Image Inpainting Using Diffusion Models

Saizalpreet Kaur

Master of Science in Artificial Intelligence

from the

University of Surrey

School of Computer Science and Electrical and Electronic Engineering

Faculty of Engineering and Physical Sciences

University of Surrey

Guildford, Surrey, GU2 7XH, UK

16 September 2025

Supervised by: Dr. Marco Volino

DECLARATION OF ORIGINALITY

I confirm that the project dissertation I am submitting is entirely my own work and that any material used from other sources has been clearly identified and properly acknowledged and referenced. In submitting this final version of my report to the JISC anti-plagiarism software resource, I confirm that my work does not contravene the university regulations on plagiarism as described in the Student Handbook. In so doing I also acknowledge that I may be held to account for any particular instances of uncited work detected by the JISC anti-plagiarism software, or as may be found by the project examiner or project organiser. I also understand that if an allegation of plagiarism is upheld via an Academic Misconduct Hearing, then I may forfeit any credit for this module or a more severe penalty may be agreed.

Image Inpainting Using Diffusion Models

Author Name: Saizalpreet Kaur

Author Signature: Saizalpreet Kaur Date: 16/9/2025

Supervisor's name: Dr. Marco Volino

WORD COUNT

Number of Pages:48

Number of Words:10012

ABSTRACT

The field of image inpainting has recently been advanced by the success of diffusion models. These models offer superior training stability and output quality compared to previous generative methods such as Generative Adversarial Networks (GANs), which often suffer from mode collapse and unstable training dynamics. This dissertation investigates the effective adaptation of pre-trained, unconditional diffusion models for the image inpainting task. The research focuses on optimizing the trade-off between image fidelity, inference speed, and computational cost. The core methodology involves fine-tuning a U-Net-based diffusion model on the CelebA-HQ and Places365 datasets using a dual-conditioning strategy. Architecturally, the model's input layer is modified to accept nine channels, accommodating the RGB image, the binary mask, and the masked image as direct inputs. Algorithmically, a crucial noise injection mechanism is employed during sampling to continuously re-insert the ground truth of known regions at each denoising step. This technique preserves image integrity and prevents the content drift that otherwise leads to catastrophic failure.

A series of systematic experiments were conducted to determine the optimal model configuration. This included a comparison of different sampling techniques (DDPM vs. DDIM), an evaluation of multiple noise schedulers (Linear, Cosine, Quadratic), and an exploration of parameter-efficient fine-tuning with LoRA. The findings demonstrate that DDIM sampling with 100 steps, paired with a model fine-tuned using a Cosine noise scheduler, achieves the best performance. This configuration attained a Fréchet Inception Distance (FID) score of 3.24 on the CelebA-HQ test set, while drastically reducing the average sampling time from 33.41 seconds to just 3.42 seconds. Furthermore, experiments confirmed that a domain-specific pre-trained model (FFHQ) yields superior results for face inpainting compared to a general-purpose model (ImageNet). The primary contribution of this work is a robust and efficient framework for adapting unconditional diffusion models for high-quality inpainting, providing a clear path for practical applications that demand both realism and speed.

Keywords: Diffusion Models, Image Inpainting, Noise Injection

CONTENTS

LIST OF FIGURES

1 INTRODUCTION

Image Inpainting is a task of filling in missing or damaged regions of an incomplete image. This process has evolved a lot from being used in art to now being a computer vision problem. The goal is not a mere pixel replacement but to generate visibly plausible images that are semantically consistent with the surrounding context. Given an Input image I, and a missing *hole*, also known as mask M, the goal is to generate a completed image I' where the reconstructed areas are visually indistinguishable from the surrounding areas. Image inpainting has several applications ranging from restoration of damaged historical photographs and artworks, to Object removal, image manipulation, rendering, and even medical imaging.

1.1 Background and Context

Early approaches cite survey to this problem were straightforward. Traditional Diffusion-based Techniques would *diffuse* pixel values from the boundary of the hole towards its insides. These worked well for small scratches but created blurry results when filling larger holes. Another traditional method was a patch-based technique. This would search the image for a similar-looking patch and copy it into the space. This was better for textures but often failed to make the overall image look coherent. The main weakness of these traditional methods was their lack of understanding of semantics and coherence. They could manipulate pixels, but did not comprehend the actual content of the image.

This field changed dramatically with the introduction of deep learning. A powerful technology called Generative Adversarial Networks (GANs) became popular for image inpainting tasks. GANs use two competing neural networks: a ”generator” that creates the new image content and a ”discriminator” that judges whether the result looks real. This competition pushes the generator to produce highly realistic images. Researchers developed many improvements, such as using special convolutions and attention mechanisms, which allowed GANs to handle complex images and irregular shapes. The main limitations of using GANs were the unstable training processes. Another major challenge was mode collapse.

Recently, Diffusion Models have emerged as a powerful solution. Diffusion models work differently from GANs. They start with an image of pure noise and gradually refine it over many steps until a clean, coherent picture appears. This step-by-step process is much more stable and easier

to train than the adversarial process of GANs. Because of their stability and the high quality of the images they produce, diffusion models offer a promising new direction for solving the image inpainting challenge. Diffusion Models have been noted to outperform GANs for image synthesis [12]

1.2 Objectives

This research explores how to best use pre-trained and unconditional diffusion models for the task of image inpainting. It focuses on finding a balance between generated image quality, sampling time, and computational costs. The main goals of this work are:

- To condition an unconditional diffusion model to inpainting by fine-tuning it to a masked dataset.
- To compare different sampling methods (DDPM and DDIM) to speed up the inference
- To experiment with different noise schedulers for better generated quality.
- Speed up the Sampling process while maintaining the quality of inpainted images.

A pre-trained diffusion model was conditioned on masks to perform image inpainting. A noise injection of ground truth to the known pixels was used ot preserve them and prevent them from drifting away. Two different Sampling Techniques, namely -DDPM (baseline) and DDIM, were employed, trying to speed up the generation process. Three different noise schedulers were tried while fine-tuning, which further decreased the computation time. Basic LoRA fine-tuning for some layers was attempted in order to speed up the training process as well.

1.3 Achievements

This research makes several important contributions to the practical use of diffusion models for image inpainting. The findings offer a clear path for optimizing these models to get high-quality results while minimizing computational costs.

This work clearly shows that DDIM sampling paired with a model trained using Cosine Scheduler gives the best results while also reducing the time dramatically (about 29 seconds less when compared to DDPM sampling). A Frechet Inception Distance (FID) score of 3.24 was achieved. Finally, a core achievement is the development of a robust method for adapting pre-trained models for inpainting. This method uses a dual conditioning strategy. The model's architecture was

first changed to accept the masked image and the mask as direct inputs. Then, during the image generation process, continuous re-insertion of the known, unmasked parts of the original image at each step was performed. This clever technique prevents the model from altering parts of the picture that should remain unchanged, ensuring a seamless final result.

1.4 Overview of Dissertation

This project is organised into 5 more chapters, which are covered in the report.

Chapter 2:Literature Review This chapter covers significant traditional and deep-learning-based image synthesising models and how they are adapted to image inpainting. Mathematical Principles behind the methods are also covered.

Chapter 3: Methodology This chapter explains the technical framework that was used to carry out this project. It covers the working of the model architecture, the changes made to adapt an unconditional diffusion model, the training/fine-tuning, and the sampling process. The metrics used to judge the work are also described in this section.

Chapter 4: Experimentation and Results This chapter covers all the experiments that were carried out to meet the objectives of the project. It presents the findings of this research.

Chapter 5: Discussion This chapter presents the resulting images and discusses the performance of the model. It provides instances where the model performed badly and where it performed impeccably.

Chapter 6: Conclusions and Future Work This chapter summarizes the key outcomes of the project. It compares the achievements against the initially set objectives and suggests future directions to continue work in this area.

2 LITERATURE REVIEW

Image inpainting is defined as the task of filling in missing or masked regions of an image using contextual information from surrounding areas. The term ”inpainting” was first introduced by [8] as mentioned in [16], who defined it as ”the technique of modifying an image in an undetectable form.”

Given an input image I with a binary mask M denoting the target region, the objective is to generate a completed image I' where the reconstructed areas are visually indistinguishable from authentic content while maintaining semantic consistency and structural coherence with the original scene's contextual meaning. There are several approaches to image inpainting; **Traditional methods** include diffusion-based methods and exemplar-based or patch-based techniques. Following that, the **Deep-learning** approaches are: Encoder-decoder Architecture, Generative Adversarial Networks, and the Diffusion Models. The working of these techniques is discussed in the respective sections below.

2.1 Traditional Image Inpainting Methods

This section covers the working of diffusion-based and patch-based methods.

2.1.1 Diffusion-based Methods

Traditional diffusion-based inpainting methods, introduced by [8], treat the missing region of an image as a boundary value problem, where the pixel intensities gradually propagate from known areas to unknown or missing areas. This propagation occurs through an iterative diffusion process, governed by partial differential equations (PDEs). These types of methods allow the visual information to flow smoothly from the surrounding context into the masked/unknown areas, guided by the edge and contour information.

In Image Inpainting by Bertalmio *et al.* [8], which is considered to be the first diffusion-based approach, PDEs were used to propagate information along isophotes (a line/ curve that joins points of equal intensity) from the boundary regions to the holes. It requires user-specified masks. The algorithm used was the following.

$$I^{n+1}(i,j) = I^{n}(i,j) + \Delta t I_t^n(i,j), \forall (i,j) \in \Omega \tag{2.1}$$

Where *n* denotes the inpainting time, (i,j) are the coordinates of the pixels in image I. Δt being the rate of improvement and $I_t^n(i,j)$ the improved version of $I^n(i,j)$. Ω is the region to be painted. The algorithm propagates the isophoto directions as perpendicular to the gradient directions and then uses a Laplacian to estimate local colour smoothness variations. This smoothness variation information is then propagated to the holes along the isophote directions.

This method works well for smaller and simpler masks, but struggles with complex textures, resulting in blurry artefacts.

Following this, Beralmio *et al.* used a method by treating the image intensity as a viscous fluid, governed by the Navier-Stokes equation [9] rather than just geometry. The isophotes are analogous to the streamlines of the fluid flow.

Few other approaches suggested formulating image inpainting as a variational problem where the image is seen as a function of bounded variation. The Total Variation (TV) image model, initially proposed for denoising, was also used for image inpainting [16]. This helped with better edge recovery. To enhance edge connectivity, Chan and Shen extended TV inpainting by introducing energy functionals, considering curve structures via [10] and [33].

Figure 2.1: Blurring effects of Diffusion- based method [16]

To summarise, diffusion-based models utilise information from boundaries to fill in the gaps. They are better suited for restoring small masks. For larger holes, they often create blurry regions.

2.1.2 Patch-based methods

Patch-based methods or Exemplar-based methods operate on the principle that natural images inherently contain redundant information, including similar textures, repeated patterns, and similar structural elements that can be borrowed intelligently and adapted to reconstruct the holes. In simple words, they copy and paste a 'best-fit' patch from the surroundings into the hole. It takes inspiration from texture synthesis [14].

One of the first methods of patch-based inpainting [11] employs an iterative process that copies and pastes patches into the holes. It employs a 'best-first' approach, guided by a priority func-

Figure 2.2: Patch-based inpainting [7]

tion that is a product of confidence and a data term. The confidence term keeps measuring the amount of known information around a patch, while the data term keeps track of linear structures (Edges, contours). The algorithm begins by calculating priorities for unfilled pixels based on their structural importance (proximity to edges) and information reliability (the amount of surrounding known content). Iteratively selects the highest-priority pixel, extracts a patch centred on that location, and searches the source region for the most similar patch using similarity metrics. The best-matching patch is copied to fill the missing portion, and confidence values are updated for newly completed areas. The process repeats until all missing regions are reconstructed.

The initial patch-based methods required exhaustive research through all possible patch locations to find the best matches, which involved a significant amount of computation and time. To solve this, PatchMatch [7] was proposed, using Nearest Neighbour Fields (NNF). It exploits natural image coherence by random initialisation, propagation, and random search (assumes that adjacent patches are likely to be optimal matches). Patch-based methods are more effective at filling large, textured areas compared to diffusion models [16]. But the patch-by-patch approach often fails to ensure global image coherence.

