Conditional GAP for Image Inverse Problems

M.Sc. in Artificial Intelligence and Machine Learning

Ankith Savio Arogya Dass

School of Computer Science

College of Engineering and Physical Sciences

University of Birmingham

2023-24

Abstract

Inverse image problems such as inpainting, colorization, and super-resolution are fundamental challenges in computer vision, often characterized by their ill-posed nature. This project introduces a novel Conditional Generative Accumulation of Photons (Conditional GAP) model designed to address these inverse problems effectively. Building upon the GAP framework, we incorporate conditional inputs to guide the generative process, enabling the model to produce plausible and diverse solutions while managing the challenges introduced by Poisson noise. The Conditional GAP model presents an effective and adaptable solution for inverse image problems, addressing the complexities of Poisson noise and image reconstruction. We evaluate our method on standard benchmarks and demonstrate significant results providing a robust new foundation for solving complex image inverse problems. The research provides a promising foundation for further exploration in Poisson based Geneartive models.

Acknowledgements

I would like to express my deepest gratitude to my supervisor, Dr. Alexander Krull, for his invaluable guidance and support throughout the research process. His expertise and insightful feedback were instrumental in shaping the direction and success of this project. Dr. Alexander Krull's dedication and commitment to my academic growth were pivotal in the completion of this research.

Abbreviations

GAP	Generative Accumulation of Photons
CGAP	Conditional Generative Accumulation of Photons
PSNR	Peak Signal to Noise Ratio
SSIM	Structural Similarity Index Measure
FID	Fretchet Inspection Distance
LPIPS	Learned Perceptual Image Patch Similarity

Contents

List of Figures

List of Tables

CHAPTER 1

Introduction

Images are an essential and universal medium for communication, making them a vital source of information. Images can be interpreted as a structured collection of pixels. In a grayscale image with a single color channel, each pixel represents a feature of the image, and a small image (e.g., a 256x256 image) can have over 65,000 features. Images are high-dimensional data, and due to the curse of dimensionality, traditional image analysis techniques become less effective in processing them. Image Analysis forms a basis for many critical applications such as medical imaging, autonomous vehicles, and scientific research.

In a simplified form, when light in the form of photons is incident on an imaging sensor, the number of photons captured is converted into the intensity of a pixel[25]. However, due to the inherent random nature of photons, the randomness induced by the collection of photons is called shot noise or Poisson shot noise. Several imaging modalities naturally follow a Poisson distribution, such as X-ray CT [55] and Fluorescence Microscopy [63].

Noise in Images is any misleading or uninformative data that obscures us from the original data we are interested in. In addition to the Poisson shot noise, several other kinds of imaging-related noise can exist, like Read Noise and compression Artifacts. Therefore, denoising has become a challenging and essential task. To remove noise, denoising algorithms must be able to deal with the ambiguity of noise, be content-aware, and handle high-dimensional data. Classical Denoising algorithms introduced Filter kernels manually designed to average over the pixels, creating a smoothing effect. Henceforth, they were replaced by learned filters from an optimization process [9].

With deep learning, denoising algorithms evolved significantly. Convolutional Neural Networks (CNNs) have become a powerful tool for denoising, as they can be trained to learn kernels at different levels of the network by minimizing the residual difference between clean and noisy data in a supervised manner [40]. However, supervised learning requires large datasets of paired noisy and clean images, which are often difficult to obtain.

Self-supervised algorithms leverage the structure of the data itself, learning

to predict and remove noise by using noisy data alone. Techniques such as Noise2Noise[27] and Noise2Void[26] have shown that it's possible to achieve competitive denoising performance by exploiting patterns within the noisy data [58]. However, neither method can provide diverse denoising solutions, i.e., sample solutions from the posterior distribution of clean solutions for any given noise image.

HDN [45] introduced diversity denoising models to address this limitation. This model utilized a variational autoencoder (VAE) [23] framework to build a posterior distribution over possible clean solutions, enabling the generation of diverse denoised outputs from a single noisy input.

Building on this concept, Krull et al.'s Generative Accumulation of Photons (GAP) model [25] introduces a novel generative approach to denoising. GAP models upon the simple image formation principle, such that each pixel value indicates the collection of photons following a Poisson distribution. GAP predicts the distribution of the next possible location for the photons, akin to increasing the exposure time in the imaging process. GAP provides a Generative Model and a Diversity Denoising Model that can provide diverse clean solutions for a given Poisson shot noise image.

While GAP excels at Denoising, its potential lies within its Diverse Generative Capability. This Report explores the Conditional GAP Model, demonstrating GAP's capability to tackle a more general challenge of Inverse Problems. By conditioning the GAP model, we can set constraints on the Generative process of the GAP model to not only denoise the images but also to synthesize valid data for incomplete or damaged images provided by the inverse problem.

The project demonstrates the effectiveness of the Conditional GAP model in solving Inverse Problems. Its contributions have the potential to advance the field of Image analysis and Generative Models in Computer Vision.

1.1 Research Question

The aim of the research project is to assess the feasibility of utilizing the Conditional GAP model for three particular Inverse problems: Colorization, Inpainting, and Super Resolution. We can define the research questions as follows:

- How can the Conditional GAP model be adapted to handle natural inverse problems while adhering to the model's assumption of Poisson shot noise?
- What are the advantages and limitations of integrating conditional inputs into the Conditional GAP model?
- How does the Conditional GAP model performan in terms of both quantitative metrics and qualitative results?
- Can the Conditional GAP model perform Diversity Denoising for complex Inverse Problems with damaged or incomplete data?

1.2 Outline

The Report is organized as follows: Chapter 2 provides a comprehensive literature review on the history of inverse problems, the evolution of generative models, and a detailed analysis of denoising generative models. Chapter 3 addresses

the legal, social, ethical, and professional issues associated with the research. Chapter 4 details the background, covering the nature of inverse problems and the fundamentals of the Generative Accumulation of Photons (GAP) model. Chapter 5 outlines the methodology for the Conditional GAP model. Chapter 6 details the experimental setup, offering precise specifications for replicating the experiments. Chapter 7 presents the experimental results. Chapter 8 discusses these results, including an ablation study to examine the impact of specific settings in the Conditional GAP model and suggests directions for future work. Finally, Chapter 9 concludes the report with a summary of the findings.

CHAPTER 2

Literature Review

In this section, we will see a brief history of Inverse problems and their evolution in Section 2.1. From the early adoption of deep learning for inverse problems in Section 2.2, we will see more recent advances using generative models in Section 2.3. We will also read about Poisson-based models and their history in Section 2.4. Then, we will introduce some datasets and metrics in sections 2.5 and 2.6, respectively.

