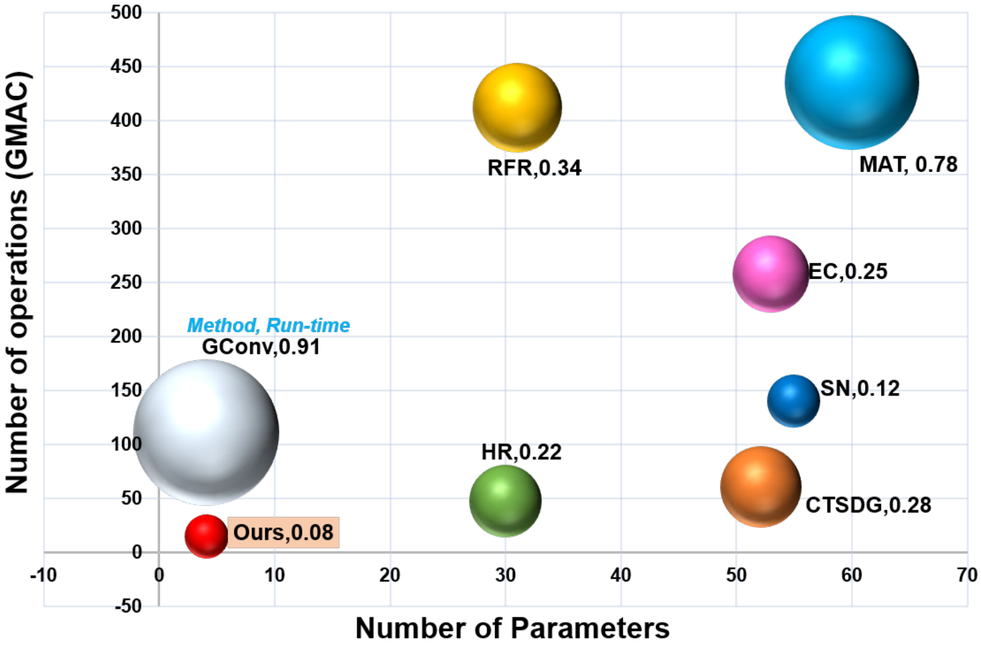


Figure 3.15: Comparison of the proposed method (ours) with existing methods (SN [10], GConv [12], EC [13], RFR [14], HR [8], CTSDG [16], MAT [17]) in terms of number of trainable parameters (*x-axis*), number of operations (GMAC) (*y-axis*) and run time complexity in *seconds per image* (*bubble size*).

### Complexity Analysis

We claim that, our propose method has low complexity with good results as compared to existing baselines. Our proposed nested deformable multi-head attention has linear complexity, since we apply the attention across channels similar to [32]. Also, the existing self attention based methods utilize number of blocks with different window sizes to capture long term dependencies in turn increasing the computational cost. Here, in this approach we come up with a single block in our NDMAL as it already consider two different feature maps to find the relative contextual information. Further, the nested attention helps the layer to extract valid content more extensively. Also, the deformable additively provide it with the larger receptive field. These points altogether allow a single block NDMAL with a $ws = 8$ to extract relevant features for image inpainting.

The computational complexity analysis in terms of number of trainable parameters, number of operations *i.e.,* Giga multiply-accumulate operations (GMAC) and average run time in terms of seconds/image is visualized in Figure 3.15. From Figure 3.15 and Tables 3.10, 3.11 and 3.12, it is clear that, with lower computational complexity, our method has good performance as compared to existing baselines.

## 3.3 FASNet: Feature Aggregation and Sharing Network for Image Inpainting

The attention mechanism generally focuses on the spatial information from the feature maps or coherence information from the non-hole region instead of focusing on the channel-wise information. The prior based methods consider the image globally neglecting the local information. The progressive or recurrent approach may suffer with output latency issue. GANs [132] now-a-days gaining more and more attention due to its ability of efficient image generation for different tasks like image-to-image translation [133], image enhancement [145], object segmentation [146], *etc.* Inspired by the fruitful results of GANs and limitations of the existing image inpainting methods, we propose a novel cascaded feature sharing and refinement adversarial architecture for image inpainting. Main contributions of the proposed work are:

- A novel feature aggregation and sharing with refinement architecture (*with 2.5M parameters*) is proposed for image inpainting.
- Multi-scale spatial channel-wise feature aggregation mechanism is proposed for extracting the efficient features from each encoder level.
- To avail the multi-scale and multi-receptive information, a decoder feature sharing mechanism is introduced.
- Further, refinement network is proposed with a novel edge extraction block to refine the inpainted image.

Comparison of the proposed method is done qualitatively and quantitatively on three benchmark datasets: CelebA-HQ [36], [1], Paris Street View (Paris_SV) [4] and Places2 [3] with existing state-of-the-art methods for image inpainting.

### 3.3.1 Proposed Framework

To inpaint an image, it is desired to consider the spatial and channel-wise information from the feature maps. Also, consideration of the maximum receptive fields is necessary to fill small to large holes efficiently. Considering that the feature maps in each of the encoder levels contain most of the relevant information, providing the features from only one encoder level may reduce the information to be forwarded to the decoder for effective reconstruction of the inpainted image. So, we aim to extract the maximum information which will help the decoder for inpainting the hole regions. To do this, we propose the extraction of features from each encoder-level and merge it to get relevant features. Similarly, taking the direct skip connections from the encoder may pass the irrelevant features of hole locations. To avoid this, we have introduced a feature aggregation block

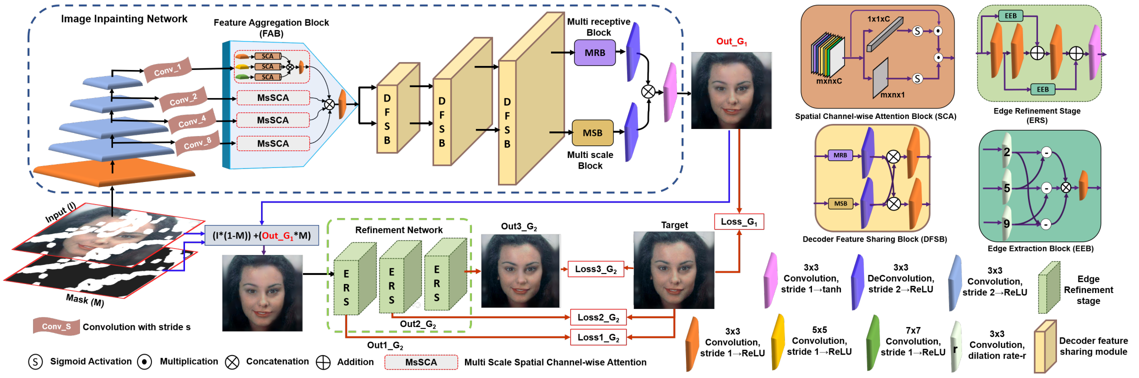


Image Inpainting Network
Feature Aggregation Block (FAB)
SCA
SCA
SCA
MsSCA
MsSCA
MsSCA
Conv_1
Conv_2
Conv_4
Conv_8
D F S B
D F S B
D F S B
Multi receptive Block
MRB
MSB
Multi scale Block
Out_G1
Input (I)
Mask (M)
(I*(1-M)) +(Out_G1*M)
Refinement Network
E R S
E R S
E R S
Out3_G2
Out2_G2
Out1_G2
Target
Loss_G1
Loss3_G2
Loss2_G2
Loss1_G2
Conv_S Convolution with stride s
S Sigmoid Activation
Multiplication
Concatenation
Addition
MsSCA Multi Scale Spatial Channel-wise Attention
1x1xC
mxnxC
mxnx1
S
Spatial Channel-wise Attention Block (SCA)
EEB
EEB
Edge Refinement Stage (ERS)
MRB
MSB
Decoder Feature Sharing Block (DFSB)
2
5
9
-
-
-
Edge Extraction Block (EEB)
3x3 Convolution, stride 1→tanh
3x3 DeConvolution, stride 2→ReLU
3x3 Convolution, stride 2→ReLU
Edge Refinement stage
3x3 Convolution, stride 1→ReLU
5x5 Convolution, stride 1→ReLU
7x7 Convolution, stride 1→ReLU
r 3x3 Convolution, dilation rate-r
Decoder feature sharing module

Figure 3.16: Flow-graph of the proposed generator framework for image inpainting.

(FAB) by combining the features from each of the encoder levels to efficiently guide the decoder for the reconstruction of the relevant content. A multi-scale multi-receptive decoder feature sharing (DFS) is proposed to reconstruct the image with small to large hole size. The inpainting network is used for generating the inpainted image from the corrupted input image. Further, in refinement network, an edge-guided refinement mechanism is proposed for enhancing and refining the inpainted image from the first network. A detailed explanation about the proposed framework is as given below:

### Inpainting Network

In the inpainting architecture, the masked image is given as input. This input is passed through $1^{st}$ convolution layer ($l$) with stride 1 and then it is processed through four convolution layers ($l \in [2, 5]$) with stride 2. The outputs of all the convolution layers from $l = 2 \rightarrow 5$ are then merged in FAB to extract the contextual information from all the encoder levels. Before passing these feature maps to FAB, they are processed through a convolution layer with stride $2^{N-l}$, $N = 5$, $l \in [2, 5]$ in order to maintain the similarity in spatial dimension between the feature maps. The merged output from FAB is then forwarded to the DFS unlike existing methods. In the DFS, at each decoder level, the multi-scale and multi-receptive features are shared in order to focus on maximum receptive field at different resolutions. The proposed inpainting architecture is as shown in Figure 3.16 (*see Image Inpainting Network architecture in blue dotted box*). The exposition of different blocks in the inpainting architecture is given in next subsections.

### Feature Aggregation Block

In this block, the features from each of the encoder levels are aggregated by processing them through the multi scale spatial channel-wise attention ($M_sSCA$) block. The output of $M_sSCA$ is (*refer FAB and $M_SSCA$ block in Figure 3.16* ):

$$O_{M_sSCA} = \mathbb{C}_1^3\{[O_{SCA}^3, O_{SCA}^5, O_{SCA}^7]\} \tag{3.20}$$

where, $\mathbb{C}_s^3$ is convolution with stride $s$ and filter size $3\times3$, [.] is concatenation operation and $O_{SCA}^k$ is output of $k^{th}$ spatial channel-wise attention block. The $M_sSCA$ is proposed to extract the features with different scales along with the spatial and channel-wise attention (SCA) block. The SCA block provides the relative weight to spatial and channel-wise (depth) feature maps. The output feature maps of the SCA block are the spatial-depth weighted outcomes. This weighing of feature map or feature map calibration helps the network to effectively correlate the spatial and channel information for inpainting the hole regions. The output of SCA is:

$$O_{SCA}^k = \sigma(S_{Avg}(f_{in}^k)) \odot (f_{in}^k \odot \sigma(C_{Avg}(f_{in}^k))) \tag{3.21}$$

where, $f_{in}^k$ is input to SCA block *i.e.,* $\mathbb{C}_1^k(En_l); k \in [3,5,7]$, $En_l$ is $l^{th}$ encoder level $(l \in (1,4))$, $\sigma$ is Sigmoid activation function, $\odot$ represents the element-wise multiplication, $S_{Avg}$ is spatial average pooling and $C_{Avg}$ is channel-wise average pooling. The SCA block extracts the spatial and depth-relevant information from the input feature map. This helps the network to learn the correlation of both spatial and depth features for the content to be inpainted. Also, the $M_sSCA$ (Eq. (3.20)) process the information with different filter size which in turn deals with the image having small to large hole size.

#### Feature Sharing Decoder

The output of feature aggregation block (FAB) is given as input to decoder feature sharing (DFS). Two parallel paths are considered, one for extracting the information from different receptive fields and the other to extract information from different scales of input feature maps. The multi-receptive ($F^{mr}$) and multi-scale ($F^{ms}$) blocks (*refer MRB and MSB in Figure 3.16* ) are explained in the Eq. (3.22) and (3.23), respectively.

$$F_l^{mr} = \mathbb{C}_{r_1}^3\{\otimes[\mathbb{C}_{r_1}^3, \mathbb{C}_{r_2}^3, \mathbb{C}_{r_3}^3, \mathbb{C}_{r_4}^3]\} \tag{3.22}$$

where, $\mathbb{C}_{r_n}^m$ represents convolution with stride $=1$, $m \times m$ filter size and $r_n$ for $n^{th}$, $n \in (1,4)$, dilation rate.

$$F_l^{ms} = \mathbb{C}_1^3\{\otimes[\mathbb{C}_1^1, \mathbb{C}_1^3, \mathbb{C}_1^5, \mathbb{C}_1^7]\} \tag{3.23}$$

where, $\mathbb{C}_1^m$ represents convolution with stride and dilation rate $=1$ with $m \times m$ filter size. As shown in Figure 3.16, in each DFS block (DFSB), dual-path features are shared at each level for extracting the relevant features. Three DFSB are used for three decoder levels and at last level the outputs of both the streams are processed through MSB and MRB respectively without feature sharing. These features are then concatenated to recover the inpainted image.

**Refinement Network**

The refinement architecture is used to enhance the edges of reconstructed image from inpainting architecture. For this purpose, the edge extraction block (EEB) is defined which extracts the edges from the input feature map (*see EEB in Figure 3.16* ). In EEB, the subtractive features from different receptive fields are extracted which are then utilized in the refinement stage (*see edge refinement stage (ERS) in Figure 3.16* ). The output of each ERS is taken for the loss calculation while training the network, which helps the network to improve the output at each stage. With such implementation, the final output will be structurally as well as visually plausible. Both these architectures, in combination give the refined inpainted image as output.

Table 3.13: Quantitative comparison of the proposed method (Ours) with SOTA methods on CelebA-HQ dataset using NVIDIA [18] masks for image inpainting.

| Mask Ratio | Method | PSNR ↑ | SSIM ↑ | $L_1$ ↓ | FID ↓ | LPIPS ↓ |
|---|---|---|---|---|---|---|
| 0.1-0.2 | PIC[11] | 30.97 | 0.965 | 0.01110 | 5.420 | 0.062 |
| | GConv [12] | 32.56 | 0.973 | 0.00880 | 5.563 | 0.061 |
| | EC [13] | 32.48 | 0.975 | 0.00880 | 4.638 | 0.064 |
| | RFRNet [14] | 33.56 | 0.981 | 0.00750 | 3.894 | 0.029 |
| | CTSDG [16] | 32.11 | 0.971 | 0.00859 | 3.326 | 0.025 |
| | **Ours** | **34.19** | **0.983** | **0.00620** | **3.095** | **0.029** |
| 0.3-0.4 | PIC[11] | 24.47 | 0.881 | 0.03140 | 25.971 | 0.172 |
| | GConv [12] | 26.72 | 0.914 | 0.02450 | 12.429 | 0.152 |
| | EC [13] | 26.62 | 0.915 | 0.02470 | 12.084 | 0.144 |
| | RFRNet [14] | 27.76 | 0.934 | 0.02120 | 17.056 | 0.158 |
| | CTSDG [16] | 26.71 | 0.929 | 0.02970 | 11.299 | 0.105 |
| | **Ours** | **28.21** | **0.942** | **0.01800** | **11.130** | **0.110** |
| 0.5-0.6 | PIC[11] | 19.29 | 0.670 | 0.07490 | 44.555 | 0.397 |
| | GConv [12] | 21.47 | 0.767 | 0.05610 | 34.980 | 0.358 |
| | EC [13] | 21.49 | 0.759 | 0.05720 | 30.277 | 0.316 |
| | RFRNet [14] | 21.80 | 0.819 | 0.04700 | 31.571 | 0.305 |
| | CTSDG [16] | 21.52 | 0.825 | 0.05692 | 27.869 | 0.197 |
| | **Ours** | **21.85** | **0.844** | **0.04390** | **21.640** | **0.144** |

### 3.3.2 Training of the Proposed Network

The proposed architecture is trained in two steps *i.e,* the inpainting network is trained first for the image inpainting task with the corrupted image as input. Further, the refinement network is trained with an inpainted image as input. The discriminator network is same as that of [133]. Weight parameters of the network are updated on NVIDIA DGX station having Tesla V100 1×16 GB GPU with the batch size of 1. The Adam optimizer [144] with the learning rate of $2 \times 10^{-4}$, $\beta_1 = 0.5$ and $\beta_2 = 0.99$ are used.

Table 3.14: Quantitative comparison of the proposed method (Ours) with SOTA methods on Places2 dataset using NVIDIA [18] masks for image inpainting.

| Mask Ratio | Method | PSNR ↑ | SSIM ↑ | $L_1$ ↓ | FID ↓ | LPIPS ↓ |
|---|---|---|---|---|---|---|
| 0.1-0.2 | PIC[11] | 27.14 | 0.932 | 0.01610 | 9.588 | 0.182 |
| | GConv [12] | 26.05 | 0.894 | 0.01680 | 9.867 | 0.172 |
| | EC [13] | 27.17 | 0.933 | 0.01570 | 6.094 | 0.184 |
| | RFRNet [14] | 27.75 | 0.939 | 0.01420 | 5.149 | 0.128 |
| | CTSDG [16] | 29.69 | 0.957 | 0.00924 | 5.129 | 0.089 |
| | **Ours** | **30.11** | **0.962** | **0.00910** | **3.095** | **0.082** |
| 0.3-0.4 | PIC[11] | 21.72 | 0.786 | 0.04410 | 34.240 | 0.236 |
| | GConv [12] | 22.45 | 0.821 | 0.04215 | 21.453 | 0.225 |
| | EC [13] | 22.18 | 0.802 | 0.04080 | 18.935 | 0.227 |
| | RFRNet [14] | 22.63 | 0.819 | 0.03810 | 15.540 | 0.218 |
| | CTSDG [16] | 23.50 | 0.826 | 0.02503 | 16.879 | 0.208 |
| | **Ours** | **24.45** | **0.831** | **0.0245** | **14.530** | **0.198** |
| 0.5-0.6 | PIC[11] | 17.17 | 0.494 | 0.09440 | 68.730 | 0.418 |
| | GConv [12] | 17.98 | 0.685 | 0.07421 | 50.450 | 0.412 |
| | EC [13] | 18.35 | 0.553 | 0.08210 | 57.677 | 0.420 |
| | RFRNet [14] | 18.92 | 0.596 | 0.07610 | 43.158 | 0.399 |
| | CTSDG [16] | 18.05 | 0.749 | 0.06573 | 41.422 | 0.384 |
| | **Ours** | **19.86** | **0.762** | **0.05510** | **35.190** | **0.326** |

#### Loss Function

While training, instead of calculating the loss on the overall image which will create the disturbances in the hole and non-hole region, we have used a separate loss function for the hole $L_1^{Holes}$ and non-hole $L_1^{NonHoles}$. Along with the adversarial loss $L_{GAN}$, perceptual loss $L_{Per}$, structural similarity index (SSIM) loss $L_{SSIM}$ and edge loss $L_{Edge}$ are used for network optimization. So, overall loss for training the proposed network is given as:

$$\begin{aligned} L_{Total} = \lambda_{Holes} L_1^{Holes} + \lambda_{NonHoles} L_1^{NonHoles} + \\ \lambda_{edge} L_{Edge} + \lambda_{Per} L_{Per} + \lambda_{SSIM} L_{SSIM} + \lambda_{GAN} L_{GAN} \end{aligned} \tag{3.24}$$

here, $\lambda_{loss}$ are the weights assigned for the respective loss functions. The values of each of the weights are $\lambda_{Holes} = 3$, $\lambda_{NonHoles} = 1$ , $\lambda_{Edge} = 1$ , $\lambda_{Per} = 0.2$ , $\lambda_{SSIM} = 0.2$ and $\lambda_{GAN} = 1$. All these mentioned losses are considered for training of both, the inpainting and refining network. For refinement architecture, a multi-stage loss is calculated at each stage of output ($Out1\_G_2$, $Out2\_G_2$, $Out3\_G_2$) and sum of all stage losses ($Loss1\_G_2 + Loss2\_G_2 + Loss3\_G_2$) is used for training the refinement network.

### 3.3.3 Experimental Analysis

In this section, the comparative analysis and ablation study are discussed in detail. For training and testing of the proposed method, we have utilized publicly available NVIDIA masks from [6]. For analysis, we have used $0.1-0.2$, $0.3-0.4$, and $0.5-0.6$ mask ratios

Table 3.15: Quantitative comparison of the proposed method (Ours) with SOTA methods on Paris Street View using NVIDIA [18] masks for image inpainting.

| Mask Ratio | Method | PSNR ↑ | SSIM ↑ | $L_1$ ↓ | FID ↓ | LPIPS ↓ |
|---|---|---|---|---|---|---|
| 0.1-0.2 | PIC[11] | 29.35 | 0.930 | 0.0140 | 21.300 | 0.053 |
| | GConv [12] | 31.32 | 0.953 | 0.0120 | 19.992 | 0.047 |
| | EC [13] | 31.19 | 0.950 | 0.0110 | 14.800 | 0.039 |
| | RFRNet [14] | 31.71 | 0.954 | 0.0110 | 11.620 | 0.015 |
| | CTSDG [16] | 32.50 | 0.949 | 0.0098 | 9.190 | 0.010 |
| | **Ours** | **32.93** | **0.960** | **0.0069** | **7.455** | **0.012** |
| 0.3-0.4 | PIC[11] | 23.97 | 0.785 | 0.0379 | 61.277 | 0.155 |
| | GConv [12] | 25.54 | 0.846 | 0.0309 | 93.584 | 0.125 |
| | EC [13] | 26.04 | 0.846 | 0.0286 | 45.480 | 0.099 |
| | RFRNet [14] | 26.44 | 0.862 | 0.0275 | 40.170 | 0.065 |
| | CTSDG [16] | 27.02 | 0.858 | 0.0265 | 32.340 | 0.058 |
| | **Ours** | **27.64** | **0.875** | **0.0196** | **31.880** | **0.057** |
| 0.5-0.6 | PIC[11] | 19.52 | 0.519 | 0.0799 | 86.624 | 0.396 |
| | GConv [12] | 20.61 | 0.621 | 0.0660 | 80.465 | 0.326 |
| | EC [13] | 21.89 | 0.646 | 0.0582 | 72.167 | 0.209 |
| | RFRNet [14] | 22.40 | 0.681 | 0.0546 | 68.613 | 0.197 |
| | CTSDG [16] | 22.11 | 0.748 | 0.0590 | 64.440 | 0.192 |
| | **Ours** | **22.83** | **0.755** | **0.0466** | **62.530** | **0.184** |

same as that of [14].

### Comparative Analysis

Comparison of the proposed method and existing methods is done on Places2, Celeb_HQ and Paris_SV dataset images corrupted using NVIDIA's mask dataset from [6]. Table 3.13, 3.15, 3.14 show the comparison in terms of PSNR, SSIM, Mean $L_1$ error, FID and LPIPS. The proposed method gives less mean $L_1$ error on Places2, CelebA-HQ and Paris_SV datasets respectively as compared to exsting methods for image inpainting. Increase in PSNR values indicate that the proposed method inpaints each pixel in hole region very efficiently (with consistent information at holes) while keeping good structural information (with good SSIM). From Table 3.13, 3.15, 3.14, it is clear that the proposed method outperforms all the existing SOTA methods. Qualitative comparison of the proposed method with the existing methods is given in Figure 3.17.

From both the quantitative and qualitative comparison with existing methods, it is clear that the proposed method gives improvement for lower to higher mask ratios. Also, it is worth to note that complexity (in terms of number of trainable parameters) of the proposed architecture is less (***2.5M***) as compared to the existing methods (***3.64*** **M** [11], ***4.05M*** [12], ***53M*** [13], ***52.14M*** [16] , ***31M***[14]). *Specifically, the proposed method outperforms the existing methods with a comparatively less number of parameters.*

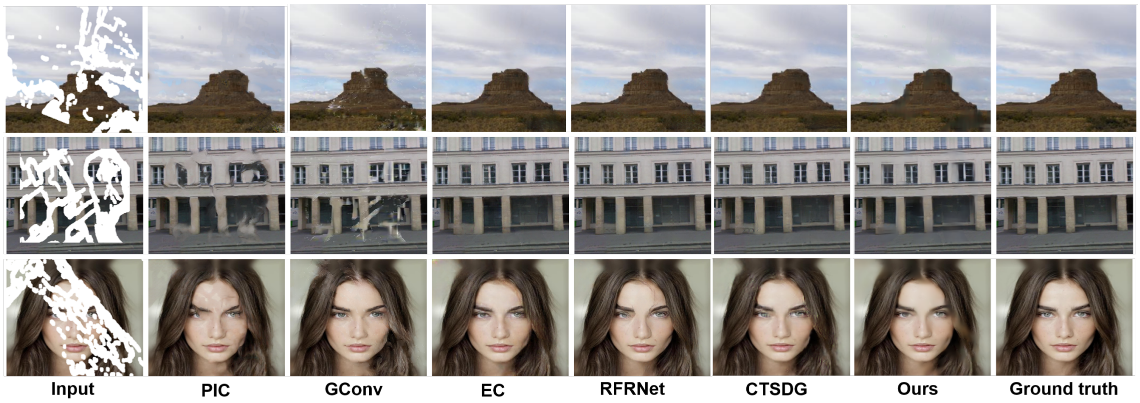


Figure 3.17: Qualitative comparison of the proposed method (Ours) with existing methods (PIC [11], GConv [12], Edge-Connect [13], RFRNet [14]) for NVIDIA [18] mask (*Row 1 -Places2, row 2 -Paris_SV, row 3 -CelebA_HQ*).

Table 3.16: Effect of the proposed refinement architecture for image inpainting using Paris_SV dataset.

| Mask Ratio → | | 0.1-0.2 | | 0.3-0.4 | | 0.5-0.6 | |
|---|---|---|---|---|---|---|---|
| Refinement | Stage | PSNR ↑ | SSIM ↑ | PSNR ↑ | SSIM ↑ | PSNR ↑ | SSIM ↑ |
| × | × | 30.26 | 0.931 | 26.52 | 0.860 | 20.68 | 0.622 |
| ✓ | 1 | 31.43 | 0.944 | 26.64 | 0.866 | 21.95 | 0.693 |
| ✓ | 2 | 31.89 | 0.956 | 26.95 | 0.871 | 22.12 | 0.704 |
| ✓ | 3 | **32.93** | **0.960** | **27.64** | **0.875** | **22.83** | **0.755** |
| ✓ | 4 | 31.87 | 0.949 | 27.04 | 0.862 | 22.04 | 0.694 |

Table 3.17: Ablation study on feature aggregation block (FAB), multi receptive block (MRB) and multi scale block (MSB) on Paris_SV dataset (*Note: ‡ and ⊎ indicate without and with decoder feature sharing respectively*)

| Mask Ratio → | | | 0.1-0.2 | | 0.3-0.4 | | 0.5-0.6 | |
|---|---|---|---|---|---|---|---|---|
| FAB | MRB | MSB | PSNR ↑ | SSIM ↑ | PSNR ↑ | SSIM ↑ | PSNR ↑ | SSIM ↑ |
| × | × | × | 30.12 | 0.906 | 25.73 | 0.858 | 20.15 | 0.640 |
| × | ✓ | ✓ | 31.26 | 0.921 | 26.13 | 0.849 | 21.28 | 0.701 |
| ✓ | × | ✓ | 30.98 | 0.954 | 25.85 | 0.862 | 20.65 | 0.695 |
| ✓ | ✓ | × | 30.92 | 0.920 | 25.97 | 0.860 | 20.83 | 0.692 |
| ✓ | ✓‡ | ✓‡ | 31.97 | 0.956 | 26.28 | 0.869 | 21.89 | 0.709 |
| ✓ | ✓⊎ | ✓⊎ | **32.93** | **0.960** | **27.64** | **0.875** | **22.83** | **0.755** |

### Ablation Study

To determine **How the refinement architecture contributes to the efficient image inpainting?** The experiment is performed without refinement network *i.e.,* inpainting network only (*row 1 of Table 3.16*) and with refinement at different stage output (*rows 2 to 5 of Table 3.16*). From Table 3.16, it can be observed that the refinement architecture with output at stage 3 gives an effective performance. If the number of the output stages is increased further, there is not much improvement in the output. Table 3.16 shows the effectiveness of the proposed refinement architecture for image inpainting. Also, it represents that, the three-stage refinement architecture performs better in terms of PSNR and SSIM.

In the proposed method, we have incorporated different blocks to address various issues in image inpainting. Here, the intuition behind proposing these blocks is explained *i.e.*, **How do the feature aggregation block, multi receptive block, multi-scale block, and decoder feature sharing approach help overall architecture for image inpainting?** To scrutinize this, experiments are performed with (✓) and without (×) respective blocks on Paris_SV dataset. Table 3.17 shows various experiments done for analysing the effect of the above-mentioned blocks as well as the effect of DFS in terms of PSNR and SSIM. From Table 3.17, it is clear that the proposed FAB, MRB, MSB, and DFS are effective for accurate image inpainting.

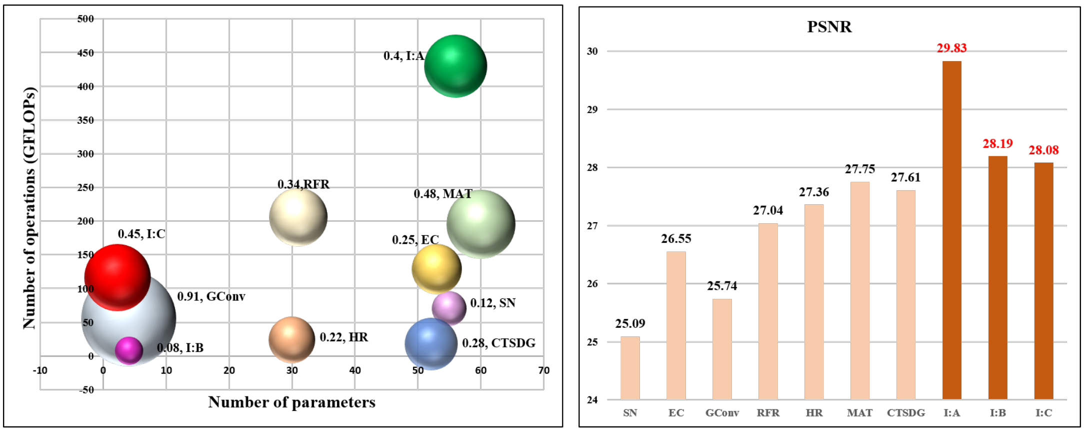


Figure 3.18: Comparison of the proposed methods (I:A-Section 3.1, I:B-Section 3.2, I:C-3.3) with existing methods. Left: in terms of the number of trainable parameters (x-axis), number of operations (GMAC) (y-axis), and run-time complexity in seconds per image (bubble size), Right: in terms of average PSNR on CelebA-HQ dataset.

## 3.4 Summary of Proposed Contribution

In this chapter, we proposed three different solutions with coarse-to-fine architectures for image inpainting. In first solution (Section 3.1), a novel spatial projection layer is proposed to inpaint the images with spatial consistencies. Also, a Canny edge detection based edge loss is proposed in order to generate detailed edges in the inpainted images. The qualitative and quantitative comparison of proposed solution is done with existing methods on three benchmark datasets corrupted using NVIDIA masks [6].

In second solution (Section 3.2), we proposed a nested deformable mutli-head attention layer (NDMAL) to effectively fuse the encoder and decoder features. This NDMAL allows the network to effectively capture the long-term dependencies and focus on valid features from different receptive fields. The qualitative and quantitative comparison is carried out on existing methods on two benchmark datasets corrupted using NVIDIA masks [6]. Also, the comparative analysis is done on corrupted images with unseen masks [7].

Third solution (Section 3.3) proposes a novel feature aggregation and sharing followed by

refinement architecture for image inpainting. In this, a multi-scale spatial channel-wise feature aggregation mechanism is proposed along with multi-scale and multi-receptive decoder feature sharing. Also, a edge refinement stage is proposed for finer edge generation in the inpainted images. The qualitative and quantittive comparison is carried out on existing methods on two benchmark datasets corrupted using NVIDIA masks [6].
The comparison of computational complexity and performance in terms of PSNR for proposed solutions and existing methods is shown in Figure 3.18.

# Chapter 4

# Single-stage Architectures

The inpainting results with coarse-to-fine architectures generally have a dependency of fine stage on coarse stage outputs. Also, most of the time, these coarse to fine architectures posses high computational complexity. There exist the single-stage architectures for image inpainting [73, 14, 102, 103] with blurry or inconsistent results. Also, these architectures have comparatively high computational complexity even-though they utilize single-stage architectures for image inpainting. With this motivation, we proposed three different solutions with single-stage architectures with relatively less computational complexity and efficient inpainting results. The proposed three solutions are:

1. Image inpainting via correlated multi-resolution feature projection.

2. Diverse receptive field based adversarial concurrent encoder network for image inpainting.

3. Pseudo decoder guided light-weight architecture for image inpainting.

## 4.1 Image Inpainting via Correlated Multi-resolution Feature Projection

Existing image inpainting architectures give notable results with deeper networks and have more computational complexity in terms of number of trainable parameters or run-time whereas the architectures with shallow networks lack in the reliability of outcomes. To inpaint an image efficiently with random hole size, we have proposed a single stage architecture with moderate complexity and remarkable outcomes. Here, we process the multi-resolution inputs to extract the information from various resolutions. These multi-resolution features are then processed through a multi-kernel non-local attention. This helps to correlate the multi-resolution features efficiently. Also, a proposed feature projection block pave a way to project correlated features from the multi-resolution inputs to decoder for effective reconstruction. Also, a valid feature fusion block is introduced to avail valid locations of encoder features for faithful reconstruction. The main contributions of the proposed work are:

- A novel single stage architecture with multi-resolution input is proposed for image inpainting.

- A multi-kernel non-local attention block is proposed to merge the encoder features from each resolution.
- To project the multi resolution processed features, a feature projection block is proposed for effective reconstruction.
- Further, a valid feature fusion block is proposed to merge the relevant features from the encoder to decoder.

Comparison of the proposed method is done qualitatively and quantitatively on two benchmark datasets: CelebA-HQ [36], [1] and Places2 [3] with existing state-of-the-art methods for image inpainting and detailed ablation study with state-of-the-art modules (non-local attention and squeeze-excitation block). Also, the comparative analysis for object removal task is discussed.