Traditional methods attempt to fill in the holes by either diffusing the boundary smoothly or copying and pasting the best-matching patches. When used for inpainting, these methods lack a semantic understanding of the image because they rely solely on low-level information. They are observed to fail for large, arbitrary masks and take a longer computational time. This led to the usage of deep-learning-based methods for image inpainting.

2.2 Deep-learning Methods

The traditional methods were effective for small-scale inpainting but faced significant limitations when working with larger masks, failing to capture the semantic context of the images. The tran-

sition to deep-learning-based methods marked a paradigm shift. The new techniques were more robust, data-drive, and could learn complex image priors from large datasets. Leveraging robust architectures such as convolutional neural networks (CNNs), generative adversarial networks (GANs), and diffusion models, these techniques excel at capturing both local textures and global contextual information. Trained on large-scale datasets, deep learning-based inpainting models can generate visually plausible content for complex scenes, large missing areas, and diverse image types. Early deep learning-based techniques focused on using Convolutional Neural Networks due to their strength in processing spatial hierarchies.

The evolution of deep learning for inpainting can be broadly categorised into four major paradigms: convolutional neural networks (CNNs), generative adversarial networks (GANs), transformer-based approaches, and, most recently, diffusion models.

2.2.1 CNN and GAN-Based Approaches

The early works utilise CNN [19] as a denoising model, which can be extended to image inpainting. However, this model only worked for grayscale images. The first successful application of GAN-based deep learning to inpainting was introduced by Pathak et al., who proposed context encoders [28]. Their encoder-decoder architecture demonstrated that neural networks could learn to predict missing content by understanding the global image context. The context encoder approach employed a bottleneck architecture that forced the network to learn compressed representations of masked images. The encoder processed the input through successive convolutional layers, reducing spatial dimensions while increasing feature depth. This bottleneck representation compelled the model to capture high-level semantic information necessary for generating believable content. The decoder then reconstructed the missing regions using transposed convolutions. A significant innovation was the integration of adversarial loss alongside pixel-wise reconstruction loss. The adversarial component addressed the inherent multi-modal nature of inpainting by encouraging the generator to produce sharp, realistic content rather than blurred averages of possible solutions. However, context encoders were constrained to small, square masks and low-resolution images, limiting their practical applicability.

Generative Adversarial Networks(GANs) revolutionised image inpainting by introducing adversarial training that pushed generated content toward photorealism. GANs constitute a fundamental framework in machine learning for synthesising data through adversarial training between

two neural networks. The methodology, established by [15], addresses the problem of learning complex probability distributions by formulating generative modelling as a competitive optimisation process.

The GAN architecture consists of two distinct neural networks operating in opposition. The generator network G transforms random noise vectors z sampled from a prior distribution $p_z(z)$ into synthetic data samples $G(z)$. The discriminator network D functions as a binary classifier that evaluates input samples and outputs probabilities indicating whether each sample originates from the true data distribution or the generator.

The training objective establishes a minimax game between the two networks through the following formulation:

$$\min_G \max_D V(D,G) = \mathbb{E}_{x\sim p_{\text{data}}}[\log D(x)] + \mathbb{E}_{z\sim p_z}[\log(1 - D(G(z)))]$$

The discriminator seeks to maximise this objective by correctly classifying real samples while identifying generated samples as artificial. The generator attempts to minimise the same objective, thereby improving its capacity to produce samples that the discriminator classifies as authentic.

Training proceeds through alternating optimisation phases. During discriminator training, the network learns to distinguish between real data samples and current generator outputs. During generator training, the network adjusts its parameters to produce samples that better deceive the discriminator. This iterative process continues until convergence is achieved.

The theoretical optimum occurs when the generator distribution p_g matches the true data distribution p_{data}. At this equilibrium, the optimal discriminator outputs probability $\frac{1}{2}$ for all samples, indicating an inability to distinguish between real and generated data. This state represents a Nash equilibrium where neither network can unilaterally improve its performance.

Building upon this foundation, Iizuka et al. advanced existing inpainting methods through a dual-discriminator approach [18]. Their method addressed the limitations of single discriminator architectures by employing both global and local discriminators to ensure comprehensive image quality assessment. The global discriminator evaluated the entire image, ensuring semantic coherence and contextual consistency across the complete scene. Simultaneously, the local dis-

criminator focused specifically on the inpainted region and its boundaries, verifying fine-grained detail quality and seamless integration with surrounding pixels. This dual evaluation strategy effectively addressed both global semantic plausibility and local visual realism. Additionally, their work introduced dilated convolutions to the inpainting domain, expanding receptive fields without increasing computational complexity. The dilated convolution architecture enabled the network to capture multi-scale contextual information essential for understanding both local textures and global structures. Despite these advances, the method still struggled with irregular mask shapes and required post-processing techniques such as Poisson blending to achieve satisfactory boundary integration.

A significant breakthrough came with the introduction of partial convolutions by Liu et al. [22]. Traditional convolution operations treated all input pixels uniformly, including invalid masked regions, which introduced artefacts and colour bleeding effects that compromise inpainting quality. Partial convolutions fundamentally reconceptualised the convolution operation for masked inputs by operating exclusively on valid pixels while automatically updating mask information throughout the network. The mathematical formulation ensures that convolution outputs depend only on unmasked input values. The mask update mechanism shrinks invalid regions as information propagates through the network, eventually producing complete feature maps at deeper layers. This innovation enabled networks to handle arbitrary hole shapes effectively while eliminating artefacts caused by treating masked regions as valid inputs. The automatic mask updating mechanism proved particularly valuable for free-form inpainting tasks where hole boundaries were irregular and complex.

Yu et al. introduced contextual attention, representing a paradigm shift in how inpainting networks utilise spatial information [38]. Their attention mechanism computed feature-level similarities between patches in unknown regions and patches in known regions, enabling explicit borrowing of relevant textures and structures from distant image areas. The contextual attention module operated by extracting feature patches from both known and unknown regions, computing cosine similarities between them, and using these similarities as attention weights for feature aggregation. This approach enabled the network to identify and copy repetitive patterns from known regions to fill missing areas. The method proved particularly effective for images containing repetitive textures, architectural elements, or periodic patterns.

The same authors subsequently introduced gated convolutions [39], which provided learnable mechanisms for feature selection and spatial attention. Unlike partial convolutions that used fixed, rule-based mask updates, gated convolutions employed learnable gates that determined feature relevance at each spatial location. The gating mechanism allowed the network to dynamically control information flow based on both spatial context and learned feature representations. Gated convolutions demonstrated superior performance across various image types and mask configurations, establishing learnable gating as a fundamental component of modern inpainting architectures.

Recognising the challenge of modelling both global structure and local detail within single-scale architectures, researchers developed pyramid-based approaches that explicitly separated these tasks across multiple resolution levels. Zeng et al. proposed the Pyramid-context Encoder Network (PEN-Net) [40], which constructed feature pyramids to capture multi-scale contextual information systematically. The architecture employed a pyramid encoder that extracted features at multiple scales, followed by an attention transfer network that propagated high-level semantic information to lower-level features. This design addressed the fundamental conflict between global context modelling, which benefits from large receptive fields, and local detail generation, which requires fine-grained spatial resolution. Multi-scale pyramid approaches demonstrated particular effectiveness for large hole inpainting, where the ratio of missing to known regions challenges traditional single-scale methods.

One of the recent and significant approaches is LaMa (Large Mask Inpainting) [36], which incorporated Fast Fourier Convolutions (FFC) to address the challenge of large receptive fields in inpainting tasks. Traditional convolutional operations suffer from limited receptive field growth, requiring deep networks to capture long-range dependencies essential for large hole completion. FFC modules process input features in both spatial and frequency domains, enabling global context modelling with computational efficiency comparable to standard convolutions. This approach proves particularly effective for inpainting large masks where understanding global image structure is crucial.

The GAN era established adversarial training as essential for photorealistic inpainting and introduced attention mechanisms that became fundamental building blocks for subsequent approaches. However, GAN-based methods often suffer from training instability, limiting their reliability and diversity. These limitations, combined with difficulties in achieving consistent convergence and the challenge of balancing generator and discriminator training, motivated the

exploration of alternative generative frameworks that could provide more stable training dynamics while maintaining high-quality output generation.

2.2.2 Transformer-Based Approaches

The success of transformers in natural language processing motivated their adaptation for image inpainting [37]. Transformers' ability to model long-range dependencies seemed naturally suited for inpainting, where filling missing regions often requires understanding relationships between distant image parts. A few significant Transformer-based approaches are mentioned below:

MAT (Mask-Aware Transformer) [21] is a hybrid image inpainting model that combines convolutional encoding with a transformer backbone tailored for large missing regions. It introduces a Multi-Head Contextual Attention (MCA) mechanism that only attends to valid tokens, using a dynamic mask to progressively expand context during inference. Unlike standard transformers, MAT removes layer norms and residuals, improving stability in sparse inputs.

Dong et al. developed Incremental Transformer structures specifically for inpainting [13]. Their approach used masked positional encoding to handle irregular hole shapes and demonstrated that transformers could capture structural relationships more effectively than CNNs for certain inpainting scenarios. However, the computational cost of full self-attention limited practical applications to relatively small images.

Liu et al. addressed the computational limitations by developing efficient transformer variants that reduced information loss through non-quantised processing [23]. Their P-VQVAE approach showed that transformers could achieve competitive inpainting quality while being more computationally tractable than naive applications of standard transformer architectures.

Zheng et al. demonstrated that transformers could effectively bridge global context interactions for high-fidelity completion [41]. Their work showed particular strength in handling images with complex global structure, where understanding distant relationships was crucial for plausible completion.

While transformer-based models highlight the importance of explicit long-range modelling, they often incur higher computational costs and have not consistently outperformed attention-augmented CNNs or GANs. Nonetheless, their influence is evident in newer hybrid architectures that aim to balance global context modelling with computational efficiency.

2.2.3 Diffusion Model Approaches

Traditional GAN training requires careful balancing between generator and discriminator networks, often resulting in unstable dynamics where one network dominates the other, leading to poor convergence or complete training failure [6]. Mode collapse, where the generator produces limited variations of outputs despite diverse training data, represents another fundamental challenge in GAN-based inpainting, particularly problematic for tasks requiring diverse plausible completions [26]. Additionally, GAN training is notoriously sensitive to hyperparameter choices, network architectures, and initialisation strategies, making reproducible results difficult to achieve [24].

Diffusion models address these fundamental limitations through a fundamentally different generative paradigm based on gradually denoising random noise through a learned reverse diffusion process [34]. Unlike adversarial minimax optimisation of GANs, diffusion models employ a stable training objective based on variational inference. This significantly improves training. [20].

Diffusion models operate on a two-part generative process inspired by non-equilibrium thermodynamics. The forward diffusion process is a fixed Markov chain that systematically adds Gaussian noise to an image, over many timesteps, until the original information is completely lost and the sample resembles pure noise. The reverse process is the generative core, where a neural network is trained to learn the inverse of this diffusion. By learning to predict and remove the noise at each step, the model can start with a random noise sample and iteratively de-noise it to produce a new, high-fidelity data point. This iterative, learnt denoising procedure effectively transforms a simple, known noise distribution into a semantic image.

2.2.3.1 Forward Diffusion Process

The forward diffusion process defines a fixed Markov chain (each step depends only on the previous one), which systematically corrupts the data by adding Gaussian noise over T steps. For an original image x_0, the forward process is denoted by $q(x_1|x_0)$.