2.1 Inverse Problems

2.1.1 Inpainting

One of the earliest algorithms for image inpainting was the Nearest Neighbors algorithm. This approach involved iteratively filling in masked regions by identifying the closest similar bright pixels, known as isotopes, and applying them to independent color channels [4]. This foundational algorithm was later extended to the Nearest Neighbour Field (NNF) [2], which considered similar patches rather than individual pixels. The NNF algorithm operates under the assumption that patches nearby often share similar nearest neighbors in the target image, and this assumption holds even when the patches are located in different photos. The method was further enhanced to improve the realism of inpainting results by dramatically increasing the dataset to include 2 million images [12]. This extension incorporated context matching, where a radius around the masked region was defined, optimizing for seamless patch integration.

2.1.2 Colorzation

The initial approach to colorization also utilized Nearest Neighbors, where neighboring pixels with similar intensities were assigned the same color, guided by user input [28]. This approach is akin to the isotopes algorithm used in image

inpainting. The method was further advanced by integrating textural information [33]. Here, similar regions were grouped, and color was applied through an optimization process, with weights assigned based on the texture within each region. This allowed for more accurate and contextually appropriate colorization.

2.1.3 Super Resolution

Traditional super-resolution techniques focused on the frequency domain of images. It involved converting multiple low-resolution frames into the frequency domain using the discrete Fourier transform and interpolating the missing frequencies to reconstruct a high-resolution image [5]. A similar strategy was proposed for addressing the spatial domain directly, particularly in face recognition [50]. This method leveraged multiple low-resolution frames from a video to create an average face component using nearest neighbors and then iteratively optimized it to achieve a higher-resolution image.

While these early methods provided significant insights and advancements, they also came with limitations. They were prone to artifacts, computationally expensive, and heavily reliant on user-defined constraints. These challenges highlighted the need for more robust and automated approaches, leading to modern machine learning-based solutions.

2.2 Deep Learning for Inverse Problems

Deep learning revolutionized the approach to inverse problems, with early solutions focusing on specific subsets within this broad domain. Unlike traditional algorithms, which required extensive feature engineering, deep learning models such as Convolutional Neural Networks (CNNs) [24] eliminated the need for manual feature extraction and user-defined constraints. By leveraging multiple layers and high nonlinearity, CNNs were able to learn complex patterns directly from the data, significantly outperforming earlier methods.

A derivative of CNNs is the Context Encoder [43], which utilizes an encoder-decoder architecture designed explicitly for reconstructing large masked regions in images. This architecture can be trained using various loss functions, including L1, L2, and Adversarial Loss, with the latter getting better results. This was extended to image colorization, where they incorporated dilated convolution to capture long-range dependencies within images. This approach trained a CNN to predict the distribution of plausible colors, thus addressing the multimodal nature of color in images. Additionally, the authors proposed colorization as a self-supervised pre-training task, demonstrating that the features learned during the process could be effectively used for other downstream tasks. Super-resolution Convolutional Neural Networks (SRCNN) [**?**] further advanced the application of deep learning to inverse problems by employing an encoder-decoder architecture; SRCNN was trained end-to-end to enhance the resolution of images, surpassing the performance of traditional super-resolution techniques. By integrating complex problem-solving capabilities into a single end-to-end network, CNNs have proven to be a substantial leap forward in inverse problems.

Despite their advantages, CNN-based approaches have a significant drawback: the requirement for paired data. Discriminative Models need large datasets of input-output pairs to train. Acquiring such paired data can be challenging and costly, particularly for complex or specialized inverse problems. This highlights an area of significant disadvantage.

2.3 Generative Models in Inverse Problems

Generative models have become a powerful tool in solving inverse problems by learning and sampling from the underlying data distribution. Unlike discriminative models, generative models can be trained in unsupervised or self-supervised ways, using the data itself as the supervisory signal. Once trained, these models can generate novel samples from the learned distribution, making them particularly suitable for reconstructing damaged or missing data.

Variational Autoencoders (VAEs) [23] are a foundational generative model that employs an encoder-decoder architecture with a latent space that learns the data distribution. The encoder maps input data into a latent space that typically represents a Gaussian distribution, while the decoder reconstructs the original data from this latent space. VAEs have been adapted for tasks such as image colorization by conditioning the model to learn a color map within the latent space [7] This approach enables the model to predict a Gaussian Mixture Model, allowing it to generate diverse plausible colorizations for grayscale images.

Generative Adversarial Networks (GANs) [10] are another advancement in generative modeling. GANs consist of two networks: a Generator that creates fake images and a Discriminator that distinguishes between real and fake images. The adversarial training process forces the Generator to improve its ability to produce realistic images, thereby learning the distribution of real data. GANs have been effectively applied to various inverse problems by incorporating conditional generative models. GANs have advanced inpainting by introducing free-form masks [61], enabling the generation of high-quality inpainted images for any arbitrary mask shapes. Similarly, GANs have been used for image colorization [38] by conditioning the model on grayscale images, and for super-resolution tasks through architectures like ESRGAN [56], which include novel modifications and the use of a Relative Discriminator [19] to enhance generative capabilities.

Diffusion Models [16] are a newer class of generative models that have gained importance for their ability to generate novel images by denoising them from pure noise. Diffusion models operate through two processes: a forward process that gradually adds Gaussian noise to the data and a reverse process that learns to remove this noise, effectively reconstructing the original image. Although slower than GANs, diffusion models offer a more stable training process and avoid issues like mode collapse. They have been applied to various inverse problems, such as inpainting, where Repaint [34] extends previous methods by incorporating masking directly within the forward and reverse processes. Additionally, diffusion models have been adapted for super-resolution tasks, with SR3 [48] demonstrating the effectiveness of cascading multiple models [17] to achieve higher magnitudes of super-resolution compared to earlier methods.

The robustness and flexibility of diffusion models have led to their adoption as a favored conditional generative model for implementing unified frameworks

that address multiple inverse problems. Palette [47], for example, introduced a framework that uses different conditioning strategies within the network to tackle various inverse problems simultaneously. Building on this, Diffusion Posterior Sampling (DPS) [6] extends the framework to handle Poisson noise by approximating it with Gaussian noise. It addressed nonlinear inverse problems and achieved state-of-the-art performance.