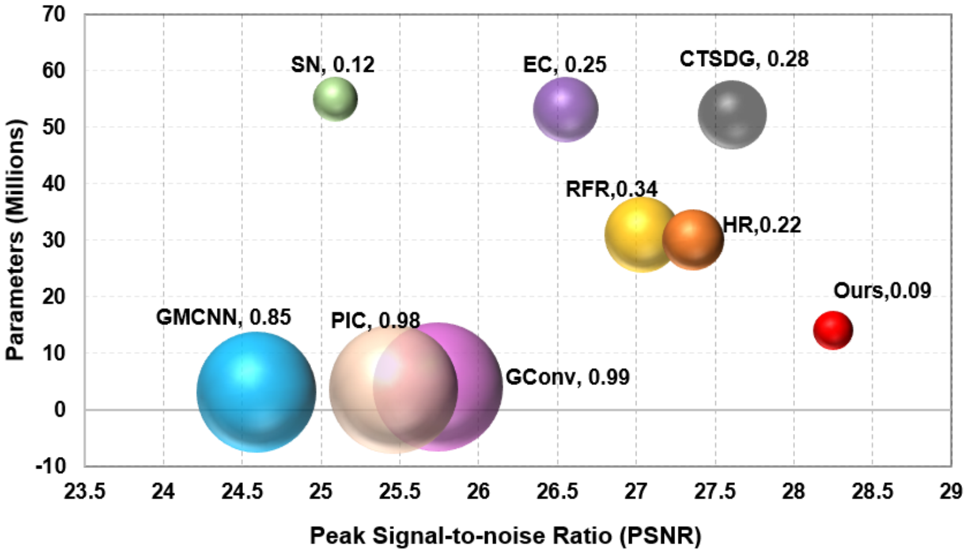


Figure 4.1: Comparison of proposed method (Ours) with existing methods (GMCNN [9], SN [10], PIC [11], GConv [12], EC [13], RFR [14], HR [8], CTSDG [16]) in terms of number of trainable parameters (*y-axis*), run time complexity and PSNR (*x-axis*) on CelebA-HQ dataset. The size of bubble indicates the run time complexity in *image/second* which is also written with bubble for respective method.

### 4.1.1 Proposed Framework

Designing very deep networks in deep learning approaches provide admirable inpainting results with a compromise of computational complexity (in terms of number of parameters or average run-time to process one image in terms of *image/second*). Whereas, the shallow networks with less computational complexity generate unpleasant results. So, we aim to propose an intermediate solution to this issue by designing a novel architecture with moderate complexity and excellent outcomes (*see Figure 4.1*). Processing multi-resolution

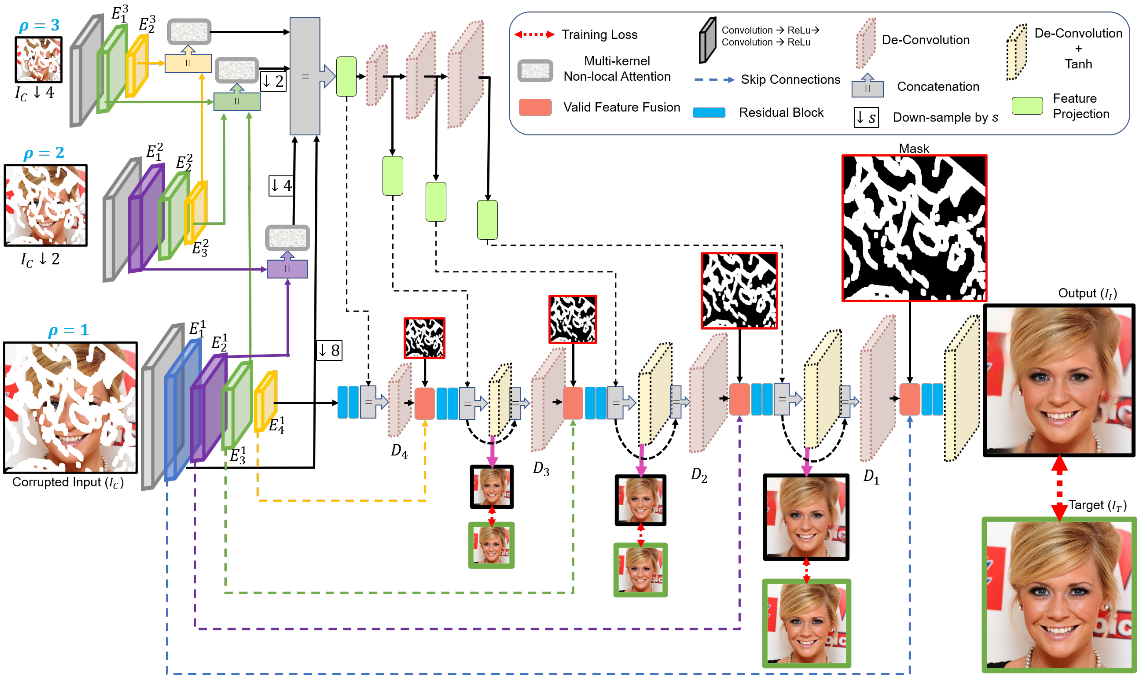


Figure 4.2: Proposed generator architecture for image inpainting.

input allows the exploitation of the features across each resolution in turn reducing the dominance of holes in the corrupted image. The proposed architecture for image inpainting basically depends on multi-resolution inputs. The encoder is designed to process the multiple inputs of different resolutions called as multi resolution encoder. All feature maps from the three resolutions are then merged to form effective feature maps for image inpainting. The fused feature maps are then processed through multi-kernel non-local attention block. This helps the network to attend the effective local and global information generation. Further, these fused features are then projected at the respective reconstruction decoder to help for better convergence. The encoder features of original resolution input are merged with the reconstruction decoder through skip connection via proposed valid feature fusion block. This merges the valid features from the encoder at the valid locations and decoder features at hole locations while reconstruction. Along with this, a multi-scale loss is also considered for training the network for efficient learning. Figure 4.2 shows the schematic of proposed generator architecture for image inpainting. The detail exposition of each of the proposed blocks is given below:

### Multi Resolution Encoder

The proposed architecture processes the corrupted input image ($I_c$) with three different resolutions. Three inputs to the network are: $I_c$, $I_c \downarrow 2$ and $I_c \downarrow 4$ respectively. Where, $\downarrow s$ indicates image down-sampled by factor $s$ with bicubic interpolation. The input image in each resolution($\rho$) is first processed through a `Convolution` $\rightarrow$ `ReLu` $\rightarrow$ `Convolution`

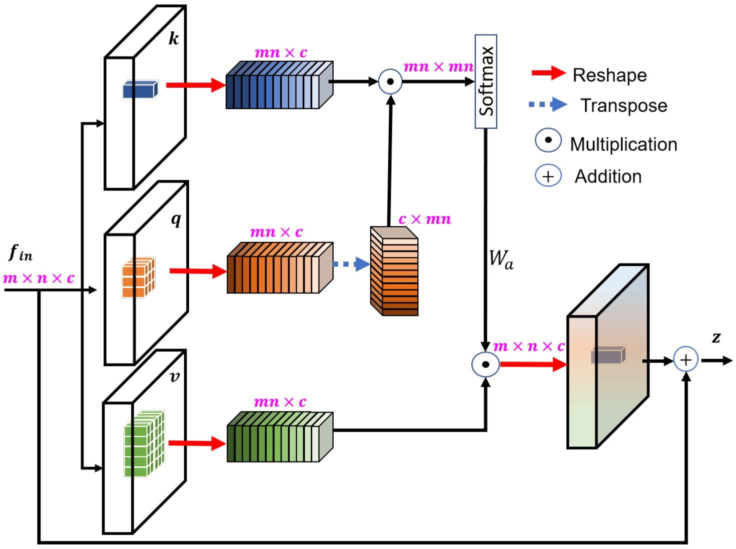


Figure 4.3: Proposed multi-kernel non-local attention block.

→ `ReLu` with *stride* = 1 layer to convert into low level feature maps. These feature maps are then processed through the encoder level *i.e.,* `Convolution` → `ReLu` → `Convolution` → `ReLu` with *stride* = 2. The number of encoder levels for each resolution input are given as: $E_l^\rho$ where $l \in (1, 5-\rho)$ and $\rho = 1, 2, 3$. So, the input at first resolution $\rho = 1$ has four encoder levels ($E_l^\rho$; $l \in (1, 4)$ and $\rho = 1$). The respective encoder features of each resolution are then concatenated and forwarded to multi-kernel non-local attention block.

### Multi Kernel Non-local Attention

To perceive the contextual information from all the pixels in input, we have proposed a multi kernel non-local (MKNL) attention for image inpainting task. The existing channel-wise and spatial features do not consider the uneven distribution of the information in the input corrupted images [19]. The non-local attention [19] helps to capture the long-range dependencies among the features and also pays attention to the difficult parts in a feature map. The existing method depending on graph convolutions [8] lacks in extracting the contextual information from all the features in a feature map. Whereas, our proposed MKNL attention helps to perceive the information from all the features in input feature maps. So, overall the proposed MKNL helps the network:

- **To focus mainly on the valid informative regions in the input feature maps.**
- **To capture the global as well as local correlations among the features.**

Let, $f_{in} \, \epsilon \, \mathbb{R}^{m \times n \times c}$ be the input feature maps to the MKNL block. Where, $m, n$ is the spatial dimension and $c$ is number of channels. At first, the proposed MKNL attention takes the input features and maps them into multi-kernel feature space. For this, they are processed with the multi kernel convolutions *i.e.,* $\theta_k$, $k \in (1, 3, 5)$. The proposed MKNL differs from existing non-local attentions [147, 148, 149] used for image super-resolution task. Where, the input features are mapped by processing with the same kernel convolutions (generally $1 \times 1$ kernel size). Here, the proposed multi-kernel delve into different receptive fields of the input feature maps. This helps to capture the long-term dependencies from the input feature maps. So, the mapped input with multi kernels is given as:

$$k = \theta_1(f_{in}); \quad q = \theta_3(f_{in}); \quad v = \theta_5(f_{in}) \tag{4.1}$$

where, $f_{in}$ is the input feature map, $\theta_m$ is `convolution with` $kernel\ size = m \times m$. Further, we collapse the spatial dimension of multi-kernel responses into a single dimension which yield a size of $mn \times c$. The correlation between $k$ and $q$ is then passed through a Sigmoid layer to form a weighted feature attention ($W_a$):

$$W_a = \sigma(k.q^T) \tag{4.2}$$

Also, to capture more long-term dependencies, the weighted feature attention $W_a$ is multiplied with $v$ and a residual connection from input is added to the response. So, the final outcome of the MKNL is given as:

$$Z = \theta_1(W_a \odot k) + f_{in} \tag{4.3}$$

where, $\odot$ is element-wise multiplication. This MKNL is applied on the concatenated encoder feature maps from different resolutions (*see Figure. 4.2 and 4.3* ).

The merged features of all the input resolutions are then forwarded to the feature projection (FP) block. The FP block, is proposed to project the merged features to the decoder in order to help for reconstructing inpainting image efficiently. These projected features are the accumulation of multi-scale and multi-resolution feature maps. This helps the network towards filling the holes with variable size. *The effect of proposed multi-kernel non-local attention is analysed in Section 4.1.3.*

### Feature Projection Block

Let, $X \in \mathbb{R}^{m \times n \times c}$ is the input feature map to the FP block. The channels of the input feature maps are first collapsed as given in Eq. 4.4 with the spatial output of size $m \times n$.

$$F_{avg} = avg_c(X_{m,n,c}) \tag{4.4}$$

where, $avg_c$ is the channel-wise average pooling. This average information is then normalized with the `layer normalization`. This converts all the averaged features in

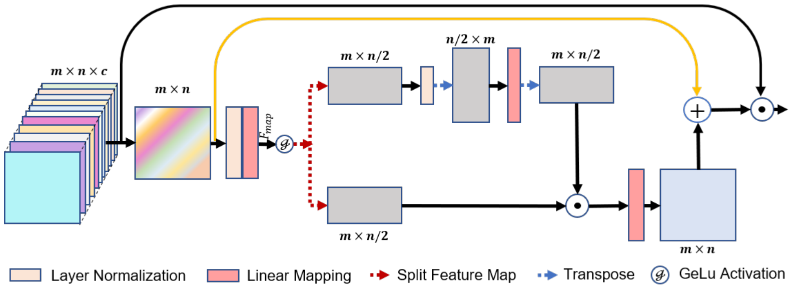


Figure 4.4: Proposed feature projection block.

the particular range. The normalized features are then mapped linearly to preserve the structure of averaged features. So, the output features after normalization and mapping are:

$$F_{map} = w(\ell(F_{avg})) + b \tag{4.5}$$

where, $\ell$ is `layer normalization`, $w$ is weight and $b$ is bias to linearly map the averaged feature map $F_{avg}$. Further, the activated features with `GeLu Activation` are split in two parts with size $m \times \frac{n}{2}$. One part is normalized and linearly mapped with linear projection. These mapped features are added to the another part with size $m \times \frac{n}{2}$. Since we are extracting the spatially weighted feature maps (unlike channel-wise gated convolution), this overall process is called as spatial gating mechanism. The gating mechanism assign the linearly mapped weight to $F_{map}$ in turn generating linearly mapped features to be projected to decoder for reconstruction. This `feature projection` of information helps the network to extract the relevant information from the non-hole feature space to fill the hole region effectively. *The effectiveness of proposed feature projection is analysed in Section 4.1.3*

### Fused Feature Decoder

Generally, the decoder is designed such that it takes an input from previous level and encoder features as skip connections to reconstruct the desired output. In our proposed architecture, we improve the decoder reconstruction by providing the feature projection from multi-resolution fused features processed with MKNL and FP. Also, similar to general decoder architectures, we also provide a skip connection. But, here instead of directly providing the skip connection we propose to utilize the valid features from the encoder and forward them with the respective decoder layer outcome. This is achieved by the valid feature fusion block in the proposed architecture (*see Figure 4.2* ). The features from encoder of original resolution ($E_4^1$) are given to the decoder for reconstruction. To effectively extract the features and to avoid the problem of vanishing gradients, these features are processed through two successive residual blocks. The residual block is represented in Figure 4.5, where each layer is a `convolution` layer. The outcome of

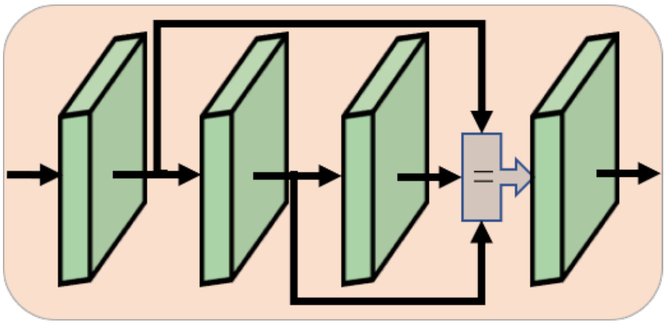


Figure 4.5: Residual block.

residual layers is then merged with the features from FP block. This makes sure that all the features from all resolutions are projected to the decoder layer helping the network to reproduce the output efficiently. The merged residual and FP block features are then fed to `deconvolution` block.

The features from each encoder of original resolution ($\rho = 1$) are taken as skip connections for respective decoder layer. Unlike existing phenomenon of utilizing the skip features as it is, we have proposed a valid feature fusion (VFF) block. In VFF block, the skip features from each encoder layer and the previous level decoder features are merged with the knowledge of hole locations with mask $M$ as input (*similar to existing approaches [10, 9, 11, 12, 13, 8, 16] we utilize mask as input*). The output of VFF block as:

$$VFF_{out} = (D_l * M^{\downarrow}_{2^{l-1}}) + (SE^1_l * (1 - M^{\downarrow}_{2^{l-1}})) \tag{4.6}$$

where, $SE^1_l$ are the skip connection from encoder layer $l$ of resolution $\rho = 1$, $M^{\downarrow}_{2^{l-1}}$ is the mask down-sampled by $2^{l-1}$ indicating *hole locations* $= 1$ and *valid locations* $= 0$, and $D_l$ is the $l^{th}$ decoder layer ($l \in (1, 4)$) (*see Figure 4.2* ). *The effectiveness of VFF block is analysed in Section 4.1.3* .

The output from each of the decoder layer is extracted by adding a `Deconvolution`→ `Tanh Activation` layer. This output from each decoder scale is further utilized for loss calculation called as multi-scale loss while training the network. The output of last decoder layer is considered as the final inpainted output.

### 4.1.2 Training of the Proposed Network

The proposed architecture is trained with the corrupted image and its mask as input and generates inpainted image as output. The discriminator network is same as that of [133]. While training, the image values are linearly scaled between the range [0 : 1]. Weight parameters of the network are updated on NVIDIA DGX station having Tesla V100 1×16 GB GPU with the batch size of 1. The ADAM optimizer [144] with the learning rate of $2 \times 10^{-4}$, $\beta_1 = 0.5$ and $\beta_2 = 0.99$ is used.

Table 4.1: Quantitative comparison of the proposed method (Ours) with state-of-the-art methods on NVIDIA [18] masks for image inpainting on CelebA-HQ dataset.

| Mask Ratio | Method | Publication | PSNR↑ | SSIM↑ | $L_1$ ↓ | LPIPS↓ | FID↓ |
|---|---|---|---|---|---|---|---|
| 0.01-0.2 | SN [10] | ECCV-18 | 30.84 | 0.961 | 2.827 | 0.0601 | 4.134 |
| | GMCNN [9] | NIPS-18 | 30.54 | 0.957 | 2.867 | 0.0572 | 7.537 |
| | PIC [11] | CVPR-19 | 32.08 | 0.967 | 2.689 | 0.0382 | 4.042 |
| | Gconv [12] | ICCV-19 | 32.06 | 0.960 | 2.681 | 0.0384 | 4.309 |
| | EC [13] | CVPRW-19 | 32.04 | 0.973 | 3.108 | 0.0387 | 4.042 |
| | RFR [14] | CVPR-20 | 33.45 | 0.973 | 1.824 | 0.0453 | 2.516 |
| | HR [8] | WACV-21 | 33.28 | 0.976 | 1.925 | 0.0418 | 2.257 |
| | CTSDG [16] | ICCV-21 | 33.57 | 0.979 | 1.329 | 0.0280 | 2.105 |
| | **Ours** | - | **34.33** | **0.985** | **0.950** | **0.0220** | **1.851** |
| 0.2-0.4 | SN [10] | ECCV-18 | 25.77 | 0.896 | 4.246 | 0.2091 | 10.643 |
| | GMCNN [9] | NIPS-18 | 24.49 | 0.894 | 4.120 | 0.1711 | 28.170 |
| | PIC [11] | CVPR-19 | 25.30 | 0.891 | 3.691 | 0.1772 | 14.376 |
| | Gconv [12] | ICCV-19 | 25.48 | 0.904 | 4.147 | 0.1668 | 11.010 |
| | EC [13] | CVPRW-19 | 26.30 | 0.901 | 3.194 | 0.1630 | 7.338 |
| | RFR [14] | CVPR-20 | 26.44 | 0.917 | 3.022 | 0.1414 | 11.767 |
| | HR [8] | WACV-21 | 26.76 | 0.935 | 3.213 | 0.1341 | 10.330 |
| | CTSDG [16] | ICCV-21 | 27.02 | 0.936 | 2.466 | 0.1020 | 7.516 |
| | **Ours** | - | **27.33** | **0.940** | **2.219** | **0.0648** | **7.135** |
| 0.4-0.6 | SN [10] | ECCV-18 | 18.65 | 0.657 | 8.852 | 0.3690 | 61.160 |
| | GMCNN [9] | NIPS-18 | 18.74 | 0.744 | 6.747 | 0.4060 | 50.981 |
| | PIC [11] | CVPR-19 | 19.01 | 0.679 | 7.011 | 0.3451 | 49.120 |
| | Gconv [12] | ICCV-19 | 19.70 | 0.860 | 5.695 | 0.3017 | 34.940 |
| | EC [13] | CVPRW-19 | 21.33 | 0.809 | 5.828 | 0.2755 | 33.011 |
| | RFR [14] | CVPR-20 | 21.23 | 0.755 | 6.354 | 0.2551 | 30.650 |
| | HR [8] | WACV-21 | 22.04 | 0.831 | 5.345 | 0.2429 | 28.498 |
| | CTSDG [16] | ICCV-21 | 22.24 | 0.845 | 4.451 | 0.1910 | 14.371 |
| | **Ours** | - | **23.09** | **0.901** | **4.302** | **0.1675** | **11.851** |

The overall loss for training the network is given as:

$$\begin{aligned} L_{Total} = \lambda_{Holes} L_1^{Holes} + \lambda_{NonHoles} L_1^{Non-holes} + \\ \lambda_{edge} L_{edge} + \lambda_P L_P + \lambda_{SSIM} L_{SSIM} + \lambda_{GAN} L_{GAN} \end{aligned} \tag{4.7}$$

here, $\lambda_{loss}$ are the weights assigned for the respective loss functions. The values of each of the weights are $\lambda_{Holes} = 3$, $\lambda_{NonHoles} = 1$ , $\lambda_{edge} = 1$ , $\lambda_P = 0.2$ , $\lambda_{SSIM} = 0.2$ and $\lambda_{GAN} = 0.1$. All these mentioned losses are considered for each scale of decoder output. So, the total loss is sum of all scale losses.

### 4.1.3 Experimental Analysis

In this section, the experiments are discussed in detail for image inpainting. The training and testing of the proposed method is done with the images corrupted using NVIDIA [6]

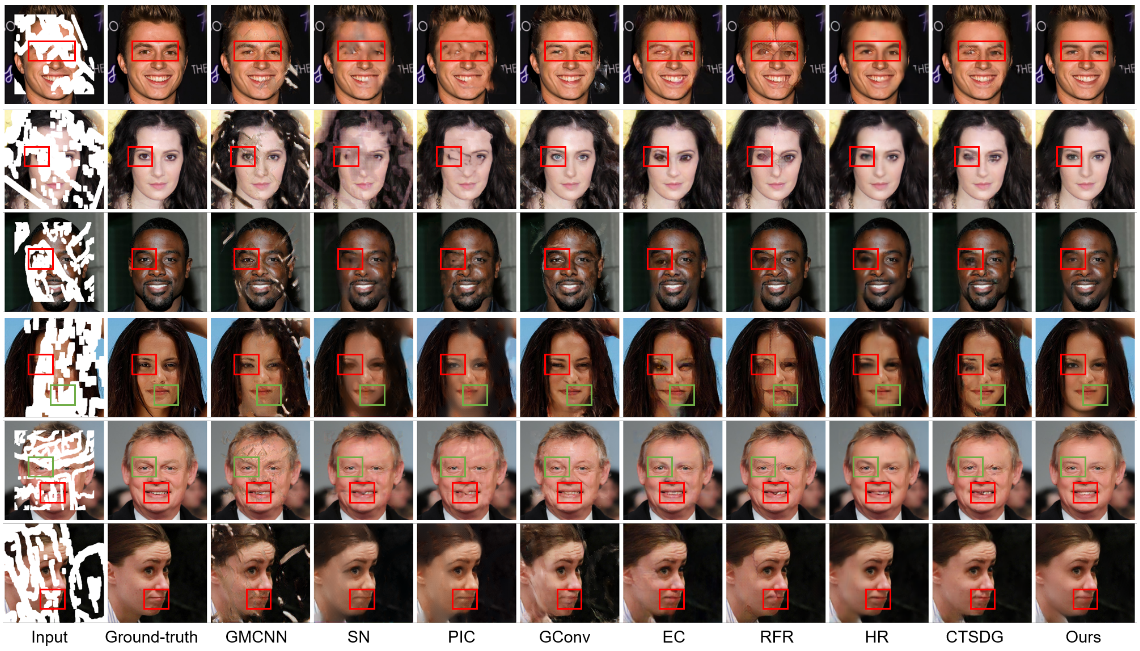


Figure 4.6: Qualitative comparison of the proposed method (Ours) with existing methods (GMCNN [9], SN [10], PIC [11], GConv [12], EC [13], RFR [14], HR [8], CTSDG [16]) on CelebA_HQ dataset for NVIDIA [18] mask.

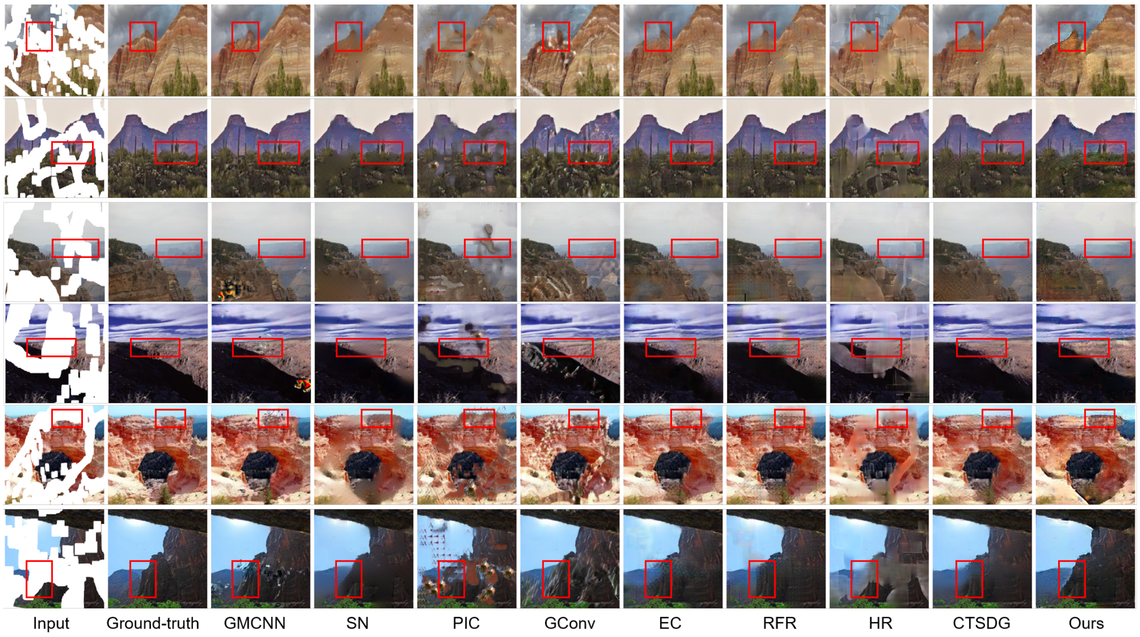


Figure 4.7: Qualitative comparison of the proposed method (Ours) with existing methods (GMCNN [9], SN [10], PIC [11], GConv [12], EC [13], RFR [14], HR [8], CTSDG [16]) on Places2 dataset for NVIDIA [18] mask.

mask dataset.

Table 4.2: Quantitative comparison of the proposed method (Ours) with state-of-the-art methods on NVIDIA [18] masks for image inpainting on Places2 dataset.

| Mask Ratio | Method | Publication | PSNR | SSIM | L1 | LPIPS | FID |
|---|---|---|---|---|---|---|---|
| 0.01-0.2 | SN [10] | ECCV-18 | 27.88 | 0.876 | 3.371 | 0.1340 | 10.763 |
| | GMCNN [9] | NIPS-18 | 28.22 | 0.894 | 3.637 | 0.1125 | 10.543 |
| | PIC [11] | CVPR-19 | 29.52 | 0.917 | 2.796 | 0.1362 | 8.447 |
| | Gconv [12] | ICCV-19 | 29.50 | 0.921 | 2.698 | 0.1269 | 7.718 |
| | EC [13] | CVPRW-19 | 29.69 | 0.915 | 2.585 | 0.1322 | 7.499 |
| | RFR [14] | CVPR-20 | 30.64 | 0.928 | 1.181 | 0.1020 | 6.104 |
| | HR [8] | WACV-21 | 30.12 | 0.936 | 1.661 | 0.0975 | 6.148 |
| | CTSDG [16] | ICCV-21 | 30.86 | 0.953 | 1.490 | 0.0660 | 4.459 |
| | **Ours** | - | **31.28** | **0.964** | **1.170** | **0.0622** | **3.968** |
| 0.2-0.4 | SN [10] | ECCV-18 | 22.67 | 0.816 | 5.173 | 0.2394 | 29.126 |
| | GMCNN [9] | NIPS-18 | 22.82 | 0.858 | 5.532 | 0.2232 | 27.398 |
| | PIC [11] | CVPR-19 | 23.46 | 0.842 | 4.410 | 0.2180 | 25.799 |
| | Gconv [12] | ICCV-19 | 23.30 | 0.872 | 4.393 | 0.2050 | 22.007 |
| | EC [13] | CVPRW-19 | 23.70 | 0.877 | 4.081 | 0.2027 | 21.018 |
| | RFR [14] | CVPR-20 | 24.22 | 0.850 | 3.828 | 0.1935 | 20.218 |
| | HR [8] | WACV-21 | 24.18 | 0.856 | 3.638 | 0.1837 | 19.326 |
| | CTSDG [16] | ICCV-21 | 25.10 | 0.877 | 3.327 | 0.1833 | 18.427 |
| | **Ours** | - | **25.38** | **0.888** | **3.172** | **0.1739** | **17.254** |
| 0.4-0.6 | SN [10] | ECCV-18 | 18.19 | 0.621 | 9.330 | 0.4468 | 74.150 |
| | GMCNN [9] | NIPS-18 | 18.19 | 0.660 | 7.499 | 0.3997 | 73.696 |
| | PIC [11] | CVPR-19 | 18.82 | 0.692 | 7.111 | 0.3713 | 73.408 |
| | Gconv [12] | ICCV-19 | 19.48 | 0.724 | 6.657 | 0.3567 | 68.005 |
| | EC [13] | CVPRW-19 | 19.52 | 0.719 | 6.361 | 0.3598 | 54.341 |
| | RFR [14] | CVPR-20 | 20.76 | 0.726 | 6.486 | 0.3426 | 49.204 |
| | HR [8] | WACV-21 | 20.83 | 0.745 | 5.999 | 0.3351 | 55.461 |
| | CTSDG [16] | ICCV-21 | 21.03 | 0.770 | 5.763 | 0.3285 | 40.266 |
| | **Ours** | - | **21.89** | **0.776** | **5.427** | **0.3117** | **37.887** |

### Baseline

The qualitative and quantitative analysis of the proposed method is done with existing state-of-the-art methods for image inpainting: multi-column image inpainting (GMCNN) [9], Shift-Net (SN) [10], pluralistic image inpainting (PIC) [11], gated convolutions (GConv) [12], EdgeConnect (EC) [13], recurrent feature reasoning (RFR) [14], hyper-realistic image inpainting with hyper-graphs (HR) [8] and CTSDG [16]. For quantitative analysis, the PSNR, SSIM, Mean $L_1$ error, FID and LPIPS metrics are used.

### Result Analysis

The comparison with existing state of the art methods is carried out qualitatively and quantitatively (*The results of existing methods are calculated from the publicly available source codes provided by respective authors*).

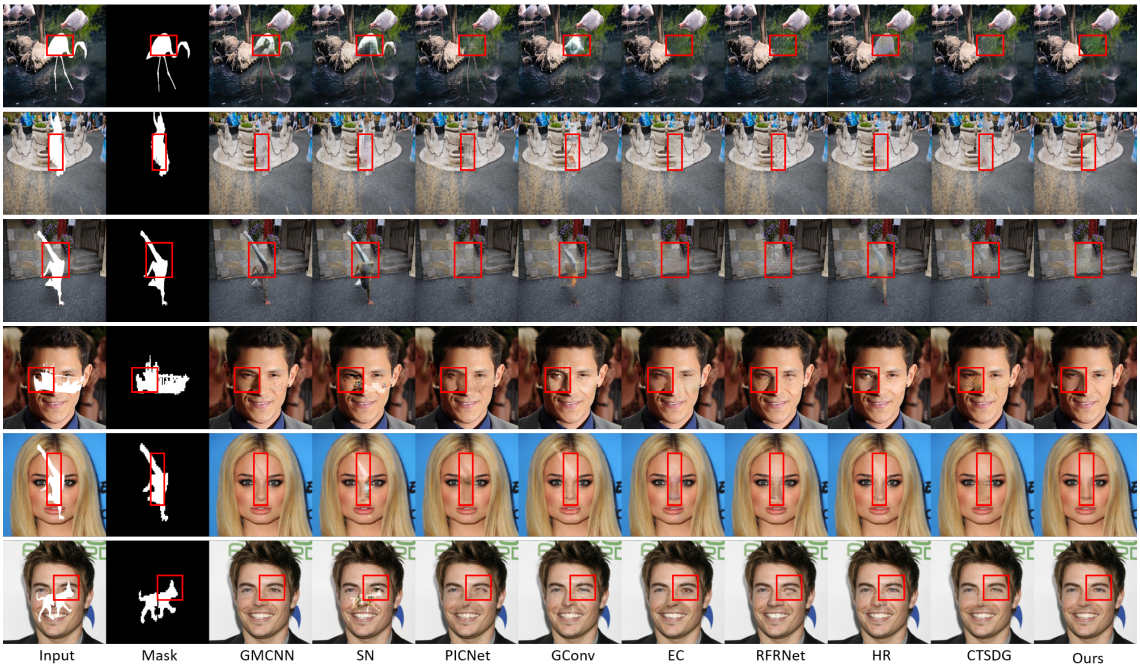


Figure 4.8: Visual comparison of the proposed method (Ours) with existing methods (GMCNN [9], SN [10], PIC [11], GConv [12], EC [13], RFR [14], HR [8], CTSDG [16]) for object removal task (*First three rows- DAVIS-2017 dataset, last three rows- CelebA-HQ dataset*).

*Quantitative Comparison:* The quantitative evaluation for both the CelebA-HQ and Places2 dataset in terms of PSNR, SSIM, $L_1$, FID and LPIPS is given in Table 4.1 and 4.2 respectively. The existing works generally use the prior information such as structural prior, edge prior, recurrent approach or two stage architectures, *etc.* These architectures lack in extracting the features from inputs with different resolutions, whereas our proposed method does so. This multi-resolution allows the network to delve into the features in detail. Also, the proposed MKNL alleviate the correlation of features from different resolutions. The existing three best methods with best results have $31M$, $30M$ and $52M$ number of trainable parameters. Whereas, our proposed method has $14M$ trainable parameters. The proposed method gives an average 0.64 dB and 0.52 dB increase in PSNR, 0.02 and 0.01 increase in SSIM for CelebA-HQ and Places2 datasets, respectively. With this improvement, further there is a noticeable decrease in the values of average $L_1$ error, LPIPS and FID for both the datasets for image inpainting.

*Qualitative Comparison:* The qualitative comparison of the proposed method with the existing methods is given in Figure 4.6 and 4.7 for Celeb_HQ and Places2 dataset respectively. For verifying the effectiveness of inpainting methods qualitatively, we have provided the visual results for $0.4 - 0.6$ mask ratio. From Figure 4.6, it is clear that our proposed method neither creates any discontinuity at hole region and valid region boundary nor or generates asymmetric results. Also, the Figure 4.7 is evidence of structurally pleasant results of proposed method as compared to existing methods for image inpainting.

From both the quantitative and qualitative comparison with state-of-the-art methods, it is clear that the proposed method gives improvement for all lower to higher mask ratios.