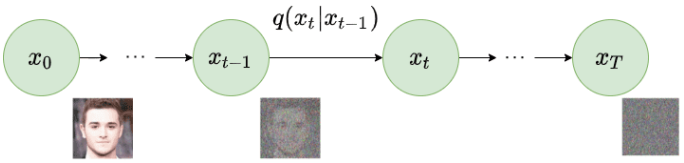

Figure 2.3: Forward Diffusion [32]

Mathematically, the forward process is defined as follows.

$$q(x_{1:T}|x_0) := \prod_{t=1}^{T} q(x_t|x_{t-1})$$

where

$$q(x_t|x_{t-1}) = \mathcal{N}(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t I)$$

This is a normal distribution with mean $\mu_t = \sqrt{1-\beta_t}x_{t-1}$ and variance $\beta_t I$. The addition of noise at each step is regulated by the variance schedule $\beta_t \in (0, 1)$. In the original paper, a linear scheduler was used, increasing from $\beta_t = 10^{-4}$ to $\beta_t = 0.002$ [17]. A reparametrisation allows direct sampling at any time step without iterating through all previous steps. $1 - \beta_t$ is defined as α_t with $\overline{\alpha_t} = \prod_{s=1}^{t} \alpha_s$, which gives us :

$$q(x_t|x_0) = \mathcal{N}(x_t; \sqrt{\overline{\alpha_t}}x_0, (1-\overline{\alpha_t})I) \tag{2.2}$$

This gives us $x_t = \sqrt{\overline{\alpha_t}}x_0 + \sqrt{1-\overline{\alpha_t}}\epsilon$ where $\epsilon \sim \mathcal{N}(0, I)$ represents standard Gaussian noise.

2.2.3.2 Reverse Diffusion Process

The reverse diffusion process learns to invert the noise added during the forward process. This process is parametrised by a neural network, usually a U-Net. We need the reverse distribution $q(x_{t-1}|x_t)$, which is approximated using a parametrised neural network p_θ. During this process, the U-net is trained to de-noise the data after the noise added at each step has been recognised.

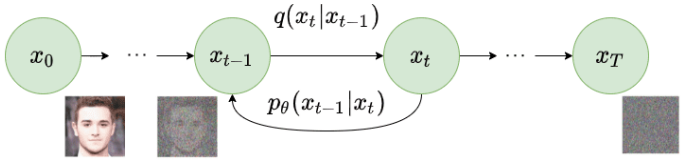

Figure 2.4: Reverse Diffusion

The process can be described as:

$$p_\theta(x_{t-1}|x_t) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \Sigma_\theta(x_t, t)) \tag{2.3}$$

It is parametrised as

$$p_\theta(x_{0:T}) = p(x_T) \prod_{t=1}^{T} p_\theta(x_{t-1}|x_t)$$

where $p(x_T) = \mathcal{N}(x_T; 0, I)$ is the prior distribution. We can choose p_θ to be Gaussian and parametrize the mean and variance. By conditioning the model on timesteps, it learns to predict the Gaussian parameters(mean and covariance) for each timestep.

2.2.3.3 Training

To calculate $p_\theta(x_0)$, we would need to keep track of $(t-1)$ variables, which is close to impossible. To solve this problem, a variational lower bound is used. So, diffusion models optimise the negative log-likelihood through a variational lower bound (ELBO). Starting from $-log p_\theta(x_0)$, we get to

$$\log p_\theta(x_0) \geq \mathbb{E}q(x_{1:T}|x_0) \left[\log \frac{p_\theta(x_{0:T})}{q(x_{1:T}|x_0)} \right]$$

Finally the loss term comes out to be [32]:

$$L_t = \mathbb{E}_{\mathbf{x}_0,t,\boldsymbol{\epsilon}} \left[\frac{\beta_t^2}{2\alpha_t(1-\bar{\alpha}_t)||\boldsymbol{\Sigma}_\theta||^2} ||\boldsymbol{\epsilon}_t - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}_t, t)||^2 \right]$$

This was simplified in the original paper because the simplified training objective resulted in stable training and better sampling. The simplified training objective is:

$$L_{\text{simple}}(\theta) := \mathbb{E}_{t,\mathbf{x}_0,\epsilon} \left[||\epsilon - \epsilon_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\epsilon, t)||^2 \right]$$

This loss function simplifies the complex problem to a simple mean squared error (MSE) loss on noise prediction. In the paper [17], the variance was kept fixed and the model only learnt the mean. The algorithms for training and sampling as given in the original paper are given below:

Algorithm 1 Training

1: **repeat**
2: $\mathbf{x}_0 \sim q(\mathbf{x}_0)$
3: $t \sim \text{Uniform}(\{1, \dots, T\})$
4: $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
5: Take gradient descent step on
$\nabla_\theta \left\| \boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t) \right\|^2$
6: **until** converged

Algorithm 2 Sampling

1: $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
2: **for** $t = T, \dots, 1$ **do**
3: $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ if $t > 1$, else $\mathbf{z} = \mathbf{0}$
4: $\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}} \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t) \right) + \sigma_t \mathbf{z}$
5: **end for**
6: **return** $\mathbf{x}_0$

[16]

Figure 2.5: Algorithm [17]

2.2.3.4 U-net

The U-Net's primary function within the reverse diffusion process is to act as a highly sophisticated **noise predictor**. At each step of the generation process, the model starts with a noisy image, x_t, and needs to estimate the Gaussian noise, ϵ, that was added to create it. The U-Net is the function, ϵ_θ, that performs this estimation. It takes two primary inputs: the noisy image x_t and the current timestep t. The timestep is a crucial piece of information because the model must behave differently depending on the level of noise; a heavily corrupted image early in the process requires a different approach than a mostly clean image near the end. The U-Net's output is not a cleaner image, but a tensor representing the predicted noise. This predicted noise is then subtracted from the input image to produce a slightly less noisy version, x_{t-1}, moving one step closer to the final, clean image [17].

The choice of the U-Net architecture is deliberate and critical to the success of diffusion models. Its design is uniquely suited for image-to-image tasks where spatial information is paramount [30]. The architecture consists of two main parts:

- Contracting Path (Encoder): This part of the network acts like a traditional convolutional network. It progressively downsamples the input image, applying a series of convolutions and pooling operations. This process allows the network to capture contextual, semantic information from the image at a low spatial resolution.
- Expansive Path (Decoder): This part symmetrically upsamples the feature maps from the encoder's low-resolution bottleneck. It learns to reconstruct a high-resolution output from the compressed contextual information.

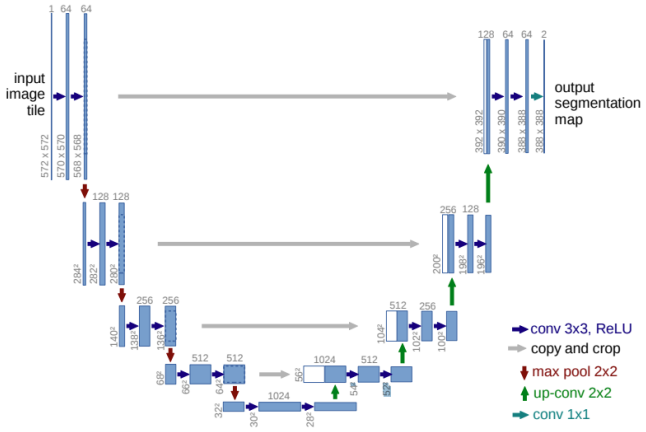

Figure 2.6: U-net Architecture [30]

The defining feature of the U-Net, however, is its use of **skip connections**. These connections create a direct link between the layers of the encoder and the corresponding layers of the decoder. As the encoder downsamples the image, it loses fine-grained spatial information (like precise edges and textures). The skip connections reintroduce this high-resolution feature information from the encoder directly into the decoder. For noise prediction, this is indispensable, as it allows the network to make a highly accurate, pixel-perfect prediction of the noise, ensuring that crucial details are not lost during the denoising process [30].

To further enhance its capabilities for diffusion, the standard U-Net is augmented with several key modifications [17]. **Timestep embeddings**, derived from the timestep t, are injected into the network's intermediate layers, conditioning the entire network on the specific noise level it is facing. Furthermore, **attention mechanisms**, originating from transformer architectures [37], are integrated into the U-Net's layers. **Self-attention** allows the model to weigh the importance of different regions of the image relative to one another, capturing long-range dependencies and improving the global coherence of the final image.

Advancements and Variants

Following the foundational work on DDPMs [17], several key advancements were proposed to address limitations such as slow sampling speed, improve sample quality, and enable control over the generation process.

2.2.3.5 Denoising Diffusion Implicit Models (DDIM)

One of the primary drawbacks of DDPMs is their slow inference time, which is due to the Markovian nature of the generative process that requires thousands of steps. To address this, Denoising Diffusion Implicit Models (DDIM) were introduced as a more efficient sampling method [35]. DDIMs achieve this by defining a non-Markovian forward process, which enables a deterministic reverse process. This allows for a flexible sampling trajectory that can skip many steps.

The key insight is to first predict the original image x_0 from the noisy image x_t, and then use that prediction to deterministically derive the image at the previous step, x_{t-1}. The prediction of the original image is given by:

$$\hat{x}_0 = \frac{1}{\sqrt{\bar{\alpha}_t}}(x_t - \sqrt{1-\bar{\alpha}_t}\epsilon_\theta(x_t, t)) \tag{2.4}$$

where $\epsilon_\theta(x_t, t)$ is the noise predicted by the U-Net. Using this, the next step x_{t-1} can be calculated deterministically:

$$x_{t-1} = \sqrt{\bar{\alpha}_{t-1}}\hat{x}_0 + \sqrt{1-\bar{\alpha}_{t-1}}\epsilon_\theta(x_t, t) \tag{2.5}$$

This formulation allows the model to take much larger steps during inference, reducing the number of required sampling steps from thousands to as few as 20-50, which drastically accelerates image generation without needing to retrain the underlying model [35].

2.2.3.6 Improved Diffusion Models

To enhance the performance and sample quality of the original DDPM framework, several key improvements were proposed [27]. These addressed both the parameterisation of the reverse process and the noise schedule of the forward process.

- **Learned Variance:** The original DDPM framework kept the variance of the reverse process, $\Sigma_\theta(x_t, t)$, fixed to a constant. The improved models demonstrated that learning this variance as a parameter of the network leads to a significant increase in sample quality and log-likelihood. Instead of predicting a fixed value, the model predicts a vector v that interpolates between the lower bound β_t and upper bound $\tilde{\beta}_t$ for the variance.

- **Cosine Noise Schedule:** The linear noise schedule used in DDPMs was found to be suboptimal, as it adds noise too aggressively at the beginning of the forward process. A cosine-based variance schedule was introduced, which adds noise more slowly and smoothly. The

schedule is defined by $\bar{\alpha}_t$:

$$\bar{\alpha}_t = \frac{f(t)}{f(0)}, \quad \text{where} \quad f(t) = \cos\left(\left(\frac{t}{T} + s\right)/(1+s) \cdot \frac{\pi}{2}\right)^2 \tag{2.6}$$

Here, T is the total number of timesteps and s is a small offset to prevent $\bar{\alpha}_t$ from being exactly zero near $t = T$. This schedule was found to significantly improve results, especially for higher-resolution images [27].

2.2.3.7 Guided Diffusion

A breakthrough for controlling the output of diffusion models came with the development of **guidance techniques**. To avoid the complexity of requiring an external classifier model, a powerful and widely adopted method known as **classifier-free guidance** was introduced [12].

This technique uses a single U-Net trained to handle both conditional and unconditional generation. This is achieved by feeding the model a class label, y, during training but randomly replacing it with a null label, $\emptyset$, a fraction of the time.

At inference, the U-Net makes two predictions at each step: one conditional, $\epsilon_\theta(x_t, y)$, and one unconditional, $\epsilon_\theta(x_t, \emptyset)$. The final noise prediction is an extrapolation from the unconditional estimate in the direction of the conditional one:

$$\hat{\epsilon}_\theta(x_t, y) = \epsilon_\theta(x_t, \emptyset) + s \cdot (\epsilon_\theta(x_t, y) - \epsilon_\theta(x_t, \emptyset)) \tag{2.7}$$

The guidance scale s controls how strongly the output should adhere to the class y. A scale of $s = 1$ performs standard conditional generation, while $s > 1$ enhances the adherence to the prompt. This method is simpler, requires only one model, and has become the standard for high-quality conditional image synthesis [12].