2.4 Handling Poisson Noise

While Gaussian noise has been widely studied and implemented as an approximation for noise in image processing tasks, it is not always a physically accurate representation, particularly in cases where noise is signal-dependent. Poisson noise, which arises in scenarios such as low-light imaging and photon-limited environments, is a more realistic model for certain types of applications. Samuel et al. [11] argue that assuming signal-independent noise (as in Gaussian models) is often unrealistic and emphasize the importance of explicitly modeling Poisson noise. Poisson noise presents unique challenges in inverse problems, as it requires specialized models that can accurately account for the signal-dependent nature of the noise.

Recent efforts have been made to adapt diffusion models for Poisson noise, as introduced by Xie et al. [60]. This approach, however, has not yet been widely studied or implemented in practice, highlighting an area for further research and development.

Generative Accumulation of Photons (GAP) [25] is a new generative model explicitly designed to handle Poisson noise. Akin to diffusion models, GAP iteratively denoises images while accounting for Poisson noise's unique characteristics. This model not only provides high-quality denoising solutions but also introduces diversity in the denoised outputs. GAP's potential lies in its ability to be applied to conditional modeling for various inverse problems while explicitly considering Poisson noise, offering an alternative to existing state-of-the-art frameworks like DPS.

CHAPTER 3

LSEPI

Legal Issues

We have ensured compliance with relevant data protection regulations, including the General Data Protection Regulation (GDPR). The FFHQ and CelebA datasets used in this research have been sourced according to their respective usage policies. The datasets are utilized in strict compliance with their data usage agreements. We have adhered to all specified terms of use, including restrictions on redistribution and commercial use, and have ensured that our modifications and analyses conform to these agreements.

Social Issues - Inherent Bias

We acknowledge that the FFHQ and CelebA datasets contain inherent biases, and our model reflects these biases as no specific measures have been taken to mitigate them. We recognize that these biases may influence the outcomes of our research, particularly in the context of tasks such as inpainting, colorization, and super resolution.

Ethical Issues

We are aware of the potential ethical implications of using biased datasets. While our model follows the inherent biases of the FFHQ and CelebA datasets without additional bias mitigation measures, we emphasize responsible reporting of our results and their limitations. We strive to prevent misuse of the technology and ensure ethical practices in its application.

Professional Issues

Our research follows the ethical guidelines set by professional organizations such as the ACM and IEEE. This includes maintaining professional integrity, respecting intellectual property rights, and adhering to ethical practices in research.

CHAPTER 4

Background

4.1 Fundamental of Inverse Problems

Forward models in Computer Vision refer to the process of creating Observations. Observed Images refer to the raw image outputs captured from various sources, reflecting the data or the source of interest. A Linear Forward Model that creates damaged observations can be formulated as follows:

$$y = Dx + \eta \tag{4.1}$$

where y represents the observed data, D is a deformation matrix, x is the clean data and η is the noise or error. In this formulation, D transforms clean data x into the observed data y, destroying some information in the x during the process.

Inverse Problems encapsulate tasks that try to reverse such processes. These tasks are inherently ill-posed, meaning multiple possible clean images can exist for a single observation. A Linear Inverse Problem that restores clean images can be formulates as follows:

$$x = D^{-1}(y - \eta) \tag{4.2}$$

where D^{-1} is the inverse of the deformation matrix.

In a probabilistic setting, we treat the observed image y as a random variable conditioned on x. Equation 4.1 can be rewritten as:

$$p(y|x) = \mathcal{N}(y; Dx, \sigma^2 I) \tag{4.3}$$

where η is assumed as Gaussian noise with a variance of σ^2. We can further rewrite Equation 4.2 using the Bayes Theorem, where the posterior distribution of possible clean images can be formulated as:

$$p(x|y) = \frac{p(y|x)p(x)}{p(y)} \tag{4.4}$$

where $p(x|y)$ is the posterior distribution of the clean image x, given the observed image y. $p(y|x)$ is the likelihood of the observed image y, given a clean image x. $p(x)$ is the prior distribution of the clean images x. $p(y)$ is the marginal likelihood or evidence.

In this project, we are interested in three inverse problems of Inpainting, Colorization and Super Resolution. The Deformation matrix D corresponds to a Binary Mask, Grayscale Transformation and the Downsampling operation respectively.

4.2 Probabilistic Modeling

In computer vision, different models serve distinct purposes based on the nature of the task. For Inverse Problems, Discriminative Models are used to learn a direct mapping from observed data y to clean data x. They model the conditional probability distribution of clean data x given observed data y. It can be formulated as follows:

$$p(x|y) \tag{4.5}$$

Here, the model does not explicitly model the underlying data distribution p(x). These models are usually trained using CNNs with regression objectives like Minimum Mean Squared Error(MMSE).

Generative Models are more sophisticated and model the joint probability distribution of the clean data x and the observed data y as follows:

$$p(x, y) \tag{4.6}$$

$$p(x) = \int p(x, y)dy \tag{4.7}$$

where p(x) is the marginalised probability distribution. These models can generate novel samples from the learned distribution p(x) and create novel clean samples x. There are different ways of training a generative model; some examples are GANs, VAEs, Autoregressive Models, and Diffusion Models. Denoising as a Generative Process was popularised by DDPM [16], where CNNs were used to model the process. Although these models can generate realistic samples, they are inherently random and do not consider any additional constraints available in inverse problems.

A solution is to use a Conditional Generative Model (CGM) that can generate realistic images x that is also constrained by some observed data y. CGMs can be formulated as follows:

$$p(x|y) = \frac{p(x, y)}{p(y)} \tag{4.8}$$

CGMs have been widely applied to Inverse Problems [3, 18]. Consequently, inverse problems are a suitable benchmark for evaluating the capabilities of new generative models.

4.3 Generative Accumulation of Photons

Generative Accumulation of Photons (GAP) is a new Denoising Generative Model that also performs Diversity Denoising. GAP assumes that every pixel

intensity is an photon count and the number of photons in a pixel follows a Poisson distribution. By addressing the formation of images as a sequential collection of photons, the objective of the GAP model is to predict the next possible locations of photons given a shot noise image y_t as the partial collection of photons at time t.

$$p(i = i_{t+1}|y_t) \tag{4.9}$$

where i is the pixel locations in the image, t represents the position in the photon sequence and the equation represents a distribution of possible locations for the next photon in the image. GAP shows that this objective is identical to performing MMSE denoising and allows it to build a generative model by iteratively denoising the shot noise image y_t from $t = 0$ to T and adding photons from the denoised images back to the noisy input y_t following a Denoising Scheduler β.