### Image inpainting for Object Removal

One of the widely used applications of the image inpainting is object removal. To verify the effectiveness of proposed method for object removal application, we have compared with existing methods for image inpainting. For this experiment we have considered an object segmentation dataset DAVIS-2017 [150]. Also, the analysis is carried out on CelebA-HQ dataset with the object masks of DAVIS-2017 dataset. Figure 4.8 depicts the object removal results. The methods GMCNN [9], SN [10], GConv [12] and HR [8] are unable to remove the object effectively. Methods PIC [11], EC [13], RFR [14] have some distortions. The proposed method gives flawless results as compared to existing methods.

### Ablation Study

To determine **Whether the multi-resolution inputs contribute to remarkable results?**, an ablation study is done and its analysis is given in Table 4.3 and Figure 4.9. Table 4.3 shows that there is improvement in average PSNR and SSIM if image with three different resolutions provided as input to the network. Also, it is clearly seen that the proposed method with $\rho = 1, 2, 3$ generate effective inpainted output as compared to input with two resolutions ($\rho = 1, 2$) or one resolution ($\rho = 1$) only. Further increase in multi resolution inputs degrade the performance (*see last row of Table 4.3* ). This also verifies that the optimized multi resolution analysis of input helps the network to extract more relevant information.

Table 4.3: Ablation study of the proposed method on CelebA-HQ dataset on $0.01 - 0.6$ mask ratio.

| Parameters (Millions) | Input resolutions | MKNL | FP | VFF | PSNR | SSIM |
|---|---|---|---|---|---|---|
| 13.42 | 1 | ✓ | ✓ | ✓ | 26.46 | 0.85 |
| 13.96 | 2 | ✓ | ✓ | ✓ | 27.01 | 0.89 |
| 15.04 | 3 | ✓ | ✓ | × | 28.04 | 0.92 |
| 13.93 | 3 | ✓ | × | ✓ | 27.86 | 0.91 |
| 13.44 | 3 | × | ✓ | ✓ | 27.52 | 0.90 |
| **14.04** | **3** | ✓ | ✓ | ✓ | **28.25** | **0.94** |
| 14.59 | 4 | ✓ | ✓ | ✓ | 27.25 | 0.92 |

The another study is carried out **to verify the effectiveness of proposed MKNL block**. The quantitative evaluation for the same is tabulated in Table 4.3 and visualized

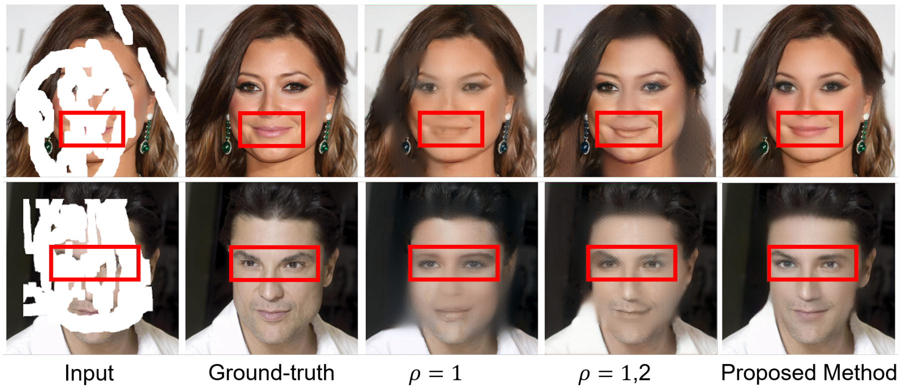


Figure 4.9: Analysis on effect of multi-resolution inputs on CelebA-HQ dataset ($\rho = 1$ *means only actual resolution image as input,* $\rho = 1, 2$ *means two inputs i.e., actual image and image down-sampled by* 2).

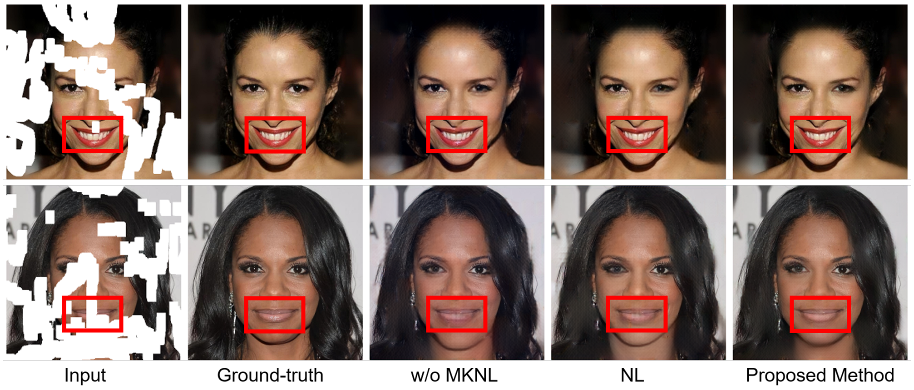


Figure 4.10: Analysis on effect of the proposed multi-kernel non-local attention block on CelebA-HQ dataset (*w/o MKNL- without proposed MKNL block, NL - existing non-local attention [19] instead of proposed MKNL*).

in Figure 4.10. In Figure 4.10, we can see that the outcomes without proposed MKNL (w/o MKNL) generated uncorrelated results (*see bounding boxes*). We also **verified the inclusion of existing non-local attention** [19] **instead of MKNL** in the proposed architecture. The difference between the existing non-local attention (NL) [19] and MKNL is that, the MKNL extracts the features of input with multiple kernels. This allows to consider the multi-scale long-range dependencies from the input feature maps. This can be easily proved from Figure 4.10, where the results of NL still have uncorrelated outcomes as compared to results of MKNL.

The valid feature fusion in the proposed architecture is utilized to extract the knowledge of hole locations while extracting the valid encoder features through skip connections and adding them with the decoder features at hole locations. This in turn helps the reconstruction of inpainted image. We carried out an experiment of **whether the VFF instead of simple `Concatenation` → `Convolution` of skip connection from encoder helps the architecture for effective reconstruction?** Table 4.3 shows that,

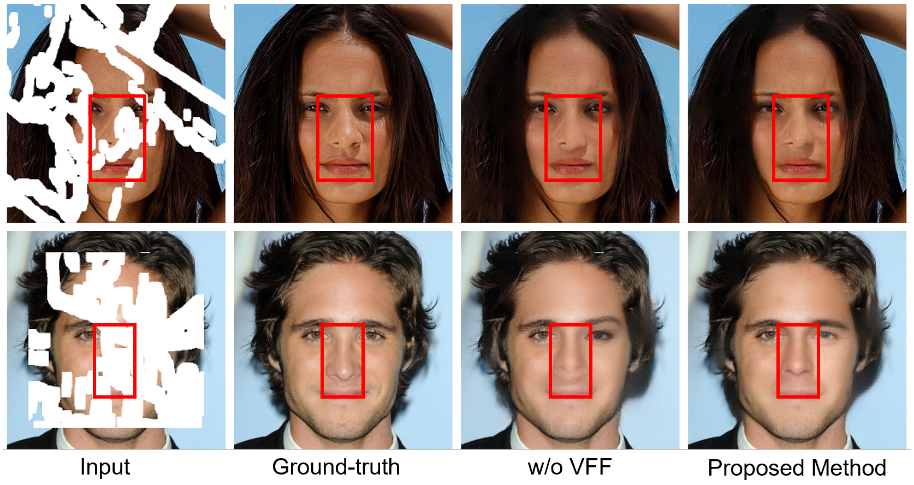


Figure 4.11: Analysis on effect of valid feature fusion on CelebA-HQ dataset (*w/o VFF- without proposed VFF i.e., only* `concatenation` $\rightarrow$ `convolution` *of features*).

without VFF the results of proposed method decreases with the increase in number of trainable parameters. Also, Figure 4.11, shows that the output of proposed method with just simple `Concatenation` → `Convolution` generate unnatural and ghosting results at hole locations. Whereas, a simple VFF can generate faithful results at the hole locations. To effectively utilize the multi-resolution features, we proposed a feature projection block (FP) which projects the weighted features from multi-resolution input to each decoder level for efficient reconstruction. **To verify the effectiveness of FP**, we carried out an experiment, where the multi-resolution features are directly forwarded to each decoder level by only up-sampling them for each level *i.e.,* **w/o FP** means direct concatenation of features from multi-resolution inputs. Table 4.3 shows that the removal of FP block affects on the performance of proposed architecture. The visual results (Figure 4.12) also shows that without feature projection there is asymmetry in the inpainted results. An additional experiment is performed by **replacing the proposed FP with existing squeeze and excitation module** [20]. In Figure 4.12, we can observe that the squeeze excitation block works well when considered to direct feature concatenation but does not provide spatial consistency as compared to FP. Training of the network with effective loss functions plays a vital role towards appropriate outcomes. **To determine the effectiveness of utilized**

Table 4.4: Analysis on the effect of losses on CelebA-HQ dataset with average PSNR and SSIM on $0.01 - 0.6$ mask ratio.

| Total Loss | PSNR | SSIM |
|---|---|---|
| $L_1$+$L_{GAN}$ | 26.94 | 0.87 |
| $L_1$+$L_{GAN}$+$L_{Edge}$ | 27.25 | 0.90 |
| $L_1$+$L_{GAN}$+$L_{Edge}$+$L_{Per}$ | 27.98 | 0.91 |
| $L_1$+$L_{GAN}$+$L_{Edge}$+$L_{Per}$+$L_{SSIM}$ | **28.25** | **0.94** |

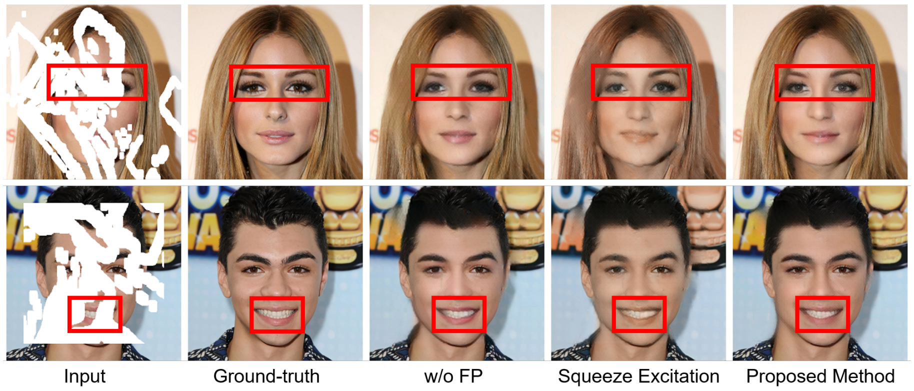


Figure 4.12: Analysis on effect of feature projection on CelebA-HQ dataset (*w/o FP- without proposed FP, Squeeze Excitation - existing squeeze excitation [20] instead of proposed FP*).

**losses, the experimental analysis with different loss functions is carried out.** Table 4.4 enlist the combinations of losses used while training the proposed network. From Table 4.4, it is clear that, utilizing the combination of $L_1$, $L_{GAN}$, $L_{Edge}$, $L_{Per}$ and $L_{SSIM}$ helps the proposed network for efficient learning for image inpainting.

## 4.2 Diverse Receptive Field based Adversarial Concurrent Encoder Network for Image Inpainting

Use of the normal encoder-decoder architecture may result in unambiguous results as the correlation between the pixel values in the non-hole region reduces when the size of the region to be filled increases [96], [12]. For this, the progressive fashion-based image inpainting method is proposed in [151] which is computationally expensive. To overcome this limitation of progressive approach, in [14] the authors proposed a recurrent feature reasoning for image inpainting. This recurrent approach may fail if the inference at the first iteration does not produce the effective boundary related feature space, which will be gathered in successive inferences to inpaint the hole regions. To overcome these limitations, focusing on the diverse receptive fields for hole region with respect to non-hole region may help to reproduce a semantically plausible inpainted image. The progressive and recurrent approaches give superior results for inpainting by ignoring an important aspect of computational complexity. The concurrent encoder feature learning may be effective as the feature learning will be independent of each other with an advantage of lower computational complexity unlike in recurrent or progressive approaches. Thus, we have proposed a lightweight concurrent encoder approach consisting of residual diverse receptive fields for image inpainting. The number of parameters in our model is far lower ($4.8M$) compared to those in the current methods for e.g. [69] uses $100M+$ parameters, [18] uses $33M$ parameters, and [14] uses $31M$ parameters. The major contributions of this work are:

- A novel lightweight adversarial framework is proposed with concurrent encoders by integrating diverse receptive fields for image inpainting.
- Concurrent processing of multi-level encoder features with an advantage of lower computational complexity.
- Design of residual diverse receptive module to effectively correlate the hole and non-hole regions in an image.

The quantitative and qualitative results' comparison with state-of-the-art approaches is carried on Places2 [3] and Paris Street View (Paris_SV) [4] datasets for image inpainting.

### 4.2.1 Proposed Framework

The available approaches for image inpainting generally follow a coarse-to-fine [69] architecture or progressive fashion [151] or a recurrent architecture [14]. In these types of architectures, if the initial inpainting stage gives irrelevant generated content, those content will be carry forwarded for further inpainting tasks. This may create an

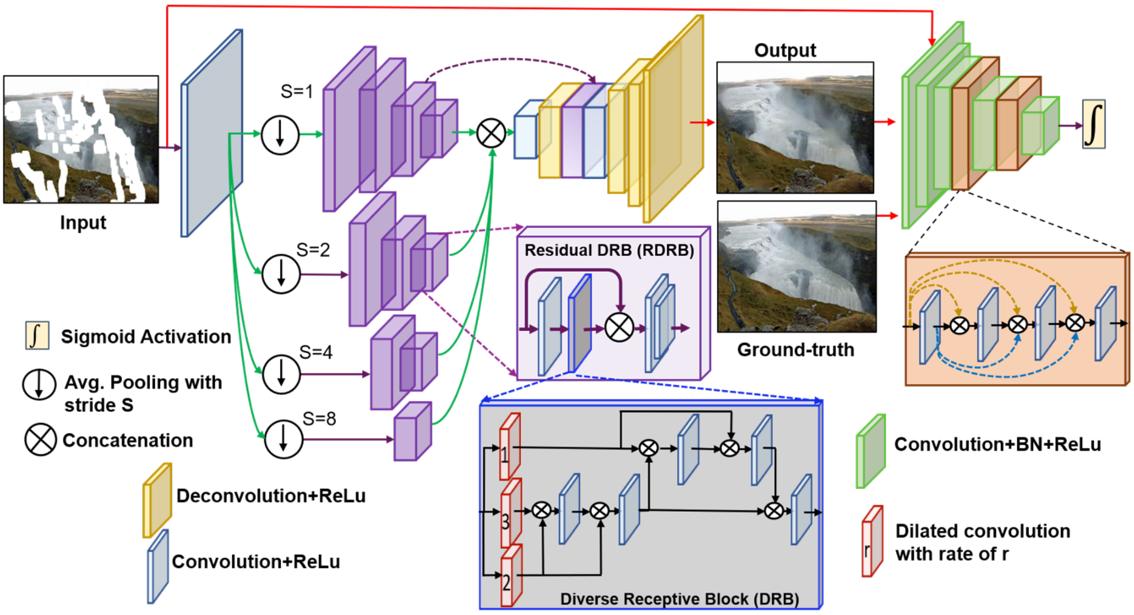


Figure 4.13: Architectural details of the proposed framework for image inpainting.

unambiguous content generation which in turn results in semantically irrelevant inpainted patches in the output image. In [152], the authors proposed a multi-scale network that works as coarse-to-fine and fine-to-coarse architecture to maximize the information flow for image de-blurring. To overcome limitations of existing inpainting methods and motivated by [152], we have proposed a lightweight concurrent encoder architecture for image inpainting based on GANs as shown in Figure 4.13. The generator architecture consisting of a concurrent encoder features processing approach with residual diverse receptive block (RDRB) and the conditioned discriminator (with input-generation and input-ground-truth pair) architecture are shown in Figure 4.13. The concurrent paths are designed to maximize the information to be provided to the decoder. Also, each of the concurrent paths comprises the residual diverse receptive blocks to educe information from various receptive fields.

The input image to be inpainted is first processed through a single convolution block with filter size $3 \times 3$ with stride 1. The feature maps are then down-sampled by average pooling operation with a stride factor of $s = 2^{P-1}$, $P \in [1, 4]$ and given to each path ($P$). These feature maps are then processed through four concurrent paths ($P \in [1, 4]$). The number of RDRB blocks in each path are given as: $RDRB_i^{f \times i}$, $i \in (1, N-P)$ where $N = 5$, $f \times i$ is number of output feature maps with $f = 32$. As previously described, the RDRB focuses on diverse receptive fields in each path to extract feature context from different receptive fields in each path. The feature maps from all the concurrent paths are then concatenated and passed through the `Convolution` → `ReLu` in order to accumulate relevant features from all the concurrent paths. These concatenated feature maps are fed to the decoder part for the reconstruction of input. The decoder is defined as $Dconv_l^{f \times (N-l)}$, $l \in (1, 4)$ with stride factor of 2. A brief explanation and significance of concurrent multi encoder path processing and RDRB in the proposed architecture are as follows:

### Concurrent Multi-encoder Processing

To overcome limitations of existing approaches, we have proposed a concurrent architecture as shown in Figure 4.13. Here, the feature maps, after convolution with stride 1, are processed through four paths. The feature maps from the first convolution layer are downscaled by the average pooling $S \in (1, 2, 3, 4)$ with kernel size $S \times S$ as shown in Figure 4.13. The concurrent processing of those feature maps is done via four paths. In each path, these feature maps are processed through a number of RDRB blocks to get efficient features by focusing on diverse receptive fields for correlating the non-hole and hole region. Each of the paths processes different feature maps concurrently, because of which the proposed architecture comes up with a lower number of trainable parameters.

### Residual Diverse-Receptive Block

For the image inpainting, the input may have missing regions with different size and shape. A single fixed receptive field may fail at filling the regions with varying size and shape. In order to mitigate this issue, it is desired that the network should learn the different receptive fields features. To focus on the diverse receptive fields for the region to be inpainted, we have proposed RDRB consisting of a diverse receptive block (DRB). The DRB generates the output feature map $Out_{DRB}$ as:

$$\begin{aligned} Out_{DRB} &= \varphi\left(Out_{DC_{123}} \otimes Out_{DC_{23}}\right) \\ Out_{DC_{123}} &= \varphi\left(\varphi\left(Out_{DC_{23}} \otimes DC_1\right) \otimes DC_1\right) \\ Out_{DC_{23}} &= \varphi\left(\varphi\left(DC_3 \otimes DC_2\right) \otimes DC_2\right) \end{aligned} \tag{4.8}$$

where, $DC_r$ is the dilated convolution with dilation rate $(r)$, $\otimes$ , $\varphi$ indicates concatenation and `Convolution`→`Relu` respectively. In DRB, the features are concatenated from different receptive fields which extract the efficient features from the required receptive field. This will help the inpainting architecture to fill the image with diverse hole regions effectively. Also, we have defined the RDRB to avoid the vanishing gradients problem and is given in Eq. (4.9)

$$RDRB_{out} = \left(\left(Conv_{s=1}^{3\times3} \rightarrow DRB\right) \otimes F_{in}\right) \rightarrow Conv_{s=1}^{1\times1} \rightarrow Conv_{s=2}^{3\times3} \tag{4.9}$$

where, $F_{in}$ is input feature map and $Conv_s^{m\times m}$ is convolution with stride $s$ and filter size $m \times m$ followed by Relu activation. The concurrent multi encoder processing with RDRB blocks pave a way to efficient image inpainting.

### 4.2.2 Training of the Proposed Network

The proposed method makes use of an adversarial training procedure because image inpainting is a similar task like image-to-image translation [133] where the goal is to fill the corrupted region with structurally plausible content. The training and testing of the proposed architecture is done on two datasets: Places2 [3] and Paris_SV [4]. The masks from [18] are used for the training and testing of the proposed architecture for image inpainting.

The overall loss for training of the network $L_{total}$ is given as follows:

$$\begin{aligned} L_{total} = \lambda_1^{holes} L_1^{holes} + \lambda_1^{non\_holes} L_1^{non\_holes} + \\ \lambda_{Adv} L_{Adv} + \lambda_{Edge} L_{Edge} + \lambda_{Per} L_{Per} \end{aligned} \tag{4.10}$$

where, $\lambda_1^{holes}$, $\lambda_1^{non\_holes}$, $\lambda_{Adv}$, $\lambda_{Edge}$ and $\lambda_{Per}$ are the weights for hole loss, non hole loss, adversarial loss, edge loss and perceptual loss respectively. The network is trained for 150 epochs using Adam optimizer with a learning rate of 0.0002 and beta = 0.5. The values of weights for the given losses are $\lambda_1^{holes} = 7$, $\lambda_1^{non-holes} = 3$, $\lambda_{ad} = 1$, $\lambda_{edge} = 1.5$, and $\lambda_P = 1.75$. Remaining settings of the model are similar to the [133]. Weights of network are updated on NVIDIA DGX station having Tesla V100 1×16 GB GPU.

### 4.2.3 Experimental Analysis

Detailed qualitative and quantitative evaluation of results estimated using the proposed network with existing SOTA methods is discussed in this section. The result analysis is done on two benchmark datasets *i.e.,* Places2 [3] and Paris_SV [4]. *The training and testing of the network is done by using the masks provided by [6]* similar to [14]. The testing masks are considered with different hole to image ratio *i.e.,* (0.1, 0.2], (0.3, 0.4], (0.5, 0.6]. Each of the mask ratio categories includes 2k masks.

#### Result Analysis

The quantitative comparison is done with CA [94], PIC [11], PConv [18], GConv [12], EC [13], PRVS [73], RFR [14] and UHR [22] in terms of PSNR and SSIM. The quantitative results for existing state-of-the art methods are taken from [14, 22]. For qualitative comparison, the results are generated by using the source code provided by respective authors. Table 4.5, 4.6 show the quantitative comparison of the proposed architecture with SOTA methods for image inpainting. From Table 4.5, 4.6, it is clear that the proposed method outperforms all the existing methods in terms of PSNR and SSIM for image inpainting. The qualitative result comparison of the proposed method with SOTA methods [11], [12], [13], [14] is shown in the Figure 4.14. From Figure 4.14, it is observed that the proposed method does not produce any patches in the generated image and produces the semantically plausible results. This shows the effectiveness of the proposed

Table 4.5: Quantitative comparison of the proposed method (Ours) with state-of-the-art methods on Places2 [3] dataset for image inpainting.

| Metric | Methods | Publication | 0.1-0.2 | 0.3-0.4 | 0.5-0.6 |
|---|---|---|---|---|---|
| PSNR↑ | CA [94] | CVPR-16 | 25.81 | 20.89 | 17.10 |
| | PIC [11] | CVPR-19 | 27.14 | 21.72 | 17.17 |
| | Pconv [18] | ECCV-18 | 27.29 | 22.15 | 18.29 |
| | GConv [12] | ICCV-19 | 27.05 | 21.55 | 16.94 |
| | EC [13] | ICCVW-19 | 27.17 | 22.18 | 18.35 |
| | PRVS [73] | ICCV-19 | 27.41 | 22.36 | 18.67 |
| | RFR [14] | CVPR-20 | 27.75 | 22.63 | 18.92 |
| | UHR [22] | CVPR-20 | 25.36 | 20.21 | 16.07 |
| | **Ours** | **SPL-21** | **30.23** | **24.84** | **20.37** |
| SSIM↑ | CA [94] | CVPR-16 | 0.906 | 0.783 | 0.648 |
| | PIC [11] | CVPR-19 | 0.932 | 0.786 | 0.494 |
| | Pconv [18] | ECCV-18 | 0.934 | 0.803 | 0.555 |
| | GConv [12] | ICCV-19 | 0.921 | 0.796 | 0.626 |
| | EC [13] | ICCVW-19 | 0.933 | 0.802 | 0.553 |
| | PRVS [73] | ICCV-19 | 0.936 | 0.81 | 0.574 |
| | RFR [14] | CVPR-20 | 0.939 | 0.819 | 0.596 |
| | UHR [22] | CVPR-20 | 0.905 | 0.762 | 0.588 |
| | **Ours** | **SPL-21** | **0.961** | **0.897** | **0.794** |

parallel diverse receptive approach on the Places2 and Paris_SV dataset.

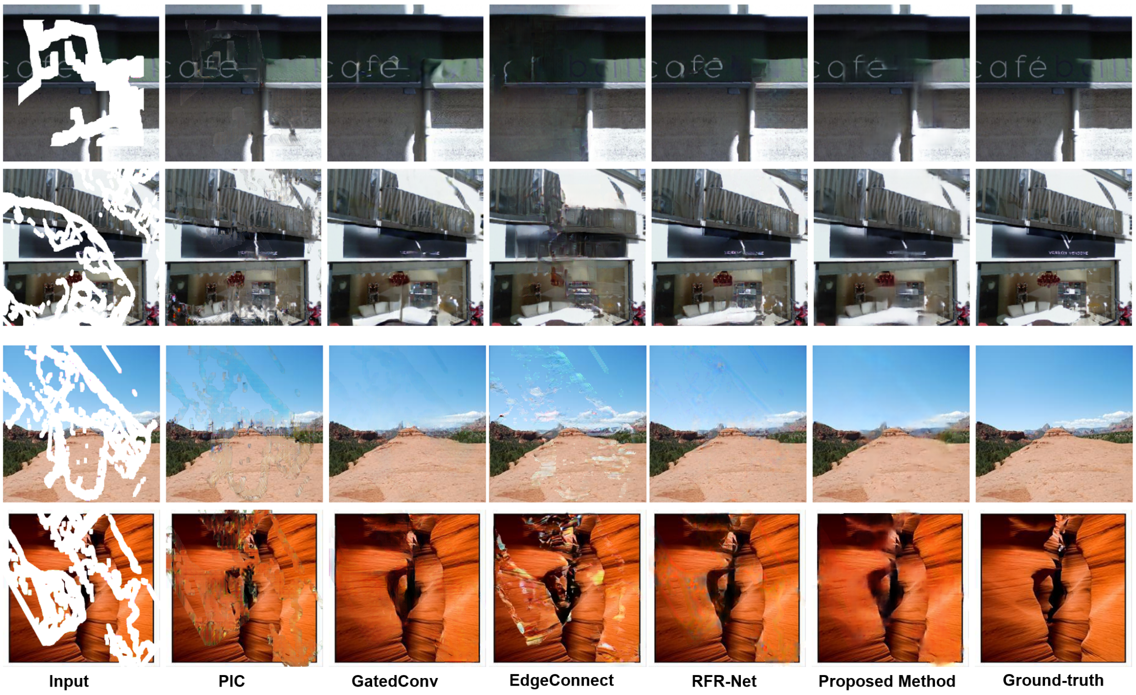


Figure 4.14: Qualitative comparison of the proposed method with state-of-the-art methods (PIC[11], GatedConv [12], EdgeConnect [13], RFR-Net[14]). *Note: Row 1, 2 - Paris_SV dataset and row 3, 4 Places2 dataset.*

Table 4.6: Quantitative comparison of the proposed method (Ours) with state-of-the-art methods Paris_SV dataset [4] for image inpainting.

| Metric | Methods | Publication | 0.1-0.2 | 0.3-0.4 | 0.5-0.6 |
|---|---|---|---|---|---|
| PSNR ↑ | PIC [11] | CVPR-19 | 29.35 | 23.97 | 19.52 |
| | Pconv [18] | ECCV-18 | 30.76 | 25.46 | 21.39 |
| | GConv [12] | ICCV-19 | 31.32 | 25.54 | 20.61 |
| | EC [13] | ICCVW-19 | 31.19 | 26.04 | 21.89 |
| | PRVS [73] | ICCV-19 | 31.49 | 26.17 | 22.07 |
| | RFR [14] | CVPR-20 | 31.71 | 26.44 | 22.40 |
| | **Ours** | **SPL-21** | **31.87** | **26.57** | **22.50** |
| SSIM ↑ | PIC [11] | CVPR-19 | 0.93 | 0.785 | 0.519 |
| | Pconv [18] | ECCV-18 | 0.947 | 0.835 | 0.619 |
| | GConv [12] | ICCV-19 | 0.953 | 0.846 | 0.621 |
| | EC [13] | ICCVW-19 | 0.95 | 0.846 | 0.646 |
| | PRVS [73] | ICCV-19 | 0.953 | 0.854 | 0.659 |
| | RFR [14] | CVPR-20 | 0.954 | 0.862 | 0.681 |
| | **Ours** | **SPL-21** | **0.954** | **0.871** | **0.744** |

Table 4.7: Computational complexity analysis on Paris Street View dataset.

| Approach | Coarse-to-fine [69] | Pconv [18] | RFR [14] | Ours |
|---|---|---|---|---|
| # Parameters | 100M+ | 33M | 31M | **4.8M** |
| PSNR | 23.1 | 23.69 | 24.6 | **26.52** |
| SSIM | 0.768 | 0.759 | 0.796 | **0.875** |

### Parameter Analysis

We have compared the proposed model with existing (Coarse-to-fine [69] , Pconv [18] and RFR [14]) state-of-the-art methods in terms of number of parameters (the results for existing method are taken from [14]). The comparison is done on Paris_SV dataset for 0.4-0.5 mask ratio. As given in Table 4.7, the proposed model has fewer parameters as compared to the existing methods. With less number of parameters, the proposed approach outperforms the existing methods in terms of PSNR and SSIM for 0.4-0.5 mask ratio on Paris_SV dataset. This shows that the proposed concurrent encoders with diverse receptive filed give the advantage of less number of parameters with superior results.

### Ablation Study

The ablation study is performed on Paris_SV dataset. The purpose of integrating DRB in the concurrent encoder is to focus on the diverse receptive fields to fill different hole regions for image inpainting. **Does the proposed DRB helps the network to inpaint different hole sizes effectively?** To examine the effectiveness of the proposed DRB, we have analysed the network performance with and without DRB for different combinations of concurrent paths (1 to 5) as given in Table 4.8. For without DRB analysis, the DRB block is removed and two convolution layers with stride 1 and 2 are used respectively.

Table 4.8: Analysis number of concurrent paths with and without DRB

| Mask Ratio | Metric | DRB | Number of Parallel Paths | | | | |
|---|---|---|---|---|---|---|---|
| | | | 1 | 2 | 3 | 4 | 5 |
| 0.5-0.6 | PSNR | Without | 16.78 | 17.62 | 19.52 | 21.16 | 20.13 |
| | | With | <u>19.14</u> | <u>19.83</u> | <u>20.14</u> | <u>**22.50**</u> | <u>21.85</u> |
| | SSIM | Without | 0.593 | 0.605 | 0.684 | 0.714 | 0.695 |
| | | With | <u>0.660</u> | <u>0.709</u> | <u>0.726</u> | <u>**0.744**</u> | <u>0.734</u> |

Here, the training is done separately for with and without DRB block. From Table 4.8, it is evident that the proposed framework with DRB gives superior results as compared to without DRB. Also, the concurrent processing of encoder features is employed in the proposed network. **Whether this concurrent processing of encoder features help the network to integrate the effective features?** To do this, the result of the proposed network is examined with combinations of concurrent paths. This analysis is depicted in Table 4.8. *From results reported in Table 4.8, it is clear that the proposed method with four concurrent paths consisting of DRBs is effective for image inpainting.*

## 4.3 Pseudo Decoder Guided Light-weight Architecture for Image Inpainting

The existing approaches [94, 153] fail to reproduce correlated global and local information in images. Because, either they make use of the latent prior which fails to reproduce high dimensional information or they use of local and global information separately which reduce the consistency in reproduced contents. Also, the prior information like noise prior [110], or edge/structure information [13], [70] based methods learn the reproduction of the information and incorporate it in the inpainting task. The distribution of these contents like noise prior, generated edges or structures may vary from the actual distribution of the required image. This may lead to lack of useful information like color or texture in the completed image. Some of the non-blind methods use knowledge of masks for passing the relevant contents from one stage to next stage. This may generate disturbances at the edges of the hole and non-hole regions in the completed images [22], [8]. The coarse-to-fine architectures require numerous resources which lead to high computational cost of the network [8]. The existing methods give effective outcomes regardless of computational complexity as shown in Table 4.9. From Table 4.9, we can see that, the existing methods have very high computational complexity. Also, the methods which use the prior information like edge information have to rely on the available methods for generation of prior information. The progressive [73] or the recurrent [14] methods give convincing results as compared to coarse-to-fine architectures. Considering these limitations of existing methods, we have proposed a light-weight architecture for image inpainting (*see Ours in Table 4.9*) without need of any prior information. The authors in [21], proposed a method based on pyramid context encoder by passing the encoder information to the decoder similar to U-Net with an attention mechanism. Motivated from this, we have used the encoder multi level information by merging the feature maps from all the encoder levels and extracting the relevant information from the encoder levels unlike [21], where the information is passed from each encoder to respective decoder. In this work, we have proposed an end-to-end multi-scale pseudo decoder weighted reconstruction architecture for image inpainting. In the proposed architecture, we have used the mask information with different scales to forward the relevant information for the regeneration of complete image. The main contributions of the proposed work are:

- An end-to-end light-weight architecture is proposed for image inpainting with very less number of parameters (0.97M).
- The concept of encoder multi level feature fusion is proposed to take out the valid context from each level (Section 4.3.1).
- A pseudo decoder is proposed to share weighted features from encoder levels to

regeneration decoder for effective image inpainting (Section 4.3.1).

- A novel regeneration decoder is proposed to merge the features from the pseudo decoder and corresponding encoder levels (Section 4.3.1).

The comparison of the proposed method is done qualitatively and quantitatively on three benchmark datasets: CelebA-HQ [36], [1], Paris Street View (Paris_SV) [4] and Places2 [3]. The corrupted images with the masks from [18] and synthetically generated mask from [8] are used for image inpainting task. Also, the effectiveness of proposed method is verified for high resolution image inpainting.