Latent Diffusion Models (LDMs) [29] signify a major step forward in model efficiency. Traditional diffusion models operate directly in the high-dimensional pixel space. This makes them computationally demanding. LDMs elegantly solve this problem. They first use a pre-trained autoencoder. This autoencoder maps the image x to a lower-dimensional latent representation, $z = \mathcal{E}(x)$. The diffusion process is then applied within this more manageable latent space, $\mathcal{Z}$. This space is not only smaller but also perceptually rich. Once the iterative denoising process in the latent space yields z_0, a decoder is used. The decoder maps the result back to the pixel space,

creating the final image, $\tilde{x} = \mathcal{D}(z_0)$. This LDM approach significantly reduces the computational resources needed. It makes high-resolution image synthesis more accessible for both training and inference.

Diffusion models emerged as a paradigm shift in generative modelling, offering superior training stability and sample quality compared to GANs [17, 35]. Their application to inpainting represented a fundamental change from direct prediction to iterative refinement through learned denoising processes.

RePaint [25] offers a clever solution for image inpainting. The method was first proposed by Lugmayr et. al. It uniquely adapts a pre-trained, unconditional Denoising Diffusion Probabilistic Model (DDPM). This removes any need for task-specific training. The core idea is to guide the generation process during inference. For any masked image, the model keeps the known pixels fixed. It then synthesizes the missing pixels. This conditioning is applied at each denoising step t. The known image regions are sampled from the tractable posterior distribution $q(x_{t-1}|x_t, x_0)$. In contrast, the unknown regions are sampled from the model's learned distribution $p_\theta(x_{t-1}|x_t)$. A key innovation within RePaint is the **resampling** strategy. This technique improves the harmony between the original and the generated content. It involves taking repetitive jumps backward and forward along the denoising timeline. This process allows the model to refine its output. The result is a more globally consistent and high-quality image. The primary benefit of RePaint is its remarkable flexibility. However, the iterative resampling makes the inference process computationally intensive and slow. This method is used in this research for comparison.

Palette [31] presents a more generalized perspective on image inpainting. It frames inpainting as a conditional image generation problem. This places it within a broader image-to-image translation framework. Palette is a conditional diffusion model. It is explicitly trained to predict the missing parts of an image. The model's architecture is typically a U-Net. It is conditioned on the masked input image. The model then learns to reverse the diffusion process for only the missing areas. Palette demonstrated that a single diffusion-based model can excel at many tasks. These include colorization, inpainting, and uncropping. State-of-the-art results are achieved by simply changing the conditioning data during training. The main strength of Palette is its ability to create high-fidelity and varied outputs. These outputs are also semantically consistent with the input. Its primary drawback is common to most of the pixel-space diffusion models: the iterative sampling process needed for generation is inherently slow.

CoPaint addresses the image inpainting task by modifying the inference process of a pre-trained, unconditional Denoising Diffusion Probabilistic Model (DDPM). Its core methodology is centered on achieving global semantic harmony between the synthesized content and the known image context. Rather than training a specialized inpainting network, CoPaint iteratively refines the entire image canvas. During each reverse diffusion step, it samples the corresponding noise level for the known image region and combines it with the model's predicted output for the masked region. This composite noisy image is then used as the input for the subsequent denoising step. This resampling strategy effectively conditions the unconditional model on the known visual data at every stage of generation, forcing the synthesized content within the masked area to become contextually and stylistically coherent with its surroundings, thereby minimizing boundary artifacts and ensuring a seamless final composition.

Diffusion models offer several key advantages for image inpainting. They address the fundamental limitations of prior techniques. Their primary benefit is superior training stability. This contrasts sharply with Generative Adversarial Networks (GANs). Diffusion models use a well-defined, stable training objective, which makes them more reliable and easier to train. Also, diffusion models excel at generating diverse outputs while GANs usually suffer from mode collapse.

This review has traced the history of inpainting techniques. The progression culminates with the rise of diffusion models. These models have overcome the key limitations of GANs. Foundational methods like RePaint and Palette set high benchmarks. However, they reveal opportunities for further refinement. A particular challenge is balancing efficiency with performance. Reconstructing complex structures in large masks remains difficult. The research in this dissertation is motivated by these limitations. It aims to address these specific challenges.s

3 METHODOLOGY

The approach is centred on a Denoising Diffusion Probabilistic Model (DDPM), a powerful class of generative models. The implementation uses two unconditional pre-trained models, trained on 256x256 Imagenet, provided by OpenAI [2], and 256x256 FFHQ, provided by [3]. These models were fine-tuned for the task of image inpainting. The base code was used from OpenAI's Improved DDPMs repository, which was manually simplified and modified to work for inpainting tasks by conditioning it to masks.
The datasets used throughout the project were CelebA-HQ[1], containing 30,000 images, and a subset of Places365 [4], containing 36,500 images. Both datasets were locally split into train, validate, and test subsets. The ratio for the split was 75:15:10.

3.1 Model Architecture

The model uses the U-net structure. The first convolutional layer was modified to have a 9 input channel rather than 3 (3 channels for RGB image, 3 channels for masks, and 3 channels for RGB masked image). The output channels give the variance along with the mean. The U-net incorporates timestep embeddings that allow it to understand where it is in the diffusion process. During training, noise is gradually added to images over multiple timesteps, and the model learns to reverse this process. Self-attention is used to model long-range dependencies. The structure of the U-net is modified to match the configuration of the pre-trained model file. For the diffusion process, standard MSE loss and perceptual loss functions are used. Different noise schedulers, like linear, cosine, and quadratic, are used to add noise over time. These are covered better in the Experimentation section. Both DDPM and DDIM sampling were used.

The conditioning occurs during the inference or sampling stage. While architectural conditioning tells the model what the problem is, algorithmic conditioning guides the model's output to ensure it remains consistent with the known image data. This is achieved using the cumulative injection of the ground truth in the unmasked regions during every step of diffusion. Without injection, the stochastic sampling process causes known regions to drift away from the ground truth. This creates visible inconsistencies where the model modifies pixels that should remain unchanged. Skipping the injection would result in inpainting outputs that fail to preserve the original image content in unmasked regions. The process is as follows:

1. **Noise Injection:** The original ground truth image is taken, and noise corresponding to the current timestep t is added to it using cached noise for consistency. This creates a ”noised ground truth” that matches the expected noise level at the current sampling step.

2. **Correction:** The known regions in the current sample are replaced with corresponding regions from the noised ground truth using the mask. This ensures known pixels maintain fidelity to the original image while preserving consistent noise levels across boundaries.

3. **Model Prediction** The model receives this corrected sample and predicts the denoising step for the entire image. The model processes both the injected known regions and the generated unknown regions uniformly, producing the next sample in the reverse diffusion trajectory.

This loop ensures that for every step of the reverse diffusion process, the known pixels are constantly corrected. This prevents the model from altering the existing image content and forces the generated solution to be coherent with its surroundings.

3.1.1 Procedural Mask Generation

To facilitate robust training and evaluation, a diverse dataset of unique binary masks was generated. This ensures that there is no data leakage or overfitting to a specific type of mask. For the CelebA-HQ dataset, most of the masks were generated to mainly cover facial features. The masks cover an image area between $5\% - 60\%$. A few masks used while fine-tuning on the CelebA-HQ dataset are displayed in 3.1. All generated masks are 256×256 pixel images where black pixels represent the regions to be inpainted. A uniqueness check using hashing is enforced to ensure that any newly generated mask is novel, preventing data leakage between the training and test sets. The masks

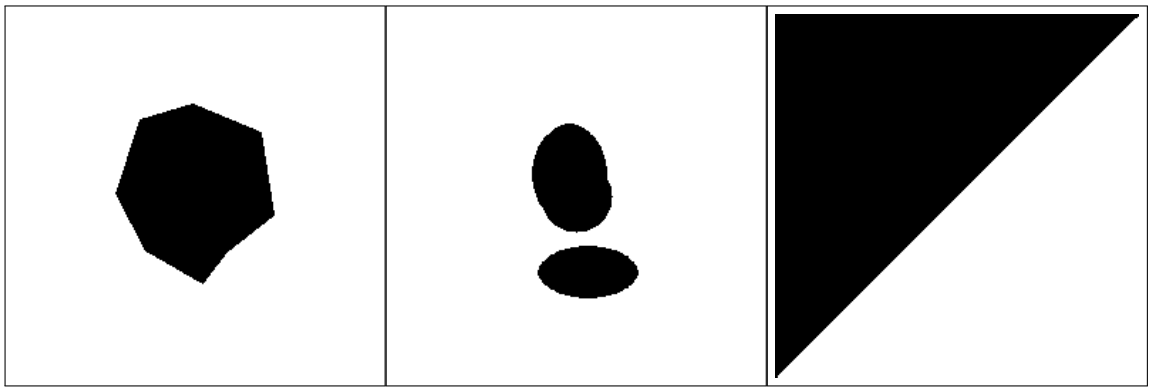

Figure 3.1: Three images with individual borders

are also split locally into the same ratio for train, validate and test sets.

3.2 Data Pipeline and Training Paradigm

Training uses a batch size of 4 and is trained for 10 epochs for full fine-tuning. The GPU used was NVIDIA RTX 4000 with 16GB VRAM for every run, including pre-trained FFHQ and CelebA-HQ. The training and sampling for all experiments using an ImageNet pre-trained model and Places365 used a single NVIDIA A100 3g.40 GB Multi-Instance GPU (MIG) partition. Convergence is monitored through validation-loss plateauing. The input data is resized to ensure a 256x256 size, and the pixel values are normalised before feeding them to the model. The binary masks have 1 for retention of pixels and 0 for the mask. Unique masks are used throughout the project and are paired with images randomly while training. The training employs a hybrid loss function combining standard MSE loss with optional perceptual loss. The MSE loss is computed exclusively over masked regions to focus learning on inpainting tasks:

$$L_{MSE} = \frac{1}{|M|} * \Sigma M \odot (\varepsilon_\theta(x_t, t) - \varepsilon)^2$$

, where M is the binary mask,ε_θ is the predicted noise, and ε is the ground truth noise. The mask area normalisation ensures consistent loss scaling regardless of mask size. When perceptual loss is employed, it operates on the predicted clean image derived from the noise prediction, weighted at 0.1 relative to MSE loss. Gradient clipping at norm 1.0 prevents instability during early training phases when the model adapts to inpainting inputs. The Optimiser used throughout is AdamW with an initial learning rate of $5 * 10^{-5}$, weight decay of 0.01, and a cosine learning rate scheduler. The injection mechanism is selectively applied during training to improve training-inference consistency. During 30% of training steps, the same injection process used in sampling is applied before model prediction. This teaches the model to generate content that harmonises with injected ground truth regions.

Validation occurs every epoch using a separate 15% subset with unique masks not seen during training. Early stopping monitors validation loss with patience of 5 epochs and minimum improvement threshold of $1 * 10^{-6}$. Model checkpoints are saved for the best validation performance and every 5 epochs. An automatic cleanup, retaining only the most recent 2 regular checkpoints plus the best model to work on a limited disk space.

3.2.1 Noise Schedulers

Noise Schedulers are the components of Diffusion Models that control how much noise is added at each timestep during a diffusion forward pass. They define the variance schedule β_t that deter-

mines the progression from clean data to pure noise over T timesteps. Three types of Schedulers were experimented with during this research.