4.3.1 Forward Process

GAP creates shot noise images y by drawing from a Poisson distribution with the clean image x as the mean:

$$p(y|x) = \frac{x^y \exp(-x)}{y!} \tag{4.10}$$

Shot noise images have an property where the mean and the variance are equal [11]:

$$E[x] = Var[x] \tag{4.11}$$

In order to denoise images of varying amount of noise, the authors propose a strategy to create shot noise images based on a Pseudo-PSNR value. Peak Signal Noise Ratio or PSNR is a quantitative metric to measure the difference between a clean and a reconstructed image. PSNR is formulated as follows:

$$PSNR = 10 \log_{10} \frac{MAX^2}{MSE} \tag{4.12}$$

where MAX is the maximum of the clean image and MSE is the Mean Squared Error between the clean and the reconstructed image. Pseudo-PSNR is an approach to calculate the PSNR of a shot noise image without the clean image. We assume that there exists a clean flat image of intensity γ, then from Equation 4.10 and 4.11 the shot noise corrupted version of that image would have an MSE $= \gamma$. We can formulate Pseudo PSNR as:

$$PSNR_{PS} = 10 \log_{10} \frac{\gamma^2}{\gamma} \tag{4.13}$$

$$PSNR_{PS} = 10 \log_{10} \gamma \tag{4.14}$$

We can further rewrite Equation 4.14 to also directly estimate the mean γ given Pseudo PSNR value as follows:

$$\gamma = 10^{\frac{PSNR_{PS}}{10}} \tag{4.15}$$

Therefore, this allows the GAP model to train for varying amounts of shot noise by randomly sampling Pseudo PSNR values from a pre-defined range and

creating shot noise images by converting the mean γ into a probability for the binomial distribution. We can then sample shot noise-corrupted images from a clean image as follows:

$$PSNR_{PS} \sim \mathcal{U}(-40, 40) \tag{4.16}$$

$$p = \frac{1}{|x|/n} \cdot 10^{\frac{PSNR_{PS}}{10}} \tag{4.17}$$

$$y \sim B(x, p) \tag{4.18}$$

where x is the clean image and y is the shot noise corrupted image. The range for the Psuedo-PSNR is defined based on the 8-bit original clean image intensity; the minimum of the range -40 represents complete corruption of the image, i.e., the image does not have any photons in it and the maximum of 40 indicates a very high-quality reconstruction of the image.

4.3.2 Generative GAP

GAP builds a generative model by iteratively denoising shot noise images. In order to denoise an shot noise image y_t from $t = 0$ to T, we input y_0 to the CNN model and predict the probability distribution for the next possible photon positions as follows:

$$\bar{x}_t = f(y_t; \theta) \tag{4.19}$$

where f is the CNN model parameterized by θ and $\sum \bar{x}_t = 1$. The GAP, sets a variable α_t that controls the number of photons to add back to the input y_t from the output $\bar{x}_t$. Another parameter β, controls the rate at which the α_t increases each iteration. The process is formulated as follows:

$$\alpha_t = \max(\beta \sum y_t, 1) \tag{4.20}$$

where the default value for $\beta = 10\%$ and the α_t considers at least one photon to add back to the input. Here, the number of photons added is proportional to the number of photons seen in the input, ensuring we denoise the image according to the noise level. Further, we sample the new photons from a Poisson distribution with the following as the mean:

$$\lambda_t = n\bar{x}_t\alpha_t \tag{4.21}$$

$$y_t^{new} \sim \mathcal{P}(\lambda_t) \tag{4.22}$$

where y_t^{new} indicates the newly sampled photons for the shot noise image y_t. Therefore, we iteratively follow this process to create shot noise images from $t = 0$ to T as follows:

$$y_{t+1} = y_t + y_t^{new} \tag{4.23}$$

To perform Diversity Denoising, initialize the input with a shot noise image y_t from any time step t to generate diverse denoising solutions.

4.4 Deep Learning

4.4.1 UNet Architecture

UNet is a Convolutional Neural Network which has been widely studied for various computer vision tasks [22, 51, 13]. It was first introduced for the purpose of Image Segmentation [46] and henceforth has been applied to various Image-to-Image Translation tasks [47, 48, 18, 64]. The UNet architecture comprises of an encoder-decoder structure and skip connections between them. The Encoder path has several Convolutional Layers that gradually downsamples the input image with ReLU activation function. The Decoder path also has symmetrical Transposed Convolutional Layers that upsamples the image to higher resolution which are also followed by ReLU activations. The skip connections send features maps of the same resolution from the Encoder to the Decoder block enabling stable learning process. All these components together make UNets a versatile architecture which have been implemented for powerful Generative Models like DDPM.

4.4.2 Residual Connections

Deep Neural Networks lead to vanishing gradients during training [42]. Vanishing gradients occur when backpropagating errors through multiple layers gradually diminish the gradients; this results in small weight updates, which makes the Training process unstable. The solution to this was introduced by the ResNet [14] architecture that constituted residual connections where the input of a layer was directly connected to the output. This enables every layer to learn an identity mapping that allows networks to have deep layers and a stable training process.

4.4.3 Fourier Feature Mapping

Fourier Feature Mapping is a technique introduced to learn high-frequency information in images more effectively [53]. This technique was adapted for MLPs [37], where the inputs were transformed using sinusoidal functions of different frequencies.They show that Fourier Feature Mapping drastically improves the performance of coordinate-based MLPs, enabling them to learn higher-frequency functions and achieve state-of-the-art results.

4.4.4 Cascaded Networks

Cascaded Networks are a type of Architecture with multiple sequentially structured models, where each model refines the output of the previous model. The hierarchical structure offers various advantages when dealing with complex computer vision tasks. Each model in a cascaded network focuses on training for a specific range of tasks, resulting in faster convergence because of the smaller and more specialized objective. This modularity allows each model to be trained in parallel and perform better than a single deep network. Cascaded Networks have been applied for Diffusion models for tasks like Super Resolution [48].

CHAPTER 5

Methodology

This chapter first details the architecture of the Conditional GAP Model. We then outline the data and training pipeline, emphasizing the modular design for adapting to different inverse problems. Finally, we provide the implementation details, covering software and hardware resources.

5.1 Conditonal GAP for Inverse Problem

GAP, in its standard form, lacks control over the generation process. We can incorporate additional conditioning to guide and constrain the generative process to Denoise the Shot Noise Image and Solve for Inverse Problems. In this project, we provide a unified framework for the conditional gap model that can be applied to inpainting, colorization, and super-resolution.