Table 4.9: Computational complexity analysis of the proposed method (Ours) and existing methods in terms of number of parameters and FLOPs for image inpainting

| Method | Publication | Parameters (Millions) | FLOPs ($\times 10^9$) |
|---|---|---|---|
| CTSDG [16] | ICCV-21 | 52.14 | 17.65 |
| HR [8] | WACV-21 | 30.25 | 23.85 |
| RFRNet [14] | CVPR-20 | 31.30 | 206.11 |
| UHR [22] | CVPR-20 | 2.70 | 41.46 |
| GConv [12] | ICCV-19 | 4.050 | 55.57 |
| PEN [21] | CVPR-19 | 10.23 | 48.07 |
| PICNet [11] | CVPR-19 | 3.630 | - |
| EC [13] | CVPRW-19 | 53.00 | 128.98 |
| SN [10] | ECCV-18 | 54.94 | 70.10 |
| PConv [18] | ECCV-18 | 33.00 | 18.95 |
| GMCNN [9] | NIPS-18 | 3.115 | - |
| **Ours** | **TIP-22** | **0.971** | **13.7** |

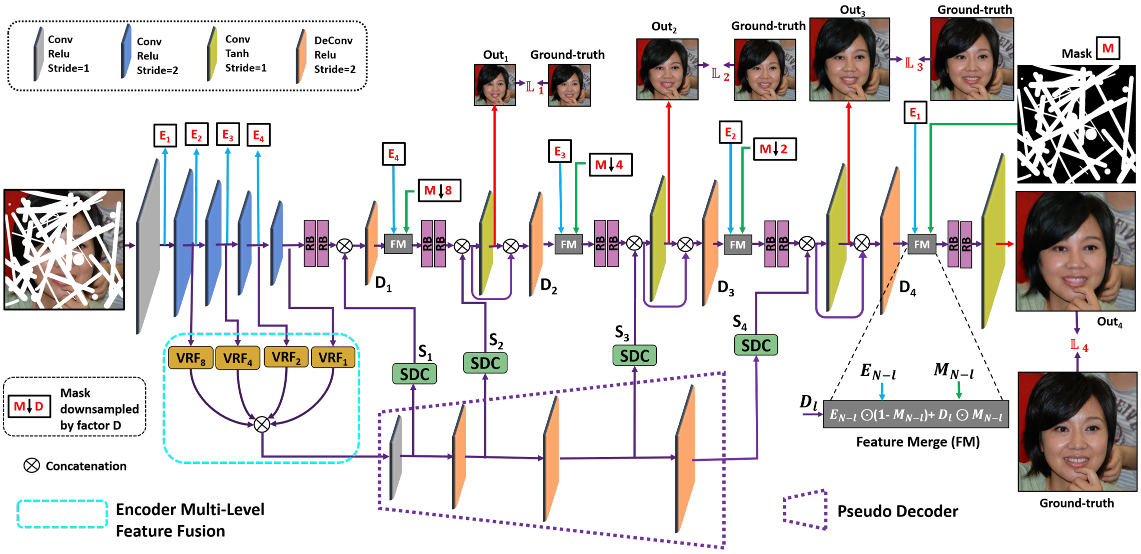


Figure 4.15: Proposed light-weight generator architecture for image inpainting. *Note:* $\mathbb{L}_1, \mathbb{L}_2, \mathbb{L}_3$ *and* $\mathbb{L}_4$ *are the losses calculated at each scale* $[1\,to\,4]$ *with respect to ground-truth while training and the total loss is the sum of all the losses. The final outcome while inference/testing is the outcome of last scale (i.e.,* $Out_4$*).*

### 4.3.1 Proposed Framework

Image inpainting desires to synthesize the missing regions in an image. In learning-based methods, specifically, in encoder-decoder based architectures when it is required to provide the skip connections for image inpainting task, the task becomes difficult as it may merge the hole content without any relevant features. Different layers like partial convolution [18] and gated convolution [12] are proposed for utilizing the hole related relevant features for image inpainting. Also, many methods [68], [8], [22] make use of coarse-to-fine architecture, where processed features from the first stage will be given to fine (second) stage for generating efficient results. In spite of exceptional results on image inpainting, these methods demand high computational complexity in terms of number of parameters and FLOPs (*see Table 4.9* ). Considering this, here, propose an end-to-end single stage pseudo decoder guided reconstruction architecture for image inpainting. The proposed method makes use of the mask at each level of regeneration decoder for merging the valid features from the encoder level and weighted features from pseudo decoder, unlike coarse-to-fine architectures. The proposed generator architecture for image inpainting is shown in Figure 4.15.

As shown in Figure 4.15, the features from each of the encoder levels ($E_1 - E_4$) are processed through the varying receptive fields blocks and then fused to form effective feature maps by focusing on different receptive fields. Further, pseudo decoder path takes the fused features from all the encoder levels and then the response of each pseudo decoder level is processed through the space-depth correlation based weighted features to provide weighted guidance for regeneration decoder. In regeneration decoder path, weighted features from pseudo decoder and from preceding decoder level are concatenated in order to forward the significant features to next decoder level with the help of weighted pseudo feature maps. This provides the correlated weight to the feature maps. The feature merge (FM) block is then used to merge the valid content from respective encoder level and highly correlated features from decoder levels for efficient learning. While training of the network, multi scale loss is calculated at each level of the regeneration decoder for learning the detailed information. Each module of the proposed architecture is explained in the next subsections.

#### Encoder Multi Level Feature Fusion

Stacking more encoder levels is one of the ways to consider maximum receptive fields for extracting valid information to inpaint the image. Further, forwarding the features of only last encoder level to inpaint the image may fail to reproduce the detailed texture and structure in the inpainted image. The shallow encoder layers contain most of the textural information whereas the deep layers contain structural information. Existing approach for image inpainting considers to process the information from encoder levels with structural

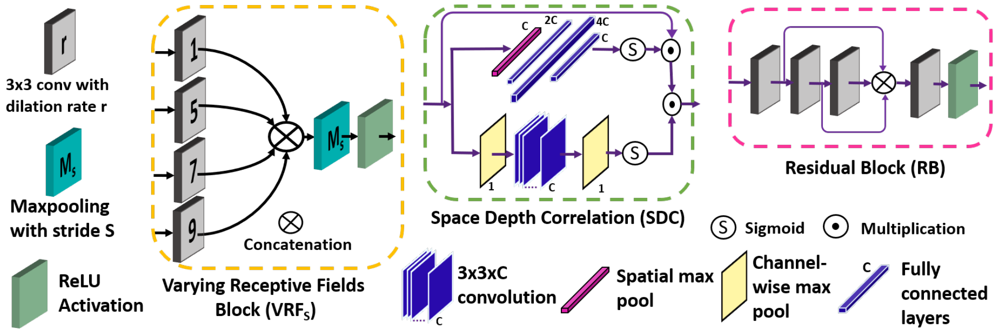


Figure 4.16: Details of varying receptive fields (VRF) block, space depth correlation (SDC) module and residual block (RB) proposed in the architecture.

and textural maps separately [154]. This leads to the uncorrelated structural and textural information. Also, we assume, the dominance of holes in the feature maps reduces as we further go deeper in the encoder layers. So, every encoder layer owns unique features (valid textural and structural both) which may help to inpaint the image efficiently. Further, direct merging of the features from encoder to decoder may pass the irrelevant hole features to the decoder leading to unpleasant results. Hence, instead of processing either the feature maps from the last encoder level or structural and textural feature maps separately or providing direct skip connections, here, we have proposed the encoder multi-level feature fusion module (EMLFF). In EMLFF, the feature maps from each of the encoder levels (`convolution with stride 2` → `ReLU`) are fused and processed to extract the feature maps relevant for filling out hole regions. This will help the network to learn structurally and texturally correlated features from encoder. Also, to extract most relevant textural and structural feature to fill the holes of any size, it is required to consider the features from different receptive fields. In [21], the authors merged the encoder level features with the corresponding decoder level by processing them through the dilated convolution blocks with direct skip connections which may transfer invalid content if the mask is very large. To avoid this, we propose to process all the encoder features with diverse receptive fields and merge them for effective feature extraction helping the network towards effective inpainting results. This will help the network to extract most relevant structural and textural information from different receptive fields. *The effect of the proposed encoder multi-level feature fusion (EMLFF) is analysed in Section 4.3.3 .*

The feature maps from each of the encoder layers are merged after passing through the varying receptive fields (VRF) block. **This merging mechanism does the exploration of relevant/valid context with various receptive fields from all the encoder levels.** In VRF block, the maximum of all the encoder-level features from different receptive fields is considered for extracting the most relevant features from different receptive fields. This block helps to extract the features not only from a single receptive field but also from different receptive fields which in turn helps for filling the holes from small to large hole size. The VRF block for each of the encoder levels can be represented as given in Eq. (4.11) (*see VRF block in Figure 4.16* ).

$$VRF_S = max_S\{\varphi^{1,1}_{3\times3}(\xi_l), \varphi^{1,5}_{3\times3}(\xi_l), \varphi^{1,7}_{3\times3}(\xi_l), \varphi^{1,9}_{3\times3}(\xi_l)\} \tag{4.11}$$

where, $max_S$ indicates max-pooling operation with stride $S$ ($S\epsilon[8,4,2,1]$ for $l^{th}$ encoder layer $l\epsilon[1,4]$ respectively), $\varphi^{s,r}_{m\times m}(\xi_l)$ indicates the convolution with filter size $m \times m$, stride $s$ and dilation rate $r$ on $l^{th}$ level feature map $\xi_l$. The feature maps from four encoder layers are merged with a concatenation operation after passing through respective VRF. Processing each level from the VRF block will extract the maximum contextual information by considering different receptive fields (*analysis on the effect of VRF is carried out in Section 4.3.3* ). Further, these merged features are forwarded to the pseudo decoder. So, in general the encoder multi level feature fusion block helps the image inpainting architecture:

- To focus on the varying receptive fields for filling out the regions with smaller to larger hole sizes.
- To extract the most relevant feature maps from each of the encoder levels.

### Pseudo Decoder Weighted Feature Sharing

The fused feature maps from the encoder are processed in the pseudo decoder to merge them at each of the respective regeneration decoder levels. As this decoder is not considered for actual image reconstruction, instead it provides the supportive features for the reconstruction of the inpainted image, so it is named as *pseudo decoder*. The pseudo decoder is defined as `convolution with stride 1` → `ReLU` and `3 deconvolution layers with stride 2` → `ReLU` as shown in the Figure 4.15. The processed features from the pseudo-decoder are further attentively forwarded to the regeneration decoder.

Existing methods [103, 79, 113, 120] generally try to find similarity in the encoded features of hole locations and valid locations by using the similarity measures between the patches of hole locations and valid locations at the encoder. This similarity attention feature maps are then utilized to forward the information to decoder. This patch based similarity measure lacks in depth-wise correlated information. The spatial and channel wise excitation blocks are proposed in [155, 156] to provide semantic and contextual information in image captioning and segmentation tasks. These methods apply aggregated spatial and channel-wise attention or employ a selective spatial and channel excitation. Unlike existing methods, the spatial and channel-wise extracted features are merged attentively **to correlate the depth wise and spatial features** from the pseudo decoder at each reconstruction scale. Hence, it is named as space depth correlation (SDC). The SDC does not depend on the patches in the feature map. Instead it processes the overall feature map both spatially and depth-wise. This helps to extract more information which will be correlated spatially as well as depth-wise (*the effectiveness of SDC is examined in Section 4.3.3*).

The outcome of each level in pseudo decoder is then passed through the SDC to weigh the pseudo decoder responses according to correlated information. To consider the maximum spatial and depth-wise correlation between the features, the merged features at each of the decoder levels are processed through the SDC before sending to the respective reconstruction level. **The SDC helps the architecture to maintain the spatial and depth (channel-wise) correlation introducing a spatial and depth-wise contextual attention at each level of pseudo decoder.** The space depth correlation block can be represented as given in Eq. (4.12) (*See SDC block in Figure 4.16* ).

$$Out_{SDC} = \{\sigma(\varphi_1(D_{max}))\} \odot \{\xi_{in} \odot \sigma(FC_3(S_{max}))\} \tag{4.12}$$

where, the $FC_l$ are $l$ fully connected layers, $\varphi_1$ is convolution layer, $\sigma$ is the sigmoid activation function. The $D_{max}$ and $S_{max}$ are the depth-wise and spatial maximum feature map from input feature map $\xi$ which are as given in Eq. (4.13) and (4.14) respectively.

$$D_{max} = \underset{C}{max}(\xi_{M,N,C}) \tag{4.13}$$

$$S_{max} = \underset{M,N}{max}(\xi_{M,N,C}) \tag{4.14}$$

From Eq. (4.13) and (4.14), it is clear that the $D_{max}$ has $M \times N \times 1$ and $S_{max}$ has $1 \times 1 \times C$ dimensions where $M \times N$ is spatial dimension and $C$ is number of feature maps *i.e.,* channels. The fully connected $FC_l$ and the convolution layers $\varphi_1$ are used on the $S_{max}$ and $D_{max}$ respectively to correlate the spatial and depth-wise features efficiently. The dimensions of fully connected layer and convolution layers applied on $S_{max}$ and $D_{max}$ respectively can be seen in detail from Figure 4.15. The responses after $FC_l$ and $\varphi_1$ are passed through a Softmax layer, to provide the sufficient weight to most relevant feature maps, which are then multiplied with input. **These weighted feature maps from each layer of pseudo decoder are then merged with respective layer of regeneration decoder to provide correlated contextual information.** Thus, the proposed pseudo decoder feature sharing with SDC helps the inpainting architecture:

- To merge the relevant weighted contextual information at each regeneration decoder level for efficient regeneration of inpainted image.

- To provide the spatial and depth-wise contextual attention.

Table 4.10: Quantitative comparison of the proposed method (Ours) with state-of-the-art methods for image inpainting on CelebA-HQ dataset [36] corrupted with NVIDIA Masks from [6].

| Mask Ratio | Method | PSNR ↑ | SSIM ↑ | $L_1$ ↓ | FID ↓ |
|---|---|---|---|---|---|
| 0.1-0.2 | PIC [11] | 28.39 | 0.953 | 2.054 | 5.420 |
| | GConv [12] | 27.56 | 0.947 | 2.216 | 5.563 |
| | EC [13] | 29.20 | 0.962 | 1.952 | 4.638 |
| | RFR [14] | 29.38 | 0.964 | 1.900 | 3.894 |
| | CTSDG [16] | 32.11 | 0.971 | 0.859 | 3.326 |
| | **Ours** | **33.70** | **0.980** | **0.780** | **3.066** |
| 0.3-0.4 | PIC [11] | 22.99 | 0.854 | 3.893 | 25.971 |
| | GConv [12] | 23.59 | 0.883 | 3.696 | 12.429 |
| | EC [13] | 24.97 | 0.904 | 3.167 | 12.084 |
| | RFR [14] | 25.06 | 0.901 | 3.116 | 17.056 |
| | CTSDG [16] | 26.71 | 0.929 | 2.970 | 11.299 |
| | **Ours** | **27.78** | **0.935** | **2.105** | **10.582** |
| 0.5-0.6 | PIC [11] | 17.21 | 0.587 | 8.758 | 44.555 |
| | GConv [12] | 18.14 | 0.775 | 5.968 | 34.980 |
| | EC [13] | 18.09 | 0.751 | 6.174 | 30.277 |
| | RFR[14] | 20.85 | 0.782 | 7.342 | 31.571 |
| | CTSDG [16] | 21.52 | 0.825 | 5.692 | 27.869 |
| | **Ours** | **22.71** | **0.862** | **5.301** | **22.504** |

### Regeneration Decoder

For the reconstruction of the inpainted image, we have proposed a regeneration decoder architecture (*see regeneration decoder in Figure 4.15* ). A residual block is used before every decoder layer to avoid the vanishing gradients problem. Consider the convolution with $m \times m$ filter size, stride $= 1$ and $n$ number of filters is defined as $Conv^{m,n}$, then the residual block is defined as: $Conv_1^{3,32} \rightarrow Conv_2^{3,32} \rightarrow Conv_3^{3,32} \rightarrow \rho\left(\left\langle Conv_1^{3,32}, Conv_2^{3,32}, Conv_3^{3,32}\right\rangle \rightarrow Conv^{3,32}\right)$, where, $\langle\cdot\rangle$ and $\rho$ are concatenation and ReLU activation function respectively (*See RB block in Figure 4.16* ). After passing the input from the two consecutive residual blocks, at each of the decoder stage, the weighted features from pseudo decoder levels followed by SDC blocks ($S_1$ to $S_4$) (Section 4.3.1) are concatenated with respective residual block output, as shown in Figure 4.15. This concatenated output is then processed in the feature merge block (FM). **The FM is used to merge the valid features from the encoder with the respective decoder layer features availing the hole locations from mask at each of the levels.** This merging helps the decoder architecture to extract the features related to the hole region processed from the the pseudo decoder (Eq. (4.12)) and valid region from the encoder efficiently. The output of the FM block at each decoder layer can be represented as:

$$FM_l^{Out} = E_{N-l} \odot (1 - M_{N-l}) + D_l \odot M_{N-l} \tag{4.15}$$

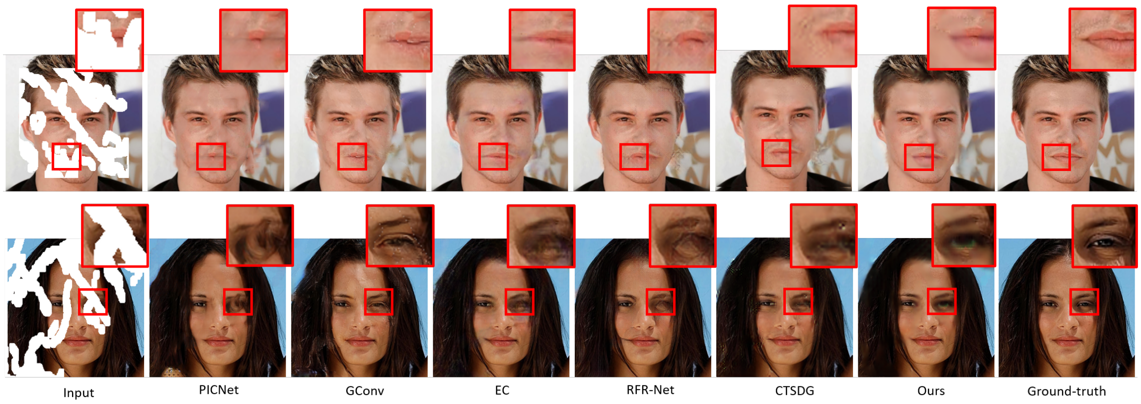


Figure 4.17: Qualitative comparison of the proposed method (Ours) with state-of-the-art methods (PICNet [11], GConv [12], EC[13], RFR-Net[14], CTSDG [16]) on CelebA_HQ dataset for image inpainting using publicly available masks from [18].

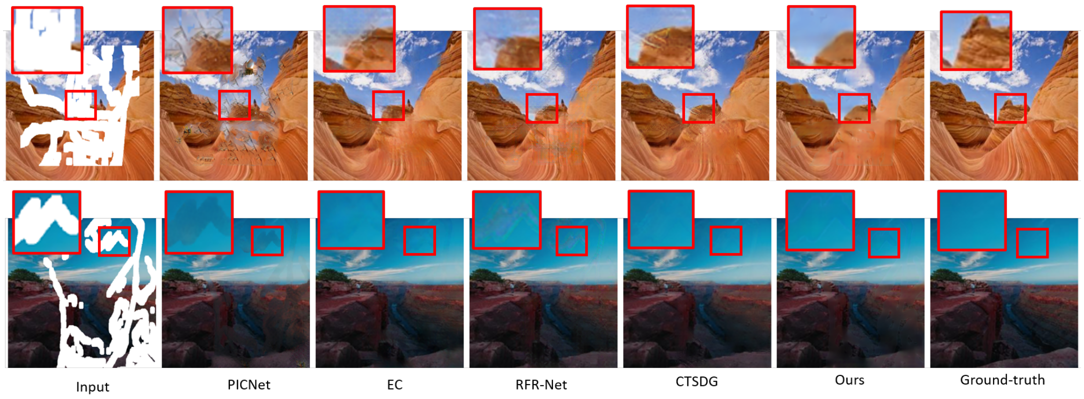


Figure 4.18: Qualitative comparison of the proposed method (Ours) with state-of-the-art methods (PICNet [11], EC[13], RFR-Net[14], CTSDG [16]) on Places2 dataset for image inpainting using publicly available masks from [18].

where, $E_l$, $D_l$ are $l_{th}$ level encoder and decoder feature maps respectively, $l\epsilon[1,4]$, $N = 5$ and $M_l$ is the down-sampled mask (by 8, 4, 2, 1) with respect to each decoder level (l = 1, 2, 3, 4 respectively). In regeneration decoder, four deconvolution layers are used to get the inpainted output image. **The proposed encoder multi level feature merging, pseudo decoder weighted feature sharing, and reconstruction decoder account for efficient image inpainting architecture.**

### 4.3.2 Training of the Proposed Network

The proposed architecture is trained end-to-end with adversarial learning for image inpainting task with the corrupted image as input. The discriminator architecture is same as that of [133].

#### Multi-Scale Loss

Existing approaches for image inpainting [157, 158] train the network with loss at different scales. In [157], the reconstruction loss (as combination of $\mathbb{L}_1$ and perceptual loss *i.e.,*

Table 4.11: Quantitative comparison of the proposed method (Ours) with state-of-the-art methods for image inpainting on Places2 dataset [3] corrupted with NVIDIA Masks from [6].

| Mask Ratio | Method | PSNR ↑ | SSIM ↑ | $L_1$ ↓ | FID ↓ |
|---|---|---|---|---|---|
| 0.1-0.2 | PIC [11] | 26.32 | 0.937 | 1.096 | 9.588 |
| | EC [13] | 27.28 | 0.943 | 1.060 | 6.094 |
| | RFR [14] | 28.28 | 0.954 | 1.033 | 5.149 |
| | CTSDG [16] | 29.69 | 0.957 | 0.924 | 5.129 |
| | **Ours** | **30.26** | **0.961** | **0.850** | **4.690** |
| 0.3-0.4 | PIC [11] | 20.77 | 0.771 | 3.447 | 34.240 |
| | EC [13] | 22.27 | 0.879 | 2.506 | 18.935 |
| | RFR [14] | 23.28 | 0.875 | 2.534 | 15.540 |
| | CTSDG [16] | 23.50 | 0.876 | 2.503 | 16.879 |
| | **Ours** | **24.19** | **0.885** | **2.360** | **14.745** |
| 0.5-0.6 | PIC [11] | 16.04 | 0.564 | 8.326 | 68.730 |
| | EC [13] | 16.26 | 0.677 | 7.950 | 57.677 |
| | RFR[14] | 17.99 | 0.684 | 7.126 | 43.158 |
| | CTSDG [16] | 18.05 | 0.749 | 6.573 | 41.422 |
| | **Ours** | **19.08** | **0.764** | **6.350** | **39.305** |

$\mathbb{L}_1 + \mathbb{L}_P$) is calculated for all output scales and adversarial loss is calculated only at last output scale. This may not help at generating good structural information. Since, at high dimension of feature maps we have more structural information. Also, calculating structural and textural loss at different scales differently [158] may produce structural and textual discontinuity. So, we propose to train the network with a multi-scale loss by considering same losses (reconstruction and adversarial both) at each output scale. We consider reconstruction loss as a combination of $\mathbb{L}_1$, perceptual loss ($\mathbb{L}_P$), and edge loss ($\mathbb{L}_{edge}$). In this context, the output of every FM block is processed through a convolution block with *tanh* activation function (*See Figure 4.13* ) with filter size $3\times3$ and $output\, channels = 3$. Every scale output ($i.e.$, $Out_l$, $l\epsilon[1,4]$) is used to calculate the loss ($\mathbb{L}_1\, to\, \mathbb{L}_4$) with the ground-truth (of respective size) while training the network. Output at the last scale ($l = 4$) is considered as the outcome of the image inpainting architecture. *Effectiveness of multi-scale loss is verified in Section 4.3.3.*

The loss at a particular reconstruction scale is given as:

$$\begin{aligned}\mathbb{L}_{scale} = \lambda_{hole}\mathbb{L}_{hole} + \lambda_{valid}\mathbb{L}_{valid} + \\ \lambda_{Adv}\mathbb{L}_{Adv} + \lambda_{Edge}\mathbb{L}_{Edge} + \lambda_{Per}\mathbb{L}_{Per}\end{aligned} \tag{4.16}$$

where, $L_{scale}$ is the loss calculated at each reconstruction scale, $\lambda_{loss}$ are the weights assigned to the respective losses. The values of each of the weights are $\lambda_{hole} = 3$ , $\lambda_{valid} = 1$

Table 4.12: Quantitative comparison of the proposed method (Ours) with state-of-the-art methods for image inpainting on Paris Street View dataset corrupted with NVIDIA Masks from [6].

| Mask Ratio | Method | PSNR ↑ | SSIM ↑ | $L_1$ ↓ | FID ↓ |
|---|---|---|---|---|---|
| 0.1-0.2 | PIC [11] | 30.71 | 0.940 | 2.116 | 21.300 |
| | GConv [12] | 26.49 | 0.891 | 2.355 | 19.992 |
| | EC [13] | 30.87 | 0.944 | 2.796 | 14.800 |
| | RFR [14] | 31.64 | 0.946 | 1.105 | 11.620 |
| | CTSDG [16] | 32.50 | 0.949 | 0.978 | 9.190 |
| | **Ours** | **32.71** | **0.956** | **0.910** | **8.490** |
| 0.3-0.4 | PIC [11] | 24.86 | 0.840 | 4.346 | 61.277 |
| | GConv [12] | 22.15 | 0.757 | 4.808 | 93.584 |
| | EC [13] | 25.66 | 0.706 | 3.350 | 45.480 |
| | RFR [14] | 26.19 | 0.799 | 2.767 | 40.170 |
| | CTSDG [16] | 27.02 | 0.858 | 2.651 | 32.340 |
| | **Ours** | **27.38** | **0.879** | **2.440** | **30.621** |
| 0.5-0.6 | PIC [11] | 17.17 | 0.501 | 10.304 | 86.624 |
| | GConv [12] | 19.14 | 0.609 | 8.002 | 80.465 |
| | EC [13] | 21.25 | 0.722 | 6.852 | 72.167 |
| | RFR[14] | 21.46 | 0.741 | 6.199 | 68.613 |
| | CTSDG [16] | 22.11 | 0.748 | 5.895 | 64.440 |
| | **Ours** | **22.53** | **0.752** | **5.401** | **61.200** |

, $\lambda_{edge} = 1$ , $\lambda_P = 0.2$ , and $\lambda_{GAN} = 1$. The overall loss of all the scales is given as:

$$\mathbb{L}_{total} = \sum_{scale=1}^{4} \mathbb{L}_{scale} \tag{4.17}$$

The input is converted in the range $[0, 1]$ before giving as input to the network. The weight parameters of the network are updated on NVIDIA DGX station having Tesla V100 $1\times16$ GB GPU with the batch size of 1. The Adam optimizer [144] with the learning rate of $2 \times 10^{-4}$, $\beta_1 = 0.5$ and $\beta_2 = 0.99$.

### 4.3.3 Experimental Analysis

The proposed architecture is compared qualitatively and quantitatively with state-of-the-art image inpainting methods in terms of PSNR, SSIM, Mean $L_1$ error, FID [159], *etc.* The comparison of proposed method on masks from [18] is done with CTSDG [16], recurrent feature reasoning (RFR) [14], gated convolutions (GConv) [12], pluralistic image inpainting (PIC) [11], and EdgeConnect (EC) [13]. The comparison of proposed method on synthetic mask from [8] is done with PIC [11], multi-column image inpainting (GMCNN) [9], GConv [12], Shift-Net (SN) [10], and Hyper-realistic image inpainting (HR) [8]. The comparison of proposed method for high resolution images is done with PEN [21], Pconv [18] and ultra-high resolution (UHR) [22] methods in terms of PSNR, SSIM, Mean

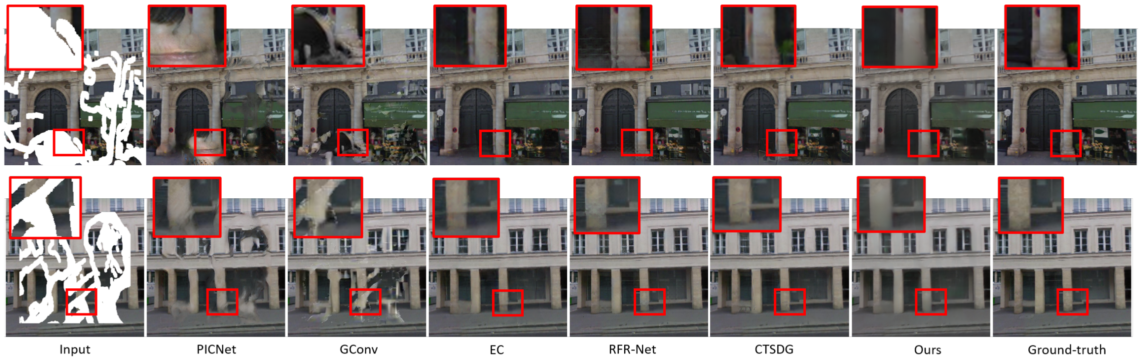


Figure 4.19: Qualitative comparison of the proposed method (Ours) with state-of-the-art methods (PICNet [11], GConv [12], EC[13], RFR-Net[14], CTSDG [16]) on Paris_SV dataset for image inpainting using publicly available masks from [18].

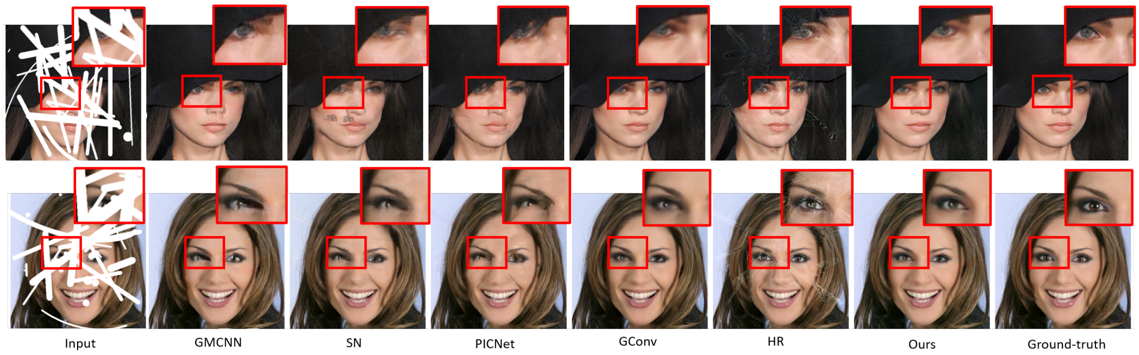


Figure 4.20: Qualitative comparison of the proposed method (Ours) with state-of-the-art methods (GMCNN [9], PICNet [11], SN [10], GConv [12], HR [8]) on CelebA_HQ dataset for image inpainting on synthetic masks.

$L_1$ error and FID. The quantitative and qualitative evaluation is done on images with $256 \times 256$ size. For quantitative evaluation on high-resolution, images with $512 \times 512$, $1k \times 1k$ and $2k \times 2k$ resolution are considered and random masks with mask ratio $\leq 0.25$ are used similar to [22].

### Results Analysis

*Experiment 1:* In this experiment, the image datasets corrupted by publicly available masks from [18] are considered for evaluation. Table 4.10, 4.11, 4.12 show the quantitative analysis of the proposed method with state-of-the-art methods for image inpainting on Places2, CelebA-HQ, and Paris_SV datasets in terms of PSNR, SSIM, Mean $L_1$ error and FID same as that of [14]. Table 4.10, 4.11, 4.12 show the results on different mask ratios same as that of [14] and average performance on given mask ratios. The proposed method gives superior performance as compared to existing state-of-the-art methods for given mask ratios. Also, the visual comparison of proposed method with state-of-the-art methods on the images corrupted using masks from [18] are shown in Figure 4.17, 4.19 and 4.18 for CelebA_HQ, Paris_SV and Places2 dataset respectively.

Table 4.13: Quantitative comparison of the proposed method (Ours) with state-of-the-art methods for image inpainting on CelebA-HQ dataset corrupted using synthetic masks

| Mask Ratio | Method | PIC | GMCNN | GConv | SN | HR | **Ours** |
|---|---|---|---|---|---|---|---|
| | Parameters | 3.63 M | 3.11 M | 4.050 M | 54.94 M | 30.25 M | **0.971 M** |
| | Publication | CVPR-19 | NIPS-18 | ICCV-19 | ECCV-18 | WACV-21 | TIP-22 |
| 0.1-0.2 | PSNR ↑ | 30.99 | 33.00 | 27.72 | 30.91 | 33.51 | **34.05** |
| | SSIM ↑ | 0.957 | 0.981 | 0.940 | 0.953 | 0.975 | **0.987** |
| | $L_1$ ↓ | 1.956 | 1.418 | 1.744 | 1.550 | 1.447 | **1.160** |
| | FID ↓ | 4.435 | 2.660 | 13.670 | 4.715 | 2.537 | **2.372** |
| 0.2-0.3 | PSNR ↑ | 28.41 | 30.54 | 25.48 | 29.12 | 30.44 | **31.73** |
| | SSIM ↑ | 0.929 | 0.969 | 0.910 | 0.933 | 0.967 | **0.979** |
| | $L_1$ ↓ | 1.848 | 1.518 | 2.606 | 1.969 | 1.625 | **1.500** |
| | FID ↓ | 8.153 | 4.616 | 22.020 | 7.281 | 7.068 | **3.881** |
| 0.3-0.4 | PSNR ↑ | 26.57 | 28.78 | 24.03 | 27.34 | 28.49 | **30.03** |
| | SSIM ↑ | 0.899 | 0.955 | 0.883 | 0.900 | 0.957 | **0.970** |
| | $L_1$ ↓ | 2.220 | 1.950 | 3.434 | 2.604 | 2.828 | **1.872** |
| | FID ↓ | 13.573 | 7.679 | 29.930 | 11.679 | 9.474 | **5.705** |
| 0.4-0.5 | PSNR ↑ | 24.64 | 26.97 | 22.59 | 24.62 | 27.30 | **28.31** |
| | SSIM ↑ | 0.845 | 0.934 | 0.849 | 0.819 | 0.942 | **0.955** |
| | $L_1$ ↓ | 3.126 | 3.223 | 4.448 | 4.094 | 2.945 | **2.375** |
| | FID ↓ | 24.955 | 13.280 | 36.166 | 28.346 | 13.109 | **8.293** |
| 0.5-0.6 | PSNR ↑ | 22.99 | 25.47 | 21.48 | 21.15 | 26.22 | **26.84** |
| | SSIM ↑ | 0.786 | 0.909 | 0.815 | 0.682 | 0.925 | **0.938** |
| | $L_1$ ↓ | 4.134 | 3.857 | 5.492 | 6.829 | 3.512 | **2.907** |
| | FID ↓ | 45.607 | 22.480 | 40.830 | 30.255 | 17.307 | **11.113** |

*Experiment 2:* Along with the comparison on publicly available mask dataset, we have compared the proposed method with state-of-the-art methods on synthetically generated masks from [8]. Table 4.14, 4.13 show the quantitative analysis of the proposed method with state-of-the-art methods for image inpainting on corrupted images using synthetic masks from [8] on CelebA-HQ and Places2 datasets in terms of PSNR, SSIM, Mean $L_1$ error and FID same as that of [8]. The Table 4.14, 4.13 prove the generalizability of proposed method on all the mask ratio sets for image inpainting. The visual comparison of proposed method on image datasets corrupted using synthetic mask are depicted in Figure 4.20 and 4.21 for CelebA-HQ and Places2 datasets respectively. *Specifically, the proposed method gives significant performance overall on all the mask ratios even though it has less computational complexity (0.97M parameters) than existing methods in the literature (see Table 4.9).* This shows that the proposed architecture reconstructs the corrupted images from smaller to larger hole regions efficiently compared to state-of-the-art methods.