- Linear Noise Scheduler: The linear scheduler provides a straightforward, uniform progression of noise addition throughout the diffusion process. It interpolates linearly between a small starting variance and a larger ending variance. While simple to implement, this approach adds noise very aggressively in the starting steps. This can lead to loss of important information during the forward process.

$$\beta_t = \beta_{\text{start}} + \frac{t}{T} \cdot (\beta_{\text{end}} - \beta_{\text{start}})$$

- Cosine Scheduler: It follows a cosine curve that starts with very small amounts of noise, gradually increases through the middle timesteps, and then plateaus toward the end. This approach preserves more structural information in the early stages of the forward process.

$$\bar{\alpha}_t = \cos^2\left(\frac{\frac{t}{T} + s}{1 + s} \cdot \frac{\pi}{2}\right)$$

- Quadratic Scheduler: This scheduler adds noise more slowly than the linear schedule in the early timesteps but more aggressively than the cosine schedule in the later timesteps. The quadratic progression means that noise increases at an accelerating rate. Quadratic

$$\beta_t = \beta_{\text{start}} + \left(\frac{t}{T}\right)^2 \cdot (\beta_{\text{end}} - \beta_{\text{start}})$$

3.2.2 Parameter-Efficient Fine-Tuning with LoRA

To address the computational constraints of full fine-tuning while maintaining adaptation quality, Low-Rank Adaptation (LoRA) was also used for parameter-efficient fine-tuning of the diffusion inpainting model. LoRA enables selective adaptation of critical model components while freezing the majority of pre-trained parameters. LoRA adapters are applied exclusively to attention mechanisms within the U-Net, specifically targeting Query-Key-Value (QKV) projections and attention output projections (proj out). This selection is motivated by attention layers' central role in spatial reasoning and contextual understanding, which are critical for high-quality inpainting results. The model architecture contains attention blocks at four key locations: input block 9, middle block 1,

and output blocks 2 and 3, all operating at 16×16 feature resolution, where semantic processing occurs. Three approaches were tried for LoRA fine-tuning:

- Applying LoRA adapters solely to attention mechanisms, creating a low-rank decomposition.
- Applying LoRA adaptation to include time embedding projection layers (emb layers), along with the attention layers throughout the U-Net hierarchy.
- - Comprehensive adaptation combining attention mechanisms with a strategic convolutional layer targeting. This configuration includes all QKV and projection layers alongside ResBlock output convolutions at key architectural positions, providing maximum adaptation capacity while maintaining substantial parameter efficiency compared to full fine-tuning.

For training, the learning rate used is $1 * 10^{-4}$, and weight decay is eliminated. During training, the base model weights remain frozen while only LoRA matrices A and B receive gradient updates. This preservation of pre-trained knowledge prevents catastrophic forgetting while enabling task-specific adaptation. For inference deployment, LoRA adapters are merged with base weights through the operation $W_{final} = W_{original} + BA$, eliminating computational overhead during sampling while retaining all learned adaptations (A and B are the LoRA matrices).

3.3 Inference and Evaluation

The evaluation employs two distinct sampling approaches to assess model performance across different computational and quality trade-offs. Both methods implement the advanced inpainting injection mechanism to ensure consistency between known and generated regions.

- DDPM Sampling: This sampling serves as the reference implementation. It uses a 1000-timestep reverse diffusion process. It operates following the standard DDPM equation :

$$x_{t-1} = \mu_\theta(x_t, t) + \sigma_t \varepsilon$$

  The inpainting injection mechanism operates after each denoising step, before advancing to the subsequent timestep. DDPM sampling requires considerable computational resources due to the complete 1000-step denoising process. Each forward pass through the U-Net is sequential. However, this extensive sampling often produces superior quality results.

- DDIM sampling: This sampling technique provides an accelerated alternative that produces samples of comparable quality while significantly reducing the computations. This employs an n-timestep schedule. This produces an evenly spaced subsequence from the original 1000-timestep approach.

$$x_{t-1} = \sqrt{\bar{\alpha}_{t-1}}\hat{x}_0 + \sqrt{1 - \bar{\alpha}_{t-1} - \sigma_t^2}\epsilon_\theta(x_t, t) + \sigma_t\epsilon$$

  where $\hat{x}_0$ is the predicted clean image, ε being the noise prediction and σ controlling the stochasticity through an ETA parameter. Injection is used in this approach as well.

The system is implemented in PyTorch with key dependencies including NumPy for numerical operations, PEFT for LoRA implementation, and specialised libraries for evaluation metrics (lpips, pytorch-fid, scikit-image for SSIM calculation). All experiments use gradient accumulation when necessary.

3.3.1 Evaluation Metrics

The model's performance was quantitatively assessed using three established image quality metrics

- **Structural Similarity Index Measure (SSIM):** The SSIM is a metric for measuring the structural fidelity of an image. It assesses quality based on three components: luminance, contrast, and structure. The metric operates on local windows within the image. It compares these characteristics between the generated and ground-truth images. This makes it effective at identifying structural inconsistencies. The SSIM score ranges from -1 to 1. A value closer to 1 signifies a higher degree of structural similarity.
- **Learned Perceptual Image Patch Similarity (LPIPS):** LPIPS measures perceptual similarity between images. Its goal is to align with human judgments of image quality. Unlike traditional metrics, LPIPS uses a pre-trained deep neural network to extract high-level features from image patches. A pre-trained AlexNet was used in this project. It then compares the features from the generated image to the original. The distance between these features quantifies perceptual differences. For this evaluation, a lower LPIPS score indicates better performance.

- **Fréchet Inception Distance (FID):** The FID is a standard metric for evaluating generative models. It measures both the quality and diversity of the generated images. This is achieved by comparing the statistical distribution of features from generated images to the distribution from real images. A pre-trained InceptionV3 network extracts these features. The FID score represents the distance between these two distributions. A lower FID score indicates that the generated images are more realistic and of higher fidelity.

In addition to these quantitative measures, a thorough qualitative analysis involving visual inspection of the inpainted results was performed to assess semantic coherence, realism, and the absence of visual artefacts.

4 EXPERIMENTATION AND RESULTS

A comprehensive evaluation of diffusion-based inpainting methodologies across multiple dimensions of optimization and performance is discussed here. The experimental design employs a systematic approach to isolate and evaluate individual components while building toward a comprehensive system assessment. Each experiment controls for specific variables while maintaining consistency in evaluation protocols, enabling direct comparison across different methodological approaches. The evaluation encompasses both quantitative metrics (LPIPS, SSIM, FID) and qualitative analysis to provide a thorough assessment of inpainting quality, computational efficiency, and practical applicability.

4.1 Different Sampling Techniques

Two sampling techniques were used as stated before: DDPM sampling and DDIM sampling. The mathematics for these has been covered in the Methodology section. Different timesteps were used for DDIM. The metrics are given in table 4.3

Table 4.1: Different Noise Schedulers

Sampling	Time Steps	FID ↓	LPIPS ↓	SSIM ↑	Time per Sample (s)
DDPM (baseline)	1000	3.70	**0.046**	0.906	33.41
DDIM	100	**3.24**	0.047	**0.921**	3.42
DDIM	50	3.62	0.047	0.910	1.75

As observed through the FID scores, DDIM sampling performs better than DDPM sampling, all the while using less computation and being faster. This could be because DDIM uses a shorter trajectory with scaled variance terms.

The results show that DDIM sampling wth 100 timesteps gives the best FID score of 3.24, while drastically reducing the time taken for sampling from DDPM. Not much change can be seen in the LPIPS and SSIM scores because both methods preserve the structure and local details well.

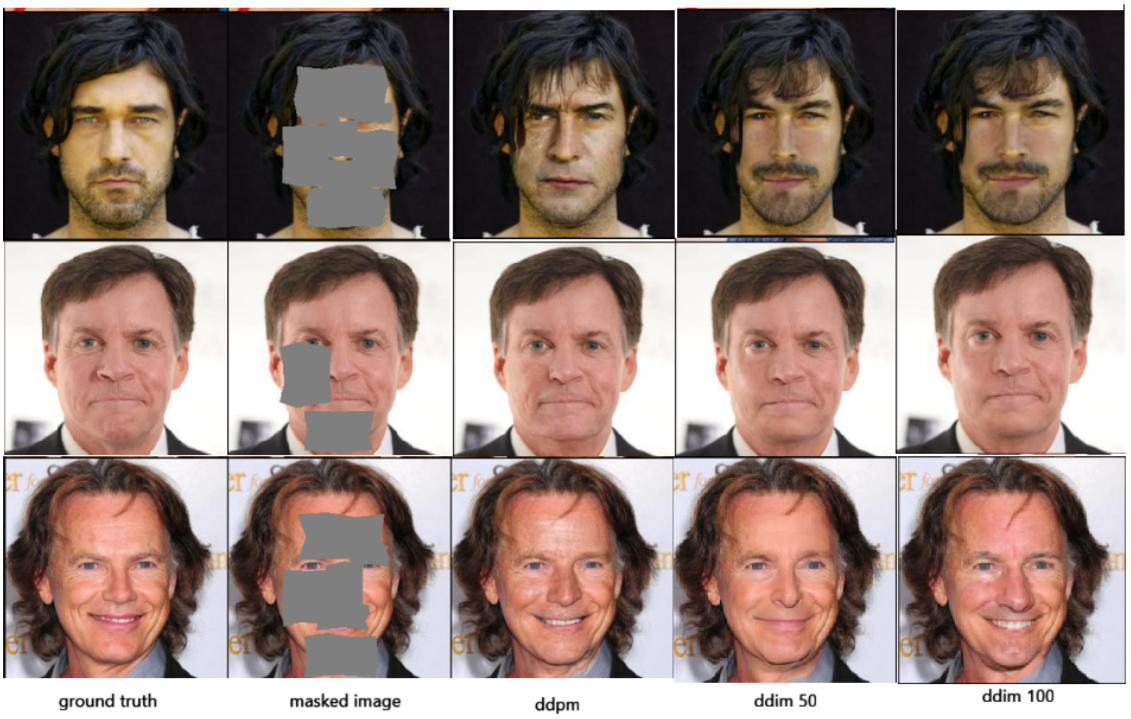

Figure 4.1: Sampling comparison for different mask coverage

4.2 Beta Noise Schedulers

Three noise schedulers were used during experimentation: Linear, Cosine, and Quadratic. DDIM sampling was used for all because of considerable good results for a significantly lesser amount of computation. These tests were performed on the CelebA-HQ test set, which comprises 3000 images. A Stochastic DDIM was used with η being 0.9 throughout.

Table 4.2: Different Noise Schedulers

Scheduler	DDIM Steps	FID ↓	LPIPS ↓	SSIM ↑	Time per Sample (s)
Linear	50	3.62	**0.047**	**0.911**	1.75
Cosine	50	**3.42**	0.048	0.907	2.61
Cosine	30	3.59	0.049	0.908	1.08
Cosine	20	5.60	0.058	0.902	0.71
Quadratic	50	3.91	0.049	0.904	1.74

As stated in [12], Cosine Scheduler is performing better than linear and Quadratic. Using the cosine Scheduler, we get better results for fewer timesteps. So, sampling time is significantly

reduced for the whole test set. The comparison of the noise addition done by these three schedulers is shown in the figure below:

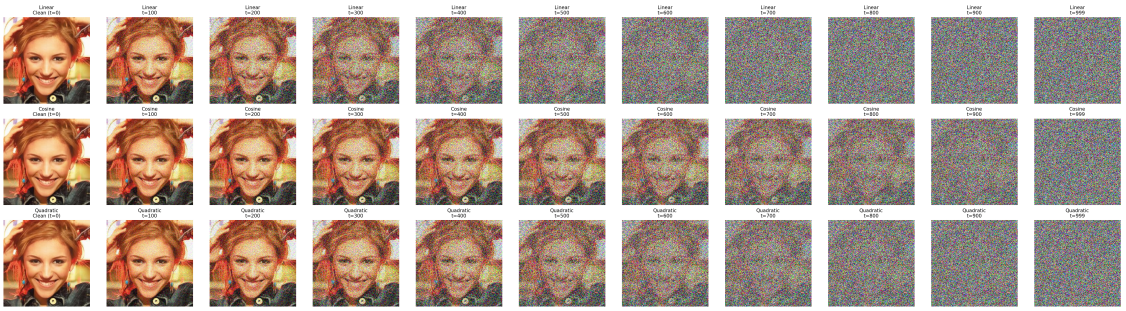

Figure 4.2: Noise Scheduler Comparison for unconditional Diffusion Models

4.3 Different Fine-tuning Techniques

The model was fully fine-tuned on the CelebA-HQ dataset, and a LoRA fine-tuning was attempted. Both were trained for the same number of epochs for comparison. Three different configurations were used for LoRA fine-tuning as mentioned in the Methodology. One training run was done for about 30 epochs to understand the training dynamics and convergence behaviour of the LoRA approach. The metrics for this experiment were not calculated because, as observed from the sampled images, the layers adapted for fine-tuning were not enough. For the 10 epoch runs, the results for a single sample are:

Figure 4.3: Lora outputs for config1: config2: config3

As observed from the results, the model has started to learn, but still requires more tweaking and training to do better. The first configuration, using only Attention layer weights for training, explains that attention is not all you need in this case. For the third configuration, attention layers, an embedding layer, and convolutional layers are used for fine-tuning. This helps the model in learning the color that we can observe in the output.
Adding onto that, the pre-trained model was not built for inpainting. So, LoRA fine-tuning a model for a different task, even having a similar dataset domain, requires more care and training.
For the third configuration, the metrics were calculated. The scores are given below:

FID : 98.24, LPIPS: 0.155, SSIM: 0.7971

4.4 Implementation of RePaint

One image inpainting approach, discussed earlier, is RePaint. This method does not require any fine-tuning of an unconditional diffusion model. Instead, this method conditions the process while sampling. The model de-noises the masked part of the image, but instead of following a linear de-noising strategy, the sampling schedule jumps back and forth in time. This back-and-forth cycle, called resampling, is repeated multiple times within an inner loop before the main process moves on to the next major timestep. The code of the palette was not public, so it could not be implemented. The authors use a pre-trained model on the same dataset, which has been provided by them. For this reason, I could not use my own test dataset, preventing Data leakage. No testing dataset for CelebA-HQ has been provided by the authors, but from the code implementation,[5], a few images were confirmed to be from their test set. Using those images, extracted from the actual dataset, RePaint was performed. The resampling parameters were the same as suggested by the authors: total timesteps being 250, length of one jump = 10, and 10 resampling for each jump. NVIDIA A100 from Google Colab was used for this purpose. The FID score was not calculated because of a very small test set. The sampled images are of good quality, indicating a low FID score. 96.5 seconds are taken to sample one image, which is a lot compared to the methods experimented with. Unfortunately, both images used for testing repaint are used for training for the developed pipeline. So, comparisons couldn't be made. The results are shown below: Ground Truth: Masked Image: Sampled output

Figure 4.4: Repaint Outputs

4.5 Other Datasets

Two sets of experiments were done using an unconditional pre-trained ImageNet model, which was fine-tuned on CelebA-HQ and Places365. The reasoning behind using this pre-trained model for CelebA-HQ was to test the impact of using a domain-specific pre-trained model and a general model. Because the pre-trained model was parameter-intensive, a limited number of experiments could be carried out for this section. The sampling time is not stated because different GPUs were used for this section.

4.5.1 CelebA-HQ

This model was also fine-tuned on 10 epochs only. The sampling used was DDIM for 100 steps.

Table 4.3: Different Pre-trained models.

Pre-training Dataset	Time Steps	FID ↓	LPIPS ↓	SSIM ↑
Image-Net	100	3.40	0.0452	0.905
FFHQ	100	3.24	0.047	0.909

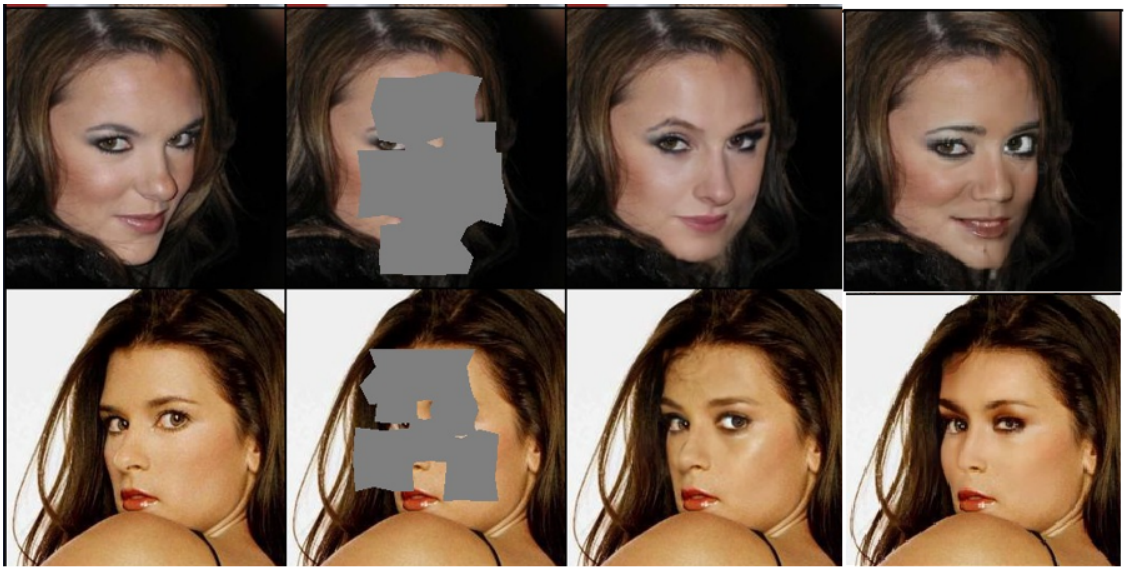

Figure 4.5: Comparison for 2 samples

It can be observed from the metrics and from the images that the model pre-trained on the FFHQ dataset performs better for extreme masks because of the domain similarity.
One thing noticed for this method was that, for the initial run on NVIDIA A100, the Imagenet model took comparatively longer to fine-tune for the same number of epochs.

4.5.2 Places365

The model was fine-tuned for 10 epochs, and ddim sampling was tested for different time-steps.

Table 4.4: Places365 sampled using DDIM

DDIM Steps	FID ↓	LPIPS ↓	SSIM ↑	Time per Sample (s)
30	2.94	0.079	0.888	2.79
50	**2.89**	**0.077**	0.886	4.32
100	2.92	0.078	0.883	**8.19**

The results are not very far apart for the three configurations. Though 50 timesteps seem to give the best results. The results are visually displayed below:

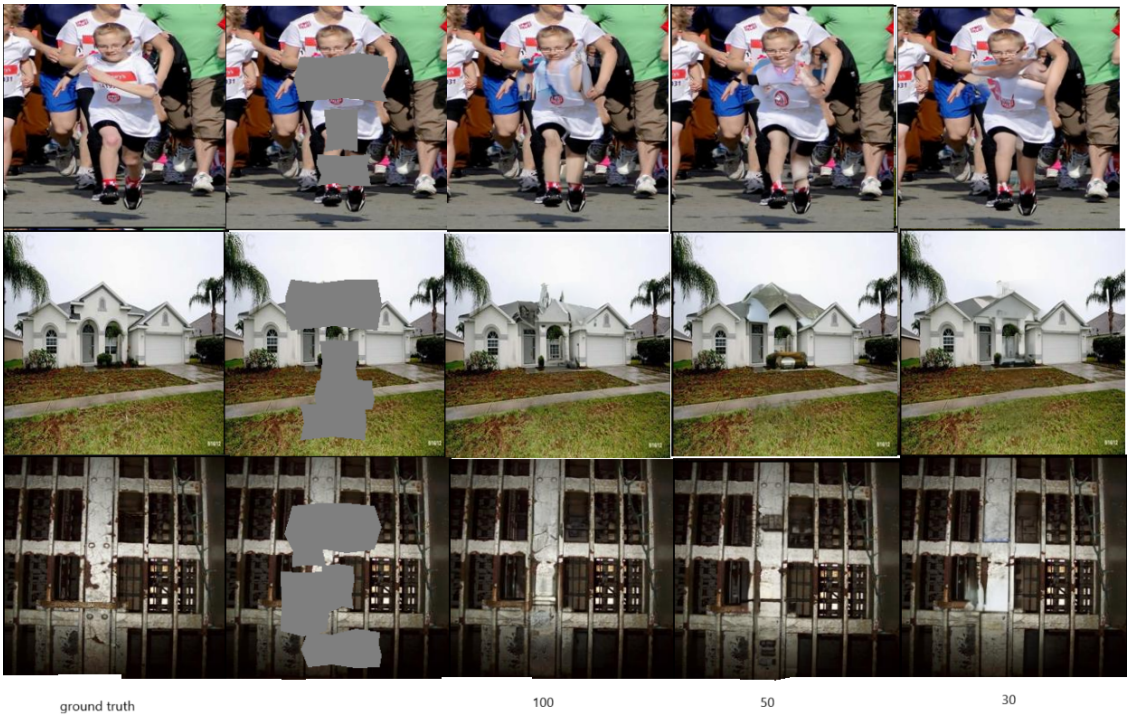

Figure 4.6: Places365

It can be seen from the results that the model performs well for simpler images, where extended regions are present. While, as seen in the first sample, the model generates a blurry artefact in place of the boy's arm.

4.6 Ablation Studies

4.6.1 Noise injection

As described in the previous section, noise injection helps save the unmasked pixels through tiffusion. When the injection guidance mechanism is omitted, the model's performance degrades substantially. The generated outputs exhibit severe visual artifacts. These include notable inconsistencies in color and illumination. In extreme cases, a catastrophic failure occurs, where the model generates incoherent textures instead of plausible facial features. This occurs because of the unguided nature of the sampling process. Without the unmasked regions present to guide the process by providing context, the generation diverges from the surroundings. The unmasked area appears to be the same because it is restored to the ground truth before saving the output. Without injection, the results received are:

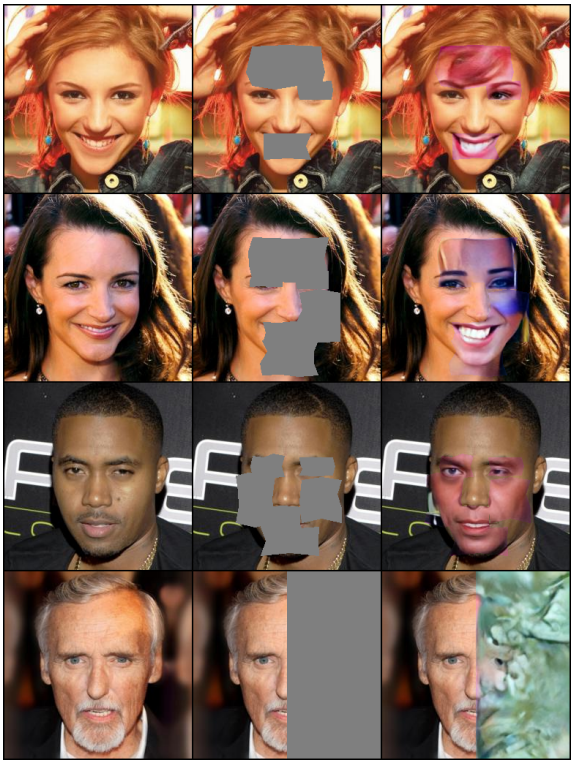

Figure 4.7: One Generated batch without inpainting injection

4.6.2 Stochastic vs. Deterministic Sampling (DDIM η)

η in Denoising Diffusion Implicit Models (DDIM) sampling controls the level of stochasticity in the generation process. It controls the amount of noise introduced at each step of reverse diffusion. $\eta = 0$ makes the process deterministic and hence reproducible. For $\eta = 1$, it behaves like DDPM sampling. Different η values were experimented with to find the one that has been used in all experiments. The same number of time-steps (50) was used.

Table 4.5: varied η

η	FID ↓	LPIPS ↓	SSIM ↑
0.90	3.42	0.048	0.907
0.75	11.91	0.077	0.8796
0.50	18.52	0.089	0.869

You need to work out how many technical chapters you will require, possibly more than two, in this template. Normally, your final chapter before the conclusion will include some clear results or proof of concepts from the end goals of your dissertation, and it is important to document

these well and show the value of what you have achieved. Frequently, many dissertations fail to document this information correctly because it is rushed at the end, so ensure you allow time to write it properly.

4.7 Discussion

Through the experimentation, it is observed that the models perform worse if judged by a human for images where most of the facial features are hidden and the mask also covers something else, say background or shadow. This seems to throw off the sampler, which produces artefacts. A few examples of these have been added here:

Figure 4.8: Larger Masking

For Places Dataset, the model performs worse when the image contains more than one kind of entity. So, if humans are meant to be inpainted from a complex background, the model fails.