5.1.1 Forward Process

Following GAP, we use the method explained in section 4.3.1 to generate the training pairs. The Input to the model is the randomly sampled Shot Noise images, and the Target is the corresponding Normalized clean image. Normalizing the target ensures we satisfy the GAPs constraints and output a probability distribution for the successive photons.

5.1.2 Deformation Process

We create the condition inputs to the model following a deformation process corresponding to each inverse problem. We utilize the image input to the Forward Process to degrade it using a Deformation Matrix elucidated in Section 4.1. When conditioned with the shot noise image, the deformed input image acts as additional guidance to the generative process.

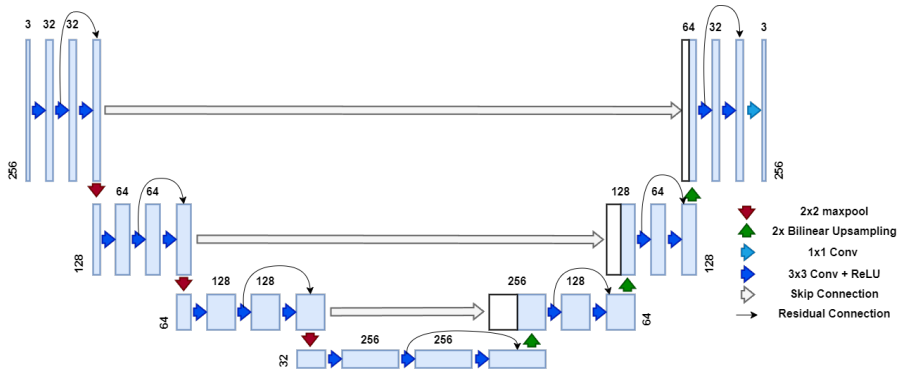

Figure 5.1: The modified UNet Architecture for Conditional GAP, we replace the transposed convolution with bilinear transformation and add residual connections. The figure depicts a simplified version of the architecture, where there are only 4 levels.

5.1.3 Architecture Modification

Following GAP, we utilize a modified UNet [46] architecture as our Denoising Model. We replace Convolutional Blocks with Residual Convolutional Blocks, with each block consisting of 3 Convoltion Layer with Residual Connection. We use the 2x2 Maxpooling operator for downsampling and following Wojna et al.[59], we use Bilinear Transformation for Upsampling to avoid artefacts from Transposed Convolutions. For Fourier Feature Mapping, we utilize 10 Sinusoids with different frequencies of power 10. Finally, we use the UNet architecture with 7 levels to accommodate our complex inverse tasks. In Figure 5.1, we provide detailed specifications of our modified UNet architecture but limit it to 4 levels.

5.1.4 Photon Loss

We follow GAP's objective of cross entropy to predict the probability distribution for the successive photos. To modify the Photon Loss, we formulate the Conditional GAP model as follows:

$$f(y_t, y_c; \theta) \approx p(i = i_{t+1} | y_t, y_c) \tag{5.1}$$

where y_c is the additional condition input created by the Deformation Process and f is the CNN parametrized by θ. The modified Photon Loss is as follows:

$$L(\theta) = \sum_{k=1}^{m} \frac{1}{n|y_{tar}^k|} \sum_{i=0}^{n} ln f_i(y_{inp}^k, y_{cinp}^k; \theta) y_{tar,i}^k \tag{5.2}$$

Equation 5.2 defines the Photon Loss for the Conditional GAP model that accommodates the additional input y_{cinp} and y_{tar} as the normalized target. In Figure 5.2, we provide the detailed implementation of the unified framework of the Conditional GAP model for Inpainting Inverse Problems; however, it is directly applicable to other inverse problems.

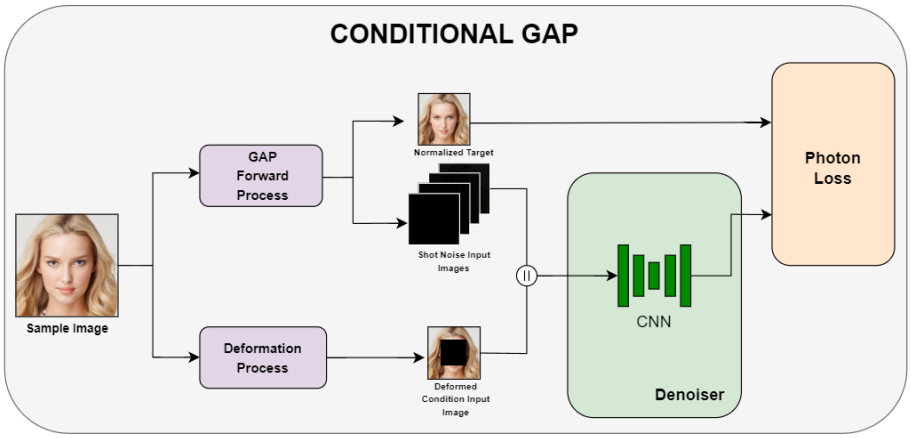

Figure 5.2: The Training process for the Conditional GAP. The process starts with creating the input, target and condition trio and training the CNN model with Photon Loss as the objective.

5.1.5 Generative Model

By adopting the formulation from Equation 5.1, we can follow the Generative Process from Section 4.3.2. We input the Conditional GAP model with a blank or empty image which describes an image with complete Poisson noise, when there is no information available in the input, the model makes an estimated guess which is the average among all different kinds of possible solutions. With the initial estimate, we sample photons and add them back to the input to predict the clean image better. We do this iteratively until the output prediction surpasses the range of noise the model was trained on. Figure 5.3 details the Generative Process with inpainting as an example. To solve other inverse problems, we replace the deformation process.

5.1.6 Diversity Denoising

Diversity Denoising follows the same procedure as the Generative Process; however, instead of providing an empty image, we provide the Conditional GAP model with the True Shot Noise Image. This demonstrates the Conditional GAP model's denoising capability by providing diverse solutions for a single Shot Noise image.

5.2 Data Pipeline

In this section, we modularize the different inverse problems and explain the preprocessing steps taken to adapt them for the Conditional GAP model. In order to simulate the Poisson Distribution for Images, we multiply the images with a constant scalar value to imitate a high exposure time. This allows us to get realistic Noise Levels in the Forward Process. We set the scalar to a value of 1000, and it is multiplied with the clean image before the Forward Process.