### Results for High Resolution Image inpainting

To verify the generalizability of the proposed method, along with the efficiency on standard resolution images, we have tested the proposed method with different resolution images.

Table 4.14: Quantitative comparison of the proposed method (Ours) with state-of-the-art methods for image inpainting on Places2 dataset corrupted using synthetic masks

| Mask Ratio | Method | PIC [11] | GMCNN [9] | GConv [12] | SN [10] | HR [8] | **Ours** |
|---|---|---|---|---|---|---|---|
| | Publication | CVPR-19 | NIPS-18 | ICCV-19 | ECCV-18 | WACV-21 | TIP-22 |
| 0.1-0.2 | PSNR ↑ | 29.31 | 30.90 | 24.68 | 28.83 | 30.60 | **32.88** |
| | SSIM ↑ | 0.936 | 0.957 | 0.846 | 0.932 | 0.945 | **0.984** |
| | $L_1$ ↓ | 1.231 | 1.002 | 2.293 | 2.183 | 1.557 | **0.692** |
| | FID ↓ | 6.398 | 5.100 | 9.517 | 7.476 | 5.264 | **4.544** |
| 0.2-0.3 | PSNR ↑ | 26.90 | 28.54 | 22.59 | 26.86 | 28.93 | **30.37** |
| | SSIM ↑ | 0.897 | 0.930 | 0.780 | 0.897 | 0.918 | **0.975** |
| | $L_1$ ↓ | 1.963 | 1.538 | 3.423 | 2.829 | 2.506 | **1.275** |
| | FID ↓ | 11.288 | 10.403 | 19.800 | 13.189 | 10.743 | **9.307** |
| 0.3-0.4 | PSNR ↑ | 25.15 | 26.89 | 21.33 | 25.09 | 27.93 | **29.50** |
| | SSIM ↑ | 0.851 | 0.901 | 0.724 | 0.855 | 0.900 | **0.923** |
| | $L_1$ ↓ | 2.752 | 2.100 | 4.445 | 3.653 | 3.311 | **1.148** |
| | FID ↓ | 17.963 | 16.807 | 28.608 | 23.208 | 17.705 | **15.660** |
| 0.4-0.5 | PSNR ↑ | 23.36 | 25.27 | 20.10 | 22.79 | 24.08 | **26.70** |
| | SSIM ↑ | 0.784 | 0.860 | 0.660 | 0.776 | 0.753 | **0.880** |
| | $L_1$ ↓ | 3.836 | 2.846 | 5.772 | 5.235 | 5.171 | **2.190** |
| | FID ↓ | 25.948 | 24.463 | 38.295 | 32.018 | 29.108 | **24.190** |
| 0.5-0.6 | PSNR ↑ | 21.84 | 24.02 | 19.42 | 20.40 | 22.48 | **25.75** |
| | SSIM ↑ | 0.707 | 0.819 | 0.608 | 0.668 | 0.725 | **0.853** |
| | $L_1$ ↓ | 4.977 | 3.562 | 6.697 | 7.772 | 3.866 | **2.252** |
| | FID ↓ | 42.647 | 29.933 | 62.610 | 65.266 | 34.451 | **27.684** |

Table 4.15: Quantitative comparison of proposed method (Ours) with state-of-the-art methods on Places2 dataset on high-resolution images (*Note - The NVIDIA masks [6] with mask ratio* $\leq 0.25$ *are similar to[22]*)

| Image Size | Method | PSNR ↑ | SSIM ↑ | $L_1$ ↓ | FID ↓ |
|---|---|---|---|---|---|
| $512 \times 512$ | PEN [21] | 32.86 | 0.912 | 0.0079 | 8.206 |
| | Pconv [18] | 33.12 | 0.935 | 0.0064 | 7.942 |
| | UHR [22] | 34.98 | 0.962 | 0.0059 | 7.001 |
| | **Ours** | **35.05** | **0.968** | **0.0054** | **6.616** |
| $1K \times 1K$ | Pconv [18] | 30.38 | 0.906 | 0.0105 | 9.542 |
| | UHR [22] | 31.82 | 0.941 | 0.0081 | 8.762 |
| | **Ours** | **32.51** | **0.950** | **0.0076** | **8.001** |
| $2K \times 2K$ | Pconv [18] | 27.24 | 0.895 | 0.0215 | 28.164 |
| | UHR [22] | 28.92 | 0.924 | 0.0166 | 27.289 |
| | **Ours** | **29.88** | **0.946** | **0.0112** | **27.046** |

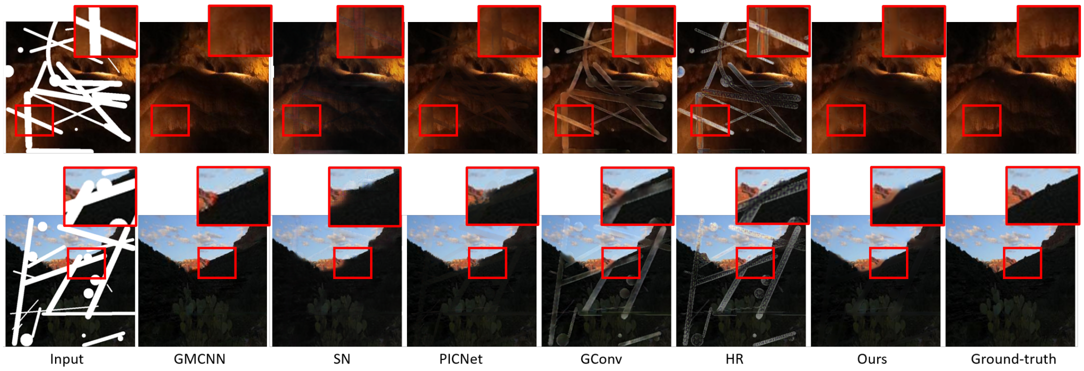


Figure 4.21: Qualitative comparison of the proposed method (Ours) with state-of-the-art methods (GMCNN [9], SN [10], PICNet [11], GConv [12], HR [8]) on Places2 dataset for image inpainting on synthetic masks.

The proposed method is tested on images with 512 × 512, 1024 × 1024, and 2048 × 2048 resolution. *The proposed network has very less number of parameters (0.97M) as compared to the existing method for high-resolution image inpainting (2.7M) [22].* We have compared different methods (similar to [22]) for high-resolution images. The methods with out of memory problem are not considered for this analysis. The PEN [21] method gives out of memory problem when tested for images with $1k$ or higher resolution images. The quantitative and qualitative comparison of the proposed method with state-of-the-art methods is provided in Table 4.15 and Figure 4.22 respectively. The comparison shows the effectiveness of the proposed method when tested on the high resolution images for image inpainting. Our proposed method generates more plausible results as compared with the existing methods. This proves that, with very less computational complexity (0.97M parameters), our proposed method performs well when tested on high-resolution images.

### Results for Object Removal

Along with inpainting the corrupted images, the image inpainting methods can be used for object removal task. Here, we verify the applicability of proposed method for object removal task where the mask of object is explicitly provided. The qualitative comparison of proposed method and existing-state-of-the-art methods is carried out to verify the efficiency for real time application of object removal. Figure 4.23 shows the visual comparison for object removal task. It can be seen from Figure 4.23, the methods PIC, GConv, SN, EC and CTSDG are unable to reproduce the significant content at hole region and the results of GMCNN, RFR and HR have inconsistencies at the boundary of hole and valid region. Whereas, the proposed method generates significant content along-with consistency at the boundary of hole and valid regions. This proves that, the proposed method gives fruitful results for object removal task as compared to existing-state-of-the-art methods for image inpainting. We give this credit to the proposed

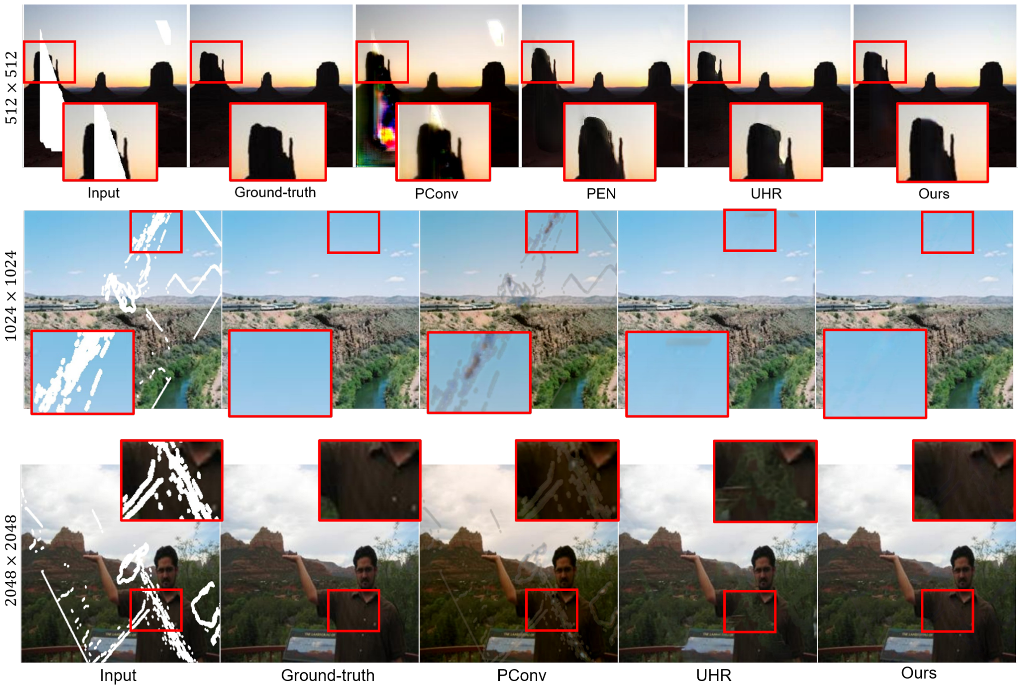


Figure 4.22: Qualitative result comparison of the proposed method (Ours) with existing methods (PConv [18], PEN [21], UHR [22]) on high resolution images

multi level feature fusion through VRF and pseudo decoder which provides correlated feature maps to reconstruction decoder for generating awesome-sauce inpainting results.

### Efficiency and Complexity

To examine the model efficiency, we have compared the proposed model with state-of-the-art methods in terms of the number of trainable parameters and FLOPs. Table 4.16 gives the comparison of the proposed method in terms of the number of parameters of the network. The comparison is done on Paris_SV dataset with 0.4 - 0.5 mask ratio in terms of PSNR and SSIM. Table 4.16 shows that the proposed method, with a very less number of parameters, outperforms the state-of-the-art methods with significant improvement for image inpainting. Also, the parameters of proposed and existing state-of-the-art methods with average PSNR are compared for CelebA_HQ dataset on synthetic mask from [8] and publicly available mask from [18]. Table 4.9 shows that, even-though the proposed network has very less number of parameters and FLOPs, it gives superior performance as compared to existing state-of-the-art methods.

Table 4.16: The hyper-parameter analysis on Paris_SV dataset for 0.4 - 0.5 mask ratio

| Method | Coarse-to-fine [69] | Pconv[18] | RFR [14] | Ours |
|---|---|---|---|---|
| #Parameters ↓ | 100M+ | 33M | 31M | **0.971M** |
| FLOPs ($\times 10^9$)↓ | 417.23 | 18.95 | 206.11 | **15.20** |
| PSNR ↑ | 23.1 | 23.69 | 24.6 | **25.16** |
| SSIM ↑ | 0.768 | 0.759 | 0.796 | **0.829** |

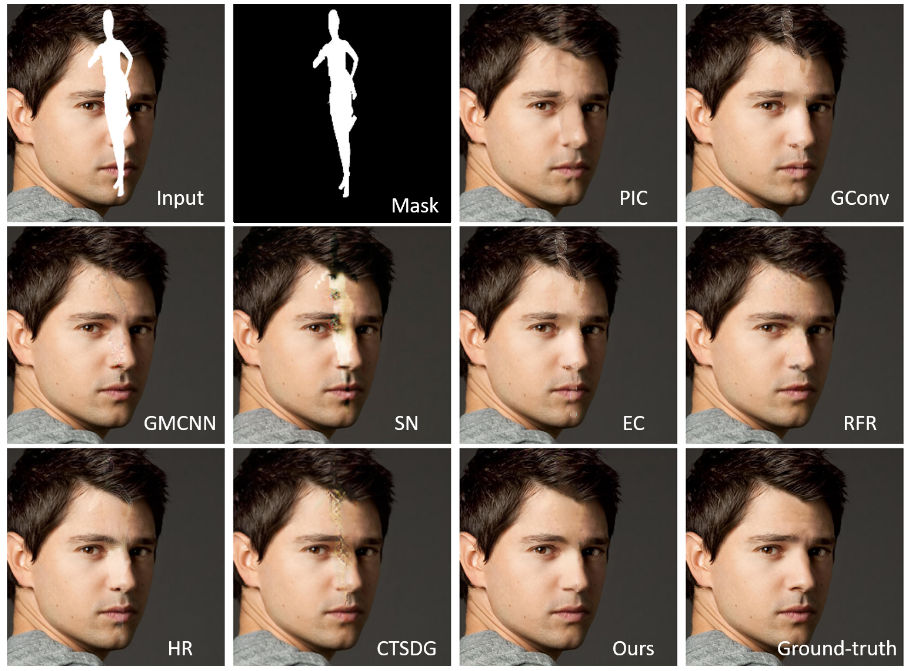


Figure 4.23: Qualitative comparison of the proposed method (Ours) with existing state-of-the-art methods (PIC [11], GConv [12], GMCNN [9], SN [10], EC[13], RFR[14], HR [8], CTSDG [16]) for object removal task.

### Ablation Study

To examine the effectiveness of proposed and used blocks in the architecture and losses used while training of the network, we have done the ablation study. For the ablation study, we have considered the Paris_SV dataset with publicly available masks [18]. The ablation study for the same is discussed in the following sections.

*Effect of multi-encoder feature fusion:* To verify effectiveness of the proposed encoder multi-level feature fusion (EMLFF), we carried out the experiment with (w/) and without (w/o) EMLFF. Here, w/o EMLFF means, direct skip connection is provided from each encoder level to respective decoder level without feature fusion. The feature fusion is designed in order to extract valid features from diverse receptive fields. Absence of EMLFF will provide encoder features to corresponding decoder level without considering the valid information. Also, hole content may be forwarded to decoder as it is and this will further deteriorate the performance for image inpainting task. The quantitative comparison of this experiment is included in Table 4.17. Also, visual comparison is provided in Figure 4.25. Without EMLFF, the network fails to reconstruct valid information at the hole locations this leads to degradation of quantitative and qualitative results with large margin. The numeric and visual comparison validates the effectiveness of the proposed encoder multi-level feature fusion.

*Effect of VRF and SDC block:*

The purpose behind utilizing the VRF and SDC block is to fill the holes of large region effectively and to provide weights to the feature maps from pseudo decoder levels

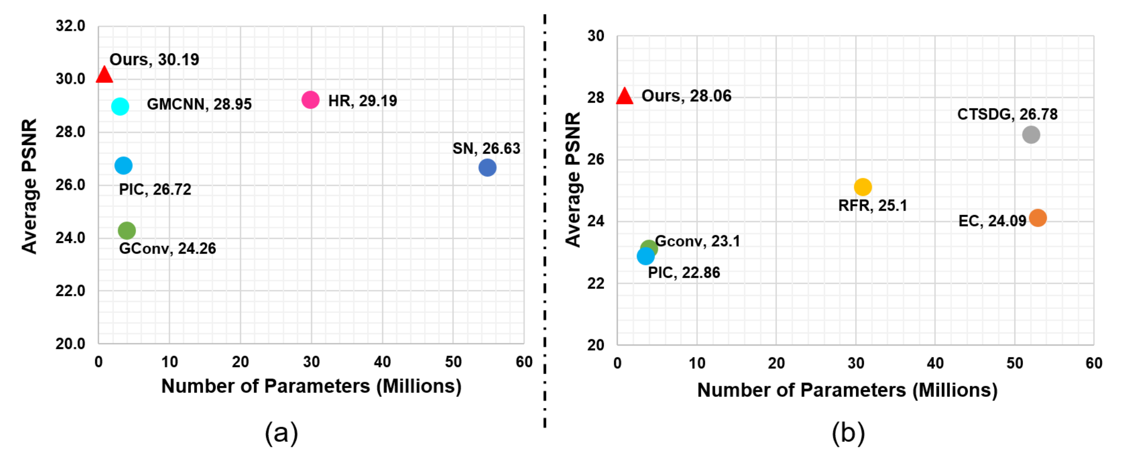


Figure 4.24: Computational complexity analysis of the proposed method (Ours) with state-of-the-art methods. (a) Comparison in terms of average PSNR and number of trainable parameters on CelebA-HQ dataset corrupted using synthetic mask from [8], (b) Comparison in terms of average PSNR and number of trainable parameters on CelebA_HQ dataset corrupted using NVIDIA mask from [6].

Table 4.17: Analysis on effect of the proposed encoder multi-level feature fusion (EMLFF) for image inpainting

| Metric | EMLFF | Mask Ratio | | |
|---|---|---|---|---|
| | | 0.1-0.2 | 0.3-0.4 | 0.5-0.6 |
| PSNR ↑ | W/o | 29.48 | 24.51 | 19.81 |
| | W/ | **32.71** | **27.38** | **22.53** |
| SSIM ↑ | W/o | 0.928 | 0.826 | 0.702 |
| | W/ | **0.956** | **0.879** | **0.752** |

respectively. To analyse the effect of the VRF and SDC block in the proposed method, four different experiments are performed (a) without (w/o) VRF and SDC, (b) w/o VRF and with (w/) SDC, (c) w/ VRF and w/o SDC (d) w/ VRF and SDC on Paris_SV dataset with different mask ratios in terms of PSNR and SSIM. From Table 4.18, it is clear that the use of both the blocks in the network gives best outcome for image inpainting on the images with large mask ratio. As proposed, these blocks helps the network to efficiently fill the hole regions of any size. Also, the qualitative comparison with these configurations is depicted in Figure 4.26. From Figure 4.26, it is clear that, use of both VRF and SDC gives effective results for image inpainting as compared to other configurations. The VRF helps to focus on larger receptive fields and the SDC helps to provide weighted pseudo decoder features to regeneration decoder path.

*Effect of multi-scale loss:*

To train the network, generally a loss is calculated at the final output. Here to train our proposed network, we used the multi-scale loss, where the total loss used for training is summation of losses calculated at each decoder scale (*see Eq. 4.16* ). The ablation study

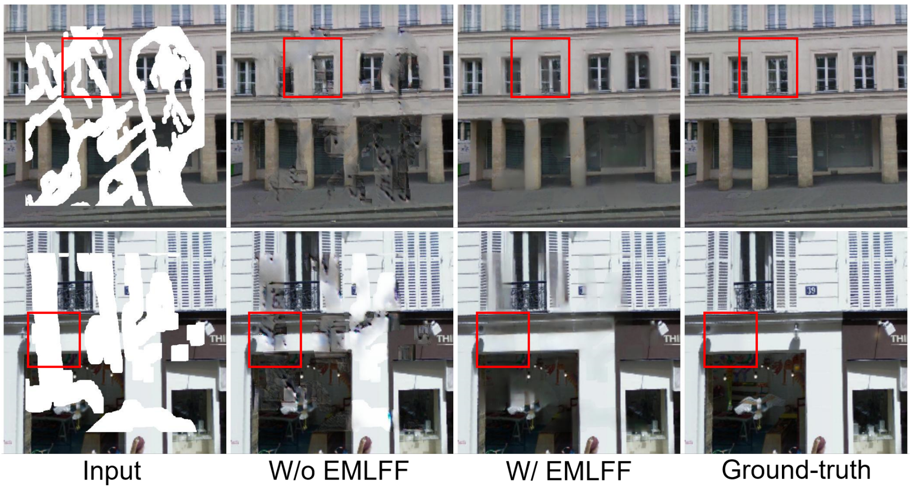


Figure 4.25: Qualitative results on the effect of the proposed encoder multi-level feature fusion (EMLFF) for image inpainting on Paris_SV dataset.

Table 4.18: Analysis on effect of VRF and SDC block on Paris_SV dataset

| Metric | VRF | SDC | Mask Ratio | | |
|---|---|---|---|---|---|
| | | | 0.1-0.2 | 0.3-0.4 | 0.5-0.6 |
| PSNR ↑ | *w/o* | *w/o* | 30.63 | 25.08 | 20.17 |
| | *w/o* | *w/* | 31.99 | 26.42 | 20.79 |
| | *w/* | *w/o* | 32.02 | 26.26 | 21.16 |
| | *w/* | *w/* | **32.71** | **27.38** | **22.53** |
| SSIM ↑ | *w/o* | *w/o* | 0.947 | 0.842 | 0.646 |
| | *w/o* | *w/* | 0.954 | 0.865 | 0.681 |
| | *w/* | *w/o* | 0.955 | 0.867 | 0.704 |
| | *w/* | *w/* | **0.956** | **0.879** | **0.752** |

Table 4.19: Analysis on effect of multi-scale loss for image inpainting

| Metric | Loss | Mask Ratio | | |
|---|---|---|---|---|
| | | 0.1-0.2 | 0.3-0.4 | 0.5-0.6 |
| PSNR ↑ | Single-scale | 28.66 | 25.01 | 20.33 |
| | Multi-scale | **32.71** | **27.38** | **22.53** |
| SSIM ↑ | Single-scale | 0.882 | 0.811 | 0.649 |
| | Multi-scale | **0.956** | **0.879** | **0.752** |

is carried out verify the efficiency of the multi-scale loss while training the network. The quantitative and qualitative results of the same are given in Table 4.19 and Figure 4.27, respectively. The overall loss in Eq. (4.16) calculated at each scale guides the network at each reconstruction stage. This provides a supervision for effective reconstruction which further helps towards faithful results. From these results, we can easily conclude that, the multi-scale loss helps the network for producing better outcome as compared to single scale loss while training.

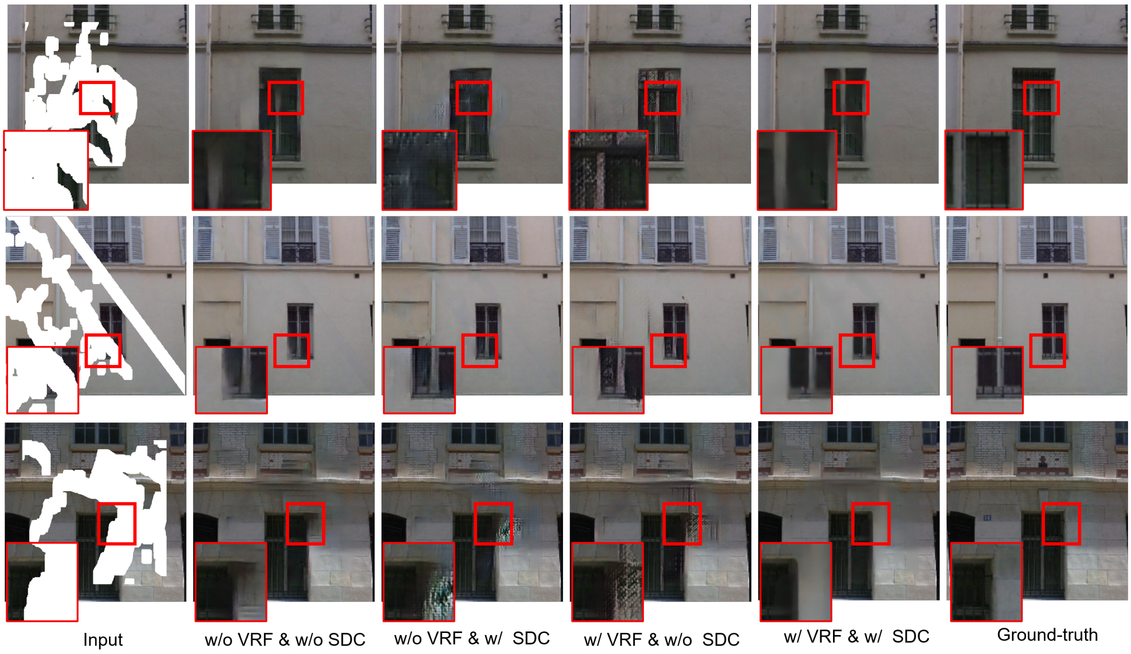


Figure 4.26: Qualitative results on the effect of VRF and SDC for image inpainting on Paris_SV dataset.

Table 4.20: Analysis on effect of loss functions using Paris_SV dataset (*Note:* $\mathbb{L}_1 = \mathbb{L}_{hole} + \mathbb{L}_{valid}$).

| Metric | Loss | Mask Ratio | | |
|---|---|---|---|---|
| | | 0.1-0.2 | 0.3-0.4 | 0.5-0.6 |
| PSNR ↑ | $\mathbb{L}_1 + \mathbb{L}_{GAN}$ | 30.66 | 25.01 | 20.33 |
| | $\mathbb{L}_1 + \mathbb{L}_{GAN} + \mathbb{L}_{Edge}$ | 31.74 | 26.06 | 21.72 |
| | $\mathbb{L}_1 + \mathbb{L}_{GAN} + \mathbb{L}_{Per}$ | 31.97 | 26.35 | 21.22 |
| | $\mathbb{L}_1 + \mathbb{L}_{GAN} + \mathbb{L}_{Edge} + \mathbb{L}_{Per}$ | **32.71** | **27.38** | **22.53** |
| SSIM ↑ | $\mathbb{L}_1 + \mathbb{L}_{GAN}$ | 0.912 | 0.841 | 0.739 |
| | $\mathbb{L}_1 + \mathbb{L}_{GAN} + \mathbb{L}_{Edge}$ | 0.947 | 0.851 | 0.749 |
| | $\mathbb{L}_1 + \mathbb{L}_{GAN} + \mathbb{L}_{Per}$ | 0.927 | 0.846 | 0.731 |
| | $\mathbb{L}_1 + \mathbb{L}_{GAN} + \mathbb{L}_{Edge} + \mathbb{L}_{Per}$ | **0.956** | **0.879** | **0.752** |

*Effect of losses:*

To analyse the effect of different losses on training of the network, an ablation study is done on Paris_SV dataset with different mask ratios. The network is trained on Paris_SV dataset with different configurations of losses as given in Table 4.20. From Table 4.20, it is clear that the effect of all the considered losses while training helps the network to learn effectively for image inpainting task. Also in Figure 4.28, we can clearly see that, with the *edge loss ($\mathbb{L}_{edge}$)* network tries to enhance the edges generated by the network. With addition of *perceptual loss ($\mathbb{L}_p$)*, the network improves the overall scene information. Whereas combination of all the losses ($\mathbb{L}_1$+$\mathbb{L}_{GAN}$+$\mathbb{L}_{edge}$+$\mathbb{L}_P$) gives perceptually enhanced outcome.

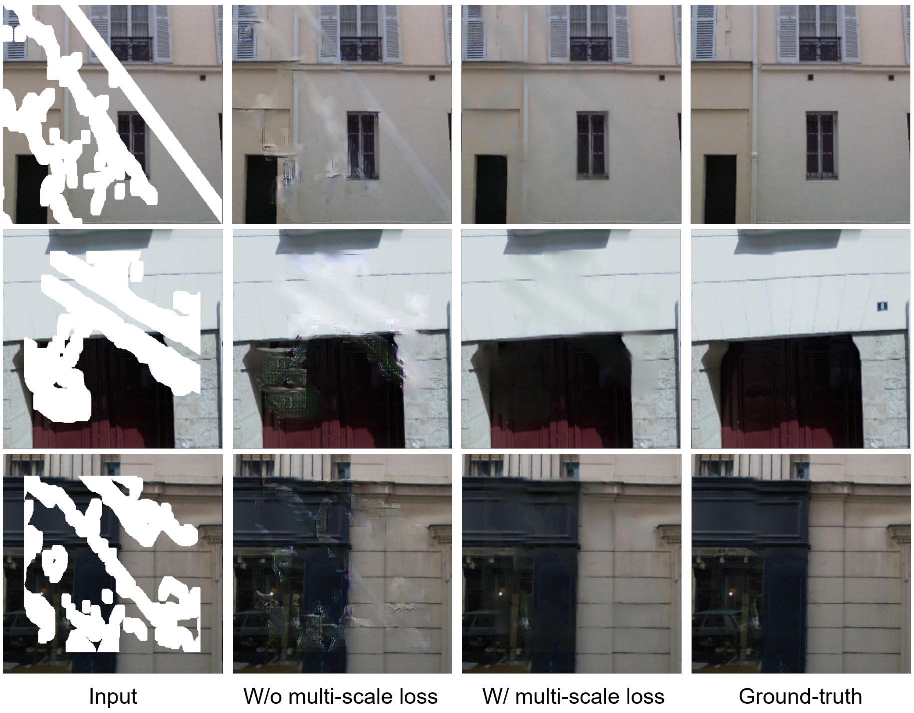


Figure 4.27: Qualitative results on the effect of multi scale loss for image inpainting on Paris_SV dataset.

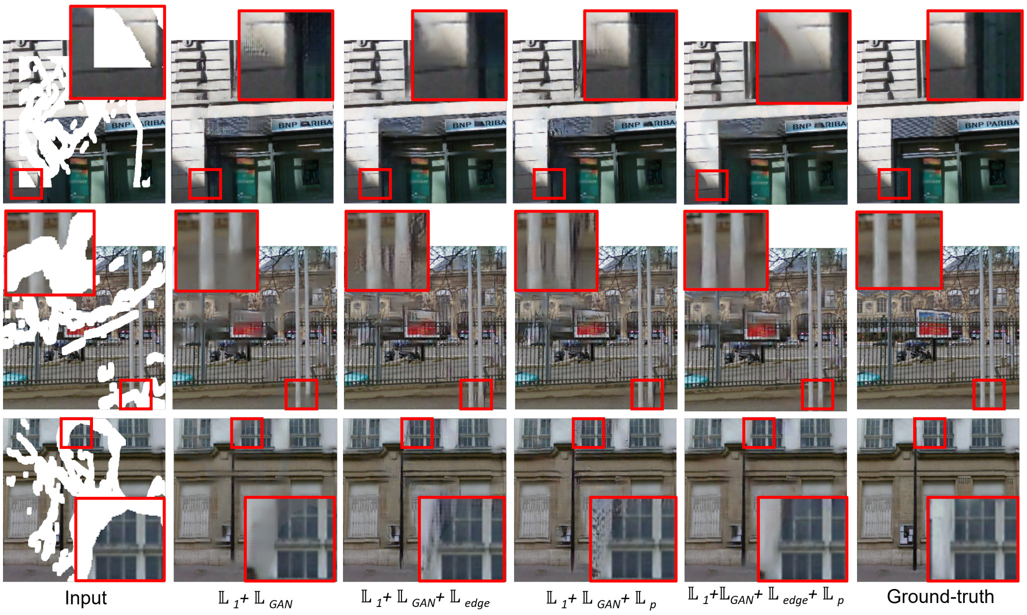

Input
$\mathbb{L}_1 + \mathbb{L}_{GAN}$
$\mathbb{L}_1 + \mathbb{L}_{GAN} + \mathbb{L}_{edge}$
$\mathbb{L}_1 + \mathbb{L}_{GAN} + \mathbb{L}_p$
$\mathbb{L}_1 + \mathbb{L}_{GAN} + \mathbb{L}_{edge} + \mathbb{L}_p$
Ground-truth

Figure 4.28: Qualitative results on the effect loss functions for image inpainting on Paris_SV dataset.

## 4.4 Summary of Proposed Contribution

In this chapter, we proposed three different solutions with single-stage architectures for image inpainting. In first solution (Section 4.1), multi-resolution inputs are utilized for image inpainting. A multi-kernel non-local attention is proposed to merge all the resolution inputs. Feature projection and valid feature fusion blocks are proposed for effective inpainting. The quantitative and qualitative comparison is done on two benchmark datasets corrupted using NVIDIA masks [6].

In second solution (Section 4.2), we proposed the concurrent processing of multi-level encoder features along with the diverse receptive fields module. The comparison is carried out on two benchmark datasets corrupted using NVIDIA masks [6].

The third solution (Section 4.3) proposes a light-weight architecture for image inpainting. Also, a pseudo-decoder in proposed to project the spatially and depth-wise correlated different encoder level features for actual reconstruction of inpainted image. The quantitative and qualitative comparison is carried out on three datasets corrupted using NVIDIA masks [6] and on two datasets corrupted using synthetic masks [8]. Further, the proposed solution is compared quantitatively and quantitatively for high-resolution image inpainting.

The computational complexity analysis and performance in terms of PSNR of existing methods and proposed solutions is given in Figure 4.29.

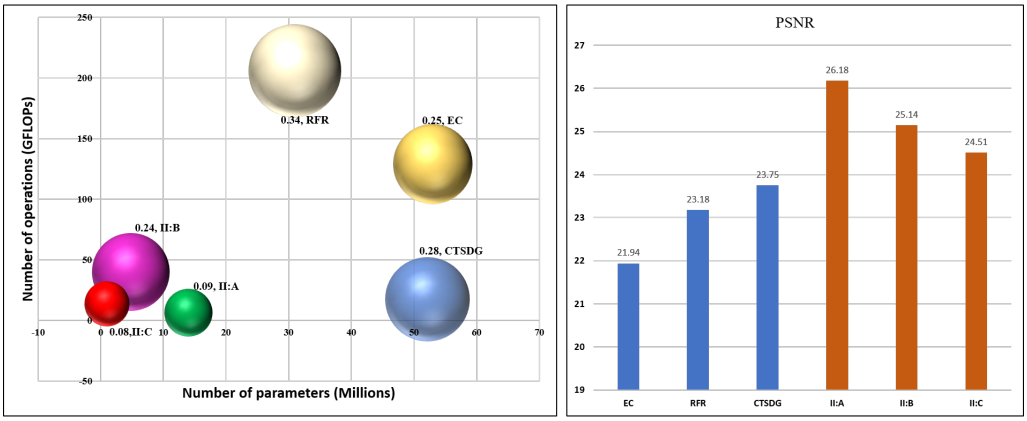


Figure 4.29: Comparison of the proposed methods (II:A-Section 4.1, II:B-Section 4.2, II:C-4.3) with existing methods. Left: in terms of the number of trainable parameters (x-axis), number of operations (GMAC) (y-axis), and run-time complexity in seconds per image (bubble size), Right: in terms of average PSNR on CelebA-HQ dataset.