Figure 4.9: Distorted Results for Places365 dataset

Looking at oit nce, it does not look very odd, but on inspection, we see that the baby is missing a leg in the first picture, there is just blended noise in the second image and the lady is missing an arm in the third image.

A few successfully generated images for larger masks are shown in the appendix.

5 CONCLUSIONS

This research explored the adaptation of pre-trained, unconditional diffusion models for the specialized task of image inpainting. The project focused on creating a practical and efficient methodology. This chapter consolidates the key findings of this investigation. It evaluates the project's achievements against its original objectives. Finally, it proposes several promising avenues for future research in this domain.

5.1 Evaluation

The central challenge of this dissertation was to repurpose powerful, unconditional generative models for a conditional task. Image inpainting requires not just generating new pixels, but ensuring those pixels are semantically and structurally coherent with existing image data. This work presents a practical investigation into making these state-of-the-art models both effective and efficient for this real-world application.

The core technical achievement is a robust pipeline for high-fidelity image inpainting. This pipeline is built upon a pre-trained unconditional diffusion model. Its success relies on a novel dual-conditioning strategy. This strategy combines an architectural modification with an algorithmic guidance mechanism. First, the model's input layer was re-engineered. It was changed to accept the masked image and the mask itself as direct inputs, providing explicit spatial context from the start of the process. Second, this was enhanced by a critical step during inference. The known, unmasked pixel information was continuously re-injected at each denoising step. The ablation study proved this re-injection is the cornerstone of the method's success. Its removal leads to a catastrophic failure of the model, resulting in incoherent and unusable outputs.
The best performance considering FID score and sampling time was achieved by DDIM sampling using 50 timesteps and a cosine scheduler. This gives an FID score of 3.42 and samping time of 2.61 seconds per sample.

5.2 Future Work

Future work could extend this research in several promising directions. The experiments with Low-Rank Adaptation (LoRA) indicated that a simple application is insufficient for this complex task, presenting a clear opportunity for deeper investigation. A more thorough exploration

of parameter-efficient fine-tuning is warranted. This could involve experimenting with different LoRA ranks, applying adapters to a wider range of convolutional and embedding layers, or exploring alternative methods entirely. Furthermore, to significantly enhance computational efficiency, the methodologies developed in this project could be applied to Latent Diffusion Models (LDMs). Operating in a compressed latent space would drastically reduce training and inference costs, making high-resolution inpainting more accessible. Finally, addressing the model's limitations in reconstructing complex, non-repetitive scenes, as observed with the Places365 dataset, remains a key challenge. Future research could focus on integrating more sophisticated guidance mechanisms or hybrid architectures that better capture long-range dependencies to improve performance on such challenging images.

BIBLIOGRAPHY

[1] URL https://github.com/tkarras/progressive_growing_of_gans.

[2] Github. URL https://github.com/openai/guided-diffusion.

[3] URL https://github.com/jychoi118/ilvr_adm.

[4] URL http://places2.csail.mit.edu/download.html.

[5] URL https://github.com/andreas128/RePaint?tab=readme-ov-file.

[6] Arjovsky, M. and Bottou, L. . Towards principled methods for training generative adversarial networks. *arXiv preprint arXiv:1701.04862*, 2017.

[7] Barnes, C. , Shechtman, E. , Finkelstein, A. , and Goldman, D. B. . PatchMatch: A randomized correspondence algorithm for structural image editing. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 28(3), Aug. 2009.

[8] Bertalmio, M. , Sapiro, G. , Caselles, V. , and Ballester, C. . Image inpainting. In *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '00, page 417–424, USA, 2000. ACM Press/Addison-Wesley Publishing Co. ISBN 1581132085. doi: 10.1145/344779.344972. URL https://doi.org/10.1145/344779.344972.

[9] Bertalmio, M. , Bertozzi, A. , and Sapiro, G. . Navier-stokes, fluid dynamics, and image and video inpainting. In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, volume 1, pages I–I, 2001. doi: 10.1109/CVPR.2001.990497.

[10] Chan, T. F. and Shen, J. . Nontexture inpainting by curvature-driven diffusions. *Journal of Visual Communication and Image Representation*, 12(4):436–449, 2001. ISSN 1047-3203. doi: https://doi.org/10.1006/jvci.2001.0487. URL https://www.sciencedirect.com/science/article/pii/S1047320301904870.

[11] Criminisi, A. , Perez, P. , and Toyama, K. . Region filling and object removal by exemplar-based image inpainting. *IEEE Transactions on Image Processing*, 13(9):1200–1212, 2004. doi: 10.1109/TIP.2004.833105.

[12] Dhariwal, P. and Nichol, A. . Diffusion models beat gans on image synthesis. In Ranzato, M. , Beygelzimer, A. , Dauphin, Y. , Liang, P. , and Vaughan, J. W. , editors, *Advances in Neural Information Processing Systems*, volume 34, pages 8780–8794. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper/2021/file/49ad23d1ec9fa4bd8d77d02681df5cfa-Paper.pdf.

[13] Dong, Q. , Cao, C. , and Fu, Y. . 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. IEEE, 2022. doi: 10.1109/CVPR52688.2022.01108. URL https://openaccess.thecvf.com/content/CVPR2022/html/Dong_Incremental_Transformer_Structure_Enhanced_Image_Inpainting_With_Masking_Positional_Encoding_CVPR_2022_paper.html.

[14] Efros, A. and Leung, T. . Texture synthesis by non-parametric sampling. In *Proceedings of the Seventh IEEE International Conference on Computer Vision*, volume 2, pages 1033–1038 vol.2, 1999. doi: 10.1109/ICCV.1999.790383.

[15] Goodfellow, I. , Pouget-Abadie, J. , Mirza, M. , Xu, B. , Warde-Farley, D. , Ozair, S. , Courville, A. , and Bengio, Y. . Generative adversarial nets. *Advances in neural information processing systems*, 27:2672–2680, 2014. URL https://proceedings.neurips.cc/paper/2014/hash/5ca3e9b122f61f8f06494c97b1afccf3-Abstract.html.

[16] Guillemot, C. and Le Meur, O. . Image inpainting : Overview and recent advances. *IEEE Signal Processing Magazine*, 31(1):127–144, 2014. doi: 10.1109/MSP.2013.2273004.

[17] Ho, J. , Jain, A. , and Abbeel, P. . Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/4c5bcfec8584af0d967f1ab10179ca4b-Abstract.html.

[18] Iizuka, S. , Simo-Serra, E. , and Ishikawa, H. . Globally and locally consistent image completion. *ACM Transactions on Graphics*, 36(4):1–14, 2017. doi: 10.1145/3072959.3073659. URL https://dl.acm.org/doi/10.1145/3072959.3073659.

[19] Jain, V. and Seung, S. . Natural image denoising with convolutional networks. In Koller, D. , Schuurmans, D. , Bengio, Y. , and Bottou, L. , editors, *Advances in Neural Information Processing Systems*, volume 21. Curran Associates, Inc., 2008. URL https://proceedings.neurips.cc/paper_files/paper/2008/file/c16a5320fa475530d9583c34fd356ef5-Paper.pdf.

[20] Kingma, D. , Salimans, T. , Poole, B. , and Ho, J. . Variational diffusion models. *Advances in neural information processing systems*, 34:21696–21707, 2021.

[21] Li, W. , Lin, Z. , Zhou, K. , Qi, L. , Wang, Y. , and Jia, J. . Mat: Mask-aware transformer for large hole image inpainting, 2022. URL https://arxiv.org/abs/2203.15270.

[22] Liu, G. , Reda, F. A. , Shih, K. J. , Wang, T.-C. , Tao, A. , and Catanzaro, B. . Image inpainting for irregular holes using partial convolutions. In *Proceedings of the European conference on computer vision (ECCV)*, pages 89–105. Springer, 2018. doi: 10.1007/978-3-030-01252-6_6. URL https://openaccess.thecvf.com/content_ECCV_2018/html/Guilin_Liu_Image_Inpainting_for_ECCV_2018_paper.html.

[23] Liu, Q. , Tan, Z. , Chen, D. , Chu, Q. , Dai, X. , Chen, Y. , Liu, M. , Yuan, L. , and Yu, N. . 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. IEEE, 2022. doi: 10.1109/CVPR52688.2022.01107. URL https://openaccess.thecvf.com/content/CVPR2022/html/Liu_Reduce_Information_Loss_in_Transformers_for_Pluralistic_Image_Inpainting_CVPR_2022_paper.html.

[24] Lucic, M. , Kurach, K. , Michalski, M. , Gelly, S. , and Bousquet, O. . Are gans created equal? a large-scale study. *Advances in neural information processing systems*, 31, 2018.

[25] Lugmayr, A. , Danelljan, M. , Romero, A. , Yu, F. , Timofte, R. , and Van Gool, L. . Repaint: Inpainting using denoising diffusion probabilistic models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11461–11471. IEEE, 2022. doi: 10.1109/CVPR52688.2022.01117. URL https://openaccess.thecvf.com/content/CVPR2022/html/Lugmayr_RePaint_Inpainting_Using_Denoising_Diffusion_Probabilistic_Models_CVPR_2022_paper.html.

[26] Metz, L. , Poole, B. , Pfau, D. , and Sohl-Dickstein, J. . Unrolled generative adversarial networks. *arXiv preprint arXiv:1611.02163*, 2016.

[27] Nichol, A. Q. and Dhariwal, P. . Improved denoising diffusion probabilistic models. *International conference on machine learning*, pages 8162–8171, 2021.

[28] Pathak, D. , Krahenbuhl, P. , Donahue, J. , Darrell, T. , and Efros, A. A. . Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2536–2544. IEEE, 2016. doi: 10.1109/CVPR.2016.278. URL https://openaccess.thecvf.com/content_cvpr_2016/html/Pathak_Context_Encoders_Feature_CVPR_2016_paper.html.

[29] Rombach, R. , Blattmann, A. , Lorenz, D. , Esser, P. , and Ommer, B. . High-resolution image synthesis with latent diffusion models, 2022. URL https://arxiv.org/abs/2112.10752.

[30] Ronneberger, O. , Fischer, P. , and Brox, T. . U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pages 234–241. Springer International Publishing, 2015. doi: 10.1007/978-3-319-24574-4_28.

[31] Saharia, C. , Chan, W. , Chang, H. , Lee, C. A. , Ho, J. , Salimans, T. , Fleet, D. J. , and Norouzi, M. . Palette: Image-to-image diffusion models, 2022. URL https://arxiv.org/abs/2111.05826.

[32] Sergios Karagiannakos, N. A. . How diffusion models work: the math from scratch. Website or Blog Title, 2022. URL https://theaisummer.com/diffusion-models/. Optional notes, like 'Accessed on [date]'.

[33] Shen, J. , Kang, S. H. , and Chan, T. F. . Euler's elastica and curvature-based inpainting. *SIAM Journal on Applied Mathematics*, 63(2):564–592, 2003. doi: 10.1137/S0036139901390088. URL https://doi.org/10.1137/S0036139901390088.

[34] Sohl-Dickstein, J. , Weiss, E. , Maheswaranathan, N. , and Ganguli, S. . Deep unsupervised learning using nonequilibrium thermodynamics. *International conference on machine learning*, pages 2256–2265, 2015.

[35] Song, J. , Meng, C. , and Ermon, S. . Denoising diffusion implicit models. *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=St1giarCHLP.

[36] Suvorov, R. , Logacheva, E. , Mashikhin, A. , Remizova, A. , Ashukha, A. , Silvestrov, A. , Kong, N. , Goka, H. , Park, K. , and Lempitsky, V. . Resolution-robust large mask inpainting with fourier convolutions, 2021. URL https://arxiv.org/abs/2109.07161.