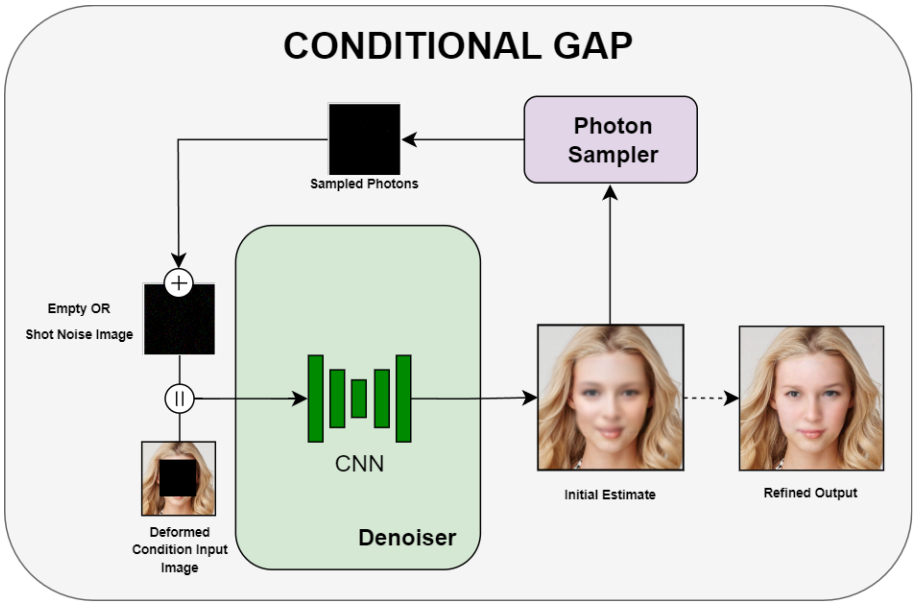

Figure 5.3: The Generative Process of the Conditional GAP. We start with an initial empty image and iteratively refine the output by adding more photons to the input image. For Diversity Denoising, we start directly with a shot noise input image.

5.2.1 Inpainting Module

This module explains the process of adapting Conditional GAP for Image Inpainting. Inpainting is a much more difficult task compared to other inverse problems as the model needs to create valid content that matches the surrounding area [4]. For any masked region there is also many valid content that can filled in. Conditional GAP can build a generative model for the final complete image given a masked image and the shot noise image.

Data Preprocessing: Masking

Conditional Masked Input images are created by multiplying a clean image x by a binary mask m produced by a Mask Generation Algorithm detailed in Algorithm 1.

$$y_c = s \times m$$

where y_c is the corrupted observation in the inverse problem. In this module, we aim to mask large rectangular regions from the image to test the Generative capability of the conditional GAP model.

Conditional Encoding and Architectural Modification

For facilitating the Conditional GAP model for inpainting, we additionally condition the model on the corrupted observation y along with the binary mask m. We concatenate all three inputs along the channel dimension. The resultant

combined input tensor is as follows:

$$I_{inp} = Concat(y_t, y_c, m)$$

where I_{inp} is formed by stacking the input data resulting in a tensor of dimensions $H \times W \times 7$. To accomodate the change the UNet architecture input channel dimension is increased to 7.

Algorithm 1 Algorithm for Mask Generation (with $imgsize = 256$, $masksize = 128$)

Initialize: $mask = zeros(imgsize, imgsize)$
$masksize = \text{random.uniform}(masksize - 0.1 \times masksize, masksize + 0.3 \times masksize)$
$maxr = imgsize - masksize$
$top = \text{random.uniform}(0, maxr)$
$left = \text{random.uniform}(0, maxr)$
$h, w = masksize$
$delta = 15$
$h\_offset = \text{random.uniform}(0, delta)$
$w\_offset = \text{random.uniform}(0, delta)$
$mask[top + h\_offset : top + h - h\_offset, left + w\_offset : left + w - w\_offset] = 1$
Return $mask$

5.2.2 Colorization Module

This Module encompasses the adaptation of Conditional GAP explained for the application of Colorization. Colorization is an ill-posed task where there are multiple plausible color interpretations for a single grayscale image. With Conditional GAP we can build a generative model for the final color image given a grayscale image and the Shot Noise images.

Data Preprocessing: Grayscale Conversion

To create the input grayscale images for the Colorization, we convert the RGB image into Grayscale using the CCIR 601 standard. We perform this conversion using the Torchvision library [35] which implements the following:

$$y_c = 0.2989 * r + 0.587 * g + 0.114 * b$$

where r, g, and b correspond to the red, green, and blue color channels, respectively. The CCIR 601 Standard has been utilized in various other Colorization solutions [36] and is a widely accepted method for Grayscale conversion.

Conditional Encoding and Architectural Modification

We adopt both the Grayscale image and the Poisson Shot Noise image as the input to the network by concatenating along the channel dimension. This results in a combined input tensor:

$$I_{inp} = Concat(y_t, y_c)$$

where I_{inp} is formed by stacking the Poisson Shot Noise image y_t and Grayscale image y_c along the channel axis, resulting in a tensor of dimensions $H \times W \times 4$. Appropriate changes are made to UNet architecture by increasing the input channel dimension to 4.

5.2.3 Super Resolution Module

This Module demonstrates how to achieve Image Super Resolution by utilizing Conditional GAP. Super Resolution is another ill-posed task where multiple feasible higher-resolution images can be obtained for a single lower-resolution image. We can build a generative model for Higher-resolution images by training the Conditional GAP model with lower-resolution images and corresponding Higher-resolution Poisson shot noise images.

Data Preprocessing: Low Resolution Images

To build an 4× Super-Resolution Model, we reduce the dimensionality of the data to 64x64 to create low-resolution images. We utilize bilinear interpolation with antialias to downsample the images. To accommodate the resolution change between the images, we resize the Low-Resolution image back to 256x256, matching the size of the Shot Noise Image using the same bilinear interpolation.

Conditional Encoding and Architectural Modification

We can build the Conditional GAP for Super Resolution by further conditioning the model on the Low-Resolution images. We concatenate the images along the channel axis, resulting in an input tensor as follows:

$$I_{inp} = Concat(y_t, Upsample(Downsample(y_c)))$$

where I_{inp} has a dimensions $H \times W \times 6$ by stacking the Poisson Shot Image y_t and Low Resolution Image y_c. The input channel dimension in the UNet Architecture is changed to 6.

5.3 Cascaded Conditional GAP

In this framework, we individually train five Conditional GAP models specializing in different ranges of Pseudo PSNR values. Rather than training for the full noise range of [-40, 40], we segment the ranges into narrower intervals of 10. we start with [-40, -30] for the first model and train till the range of [0, 10] for the fifth model. This allows us to train multiple parallel models that converge faster than training a full-range model. Other than the noise range, the Cascaded Conditional GAP Model share the same specification as the Conditional GAP model.