# Chapter 5

# Blind Image Inpainting

Typically, image inpainting methods require information about the corrupted regions in the form of masks to guide the restoration process. These methods are known as non-blind image inpainting methods. However, in many real-world applications, such as photo editing, unwanted object removal, mesh-face verification, etc., it is often difficult to obtain masks for guidance. This has led to the development of a new technique called blind image inpainting, which does not require any prior knowledge about the corrupted regions or masks to perform image restoration.

The existing approaches for blind image inpainting [131, 23] try to predict the corrupted regions first and then inpaint the image using predicted mask and input corrupted image. Also, in [24], the authors try to attentively predict the mask to inpaint the corrupted image. Though both the stage-wise mask prediction followed by inpainting [131, 23] and intermediate attentive mask guidance inpainting [24] seem to be different, they follow almost similar approach to make the inpainting task directly or indirectly dependent on the identification of inconsistent regions. Also, when image with large corrupted regions is provided as input to these methods [131, 23, 24], they fail at inpainting the globally consistent image. In this work, we propose an end-to-end training approach independent of any identification of masked and non masked regions for blind image inpainting. In this regards, we propose two contributions for blind image inpainting as:

- Blind Image Inpainting via Omni-dimensional Gated Attention and Wavelet Queries
- Blind Image Inpainting via High-frequency Attentive Deformable Convolution

These solutions are explained in next sections.

## 5.1 Blind Image Inpainting via Omni-dimensional Gated Attention and Wavelet Queries

The transformers are well known for their ability to exploit long-range dependencies. With this ability, transformers have shown better convergence in numerous applications of image restoration [17, 88, 123, 131, 32, 34] including image inpainting. Further, the queries are the inputs to the transformer multi-head attention for which the attention is calculated. Providing appropriate queries to the transformer block may further enhance their convergence capability. With this assumption, in this work, we propose a wavelet query-based multi-head attention mechanism in the transformer block. The processed wavelet coefficients will provide less degraded information as a query to the multi-head attention mechanism. Also, to forward the encoder features to the respective decoder, we propose a gated omni-dimensional attention block. This block provides the all dimensional attentive information to the features which may help the network for efficient reconstruction. The contributions of our work are:

- An end-to-end transformer based architecture is proposed for blind image inpainting.
- A novel wavelet query multi-head attention mechanism is introduced in the transformer block.
- A omni-dimensional gated attention mechanism is proposed to forward different dimensional attentive features from encoder to respective decoder for effective reconstruction of inpainted image.

Our proposed approach achieves remarkable performance improvement as compared to existing state-of-the-art blind image inpainting methods.

### 5.1.1 Proposed Method

In this work, we propose a single-stage end-to-end transformer architecture for blind image inpainting (*see Figure 5.1*). Here, we propose two major components namely: (a) wavelet query multi-head attention mechanism in transformer: *to provide processed query as input to the multi-head attention*, and (b) omnidimensional gated attention: for *providing all dimensional attentive features* in order to achieve a plausible outcome. In this section, we will first give a detailed exposition of the proposed architecture for blind image inpainting and then we detail the proposed modules.

Overview of proposed transformer based architecture with the wavelet query multi-head attention (WQMA) and omnidimensional gated attention (OGA) is shown in Figure 5.1. To convert input image into feature space, we first apply the convolution layer. These convolved features are processed through three successive transformer blocks followed by down-sampler. The input with spatial size $m, n$ is then converted into $\frac{m}{8}, \frac{n}{8}$ sized feature maps at $4^{th}$ transformer block. In this transformer block, we propose *a wavelet query multi-head attention (WQMA) to provide processed features as a query to the multi-head attention.* These feature maps are then forwarded again to the successive transformer blocks but now these blocks are followed with an up-sampler to come up with the actual spatial dimension $(m, n)$ at the last stage. Here, in the decoding stage, we apply the proposed omni-dimensional gated attention (OGA) on encoder features while giving a

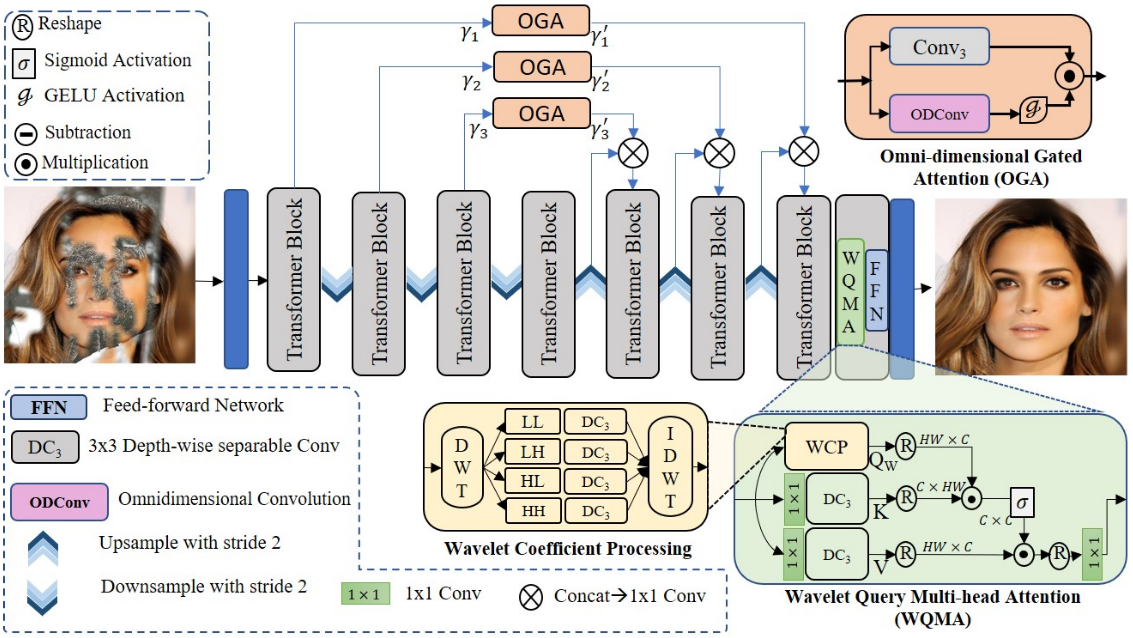


Figure 5.1: Overview of the proposed architecture for blind image inpainting. The architecture includes of transformer block consisting of the proposed wavelet query-based multi-head attention for providing prominent information as a query. Further, omni-dimensional gated attention is proposed in order to forward efficient attentive features from encoder to the respective decoder.

skip connection from the respective encoder to the subsequent decoder level. *The OGA helps the network to provide multidimensional attentive features to the decoder for effective reconstruction.* The structure of the transformer block consists of the proposed WQMA and a feed-forward network [32]. Finally, we again apply a convolution layer to generate final output $O$.

### Wavelet Query Multi-head Attention

In the existing transformer approach [32], generally the query, key, and values are considered from the same input without any separate processing to generate them. In a transformer block, the query is used to which attention is calculated and the key is from which the attention is calculated. So, here query plays an important role in overall multi-head attention for which attention is calculated. Providing effective features as a query may help the transformer block to further improve its performance. The contaminations in the inputs for a blind image inpainting task are considered as the noise appended on top of the clear image. Wavelets are well known for the task of image denoising where each of the decomposed wavelet coefficients is processed separately to reduce the noise. The wavelet-based attention mechanism is proposed in [160] for the task of image classification where the attention mechanism is applied in wavelet coefficient space. In the case of image inpainting, the input image has some corrupted regions present in it. Directly applying the attention in wavelet coefficient space may consider the corrupted regions also. Since the wavelet coefficient space also has corrupted regions in it. In order to avoid forwarding the noisy wavelet coefficients, we propose the processing of each wavelet coefficient. Further, the multi-head attention mechanism plays an essential role in capturing the long-term dependencies in the transformer block. The 2D wavelet coefficients are first calculated using forward discrete wavelet transform (DWT) as:

$$LL, LH, HL, HH = DWT(\mathbb{F}_{in}) \tag{5.1}$$

where, $LL$, $LH$, $HL$, and $HH$ are approximate, horizontal, vertical, and diagonal coefficients respectively of input feature maps $\mathbb{F}_{in}$ calculated using DWT. Each of the coefficients is separately processed as:

$$\begin{aligned} LL' = \psi_a(LL); LH' = \psi_h(LH) \\ HL' = \psi_v(HL); HH' = \psi_d(HH) \end{aligned} \tag{5.2}$$

where, $\psi$ is depth-wise separable convolution with kernel size $3 \times 3$. Further, these processed wavelet coefficients are utilized to form the output feature map by passing them through the inverse discrete wavelet transform (see Wavelet Coefficient Processing block in Figure 5.1). These processed wavelet coefficients are considered as the queries ($Q_W$) to the multi-head attention. This may help the network to calculate the attention with less effect of contaminations. The overall attention using wavelet queries is calculated as:

$$Attention(\mathbb{F}_{in}) = \sigma\left(\frac{Q_W K^T}{\sqrt{d}}\right) V \tag{5.3}$$

where, $K = C_1(\psi(\mathbb{F}_{in}))$, $V = C_1(\psi(\mathbb{F}_{in}))$, $C_1$ is convolution with kernel size $1 \times 1$. This proposed approach helps the network to effectively capture long-term dependencies with the minimum effect of corrupted regions.

### Omni-dimensional Gated Attention

In order to forward the encoder features to the respective decoder, we propose an omni-dimensional gated attention mechanism. This attention mechanism is given as:

$$\gamma_i' = C_3(\gamma_i) \odot \mathcal{G}(ODC_3(\gamma_i)) \tag{5.4}$$

where, $\gamma_i$ are the encoder features with $i \in (1, 2, 3)$, $C_3$ is convolution with kernel size $3 \times 3$, $\mathcal{G}$ is a GELU activation function, $ODC$ is omni-dimensional convolution with kernel size $3 \times 3$. This omni-dimensional gated attention provides the weighted feature from four different dimensions to the input encoder features.

The omni-dimensional convolution is a dynamic convolution that considers all the different dimensions of the input feature maps. Here, the omni-dimensional refers to the four different dimensions *i.e.* spatial, channel, filter, and kernel-wise attention. Let, for a dynamic convolution there are $n$ different convolutional kernels, each of the kernels has the spatial dimension $k \times k$, the number of input channels is $c_{in}$, and the number of output filters is $c_{out}$. Input ($\gamma_i$) to the ODC is first processed through a global average pooling operation followed by a fully connected layer and the ReLU activation function. These processed $1D$ features are used to generate different attentions like (i) spatial attention ($\alpha_s$) of size $k \times k$ to the spatial dimension of convolution kernel, (ii) channel attention ($\alpha_c$) of size $1 \times 1 \times c_{in}$ to the input channels $c_{in}$, (iii) filter attention ($\alpha_f$) of size $1 \times 1 \times c_{out}$ to the output number of filters $c_{out}$, and (iv) kernel attention ($\alpha_w$) to the $n$ dynamic convolution kernels. These attentions are calculated by applying a fully connected layer (to generate the required dimension) followed by the Sigmoid activation function. The output of ODC is formulated as:

$$Y = \left( \sum_{i=1}^{n} \alpha_{w_i} \odot \alpha_{f_i} \odot \alpha_{c_i} \odot \alpha_{s_i} \odot W_i \right) * \gamma_i \tag{5.5}$$

where, $\alpha_{w_i}$ is the attention applied to $i_{th}$ convolution kernel, $\alpha_f$ is the attention applied to the $c_{out}$ convolution filters, $\alpha_c$ is the attention applied to the $c_{in}$ convolution filters, and $\alpha_s$ is attention applied to spatial dimension $k \times k$ of convolution filter [161]. This ODA provides the network with the ability to learn attentive features from all the dimensions, unlike existing only spatial or channel-wise attentions.

### 5.1.2 Experiments and Results Discussion

In this section, we will discuss different experimental datasets, evaluation metrics, and quantitative and qualitative results of the proposed and existing state-of-the-art methods.

#### Datasets and Evaluation Metrics

For blind image inpainting, we use four datasets: FFHQ [2], CelebA-HQ [162], Places2 [3], and Paris Street View(ParisSV) [4]. The comparative analysis for blind image inpainting is done with VCNet [23], TransCNNHAE [24] (blind inpainting methods) and CTSDG [16] (non-blind inpainting method as provided in [24]). For fair comparison we have compared methods with publicly available source codes on all the blind/non-blind image inpainting datasets.

For quantitative results comparison of the proposed method and existing state-of-the-art methods on blind image inpainting, peak signal-to-noise ratio (PSNR), structural similarity index measure (SSIM), mean $L_1$ error and Fréchet inception distance (FID) metrics are used.

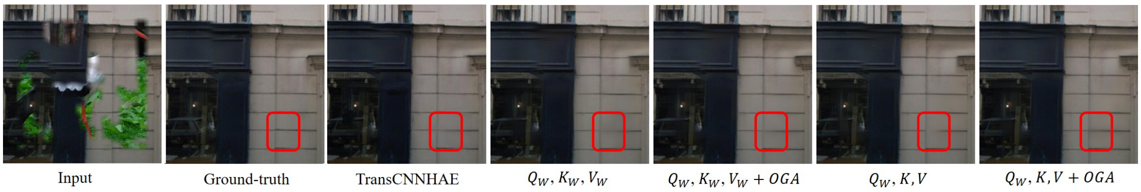


Figure 5.2: Qualitative result analysis of ablation study on different configurations of the proposed network for blind image inpainting.

Table 5.1: Ablation study on different configurations of the proposed network on ParisSV dataset for blind image inpainting. (Note: ↑- Higher is better, ↓- Lower is better).

| Network Configuration | PSNR↑ | SSIM↑ | $L_1$ ↓ | FID ↓ |
|---|---|---|---|---|
| TransCNNHAE [24] | 26.72 | 0.896 | 0.0352 | 41.50 |
| $Q_W$, $K_W$, $V_W$ | 26.89 | 0.885 | 0.0347 | 46.63 |
| $Q_W$, $K_W$, $V_W$ $+OGA$ | 27.50 | 0.901 | 0.0324 | 43.11 |
| $Q_W$, $K$, $V$ | 27.05 | 0.898 | 0.0328 | 44.32 |
| **$Q_W$, $K$, $V$ $+OGA$** | **27.81** | **0.905** | **0.0301** | **40.646** |

### Implementation and Training Details

To train the proposed blind image inpainting approach, we use AdamW optimizer with $3e^{-4}$ learning rate which is gradually reduced with the cosine annealing strategy. We train the proposed network using the $L_1$ loss. Also, to guide the network for textural and structural information by extracting effective features, the perceptual loss ($L_P$) is calculated between the deep feature maps of the ground-truth and inpainted images by passing them through the pre-trained VGG16 model [134] as:

$$L_P = \sum_{s=1}^{S} (\|\phi_s(G_t) - \phi_s(O)\|_1) \tag{5.6}$$

where, $G_t$ is ground-truth, $O$ is the output, $\phi_s$ are the feature maps ($s \in (1, S)$) of the VGG16 model. The edge loss ($L_e$) is also considered to focus on edge enhancement while training. The edge loss with sobel operator $\mathbb{S}$ is formulated as:

$$L_e = \|\mathbb{S}(G_t) - \mathbb{S}(O)\|_1 \tag{5.7}$$

For structurally consistent output generation we utilized the structural similarity loss ($L_S$), given as:

$$L_S = 1 - SSIM(O) \tag{5.8}$$

where, $SSIM$ is structural similarity index metric. So the overall loss to train the network is given as:

$$L_T = \lambda_1 L_1 + \lambda_P L_P + \lambda_e L_e + \lambda_S L_S \tag{5.9}$$

where, $\lambda_{Loss}$ is the weight assigned to respective $Loss$ function which is verified experimentally as: $\lambda_1 = 10$, $\lambda_P = 0.6$, $\lambda_e = 0.4$, $\lambda_S = 0.5$.

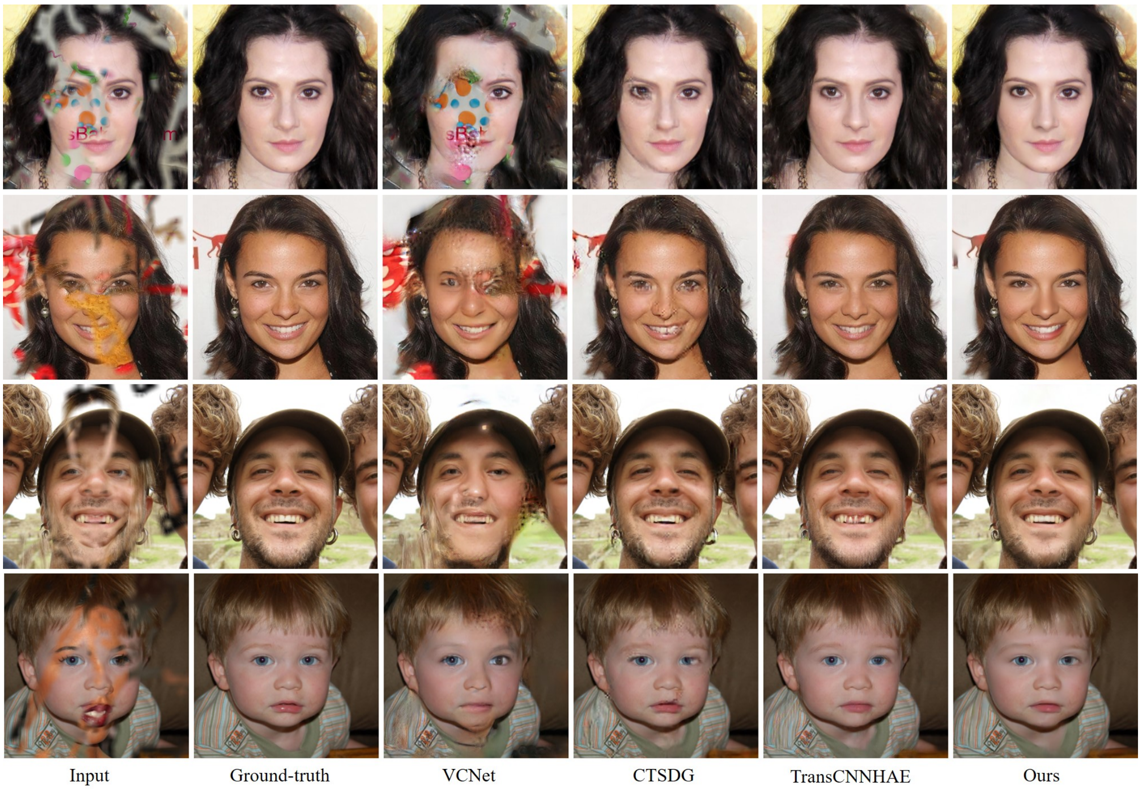


Figure 5.3: Qualitative results comparison of the proposed method (Ours) with existing state-of-the-art methods (VCNet [23], CTSDG [16], TransCNNHAE [24]) on Celeb (first two rows) and FFHQ (last two rows) dataset for blind image inpainting.

Table 5.2: Comparison of the proposed method (Ours) and existing state-of-the-art methods for blind image inpainting (↑- Higher is better, ↓- Lower is better).

| Metric | Dataset | VCNet [23] | CTSDG [16] | TransCNNHAE [24] | Ours |
|---|---|---|---|---|---|
| PSNR ↑ | CelebA-HQ | 25.59 | 26.94 | 27.71 | **28.21** |
| | FFHQ | 23.62 | 24.62 | 27.05 | **28.19** |
| | ParisSV | 23.62 | 26.08 | 26.72 | **27.81** |
| | Places2 | 24.09 | 26.05 | 26.87 | **27.55** |
| SSIM ↑ | CelebA-HQ | 0.874 | 0.934 | 0.949 | **0.951** |
| | FFHQ | 0.861 | 0.935 | 0.941 | **0.952** |
| | ParisSV | 0.824 | 0.861 | 0.896 | **0.905** |
| | Places2 | 0.869 | 0.905 | 0.910 | **0.918** |
| $L_1$ ↓ | CelebA-HQ | 0.0396 | 0.0318 | 0.0250 | **0.0221** |
| | FFHQ | 0.0482 | 0.0392 | 0.0281 | **0.0234** |
| | ParisSV | 0.0527 | 0.0412 | 0.0352 | **0.0301** |
| | Places2 | 0.0429 | 0.0308 | 0.0261 | **0.0231** |
| FID ↓ | CelebA-HQ | 9.275 | 8.561 | 7.251 | **7.235** |
| | FFHQ | 10.148 | 9.586 | 9.424 | **8.639** |
| | ParisSV | 64.215 | 43.015 | 41.505 | **40.646** |
| | Places2 | 28.821 | 18.685 | 17.640 | **17.521** |

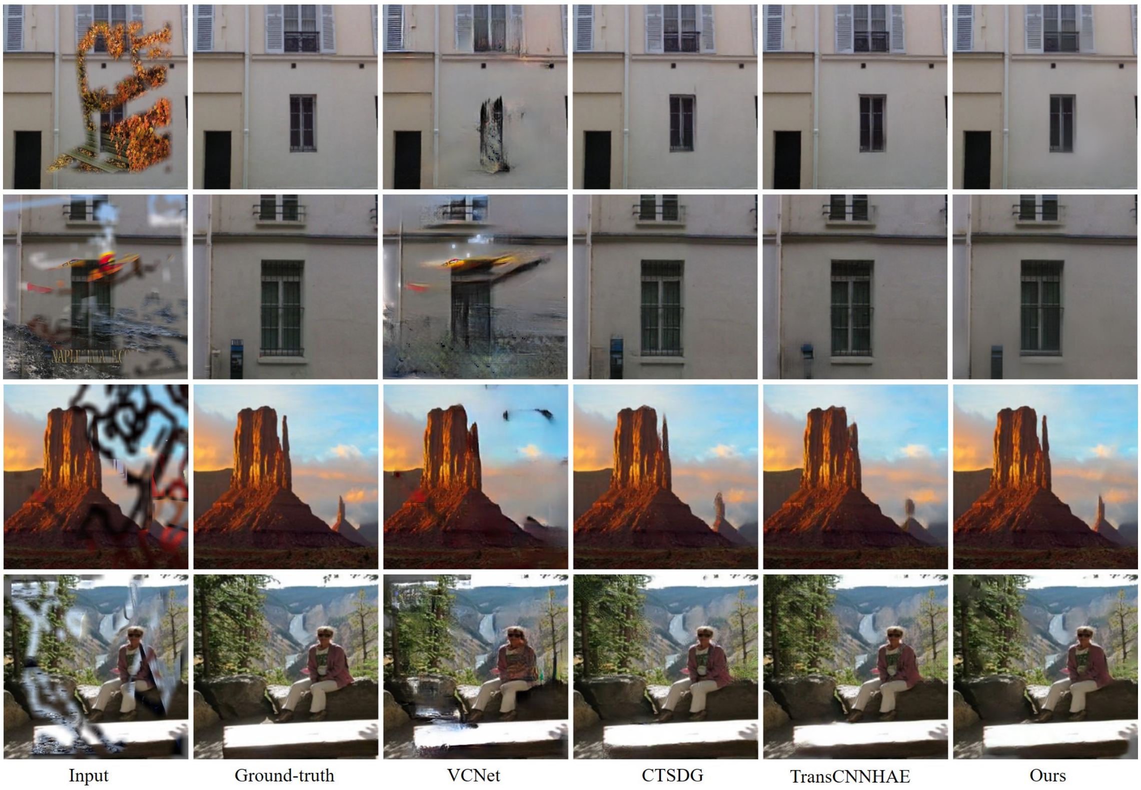


Figure 5.4: Qualitative results comparison of the proposed method (Ours) with existing state-of-the-art methods (VCNet [23], CTSDG [16], TransCNNHAE [24]) on Paris_SV (first two rows) and Places2 (last two rows) datasets for blind image inpainting.

### Ablation Study

To determine the design choices of the network for blind image inpainting, we performed various experiments on the Paris_SV dataset. How the each of proposed modules led to performance improvement is discussed in this section.

*Effect of the wavelet-based query to multi-head attention*

Wavelet base attention mechanism in transformer block has proved its efficiency for the image classification task [160]. With this motivation, at first, we aimed to provide wavelet query ($Q_W$), keys ($K_W$), and values ($K_W$) to the multi-head attention. For comparison purpose, we considered the existing best blind image inpainting method (TransCNNHAE [24]). The results improved in terms of PSNR, SSIM, and $L_1$ error. But there was no improvement in FID due to structural inconsistencies. Further, we evaluated the importance of providing wavelet processed query only to the multi-head attention with a combination of $Q_W, K, V$ which resulted in better convergence as compared to $Q_W, K_W, V_W$ (see row 2 and 4 of Table 5.1)

*Effect of omni-dimensional gated attention*

Further, to help the network for better reconstruction and structural information, we proposed omni-dimensional gated attention (OGA). The experiments are carried out with both the above discussed wavelet conditions *i.e.,* $Q_W, K_W, V_W$ + *OGA* and $Q_W, K, V$

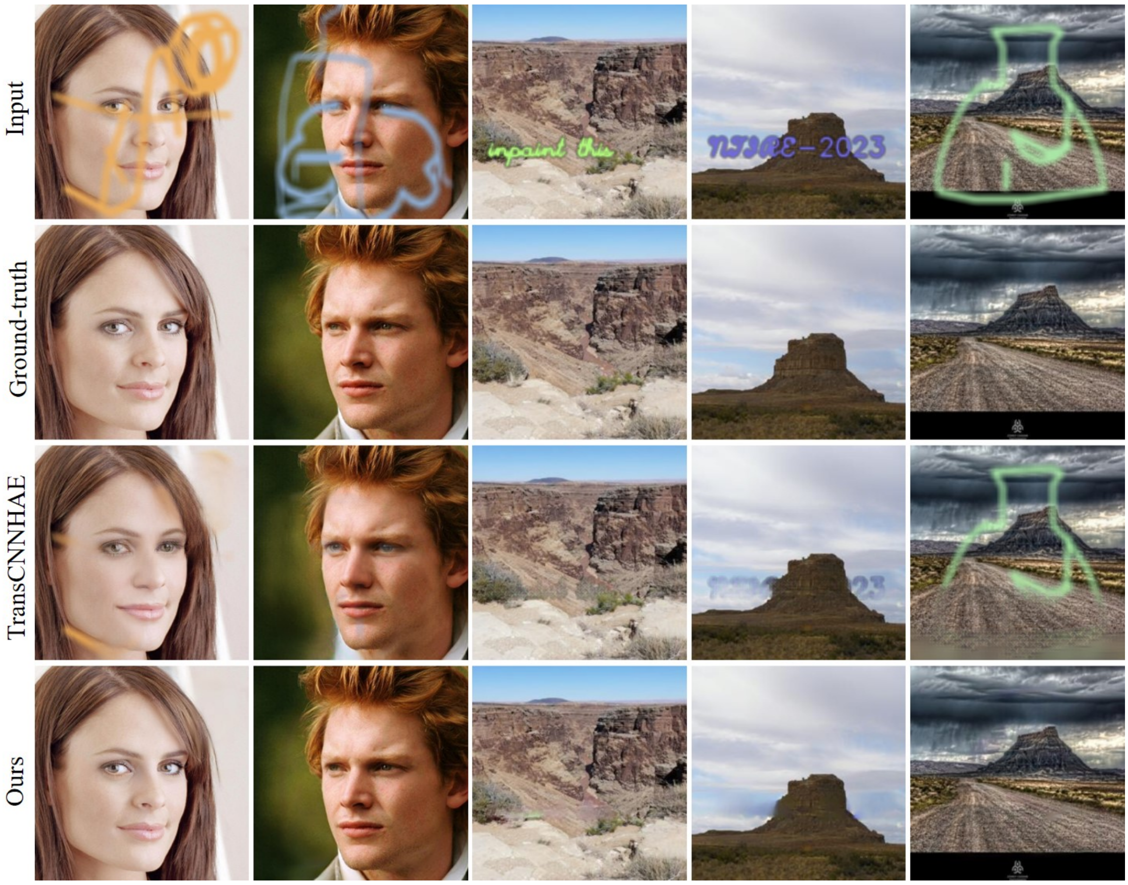


Figure 5.5: Qualitative results comparison of the proposed method (Ours) with existing state-of-the-art method (TransCNNHAE [24]) on unseen patterns.

+ *OGA* to verify the effectiveness of both the proposed modules. The inclusion of the proposed OGA to forward the encoder features to the respective decoder performed well by improving in terms of PSNR and SSIM. Along with these parameters improvement, there is a lot of improvement in the FID value (see Table 5.1).

Overall, our proposed modules ($Q_W, K, V$ + *OGA*) effectively help the network with improved performance for the task of blind image inpainting. Also, the visual results of the ablation study are provided in Figure 5.2.

### Blind Image Inpainting Results Analysis

For the task of blind image inapinting, we considered four different datasets covering large variety of cases like natural places scenes, facial images. The comparison in terms of PSNR, SSIM $L_1$ error and FID is provided in Table 5.2. Along with state-of-the-art blind image inpainting methods [23, 24] ([24] is retrained on respective datasets as per the configurations provided due to unavailability of pre-trained checkpoints), we considered the existing non-blind image inpainting method [16] with best performance (*as provided in [24]*). Since, it is worth to note that, the existing non-blind method may not work feasibly for blind image inpainting task, we provided the ground-truth masks as inputs to these methods as suggested in [24]. From Table 5.2, it is clear that the proposed approach for blind image inpainting performs remarkably as compared to state-of-the-art blind and non-blind methods.

The visual results comparison for blind image inpainting is provided in Figure 5.3 and 5.4. When compared qualitatively, our proposed method generates comparatively plausible results on all the datasets for blind image inpainting.

### Unseen Contamination Result Analysis

Here, we have evaluated the performance of our proposed approach for unseen contamination such as random scratches and text. The comparison is done with the existing state-of-the-art (TransCNNHAE [24]) for blind image inpainting. Figure 5.5 shows the performance of our proposed approach on unseen patterns as compared to existing approach for blind image inpainting.

## 5.2 Blind Image Inpainting via High-frequency Attentive Deformable Convolution

In order to inpaint the corrupted image with large masked regions, it is necessary to focus on capturing the higher receptive fields which further helps the network for attaining global consistency in the output. In this contribution, *we propose a multi-kernel transformer block which mainly does the task of considering the maximum receptive fields.* The corrupted inputs for these tasks have almost similar degradation where the valid image contents are blended with noisy content. In this kind of restoration, it is desired to focus on the pixel variations or inconsistencies so that the high variations (residuals) can be captured easily. In order to achieve this, and inspired from existing image restoration tasks such as image rain removal, snow removal, we propose a novel *high-frequency offset deformable feature merging* approach to capture the variations in the input feature maps. The contributions of our work are:

- We propose an end-to-end transformer architecture for blind image inpainting independent of intermediate mask prediction.
- A *high receptive field extracting (multi-kernel) multi-head attention* is proposed to focus on the large contextual information for blind image inpainting.
- A *novel high-frequency offset deformable feature merging module* is proposed to highlight the disturbances in the input image.

This proposed approach achieves remarkable performance improvement as compared to existing state-of-the-art blind image inpainting methods. To further verify the efficiency of our proposed approach, we evaluate the performance of our approach on different image restoration tasks like rain and snow removal.

### 5.2.1 Proposed Method

The key consideration behind designing the proposed architecture is to *introduce an efficient blind image inpainting method which does not rely on any kind of mask prediction* whether it is of two-stage [23] or single-stage [24]. Here we propose a single-stage end-to-end transformer architecture for blind image inpainting (*see Figure 5.6*). In this single stage-architecture, we propose two major components namely: (a) high receptive field extracting multi-head attention: *to capture the global context* efficiently, and (b) high-frequency offset deformable feature merging: for *processing the contaminated regions by highlighting the high variations in the input feature maps.* In this section, we will first give the detailed exposition of the proposed architecture for blind image inpainting and then we detail the proposed modules.

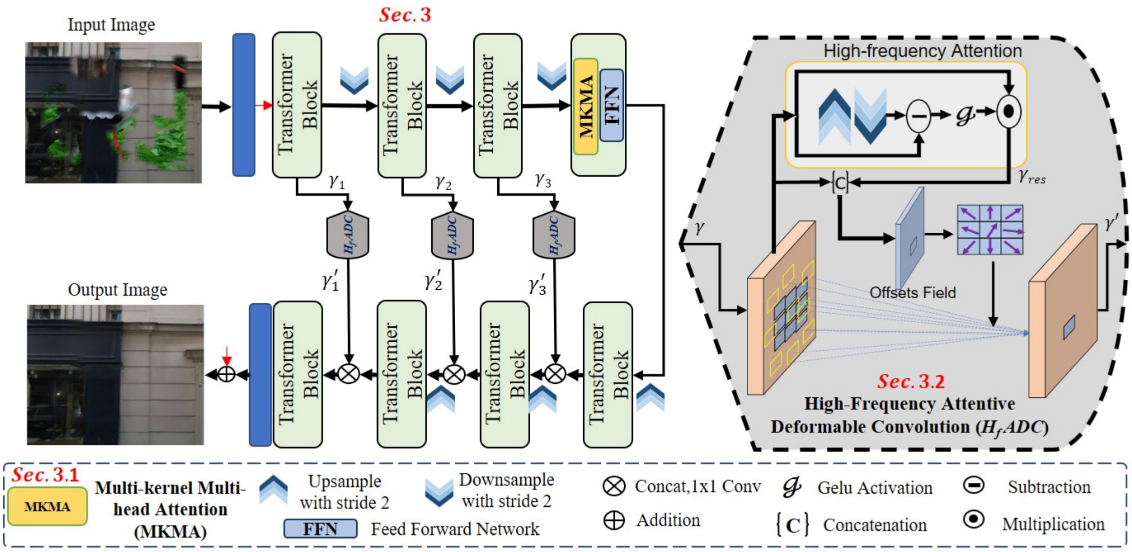


Figure 5.6: Overview of the proposed architecture for blind image inpainting.

The existing multi-stage [131, 23] and single-stage [24] blind image inpainting methods directly or indirectly depend on predicting the masked regions. Unlike these methods, we urge to propose an approach which directly inpaints the image without any kind of prediction of masked regions. To do this, we observed that, in case of blind image inpainting, the contaminations are more like the rain and snow degradations in the weather degraded images which are nothing but the unusual variations (high-frequency) in the image [34]. Highlighting these variations may help the network towards efficient inpainting results. This motivated us to propose a novel high-frequency offset deformable feature merging module to finely focus on the high-frequency (pixel variations) degradations while inpainting the image. Also, we include the high receptive field extracting multi-head attention to capture the global context enabling the network to inpaint the image with large corruptions.