[37] Vaswani, A. , Shazeer, N. , Parmar, N. , Uszkoreit, J. , Jones, L. , Gomez, A. N. , Kaiser, Ł. , and Polosukhin, I. . Attention is all you need. In Guyon, I. , Luxburg, U. V. , Bengio, S. , Wallach, H. , Fergus, R. , Vishwanathan, S. , and Garnett, R. , editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

[38] Yu, J. , Lin, Z. , Yang, J. , Shen, X. , Lu, X. , and Huang, T. S. . Generative image inpainting with contextual attention. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5505–5514. IEEE, 2018. doi: 10.1109/CVPR.2018.00577. URL https://openaccess.thecvf.com/content_cvpr_2018/html/Yu_Generative_Image_Inpainting_CVPR_2018_paper.html.

[39] Yu, J. , Lin, Z. , Yang, J. , Shen, X. , Lu, X. , and Huang, T. S. . Free-form image inpainting with gated convolution. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4471–4480. IEEE, 2019. doi: 10.1109/ICCV.2019.00457. URL https://openaccess.thecvf.com/content_ICCV_2019/html/Yu_Free-Form_Image_Inpainting_With_Gated_Convolution_ICCV_2019_paper.html.

[40] Zeng, Y. , Fu, J. , Chao, H. , and Guo, B. . Learning pyramid-context encoder network for high-quality image inpainting, 2019. URL https://arxiv.org/abs/1904.07475.

[41] Zheng, C. , Cham, T.-J. , Cai, J. , and Phung, D. . Bridging global context interactions for high-fidelity image completion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11512–11522. IEEE, 2022. doi: 10.1109/CVPR52688.2022.01124. URL https://openaccess.thecvf.com/content/CVPR2022/html/Zheng_Bridging_Global_Context_Interactions_for_High-Fidelity_Image_Completion_CVPR_2022_paper.html.

APPENDIX

Figure 5.1: Using DDIM 50

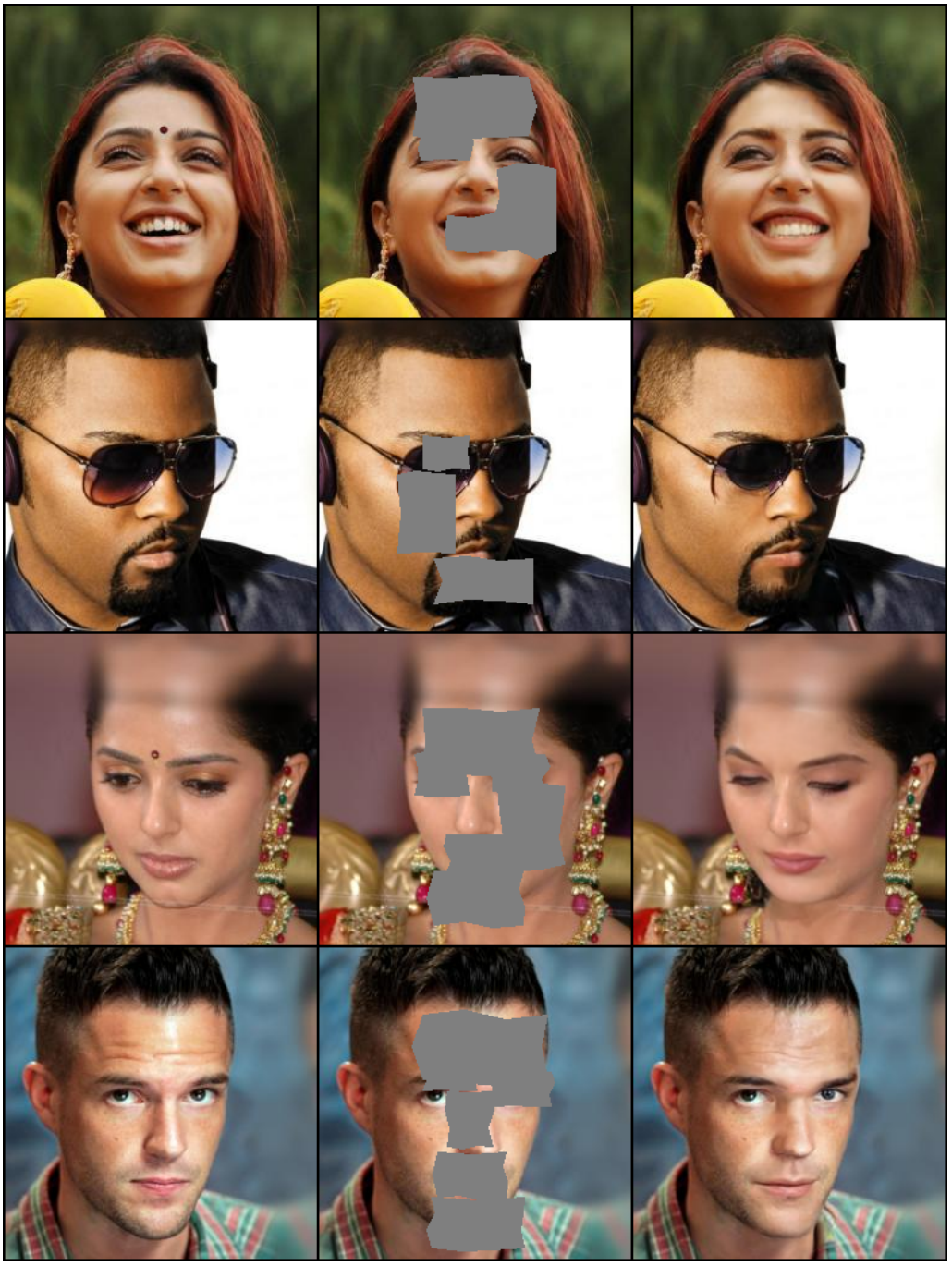

Figure 5.2: Using DDIM 50

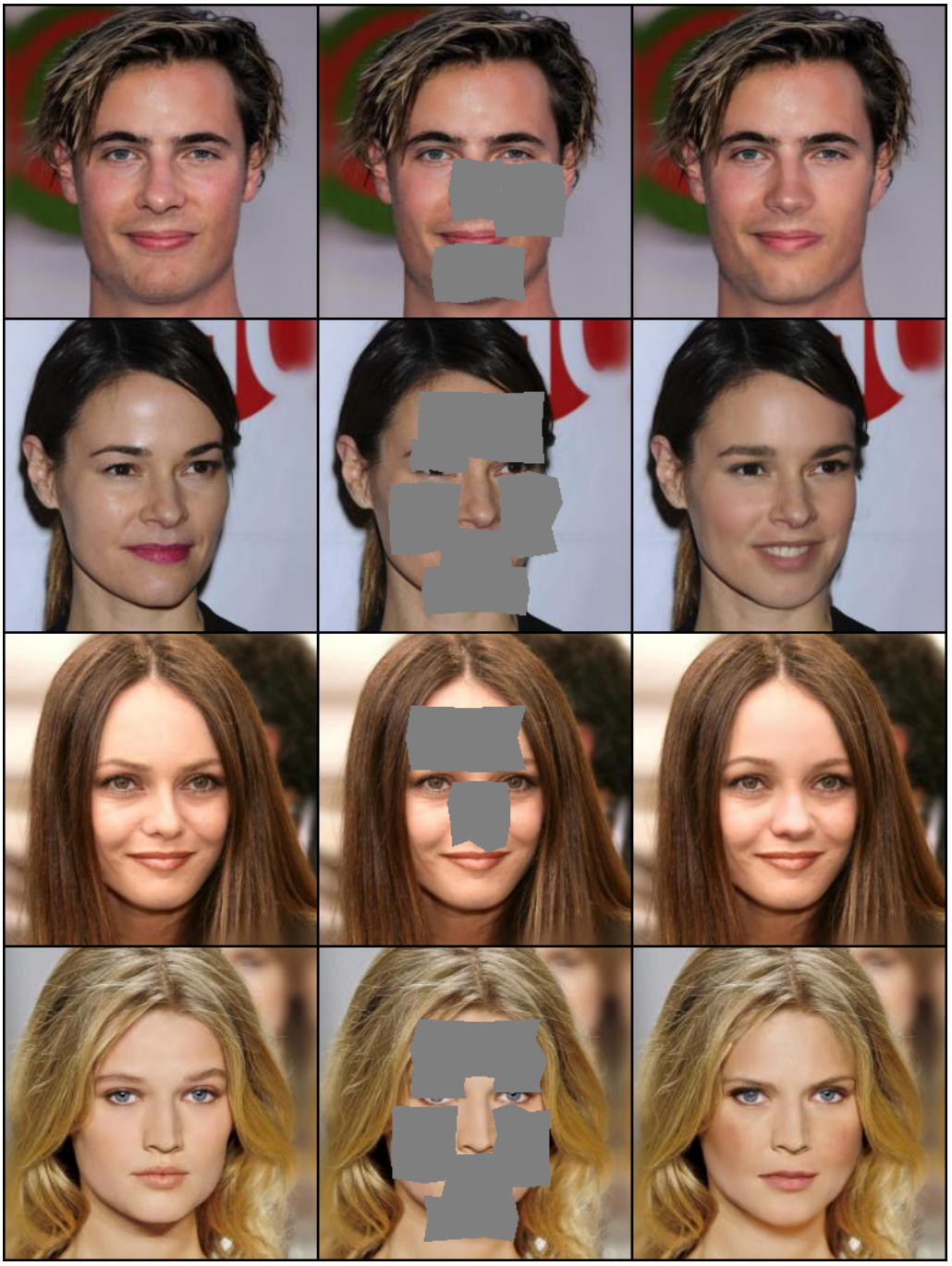

Figure 5.3: Using DDIM 50

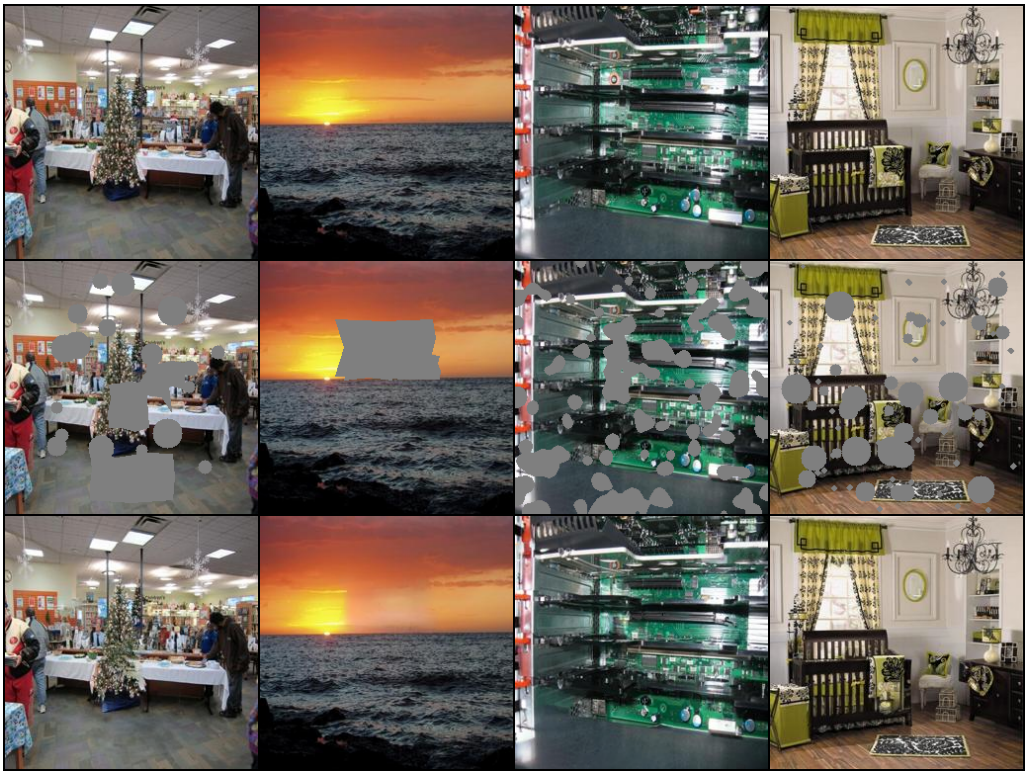

Figure 5.4: Places Using DDIM 100