5.4 Implementation Details

Software specifications

PyTorch [1] is the core deep learning framework used for model implementation. PyTorch provides an efficient platform for building neural networks due to its dynamic computation graph and extensive support for custom model development. For managing training loops and other tasks such as checkpointing, and logging, we used PyTorch Lightning [8]. To preprocess the Image datasets elucidated in Section 5.2, we utilized Torchvision [35].

Hardware Specification

The model was trained on an NVIDIA Tesla T4 GPU, which provided the computational power to train complex neural networks with large image datasets. This GPU performed all the training and evaluation in this project.

Training Specification

We trained the Conditional GAP model on Tesla T4 using a batch size of 32 for 20 epochs for each inverse problem, and each training run took approximately 10 hours. We used Adam Optimizer with a learning rate of $1e-4$ and a learning rate decay scheduler from Pytorch, *ReduceLROnPlateau*.

CHAPTER 6

Experiment Setup

In this chapter, we outline the experimental setup used to evaluate the project. We detail the datasets and metrics employed to assess the performance of the Conditional GAP model across various inverse problems. By providing a comprehensive overview of the experimental framework, we aim to ensure reproducibility and clarity in the results presented.

6.1 Dataset

Training Data

We utilize the Flickr-Faces-HQ Dataset (FFHQ) [21] 256x256, which is a human faces dataset created to evaluate Generative Adversarial Networks (GANs) for their Generative capabilities. The dataset consists of aligned human faces of dimensions 256x256 scraped from the Flickr website. The dataset consists of images of varying age and ethnic backgrounds; a more detailed analysis of the diversity can be found in the following paper by Perera et.al [44]. For each Inverse problem, the dataset is preprocessed differently, explained in their respective Modules in Chapter 5. The dataset will serve as the Ground Truth or the Clean Image s for the Conditional GAP model.

Evaluation Data

For evaluation, we follow SR3 [48] and use the High-Quality version of the Large-scale CelebFaces Attributes Dataset (CelebA-HQ) [31, 20] 256x256, which is another similarly aligned human celebrity faces dataset created to train GANs [20]. Evaluating on a separate dataset allows us to test the capabilities of our Conditional GAP Model more robustly.

6.2 Metrics

To evaluate the conditional GAP model quantitatively, we use standard metrics that compare the pixel values from the results directly with the ground truth and perceptual metrics that align with human perception and use different patterns in the images [62]. By showing both metrics, we account for technical accuracy and perceptual quality, resulting in a complete assessment of its performance for the Inverse Problems.

6.2.1 Standard Metrics

PSNR

We use the PSNR (Peak Signal-to-Noise Ratio) metric to evaluate how closely a reconstructed image matches the original regarding pixel-level accuracy. It measures the ratio between the maximum possible pixel value and the error between the reconstructed and ground truth images. PSNR is useful for quantifying the fidelity of image reconstructions, particularly in tasks like image compression and super-resolution, where preserving the exact pixel values is critical. We reformulate the PSNR explained in Equation 4.12 to fit our problem:

$$MSE = \frac{1}{n}\sum_{i=0}^{n}[x_i - \hat{x_i}]^2$$

$$PSNR = 10\log_{10}\frac{\bar{x}^2}{MSE}$$

where) is the clean image, $\hat{x}$ is the estimated image, n is the size of the images and $\bar{x}$ is the maximum of the clean image. PSNR is measured in decibels dB, where higher values indicate better image quality, with typical ranges above 30 dB considered good. Low PSNR values (below 20 dB) suggest significant deviations from the ground truth, meaning the method struggles to preserve details.

SSIM

Structural Similarity Index Measure (SSIM) [57] is another quantifiable metric that calculates the similarity between two images by comparing their local patches. SSIM considers the image luminance, contrast, and local patterns to calculate the measure. SSIM goes beyond pixel differences by assessing the perceived quality of images based on structural information. It is valuable for capturing important structural features, making it more aligned with human perception than pure pixel-based metrics like PSNR. It can be formulated as follows:

$$SSIM = \frac{(2\mu_x\mu_{\hat{x}} + c_1)(2\sigma_{x\hat{x}} + c_2)}{(\mu_x^2 + \mu_{\hat{x}}^2 + c_1)(\sigma_x^2 + \sigma_{\hat{x}}^2 + c_2)}$$

where x is the clean image, $\hat{x}$ is the estimated image, μ are the mean and σ^2 are the variance of the image features, $\sigma_{x\hat{x}}$ is the covariance of x and $\hat{x}$. The c_1 and c_2 avoid division by zero or very small values. SSIM ranges from 0 to 1, where 1 indicates perfect structural similarity between the reconstructed and ground truth image. Values closer to 1 represent that the model retains

important structural details and lower SSIM values suggest poor preservation of image structures.

6.2.2 Perceptual Metrics

FID

Fretchet Inspection Distance (FID) [15] is a metric used to evaluate the quality of images from Generative Models [48, 6]. FID score measures the similarity between distributions of generated and real images by comparing their feature representations. FID assesses the realism of generated images by evaluating how closely their distribution matches that of real-world images. It is particularly useful in generative tasks like inpainting and colorization, where the goal is to produce images that appear naturally plausible. We use a pre-trained Inception-v3 [52] as an image Feature extractor and to match its configuration, images are resized to 299x299. The extracted feature size is set to 2048, corresponding to the model's last layer. We can formulate the FID as follows:

$$FID = \| \mu_x - \mu_{\hat{x}} \|^2 + tr(\Sigma_x + \Sigma_{\hat{x}} - 2(\Sigma_x \Sigma_{\hat{x}})^{\frac{1}{2}})$$

where $\mathcal{N}(\mu_x, \Sigma_x)$ is the multivariate normal distribution of the clean images x and $\mathcal{N}(\mu_{\hat{x}}, \Sigma_{\hat{x}})$ is the multivariate distribution of the estimated images $\hat{x}$ from the inception-v3 network. Lower FID values indicate better performance. An FID of 0 means the generated and real image distributions are identical, with good values ranging between 1 and 30. Higher FID scores (> 100) suggest the model produces unrealistic images far from the clean data distribution.