The overview of proposed transformer based architecture with the multi-kernel multi-head attention (MKMA) and high-frequency attentive deformable merging ($H_fADM$) is shown in Figure 5.6. To convert input image into feature space, we first apply the convolution layer. These convolved features are processed through successive transformer blocks followed by down-sampler. The input with spatial size $m, n$ is then converted into $\frac{m}{8}, \frac{n}{8}$ sized feature maps at $4^{th}$ transformer block. In this transformer block, we propose *a multi-kernel multi-head attention (MKMA) to focus on large receptive fields efficiently.* These feature maps are then forwarded again to the successive transformer blocks but now these blocks are followed with up-sampler to come up with actual spatial dimension $(m, n)$ at the last stage. Here, in the decoding stage, we apply the proposed high-frequency attentive deformable merging ($H_fADM$) of features while giving skip connection from respective encoder to subsequent decoder level. *The $H_fADM$ helps the network to attentively identify the high variations in the input feature maps and guide the decoder* *for efficient residual generation.* Finally, we again apply a convolution layer to generate the high-frequency (residual) image in terms of negative residual. This high-frequency residual image is then added with the input to generate final output ($O$).

### Multi-Kernel Multi-head Attention (MKMA)

Blind image inpainting is a restoration task where the contaminations in the input appear at random *i.e.* random in shape, size and locations *etc*. In order to tackle this randomness, it is required to consider the maximum receptive fields and long-term dependencies. The transformers are well known to consider long range dependencies effectively [17, 32, 24]. We consider this property of transformers to advance the blind image inpainting performance by effectively extracting the features and capturing long-term dependencies. In existing work [32], the multi-head attention is introduced with less computational overhead. On top of this, *we introduce the multi-head attention with the capability of capturing more receptive fields.* This high receptive fields capturing capability may help the network to inpaint the image with varying sized corrupted regions.

At first, the input feature map ($\mathbb{F}$) is converted into query ($Q$), key ($K$) and values ($V$) by processing through the $1 \times 1$ convolution. Then, in order to capture maximum receptive fields, the $Q$, $K$ and $V$ are processed through the multi-kernel depth-wise separable convolutions (MKDC) (*see Figure 5.6*) which is represented as:

$$Q = \Psi_D^1 \left( \mathbb{C} \left[ \Psi_D^{\mathtt{k}} \left( \Psi_D^1(\mathbb{F}) \right) \right] \right); \mathtt{k} \in (1, 3, 5, 7) \tag{5.10}$$

where, $\mathbb{C}[\cdot]$ is concatenation, $\Psi_D^{\mathtt{k}}$ is the depth-wise separable convolution with $\mathtt{k} \times \mathtt{k}$ kernel size. The $K$ and $V$ are calculated in similar way with multi kernel depth-wise separable convolutions (MKDC). Further, the $Q$, $K$ and $V$ are reshaped and the attentive dot product of $Q$, $K$ is projected on $V$. So finally the multi-kernel multi-head attention (MKMA) on input feature $\mathbb{F}$ is represented as:

$$MKMA(\mathbb{F}) = \sigma(Q \cdot K^T) \cdot V \tag{5.11}$$

*Applying, the MKDC on $Q$, $K$ and $V$ allows the network to delve into the maximum receptive fields while capturing the long-term dependencies.* Overall, this leads to globally consistent inpainted image. Further, similar to existing transformer blocks, the output of MKMA is fed to feed forward network (FFN) and layer-norm.

### High-frequency Attentive Deformable Merging

The existing blind image inpainting methods aim to predict the corrupted locations [23, 131]. Due to failure in appropriate prediction, these methods are unable to inpaint the image efficiently. We assume, the blind image inpainting task is equivalent to the

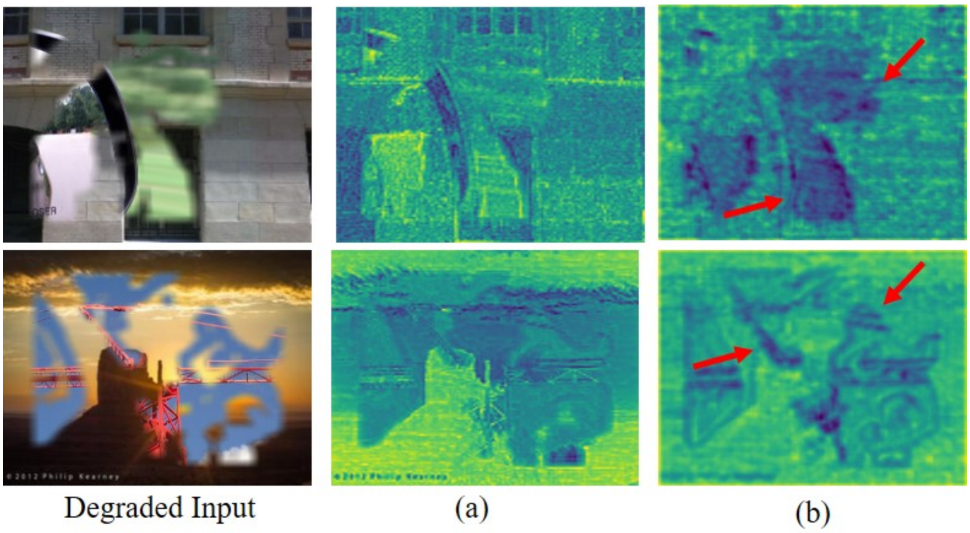


Figure 5.7: The feature map visualization of (a) conventional convolution and (b) the proposed high-frequency as offsets to deformable convolution.

image restoration tasks such as image snow removal and rain removal as the noise is blended on top of the actual information. The images with these types of degradations, generally have high variations (residuals) which should be highlighted while restoring the image. With this intuition, we propose high-frequency attentive deformable feature merging module ($H_fADM$). *The $H_fADM$ allows the network to effectively identify the irrelevant variations present in the input.* Attentively forwarding these input variations to the reconstruction decoder may help for structurally consistent inpainting. In order to highlight the residuals, we apply the phenomenon of residual generation when the input is processed through a forward and inverse function successively. So we generate the high-frequency variations in the feature map $\mathbb{F}$ as: $\mathbb{F}_{res} = \mathbb{F} - (\uparrow\downarrow \mathbb{F})$, where $\uparrow$ and $\downarrow$ are up-sampling and down-sampling with `convolution→pixel-shuffle` and `convolution→pixel-unshuffle`, respectively to highlight the variations in the input (see Figure 5.7 (b)).

In order to utilize these residuals in an efficient way, it is required to adaptively process these variations in the input feature maps. The deformable convolution is the key to process the input features adaptively [163]. In deformable convolution, offsets play an important role in determining the locations to be considered for further processing. *We provide the high-frequency attentive offsets to the deformable convolution in the proposed $H_fADC$ block. The $H_fADC$ attentively picks the locations with respect to the high-frequency (variations) in the input feature maps.* Figure 5.7 shows that the proposed approach effectively provides high variation offsets to deformable convolution. The output of $H_fADC$ is concatenated with the respective decoding transformer block for generation of the negative residual (high-frequency) image which is then added with the input to generate final inpainted image ($O$). So, overall the $H_fADC$ for every position $n$ on input feature map $\psi$ is represented as:

$$\gamma' = H_f ADC(\gamma_n) = \sum_{i=1}^{N} \psi(n_i) \cdot \gamma(n + n_i + \Delta n_i) \cdot \Delta m_{n_i} \tag{5.12}$$

where, $\psi$ is kernel weights, $\Delta n_i$ and $\Delta m_{n_i}$ are the high-frequency attentive extracted offsets and modulator scalars respectively, $n_i \in \{(-1,-1),(-1,0),......,(0,1),(1,1)\}$. Extraction of high-frequency attentive offsets is given as:

$$\{\Delta n_i, \Delta m_{n_i}\} = \Theta = \psi_{off}(\langle \gamma, GELU(\gamma_{hf}) \odot \gamma \rangle) \tag{5.13}$$

where, $\psi_{off}$ is offset convolution with kernel size $3 \times 3$, and $GELU$ is Gaussian error linear unit activation function, $\langle a, b \rangle$ is concatenation operation on $a$ and $b$, and $\odot$ is the multiplication operation.

### 5.2.2 Experiments and Results Discussion

In this section we will discuss different experimental datasets, evaluation metrics, quantitative and qualitative results of the proposed and existing state-of-the-art methods.

#### Datasets and Evaluation Metrics

Here: FFHQ [2], CelebA-HQ [162], Placs2 [3], Paris Street View (Paris_SV) [4], and ImageNet [5]. The comparative analysis for blind image inpainting is done with VCNet [23], TransCNNHAE [24]. Since the image corrupted regions may vary, for fair comparison we have compared methods with publicly available source codes on all the blind image inpainting datasets.

In addition, to further verify the network performance further, we trained and tested the proposed approach for rain and snow removal. The Rain 13k and Test1200 datasets [32] are used to train and test the proposed network for rain removal application respectively. For image desnowing, we use Snow100K dataset for training and SnowTest100K-L dataset for testing [37]. For rain and snow removal, the quantitative values are collected from [32] and [34] respectively. The qualitative results are collected from the official github repository.

For quantitative results comparison of the proposed method and existing state-of-the-art methods on blind image inpainting, peak signal-to-noise ratio (PSNR), structural similarity index measure (SSIM), mean $L_1$ error and Fréchet inception distance (FID) metrics are used. Further, PSNR evaluated on Y channel and SSIM metrics are used for comparison on rain and snow removal [32].

Table 5.3: Ablation study on different configurations of the proposed network on FFHQ dataset for **blind image inpainting**.

| Network Configuration | PSNR↑ | SSIM↑ | $L_1$ ↓ | FID ↓ |
|---|---|---|---|---|
| (a) Transformer (MHA) | 26.86 | 0.933 | 0.0297 | 9.124 |
| (b) *Transformer (MKMA)* | 27.58 | 0.942 | 0.0257 | 8.956 |
| (c) High-frequency merging | 28.05 | 0.951 | 0.0228 | 8.246 |
| (d) Deformable merging | 28.96 | 0.958 | 0.0189 | 7.978 |
| (e) $H_f ADM$ | **29.38** | **0.960** | **0.0176** | **7.700** |

### Implementation and Training Details

To train the network, we use AdamW optimizer with $3e^{-4}$ learning rate which is gradually reduced with the cosine annealing strategy. Instead of utilizing the general $L_1$ loss function, we calculate the region based loss function while training the network *i.e.* $L_1$ loss for corrupted ($L_1^c$) regions and $L_1$ loss for valid regions ($L_1^v$). As we are focused to highlight the residual information in the corrupted regions, more weight is given to $L_1^c$. Along with this, we use the perceptual loss ($L_{Per}$) to guide the network for efficient feature extraction. The edge ($L_{Edge}$) and structural similarity ($L_{SSIM}$) losses are used to focus on the edge and structural information while training the network. So the overall loss to train the network is given as:

$$L_T = \lambda_c L_1^c + \lambda_v L_1^v + \lambda_{Per} L_{Per} + \lambda_{Edge} L_{Edge} + \lambda_{SSIM} L_{SSIM} \tag{5.14}$$

where, $\lambda_{Loss}$ is the weight assigned to respective *Loss* function which is verified experimentally as: $\lambda_c = 0.9$, $\lambda_v = 0.2$, $\lambda_{Per} = 0.6$, $\lambda_{Edge} = 0.4$, $\lambda_{SSIM} = 0.5$. To train the network for rain and snow removal, instead of separate $L_1^c$ and $L_1^v$, we used $L_1$ loss for overall image.

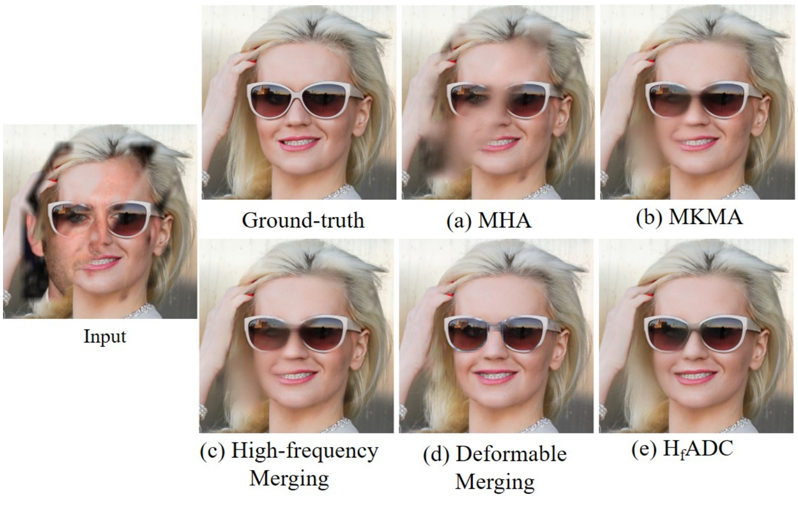


Figure 5.8: Qualitative result analysis of ablation study on different configurations of the proposed network for blind image inpainting.

### Ablation Study

To determine the design choices of the network for blind image inpainting, we performed various experiments on FFHQ dataset. How the successive inclusion of each proposed module led to the performance improvement is discussed in this section.

*Need of Multi Kernel Multi-head Attention:*

The first observed limitation of existing blind inpainting methods is, they fail at inpainting the image with variable contaminations. In order to mitigate this limitation, unlike [24], we used the end-to-end transformer network architecture, which helped to extract the long-term dependencies. The results for the end-to-end transformer with conventional multi-head attention (MHA) [32] configuration are considered as baseline for all the experiments (*see configuration (a) in Table 5.3*). So, we proposed a multi kernel multi-head attention module in the transformer block which helped to improve the performance in terms of all the evaluation metrics (*see configuration (b) in Table 5.3*). Also, from Figure 5.8, we can see that transformer block with existing multi-head attention (MHA) [32] is not able to capture maximum information. Whereas, the multi-kernel multi-head attention (MKMA) improve the results (*see Figure 5.8 (a) and (b)*).

*Effect of High-frequency Attentive Deformable Merging:*

Embedding the encoder features while reconstruction plays an important role in image restoration task. Efficient feature extraction from encoder may further enhance the performance. In order to evaluate the performance of proposed high-frequency attentive deformable merging, we performed the experiments in which various configurations are used as given in Table 5.3 (*configuration (c), (d) and (e)*). Utilizing the high-frequency feature merging (configuration (c)) improved the performance a bit as compared to direct merging of features (configuration (a) and (b) in Table 5.3). Further, applying the conventional deformable convolution (*see configuration (d) in Table 5.3*) helped the network to improve the performance over direct merging of the features. So, we designed the **amalgamation of attentive high-frequency features with deformable convolution** which improved the inpainting performance (*see configuration (e) in Table 5.3*). From Figure 5.8 (c), (d) and (e) it is clear that the amalgamation of high-frequency attention in deformable convolution layer generate plausible results for image inpainting. Overall, our proposed modules effectively help the network with the improved performance for the task of blind image inpainting.

### Blind Image Inpainting Results

For the task of blind image inapinting, we considered five different datasets covering large variety of cases like natural places scenes, objects and faces. The evaluation in terms of PSNR, SSIM, $L_1$ error and FID is tabulated in Table 5.4. Along with state-of-the-art blind image inpainting methods [23, 24] ([24] is retrained on respective datasets as per the

Table 5.4: Comparison of the proposed method (ours) and existing state-of-the-art methods for blind image inpainting.

| Method | PSNR↑ | SSIM↑ | $L_1$ ↓ | FID↓ |
|---|---|---|---|---|
| FFHQ | | | | |
| VCNet [23] | 23.62 | 0.861 | 0.0482 | 10.148 |
| TransCNNHAE[24] | 27.05 | 0.941 | 0.0281 | 9.424 |
| **Ours** | **29.38** | **0.960** | **0.0176** | **7.700** |
| CelebA-HQ | | | | |
| VCNet [23] | 25.59 | 0.874 | 0.0396 | 9.275 |
| TransCNNHAE[24] | 27.71 | 0.949 | 0.0250 | 7.251 |
| **Ours** | **29.12** | **0.959** | **0.0179** | **7.209** |
| ImageNet | | | | |
| VCNet [23] | 22.46 | 0.856 | 0.0518 | 21.984 |
| TransCNNHAE[24] | 24.68 | 0.903 | 0.0357 | 26.655 |
| **Ours** | **26.65** | **0.931** | **0.0250** | **21.510** |
| Places2 | | | | |
| VCNet [23] | 24.09 | 0.869 | 0.0429 | 28.821 |
| TransCNNHAE[24] | 26.87 | 0.910 | 0.0261 | 17.640 |
| **Ours** | **28.62** | **0.928** | **0.0181** | **17.263** |
| ParisSV | | | | |
| VCNet[23] | 23.62 | 0.824 | 0.0527 | 64.215 |
| TransCNNHAE[24] | 26.72 | 0.896 | 0.0352 | 41.505 |
| **Ours** | **28.71** | **0.919** | **0.0224** | **39.414** |

configurations provided due to unavailability of pre-trained checkpoints), we considered the existing non-blind image inpainting method [16] with best performance (*as provided in [24]*). Since, it is worth to note that, the existing non-blind method may not work feasibly for blind image inpainting task, we provided the ground-truth masks as inputs to these methods as suggested in [24]. From Table 5.4, it is clear that the proposed approach for blind image inpainting performs remarkably as compared to state-of-the-art blind and non-blind methods.

The visual results comparison for blind image inpainting is provided in Figure 5.9. When compared qualitatively, our proposed method generates comparatively plausible results on all the datasets for blind image inpainting.

### Rain Removal Results

As the design of the proposed architecture is mainly motivated with the degradation similarity to rain and snow removal tasks, the performance of proposed method is compared with existing state-of-the-art methods for rain removal. The quantitative results tabulated in Table 5.5 show that the proposed architecture achieves significant improvement over existing methods for rain removal. Specifically, our proposed architecture obtain 1.40 *dB* PSNR and 0.012 SSIM gain as compared to existing best method [32] for rain removal. Also, the qualitative results comparison is provided in Figure 5.10. The proposed method restores the image faithfully as compared to state-of-the-art

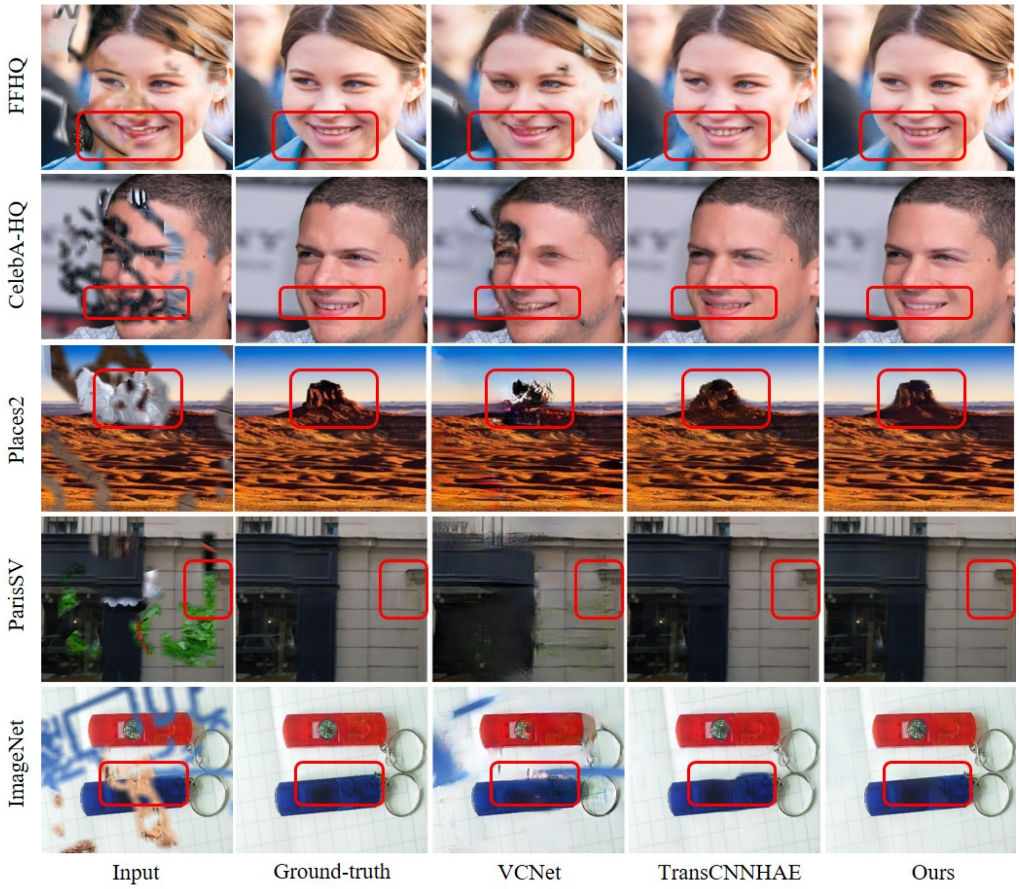


Figure 5.9: Qualitative results comparison of the proposed method (Ours) with existing state-of-the-art methods (VCNet [23], CTSDG [16], TransCNNHAE [24]) for blind image inpainting.

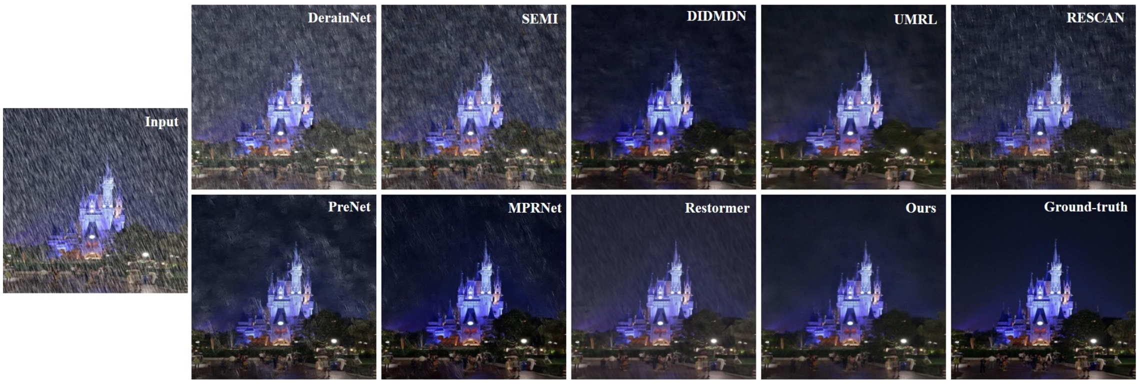


Figure 5.10: Qualitative results comparison of the proposed method (Ours) and existing state-of-the-art methods (DerainNet[25], SEMI[26], DIDMDN[27], UMRL [28], RESCAN[29], PreNet[30], MPRNet[31], Restormer[32]) for rain removal.

methods for rain removal.

### Snow Removal Results

We evaluate the performance of proposed method on Snow100K-L test dataset with existing state-of-the-art methods for snow removal. The quantitative comparison provided in Table 5.6 shows that, our proposed method achieves the best performance as compared

Table 5.5: Comparison of the proposed method (Ours) and existing state-of-the-art methods on Test1200 [27] dataset for **rain removal**.

| Method | Publication | PSNR↑ | SSIM↑ |
|---|---|---|---|
| DerainNet[25] | TIP-17 | 23.38 | 0.835 |
| SEMI[26] | CVPR-19 | 26.05 | 0.822 |
| DIDMDN[27] | CVPR-18 | 29.65 | 0.901 |
| UMRL [28] | CVPR-19 | 30.55 | 0.910 |
| RESCAN[29] | ECCV-18 | 30.51 | 0.882 |
| PreNet[30] | CVPR-19 | 31.36 | 0.911 |
| MSPFN[164] | CVPR-20 | 32.39 | 0.916 |
| MPRNet[31] | CVPR-21 | 32.91 | 0.916 |
| SPAIR[165] | ICCV-21 | 33.04 | 0.922 |
| Restormer[32] | CVPR-22 | 33.19 | 0.926 |
| **Ours** | - | **34.59** | **0.938** |

Table 5.6: Comparison of the proposed method (Ours) and existing methods on SnowTest100k-L test dataset [37] for **snow removal**.

| Method | Publication | PSNR↑ | SSIM↑ |
|---|---|---|---|
| DetailsNet[166] | CVPR-17 | 19.18 | 0.7495 |
| DesnowNet[37] | TIP-18 | 27.17 | 0.8983 |
| JSTASR[167] | ECCV-20 | 25.32 | 0.8076 |
| WiperNet [33] | ITS-22 | 27.64 | 0.8857 |
| Swin-IR[141] | ICCV-21 | 28.18 | 0.8800 |
| DDMSNET[168] | TIP-21 | 28.85 | 0.8772 |
| All-in-One[169] | CVPR-20 | 28.33 | 0.8820 |
| TransWeather[34] | CVPR-22 | 28.48 | 0.9308 |
| **Ours** | - | **30.02** | 0.9261 |

to existing state-of-the-art methods for snow removal. Also, it is worth to note that our proposed method achieves the improvement of 1.48 *dB* PSNR as compared to existing best method [34]. The visual results comparison for snow removal is provided in Figure 5.11. This comparison shows the ability of the proposed approach towards effective snow removal as compared to existing state-of-the-art methods.

## 5.3 Summary of Proposed Contribution

In this chapter we proposed two solutions for blind image inpainting. The first solution (Section 5.1) proposes an end-to-end transformer with wavelet coefficient processing and providing them as a query to multi-head attention in the transformer block. Further, the gated omni-dimensional attention is proposed to forward the encoder features to the respective decoder as a skip connection. Along with the comparison with blind and non-blind image inpainting methods, the performance of the proposed approach is verified for unseen contamination.

Similarly, the second contribution inpaint the images with arbitrary size, shape and

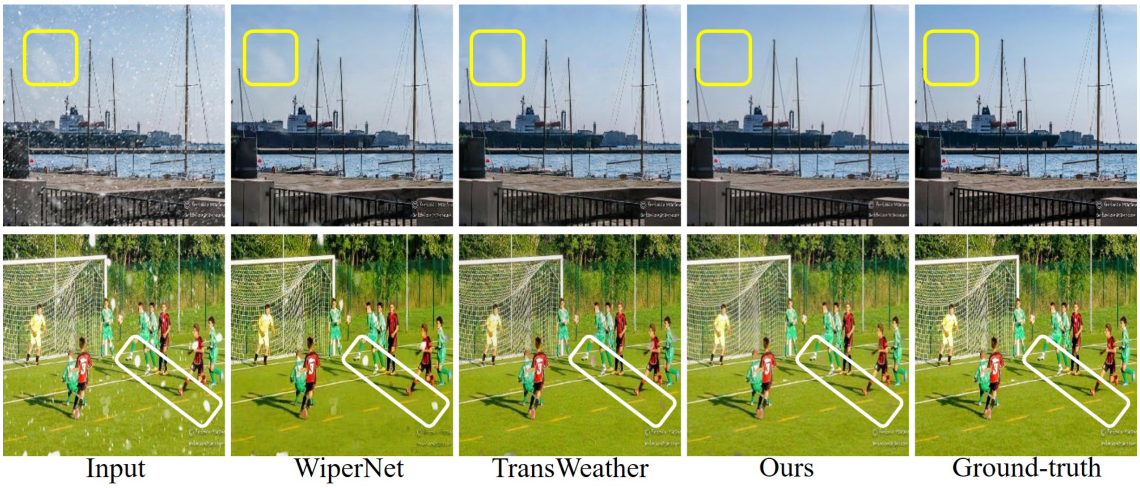


Figure 5.11: Qualitative results comparison of the proposed method (Ours) and existing state-of-the-art methods (WiperNet [33], TransWeather[34]) for snow removal.

Table 5.7: Computational complexity analysis (the **best** and <u>second best</u> are shown in **bold** and <u>underline</u>).

| Method | Parameters (M) ↓ | FLOPs (G) ↓ |
|---|---|---|
| VCNet [23] | 3.79 | 65.25 |
| CTSDG [16] | 52.14 | 53.38 |
| TransCNNHAE [24] | **2.75** | <u>19.71</u> |
| III:A | <u>3.24</u> | **16.61** |
| III:B | 4.06 | 29.01 |

locations by proposing a multi-kernel multi head attention. With the observation of similarity in blind image inpainting and other restoration tasks (image rain/snow removal), we proposed a high-frequency attentive deformable merging to highlight the undesired disturbances in the input image. The quantitative and qualitative comparison is carried out with the blind image inpainting methods. Further, the experiments on rain and snow removal proves the efficiency of the proposed method when compared with existing state-of-the-art methods for rain and snow removal.

The computational complexity comparison of the proposed approaches and existing methods is given in Table 5.7 in terms of the number of trainable parameters and the number of floating point operations (FLOPs). Although moderately complex in terms of the number of trainable parameters and FLOPs, our proposed approaches perform better as compared to compared to state-of-the-arts.

# Chapter 6

# Conclusion and Future Scope

## 6.1 Conclusion of Proposed Work

The main aim of this thesis work is to design novel deep generative architectures for image inpainting and mask prediction deep generative architectures for blind image inpainting. The major concern of image inpainting task is the balance between the quality and complexity of the existing approaches for image inpainting. Specifically, for blind image inpainting case, the existing approaches directly or indirectly depend on intermediate mask prediction failing in which leads to undesired inpainted results. Many approaches exist with coarse-to-fine or single-stage architectures for image inpainting having either inconsistent results or high complexity.

In order to have an architecture producing highly plausible results, we proposed different coarse to fine architectures for image inpainting. The main objective of these proposed coarse-to-fine architectures is to generate spatially consistent inpainting results with remarkable performance.

Also, to have computationally efficient architecture for image inpainting, we proposed different architectures with less computational cost as compared to existing approaches for image inpainting. These proposed architectures generate remarkable outcomes along with having less complexity as compared to existing approaches.

The comparison of all the proposed approaches for image inpainting in terms of computational complexity and PSNR, SSIM on three different datasets corrupted using NVIDIA masks is given in Table 6.1.

Further, the blind image inpainting is a task where the inpainting architecture should be ideally independent of knowledge of masked regions. In this regard we proposed two mask prediction independent approaches for blind image inpainting. The proposed blind image inpainting architectures give remarkable performance on five different datasets (CelebA-HQ, FFHQ, Places2, Paris Street View and ImageNet) as compared to existing blind image inpainting methods.

The comparison of the proposed approaches for blind image inpainting is provided in Table 6.2 in terms of computational complexity, PSNR, and SSIM on four different datasets.

Table 6.1: The quantitative comparison between the proposed approaches in terms of computational complexity (MS: Model Size in number of trainable parameters in Millions, GFLOPs: Giga FLOPs number of operations, RT: Run-time in sec/image), PSNR and SSIM for CelebA-HQ, Places2 and Paris_SV datasets corrupted using NVIDIA masks [6] for image inpainting.

| Methods | | I:A<br>(Sec. 3.1) | I:B<br>(Sec. 3.2) | I:C<br>(Sec. 3.3) | II:A<br>(Sec. 4.1) | II:B<br>(Sec. 4.2) | II:C<br>(Sec. 4.3) |
|---|---|---|---|---|---|---|---|
| Complexity | MS | 56.00 | 4.10 | 2.30 | 14.01 | 4.80 | 0.97 |
| | GFLOPs | 430.0 | 7.5 | 116.0 | 6.5 | 40.1 | 13.7 |
| | RT | 0.40 | 0.08 | 0.45 | 0.09 | 0.24 | 0.08 |
| CelebA-HQ | PSNR | 29.83 | 28.19 | 28.08 | 28.25 | - | 28.06 |
| | SSIM | 0.946 | 0.929 | 0.923 | 0.942 | - | 0.926 |
| Places2 | PSNR | 26.22 | 26.87 | 24.81 | 26.18 | 25.14 | 24.51 |
| | SSIM | 0.905 | 0.879 | 0.852 | 0.876 | 0.884 | 0.870 |
| Paris_SV | PSNR | 28.87 | - | 27.8 | - | 26.98 | 27.54 |
| | SSIM | 0.910 | - | 0.863 | - | 0.856 | 0.862 |

Table 6.2: The quantitative comparison between the proposed approaches (III:A, III:B) in terms of computational complexity (MS: Model Size in number of trainable parameters in Millions, GFLOPs: Giga FLOPs number of operations, RT: Run-time in sec/image), PSNR and SSIM for CelebA-HQ, FFHQ, Places2, Paris_SV and ImageNet datasets for blind image inpainting

| Method | III:A | III:B | III:A | III:B |
|---|---|---|---|---|
| Complexity | MS | | FLOPs | |
| | 3.24 | 4.06 | 16.61 | 29.01 |
| Dataset | PSNR | | SSIM | |
| CelebA-HQ | 28.21 | 29.12 | 0.951 | 0.959 |
| FFHQ | 28.19 | 29.38 | 0.952 | 0.960 |
| Places2 | 27.55 | 28.62 | 0.918 | 0.928 |
| Paris_SV | 27.81 | 28.71 | 0.905 | 0.919 |

## 6.2 Future Scope

In this work, we proposed different approaches for image inpainting and blind image inpainting. The blind image inpainting task can be further extended in order to have computationally efficient architectures. Also, the proposed work can be extended for image outpainting where the image is painted outwards from the inside to enlarge the view. Considering efficiency of the proposed architectures, this work can be extended for video inpainting task.

# List of Publications

## International Journals

1. Phutke Shruti S. and S. Murala, "Pseudo decoder guided light-weight architecture for image inpainting", IEEE Transactions on Image Processing, vol. 31, pp. 6577-6590, 2022 (**Impact Factor=11.041**).

2. Phutke Shruti S. and S. Murala, "Image inpainting via spatial projections", Pattern Recognition, vol. 133, p. 109040, 2022 (**Impact Factor =8.518**).

3. Phutke Shruti S. and S. Murala, "FASNET: Feature aggregation and sharing network for image inpainting", IEEE Signal Processing Letters, vol. 29, pp. 1664-1668, 2022 (**Impact Factor =3.109**).

4. Phutke Shruti S. and S. Murala, "Diverse receptive field based adversarial concurrent encoder network for image inpainting", IEEE Signal Processing Letters, vol. 28, pp. 1873-1877, 2021 (**Impact Factor =3.109**).

5. Phutke Shruti S. and S. Murala, "Image inpainting via correlated multi-resolution feature projection", IEEE Transactions on Visualization and Computer Graphics (Minor-revision submitted).

## International Conference

1. Phutke Shruti S. and S. Murala, "Nested deformable multi-head attention for facial image inpainting", IEEE/CVF Winter Conference Computer Vision (WACV)-2023, pp. 6078-6087 (**h-index=76**).

2. Phutke Shruti S., A Kulkarni, S K Vipparthi, and S. Murala, "Blind Image Inpainting via Omni-dimensional Gated Attention and Wavelet Queries", IEEE/CVF Computer Vision and Pattern Recognition Workshop (CVPRW)-2023, pp. 1251-1260 (**h-index=106**).

3. Phutke Shruti S., A Kulkarni , S. K. Vipparthi, and S. Murala, "Blind Image Inpainting", IEEE International Conference on Computer Vision (ICCV)-2023 (Submitted).

# References

[1] 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.

[2] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4401–4410, 2019.

[3] 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*, 40(6):1452–1464, 2017.

[4] Carl Doersch, Saurabh Singh, Abhinav Gupta, Josef Sivic, and Alexei Efros. What makes paris look like paris? *ACM Transactions on Graphics*, 31(4), 2012.

[5] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[6] Guilin Liu, Fitsum A. Reda, Kevin J. Shih, Ting-Chun Wang, Andrew Tao, and Bryan Catanzaro. Nvidia irregular mask dataset. In *https://nv-adlr.github.io/publication/partialconv-inpainting*, 2018.

[7] David Ha and Douglas Eck. A neural representation of sketch drawings. *arXiv preprint arXiv:1704.03477*, 2017.

[8] Gourav Wadhwa, Abhinav Dhall, Subrahmanyam Murala, and Usman Tariq. Hyperrealistic image inpainting with hypergraphs. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3912–3921, 2021.

[9] Yi Wang, Xin Tao, Xiaojuan Qi, Xiaoyong Shen, and Jiaya Jia. Image inpainting via generative multi-column convolutional neural networks. *Advances in neural information processing systems*, 31, 2018.

[10] 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.

[11] Chuanxia Zheng, Tat-Jen Cham, and Jianfei Cai. Pluralistic image completion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1438–1447, 2019.

[12] 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.

[13] 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*, pages 0–0, 2019.

[14] 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*, pages 7760–7768, 2020.

[15] Ning Wang, Sihan Ma, Jingyuan Li, Yipeng Zhang, and Lefei Zhang. Multistage attention network for image inpainting. *Pattern Recognition*, 106:107448, 2020.

[16] 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.

[17] 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.

[18] 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.

[19] Y Zhang, K Li, B Zhong, and Y Fu. Residual non-local attention networks for image restoration. In *International Conference on Learning Representations*, 2019.

[20] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141, 2018.

[21] 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.

[22] 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.

[23] Yi Wang, Ying-Cong Chen, Xin Tao, and Jiaya Jia. Vcnet: A robust approach to blind image inpainting. In *European Conference on Computer Vision*, pages 752–768. Springer, 2020.

[24] Haoru Zhao, Zhaorui Gu, Bing Zheng, and Haiyong Zheng. Transcnn-hae: Transformer-cnn hybrid autoencoder for blind image inpainting. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 6813–6821, 2022.

[25] Xueyang Fu, Jiabin Huang, Xinghao Ding, Yinghao Liao, and John Paisley. Clearing the skies: A deep network architecture for single-image rain removal. *IEEE Transactions on Image Processing*, 26(6):2944–2956, 2017.

[26] Wei Wei, Deyu Meng, Qian Zhao, Zongben Xu, and Ying Wu. Semi-supervised transfer learning for image rain removal. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3877–3886, 2019.

[27] He Zhang and Vishal M Patel. Density-aware single image de-raining using a multi-stream dense network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 695–704, 2018.

[28] Rajeev Yasarla and Vishal M Patel. Uncertainty guided multi-scale residual learning-using a cycle spinning cnn for single image de-raining. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8405–8414, 2019.

[29] Xia Li, Jianlong Wu, Zhouchen Lin, Hong Liu, and Hongbin Zha. Recurrent squeeze-and-excitation context aggregation net for single image deraining. In *Proceedings of the European conference on computer vision (ECCV)*, pages 254–269, 2018.

[30] Dongwei Ren, Wangmeng Zuo, Qinghua Hu, Pengfei Zhu, and Deyu Meng. Progressive image deraining networks: A better and simpler baseline. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3937–3946, 2019.

[31] Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, Ming-Hsuan Yang, and Ling Shao. Multi-stage progressive image restoration. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14821–14831, 2021.

[32] Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, and Ming-Hsuan Yang. Restormer: Efficient transformer for high-resolution image restoration. In *Proceedings of the IEEE CVPR*, pages 5728–5739, 2022.

[33] Ashutosh Kulkarni and Subrahmanyam Murala. Wipernet: A lightweight multi-weather restoration network for enhanced surveillance. *IEEE Transactions on Intelligent Transportation Systems*, 2022.

[34] Jeya Maria Jose Valanarasu, Rajeev Yasarla, and Vishal M Patel. Transweather: Transformer-based restoration of images degraded by adverse weather conditions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2353–2363, 2022.

[35] Hanxiao Liu, Zihang Dai, David So, and Quoc V Le. Pay attention to mlps. *Advances in Neural Information Processing Systems*, 34:9204–9215, 2021.

[36] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of the IEEE ICCV*, pages 3730–3738, 2015.

[37] Yun-Fu Liu, Da-Wei Jaw, Shih-Chia Huang, and Jenq-Neng Hwang. Desnownet: Context-aware deep network for snow removal. *IEEE Transactions on Image Processing*, 27(6):3064–3073, 2018.

[38] Luxi Li, Qin Zou, Fan Zhang, Hongkai Yu, Long Chen, Chengfang Song, Xianfeng Huang, and Xiaoguang Wang. Line drawing guided progressive inpainting of mural damages. *arXiv preprint arXiv:2211.06649*, 2022.

[39] Chongjian Ge, Yibing Song, Yuying Ge, Han Yang, Wei Liu, and Ping Luo. Disentangled cycle consistency for highly-realistic virtual try-on. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16928–16937, June 2021.

[40] Xintong Han, Zuxuan Wu, Weilin Huang, Matthew R. Scott, and Larry S. Davis. Finet: Compatible and diverse fashion image inpainting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[41] Dahun Kim, Sanghyun Woo, Joon-Young Lee, and In So Kweon. Deep blind video decaptioning by temporal aggregation and recurrence. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4263–4272, 2019.

[42] Pengpeng Chu, Weize Quan, Tong Wang, Pan Wang, Peiran Ren, and Dong-Ming Yan23. Deep video decaptioning. 2021.

[43] 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.

[44] Thomas Schoenemann, Fredrik Kahl, and Daniel Cremers. Curvature regularity for region-based image segmentation and inpainting: A linear programming relaxation. In *2009 IEEE 12th International Conference on Computer Vision*, pages 17–23. IEEE, 2009.

[45] Olivier Le Meur, Mounira Ebdelli, and Christine Guillemot. Hierarchical super-resolution-based inpainting. *IEEE transactions on image processing*, 22(10): 3779–3790, 2013.

[46] Jingang Shi and Chun Qi. Sparse modeling based image inpainting with local similarity constraint. In *2013 IEEE International Conference on Image Processing*, pages 1371–1375. IEEE, 2013.

[47] Masaya Hasegawa, Takahiro Kako, Shigeki Hirobayashi, Tadanobu Misawa, Toshio Yoshizawa, and Yasuhiro Inazumi. Image inpainting on the basis of spectral structure from 2-d nonharmonic analysis. *IEEE transactions on image processing*, 22(8): 3008–3017, 2013.

[48] Liangtian He and Yilun Wang. Iterative support detection-based split bregman method for wavelet frame-based image inpainting. *IEEE Transactions on Image Processing*, 23(12):5470–5485, 2014.

[49] Fang Li and Tieyong Zeng. A universal variational framework for sparsity-based image inpainting. *IEEE Transactions on Image Processing*, 23(10):4242–4254, 2014.

[50] Connelly Barnes, Eli Shechtman, Adam Finkelstein, and Dan B Goldman. Patchmatch: A randomized correspondence algorithm for structural image editing. *ACM Trans. Graph.*, 28(3):24, 2009.

[51] Marin Köppel, Mehdi Ben Makhlouf, Karsen Müller, and Thomas Wiegand. Fast image completion method using patch offset statistics. In *2015 IEEE International Conference on Image Processing (ICIP)*, pages 1795–1799. IEEE, 2015.

[52] Tijana Ruzic and Aleksandra Pizurica. Context-aware patch-based image inpainting using markov random field modeling. *IEEE transactions on image processing*, 24 (1):444–456, 2015.

[53] Zhidan Li, Hongjie He, Heng-Ming Tai, Zhongke Yin, and Fan Chen. Color-direction patch-sparsity-based image inpainting using multidirection features. *IEEE Transactions on Image Processing*, 24(3):1138–1152, 2014.

[54] Mrinmoy Ghorai, Soumitra Samanta, Sekhar Mandal, and Bhabatosh Chanda. Multiple pyramids based image inpainting using local patch statistics and steering kernel feature. *IEEE Transactions on Image Processing*, 28(11):5495–5509, 2019.

[55] Darui Jin and Xiangzhi Bai. Patch-sparsity-based image inpainting through a facet deduced directional derivative. *IEEE Transactions on Circuits and Systems for Video Technology*, 29(5):1310–1324, 2018.

[56] Sandhya Thaskani, Shirish Karande, and Sachin Lodha. Multi-view image inpainting with sparse representations. In *2015 IEEE International Conference on Image Processing (ICIP)*, pages 1414–1418. IEEE, 2015.

[57] Wallace Casaca, Danilo Motta, Gabriel Taubin, and Luis Gustavo Nonato. A user-friendly interactive image inpainting framework using laplacian coordinates. In *2015 IEEE International Conference on Image Processing (ICIP)*, pages 862–866. IEEE, 2015.

[58] Tian-Hui Ma, Yifei Lou, Ting-Zhu Huang, and Xi-Le Zhao. Group-based truncated l 1–2 model for image inpainting. In *2017 IEEE International Conference on Image Processing (ICIP)*, pages 2079–2083. IEEE, 2017.

[59] Naoufal Amrani, Joan Serra-Sagristà, Pascal Peter, and Joachim Weickert. Diffusion-based inpainting for coding remote-sensing data. *IEEE Geoscience and Remote Sensing Letters*, 14(8):1203–1207, 2017.

[60] Xin Zhang, Bernd Hamann, Xiao Pan, and Caiming Zhang. Superpixel-based image inpainting with simple user guidance. In *2017 IEEE International Conference on Image Processing (ICIP)*, pages 3785–3789. IEEE, 2017.

[61] Jiaying Liu, Shuai Yang, Yuming Fang, and Zongming Guo. Structure-guided image inpainting using homography transformation. *IEEE Transactions on Multimedia*, 20(12):3252–3265, 2018.

[62] Ding Ding, Sundaresh Ram, and Jeffrey J Rodríguez. Image inpainting using nonlocal texture matching and nonlinear filtering. *IEEE Transactions on Image Processing*, 28(4):1705–1719, 2018.

[63] Ashutosh Kulkarni, Prashant W Patil, and Subrahmanyam Murala. Progressive subtractive recurrent lightweight network for video deraining. *IEEE Signal Processing Letters*, 29:229–233, 2021.

[64] Akshay Dudhane, Kuldeep M Biradar, Prashant W Patil, Praful Hambarde, and Subrahmanyam Murala. Varicolored image de-hazing. In *proceedings of the*

*IEEE/CVF conference on computer vision and pattern recognition*, pages 4564–4573, 2020.

[65] Prashant W. Patil, Kuldeep M. Biradar, Akshay Dudhane, and Subrahmanyam Murala. An end-to-end edge aggregation network for moving object segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[66] Prashant W. Patil, Akshay Dudhane, and Subrahmanyam Murala. Multi-frame recurrent adversarial network for moving object segmentation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 2302–2311, January 2021.

[67] Prashant W Patil, Akshay Dudhane, Ashutosh Kulkarni, Subrahmanyam Murala, Anil Balaji Gonde, and Sunil Gupta. An unified recurrent video object segmentation framework for various surveillance environments. *IEEE Transactions on Image Processing*, 30:7889–7902, 2021.

[68] 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.

[69] Hongyu Liu, Bin Jiang, Yi Xiao, and Chao Yang. Coherent semantic attention for image inpainting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4170–4179, 2019.

[70] Yurui Ren, Xiaoming Yu, Ruonan Zhang, Thomas H Li, Shan Liu, and Ge Li. Structureflow: Image inpainting via structure-aware appearance flow. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 181–190, 2019.

[71] Junyuan Xie, Linli Xu, and Enhong Chen. Image denoising and inpainting with deep neural networks. *Advances in neural information processing systems*, 25, 2012.

[72] Rolf Köhler, Christian Schuler, Bernhard Schölkopf, and Stefan Harmeling. Mask-specific inpainting with deep neural networks. In *German conference on pattern recognition*, pages 523–534. Springer, 2014.

[73] 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*, pages 5962–5971, 2019.

[74] 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.

[75] Eunhye Lee, Jeongmu Kim, Jisu Kim, and Tae Hyun Kim. Restore from restored: Single-image inpainting. *arXiv preprint arXiv:2102.08078*, 2021.

[76] Andrey Moskalenko, Mikhail Erofeev, and Dmitriy Vatolin. Deep two-stage high-resolution image inpainting. *arXiv preprint arXiv:2104.13464*, 2021.

[77] Hongyu Liu, Ziyu Wan, Wei Huang, Yibing Song, Xintong Han, and Jing Liao. Pd-gan: Probabilistic diverse gan for image inpainting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9371–9381, 2021.

[78] 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.

[79] Yu Zeng, Zhe Lin, Huchuan Lu, and Vishal M Patel. Cr-fill: Generative image inpainting with auxiliary contextual reconstruction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14164–14173, 2021.

[80] Cairong Wang, Yiming Zhu, and Chun Yuan. Diverse image inpainting with normalizing flow. In *European Conference on Computer Vision*, pages 53–69. Springer, 2022.

[81] Wentao Wang, Li Niu, Jianfu Zhang, Xue Yang, and Liqing Zhang. Dual-path image inpainting with auxiliary gan inversion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11421–11430, 2022.

[82] Dongsik Yoon, Jeong-Gi Kwak, Yuanming Li, David Han, and Hanseok Ko. Difai: Diverse facial inpainting using stylegan inversion. In *2022 IEEE International Conference on Image Processing (ICIP)*, pages 1141–1145. IEEE, 2022.

[83] Chenjie Cao and Yanwei Fu. Learning a sketch tensor space for image inpainting of man-made scenes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14509–14518, 2021.

[84] 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.

[85] Mohamed Abbas Hedjazi and Yakup Genc. Efficient texture-aware multi-gan for image inpainting. *Knowledge-Based Systems*, 217:106789, 2021.

[86] Shuyi Qu, Zhenxing Niu, Kaizhu Huang, Jianke Zhu, Matan Protter, Gadi Zimerman, and Yinghui Xu. Structure first detail next: Image inpainting with pyramid generator. *arXiv preprint arXiv:2106.08905*, 2021.

[87] Soo Ye Kim, Kfir Aberman, Nori Kanazawa, Rahul Garg, Neal Wadhwa, Huiwen Chang, Nikhil Karnad, Munchurl Kim, and Orly Liba. Zoom-to-inpaint: Image inpainting with high-frequency details. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 477–487, 2022.

[88] 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*, pages 11358–11368, 2022.

[89] Weize Quan, Ruisong Zhang, Yong Zhang, Zhifeng Li, Jue Wang, and Dong-Ming Yan. Image inpainting with local and global refinement. *IEEE Transactions on Image Processing*, 31:2405–2420, 2022.

[90] Jiayin Cai, Changlin Li, Xin Tao, and Yu-Wing Tai. Image multi-inpainting via progressive generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 978–987, 2022.

[91] Xiaoguang Li, Qing Guo, Di Lin, Ping Li, Wei Feng, and Song Wang. Misf: Multi-level interactive siamese filtering for high-fidelity image inpainting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1869–1878, 2022.

[92] Yohei Yamashita, Kodai Shimosato, and Norimichi Ukita. Boundary-aware image inpainting with multiple auxiliary cues. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 619–629, 2022.

[93] Tengfei Wang, Hao Ouyang, and Qifeng Chen. Image inpainting with external-internal learning and monochromic bottleneck. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5120–5129, 2021.

[94] 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.

[95] Raymond A Yeh, Chen Chen, Teck Yian Lim, Alexander G Schwing, Mark Hasegawa-Johnson, and Minh N Do. Semantic image inpainting with deep generative models. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5485–5493, 2017.

[96] Chaohao Xie, Shaohui Liu, Chao Li, Ming-Ming Cheng, Wangmeng Zuo, Xiao Liu, Shilei Wen, and Errui Ding. Image inpainting with learnable bidirectional attention maps. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8858–8867, 2019.

[97] Yu-Zhe Su, Tsung-Jung Liu, Kuan-Hsien Liu, Hsin-Hua Liu, and Soo-Chang Pei. Image inpainting for random areas using dense context features. In *2019 IEEE International Conference on Image Processing (ICIP)*, pages 4679–4683. IEEE, 2019.

[98] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[99] Haofeng Li, Guanbin Li, Liang Lin, Hongchuan Yu, and Yizhou Yu. Context-aware semantic inpainting. *IEEE transactions on cybernetics*, 49(12):4398–4411, 2018.

[100] Ning Wang, Yipeng Zhang, and Lefei Zhang. Dynamic selection network for image inpainting. *IEEE Transactions on Image Processing*, 30:1784–1798, 2021.

[101] Xue Zhou, Tao Dai, Yong Jiang, and Shu-Tao Xia. Bishift-net for image inpainting. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2470–2474. IEEE, 2021.

[102] Min-cheol Sagong, Yong-goo Shin, Seung-wook Kim, Seung Park, and Sung-jea Ko. Pepsi: Fast image inpainting with parallel decoding network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11360–11368, 2019.

[103] Yong-Goo Shin, Min-Cheol Sagong, Yoon-Jae Yeo, Seung-Wook Kim, and Sung-Jea Ko. Pepsi++: Fast and lightweight network for image inpainting. *IEEE transactions on neural networks and learning systems*, 32(1):252–265, 2020.

[104] 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.

[105] Xin Ma, Xiaoqiang Zhou, Huaibo Huang, Zhenhua Chai, Xiaolin Wei, and Ran He. Free-form image inpainting via contrastive attention network. In *2020 25th*

*International Conference on Pattern Recognition (ICPR)*, pages 9242–9249. IEEE, 2021.

[106] Wentao Wang, Jianfu Zhang, Li Niu, Haoyu Ling, Xue Yang, and Liqing Zhang. Parallel multi-resolution fusion network for image inpainting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14559–14568, 2021.

[107] Chao Yang, Xin Lu, Zhe Lin, Eli Shechtman, Oliver Wang, and Hao Li. High-resolution image inpainting using multi-scale neural patch synthesis. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6721–6729, 2017.

[108] Ahmed Ben Saad, Youssef Tamaazousti, Josselin Kherroubi, and Alexis He. Where is the fake? patch-wise supervised gans for texture inpainting. In *2020 IEEE International Conference on Image Processing (ICIP)*, pages 568–572. IEEE, 2020.

[109] Siyuan Li, Lu Lu, Zhiqiang Zhang, Xin Cheng, Kepeng Xu, Wenxin Yu, Gang He, Jinjia Zhou, and Zhuo Yang. Interactive separation network for image inpainting. In *2020 IEEE International Conference on Image Processing (ICIP)*, pages 1008–1012. IEEE, 2020.

[110] Avisek Lahiri, Arnav Kumar Jain, Sanskar Agrawal, Pabitra Mitra, and Prabir Kumar Biswas. Prior guided gan based semantic inpainting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13696–13705, 2020.

[111] Lei Zhao, Qihang Mo, Sihuan Lin, Zhizhong Wang, Zhiwen Zuo, Haibo Chen, Wei Xing, and Dongming Lu. Uctgan: Diverse image inpainting based on unsupervised cross-space translation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5741–5750, 2020.

[112] Liang Liao, Jing Xiao, Zheng Wang, Chia-Wen Lin, and Shin'ichi Satoh. Uncertainty-aware semantic guidance and estimation for image inpainting. *IEEE Journal of Selected Topics in Signal Processing*, 15(2):310–323, 2020.

[113] Liang Liao, Jing Xiao, Zheng Wang, Chia-Wen Lin, and Shin'ichi Satoh. Image inpainting guided by coherence priors of semantics and textures. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6539–6548, 2021.

[114] Minyu Chen and Zhi Liu. Edbgan: Image inpainting via an edge-aware dual branch generative adversarial network. *IEEE Signal Processing Letters*, 28:842–846, 2021.

[115] Chong Han and Junli Wang. Face image inpainting with evolutionary generators. *IEEE Signal Processing Letters*, 28:190–193, 2021.

[116] 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.

[117] Zeyu Lu, Junjun Jiang, Junqin Huang, Gang Wu, and Xianming Liu. Glama: Joint spatial and frequency loss for general image inpainting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1301–1310, 2022.

[118] 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.

[119] Yingchen Yu, Fangneng Zhan, Shijian Lu, Jianxiong Pan, Feiying Ma, Xuansong Xie, and Chunyan Miao. Wavefill: A wavelet-based generation network for image inpainting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14114–14123, 2021.

[120] Maitreya Suin, Kuldeep Purohit, and AN Rajagopalan. Distillation-guided image inpainting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2481–2490, 2021.

[121] 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.

[122] Haitian Zheng, Zhe Lin, Jingwan Lu, Scott Cohen, Eli Shechtman, Connelly Barnes, Jianming Zhang, Ning Xu, Sohrab Amirghodsi, and Jiebo Luo. Image inpainting with cascaded modulation gan and object-aware training. In *European Conference on Computer Vision*, pages 277–296. Springer, 2022.

[123] Qiankun Liu, Zhentao Tan, Dongdong Chen, Qi Chu, Xiyang Dai, Yinpeng Chen, Mengchen Liu, Lu Yuan, and Nenghai Yu. Reduce information loss in transformers for pluralistic image inpainting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11347–11357, 2022.

[124] Yuliang Fan, Yue Zhou, Zonghao Yang, and Zhenyu Tong. Sltfill: Spatial and light transformer for multi-reference image inpainting. In *2022 IEEE International Conference on Image Processing (ICIP)*, pages 4173–4177. IEEE, 2022.

[125] Yuqian Zhou, Connelly Barnes, Eli Shechtman, and Sohrab Amirghodsi. Transfill: Reference-guided image inpainting by merging multiple color and spatial transformations. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2266–2276, 2021.

[126] Ziyu Wan, Jingbo Zhang, Dongdong Chen, and Jing Liao. High-fidelity pluralistic image completion with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4692–4701, 2021.

[127] 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.

[128] Nian Cai, Zhenghang Su, Zhineng Lin, Han Wang, Zhijing Yang, and Bingo Wing-Kuen Ling. Blind inpainting using the fully convolutional neural network. *The Visual Computer*, 33(2):249–261, 2017.

[129] Yang Liu, Jinshan Pan, and Zhixun Su. Deep blind image inpainting. In *International Conference on Intelligent Science and Big Data Engineering*, pages 128–141. Springer, 2019.

[130] Jimmy S Ren, Li Xu, Qiong Yan, and Wenxiu Sun. Shepard convolutional neural networks. *Advances in neural information processing systems*, 28, 2015.

[131] Junke Wang, Shaoxiang Chen, Zuxuan Wu, and Yu-Gang Jiang. Ft-tdr: Frequency-guided transformer and top-down refinement network for blind face inpainting. *IEEE Transactions on Multimedia*, 2022.

[132] Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *arXiv preprint arXiv:1406.2661*, 2014.

[133] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017.

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

[135] George Seif and Dimitrios Androutsos. Edge-based loss function for single image super-resolution. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1468–1472. IEEE, 2018.

[136] Yang Chen, Yu-Kun Lai, and Yong-Jin Liu. Cartoongan: Generative adversarial networks for photo cartoonization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[137] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[138] 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.

[139] Xuezhe Ma, Xiang Kong, Sinong Wang, Chunting Zhou, Jonathan May, Hao Ma, and Luke Zettlemoyer. Luna: Linear unified nested attention. *Advances in Neural Information Processing Systems*, 34:2441–2453, 2021.

[140] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[141] Jingyun Liang, Jiezhang Cao, Guolei Sun, Kai Zhang, Luc Van Gool, and Radu Timofte. Swinir: Image restoration using swin transformer. In *Proceedings of the IEEE ICCV*, pages 1833–1844, 2021.

[142] Zhendong Wang, Xiaodong Cun, Jianmin Bao, Wengang Zhou, Jianzhuang Liu, and Houqiang Li. Uformer: A general u-shaped transformer for image restoration. In *Proceedings of the IEEE CVPR*, pages 17683–17693, 2022.

[143] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *Proceedings of the IEEE ICCV*, pages 764–773, 2017.

[144] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[145] Ruoteng Li, Loong-Fah Cheong, and Robby T Tan. Heavy rain image restoration: Integrating physics model and conditional adversarial learning. In *Proceedings of*

*the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1633–1642, 2019.

[146] Joakim Johnander, Martin Danelljan, Emil Brissman, Fahad Shahbaz Khan, and Michael Felsberg. A generative appearance model for end-to-end video object segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8953–8962, 2019.

[147] Tong Chen, Haojie Liu, Zhan Ma, Qiu Shen, Xun Cao, and Yao Wang. End-to-end learnt image compression via non-local attention optimization and improved context modeling. *IEEE Transactions on Image Processing*, 30:3179–3191, 2021.

[148] Bryan Ning Xia, Yuan Gong, Yizhe Zhang, and Christian Poellabauer. Second-order non-local attention networks for person re-identification. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3760–3769, 2019.

[149] Yiqun Mei, Yuchen Fan, Yuqian Zhou, Lichao Huang, Thomas S Huang, and Honghui Shi. Image super-resolution with cross-scale non-local attention and exhaustive self-exemplars mining. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5690–5699, 2020.

[150] Jordi Pont-Tuset, Federico Perazzi, Sergi Caelles, Pablo Arbeláez, Alex Sorkine-Hornung, and Luc Van Gool. The 2017 davis challenge on video object segmentation. *arXiv preprint arXiv:1704.00675*, 2017.

[151] Zongyu Guo, Zhibo Chen, Tao Yu, Jiale Chen, and Sen Liu. Progressive image inpainting with full-resolution residual network. In *Proceedings of the 27th ACM International Conference on Multimedia*, pages 2496–2504, 2019.

[152] Jianrui Cai, Wangmeng Zuo, and Lei Zhang. Dark and bright channel prior embedded network for dynamic scene deblurring. *IEEE Transactions on Image Processing*, 29:6885–6897, 2020.

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

[154] Hongyu Liu, Bin Jiang, Yibing Song, Wei Huang, and Chao Yang. Rethinking image inpainting via a mutual encoder-decoder with feature equalizations. In *European Conference on Computer Vision*, pages 725–741. Springer, 2020.

[155] Long Chen, Hanwang Zhang, Jun Xiao, Liqiang Nie, Jian Shao, Wei Liu, and Tat-Seng Chua. Sca-cnn: Spatial and channel-wise attention in convolutional

networks for image captioning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5659–5667, 2017.

[156] Abhijit Guha Roy, Nassir Navab, and Christian Wachinger. Recalibrating fully convolutional networks with spatial and channel "squeeze and excitation" blocks. *IEEE transactions on medical imaging*, 38(2):540–549, 2018.

[157] Liang Liao, Jing Xiao, Zheng Wang, Chia-Wen Lin, and Shin'ichi Satoh. Guidance and evaluation: Semantic-aware image inpainting for mixed scenes. In *European Conference on Computer Vision*, pages 683–700. Springer, 2020.

[158] Xin Hong, Pengfei Xiong, Renhe Ji, and Haoqiang Fan. Deep fusion network for image completion. In *Proceedings of the 27th ACM international conference on multimedia*, pages 2033–2042, 2019.

[159] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.

[160] Yufan Zhuang, Zihan Wang, Fangbo Tao, and Jingbo Shang. Waveformer: Linear-time attention with forward and backward wavelet transform. *arXiv preprint arXiv:2210.01989*, 2022.

[161] Chao Li, Aojun Zhou, and Anbang Yao. Omni-dimensional dynamic convolution. *arXiv preprint arXiv:2209.07947*, 2022.

[162] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. In *International Conference on Learning Representations*, 2018.

[163] Xizhou Zhu, Han Hu, Stephen Lin, and Jifeng Dai. Deformable convnets v2: More deformable, better results. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9308–9316, 2019.

[164] Kui Jiang, Zhongyuan Wang, Peng Yi, Chen Chen, Baojin Huang, Yimin Luo, Jiayi Ma, and Junjun Jiang. Multi-scale progressive fusion network for single image deraining. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8346–8355, 2020.

[165] Kuldeep Purohit, Maitreya Suin, AN Rajagopalan, and Vishnu Naresh Boddeti. Spatially-adaptive image restoration using distortion-guided networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2309–2319, 2021.

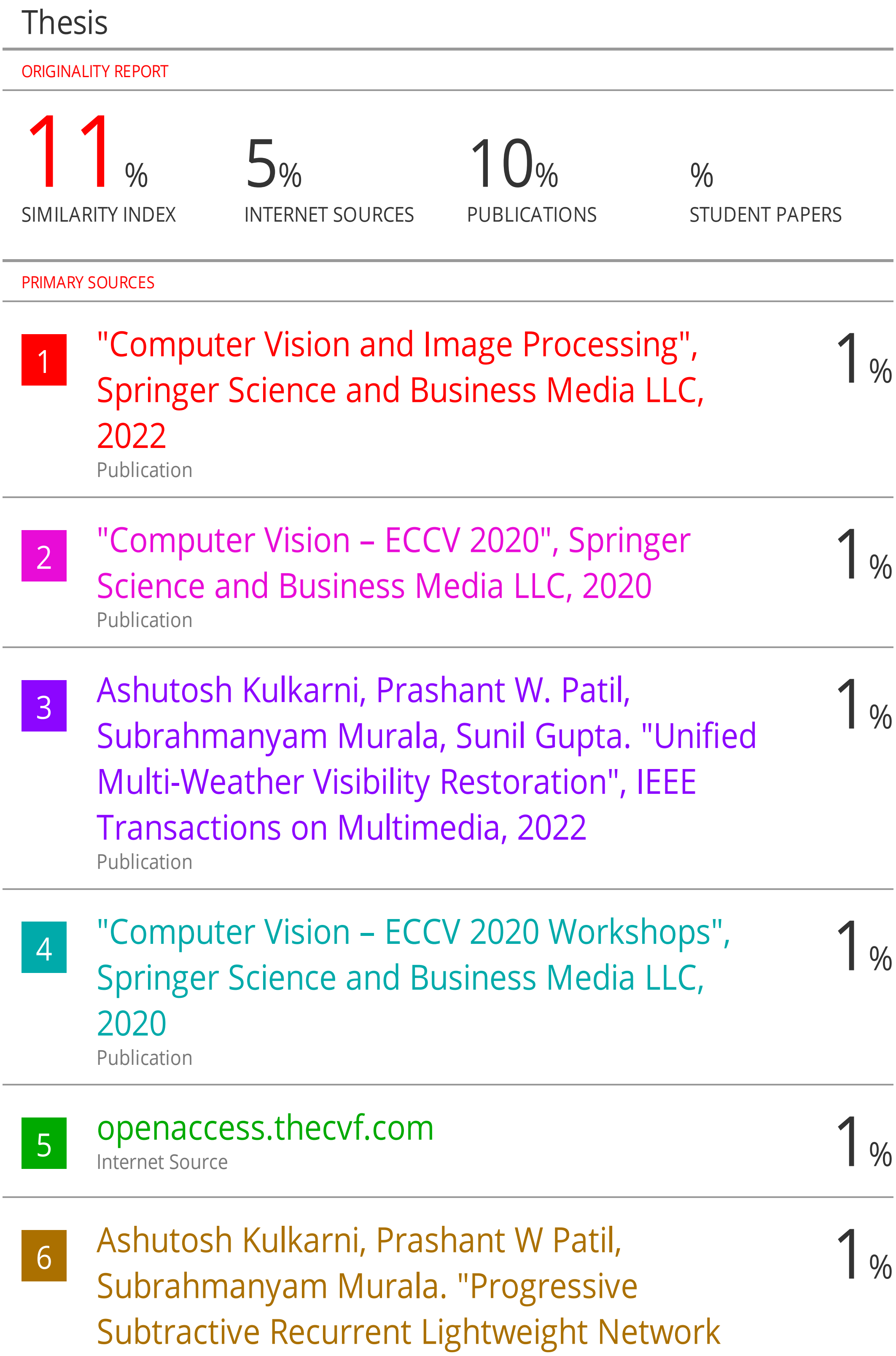
[166] Xueyang Fu, Jiabin Huang, Delu Zeng, Yue Huang, Xinghao Ding, and John Paisley. Removing rain from single images via a deep detail network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3855–3863, 2017.

[167] Wei-Ting Chen, Hao-Yu Fang, Jian-Jiun Ding, Cheng-Che Tsai, and Sy-Yen Kuo. Jstasr: Joint size and transparency-aware snow removal algorithm based on modified partial convolution and veiling effect removal. In *European Conference on Computer Vision*, pages 754–770. Springer, 2020.

[168] Kaihao Zhang, Rongqing Li, Yanjiang Yu, Wenhan Luo, and Changsheng Li. Deep dense multi-scale network for snow removal using semantic and depth priors. *IEEE Transactions on Image Processing*, 30:7419–7431, 2021.

[169] Ruoteng Li, Robby T Tan, and Loong-Fah Cheong. All in one bad weather removal using architectural search. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3175–3185, 2020.

# Thesis Plagiarism



Thesis

ORIGINALITY REPORT

| 11 % | 5% | 10% | % |
|---|---|---|---|
| SIMILARITY INDEX | INTERNET SOURCES | PUBLICATIONS | STUDENT PAPERS |

PRIMARY SOURCES

| | | |
|---|---|---|
| 1 | "Computer Vision and Image Processing", Springer Science and Business Media LLC, 2022<br>Publication | 1 % |
| 2 | "Computer Vision – ECCV 2020", Springer Science and Business Media LLC, 2020<br>Publication | 1 % |
| 3 | Ashutosh Kulkarni, Prashant W. Patil, Subrahmanyam Murala, Sunil Gupta. "Unified Multi-Weather Visibility Restoration", IEEE Transactions on Multimedia, 2022<br>Publication | 1 % |
| 4 | "Computer Vision – ECCV 2020 Workshops", Springer Science and Business Media LLC, 2020<br>Publication | 1 % |
| 5 | openaccess.thecvf.com<br>Internet Source | 1 % |
| 6 | Ashutosh Kulkarni, Prashant W Patil, Subrahmanyam Murala. "Progressive Subtractive Recurrent Lightweight Network | 1 % |