LPIPS

Learned Perceptual Image Patch Similarity (LPIPS) [62], unlike FID, measures the actual difference between the features extracted from intermediate layers of a pre-trained VGG16 [49]. It evaluates the perceptual similarity between images based on VGG16 outputs. It focuses on comparing the overall visual similarity of image patches, making it sensitive to the perceptual quality and texture differences. It can be formulated as follows:

$$LPIPS = \sum_l \frac{1}{n_l} \sum_{i=0}^{n} w_l \| \phi_l(x)_i - \phi_l(\hat{x})_i \|_2^2$$

where x is the clean image, $\hat{x}$ is the estimated image, l denotes a layer of the VGG16 network, ϕ_l is the features extracted from the images, w_l is a learned weight for the features and n_l is the feature size for the layers l. LPIPS values also range from 0 to 1, where a lower LPIPS score suggests that the generated images are visually close to the ground truth regarding texture and perceptual quality. Higher LPIPS values indicate a loss of visual quality.

6.3 Qualitative Evaluation

To showcase the qualitative performance of the Conditional GAP model, we conduct two different types of experiments that align with the GAP framework, providing visual examples to highlight the model's effectiveness in various image inverse tasks.

6.3.1 Image Generation

We conduct full generative experiments following the method outlined in Section 5.1.5, focusing on both the Conditional GAP model trained across the full noise range and the Cascaded Conditional GAP model. The full-range Conditional GAP generates images across a wide spectrum of noise levels, offering insight into its flexibility. In contrast, the Cascaded model, designed to handle narrower noise ranges, demonstrates its ability to maintain precision in progressively refined noise intervals. These generative samples emphasize the model's ability to recover fine details in the Inverse problem tasks.

6.3.2 Diversity Denoising

Following the GAP model, we perform diversity denoising experiments where multiple plausible outputs are generated for the same shot-noise-degraded input image. These samples reflect the conditional GAP model's ability to offer diverse solutions for the inverse problems of inpainting and colorization where diversity is more prevalent. We provide three distinct denoised outputs for each shot noise image, represented by photons per pixel. Using the Cascaded Conditional GAP model, we emphasize the model's robustness in handling various noise levels by showcasing these diverse outputs, particularly for challenging noise ranges where conventional models might struggle.

CHAPTER 7

Results

In this Chapter, we provide the results for the Generative Process and the Diversity Denoising for each inverse problem module defined in Chapter 5.

7.1 Inpainting

Generative CGAP

We provide the quantitative results for the task of Generative Inpainting using the Algorithm 1 in Table 7.1 for both CGAP and Cascaded CGAP. To better visualize the qualitative results, we use a binary center mask of size 128x128 in Figure 7.1. We notice that the Cascaded cGAP drastically outperforms the base CGAP model on all the metrics.

Diversity Denoising CGAP

We provide qualitative results for the task of Diversity Denoising for Inpainting using the binary centre mask of size 128x128 in Figure 7.4. We provide three samples for each input shot noise image using the Cascaded cGAP. We notice the diversity of the denoising decreasing with an increase in the number of photons in the input image.

Method	**PSNR** ↑	**SSIM** ↑	**FID** ↓	**LPIPS** ↓
CGAP	20.49 ± 3.05	0.78 ± 0.05	42.054	0.17 ± 0.05
CGAP Cascaded	**23.40±2.52**	**0.85±0.03**	**26.630**	**0.10±0.03**

Table 7.1: Quantitative evaluation (PSNR, SSIM, FID, LPIPS) on CelebA-HQ 256×256-1k validation dataset for Inpainting. **Bold**: best.

Figure 7.1: Qualitative Results for Inpainting using Conditional GAP and Cascaded Conditional GAP. This figure showcases the inpainting performance of our proposed models. The top row displays input images with masked regions. The second row presents inpainting results using the base conditional GAP model (CGAP). The third row shows results obtained with the cascaded conditional GAP model (Cascaded CGAP), demonstrating further improvements in reconstructing realistic and detailed facial features. The bottom row provides the ground truth images for reference.

7.2 Colorization

Generative CGAP

We provide the quantitative results for the tasks of Generative Colorization for CGAP and the Cascaded CGAP in Table 7.2. Similarly, we provide the qualitative results in the Figure 7.2. Here, we again notice that the Cascaded CGAP outperforms the cGAP model.

Figure 7.2: Qualitative Results for Colorization using Conditional GAP and Cascaded Conditional GAP. This figure presents colorization results for the CelebA dataset. The top row displays the grayscale input images. The second row showcases colorized outputs generated by the conditional GAP model (CGAP). The third row presents results obtained with the cascaded conditional GAP model (Cascaded CGAP), which further enhances the vibrancy and realism of the colors. The bottom row provides the ground truth color images for comparison.

Diversity Denoising CGAP

We utilize the Cascaded CGAP to perform diversity denoising for Colorization in Figure 7.5. Similarly, we provide three samples for varying shot noise images and notice the diversity decreasing with increased photon count in the input.

Method	**PSNR ↑**	**SSIM ↑**	**FID ↓**	**LPIPS ↓**
CGAP	22.18± 3.60	0.88 ± 0.03	39.937	0.14 ± 0.05
CGAP Cascaded	**23.83±3.76**	**0.94±0.03**	**26.685**	**0.09±0.04**

Table 7.2: Quantitative evaluation (PSNR, SSIM, FID, LPIPS) on CelebA-HQ 256×256-1k validation dataset for Colorization. **Bold**: best.

7.3 Super Resolution

Generative CGAP

We provide the quantitative results for the final module in Table 7.3 for both models. Qualitative results are provided in Figure 7.3. Like the other modules, we again see that Cascaded CGAP outperforms the CGAP model. While our model demonstrated promising diversity denoising results for other inverse problems, the application to Super Resolution did not yield significant diversity changes. This could be attributed to Super Resolution's nature, as the task of four times magnification is less complex. As a result, we focus our analysis on the quantitative results.

Figure 7.3: Qualitative Results for Super-resolution using Conditional GAP and Cascaded Conditional GAP. This figure presents 4×super-resolution results on the CelebA dataset. The top row displays low-resolution input images. The second row showcases the upscaled outputs generated by the conditional GAP model (CGAP). The third row presents results obtained with the cascaded conditional GAP model (Cascaded CGAP), exhibiting sharper details and improved visual fidelity. The bottom row provides the ground truth high-resolution images for reference.

Method	PSNR ↑	SSIM ↑	FID ↓	LPIPS ↓
CGAP	26.61 ± 3.33	0.79 ± 0.04	47.330	0.14 ± 0.04
CGAP Cascaded	**26.65±3.20**	**0.81±0.04**	**37.531**	**0.13±0.04**

Table 7.3: Quantitative evaluation (PSNR, SSIM, FID, LPIPS) on CelebA-HQ 256×256-1k validation dataset for Super Resolution. **Bold**: best.

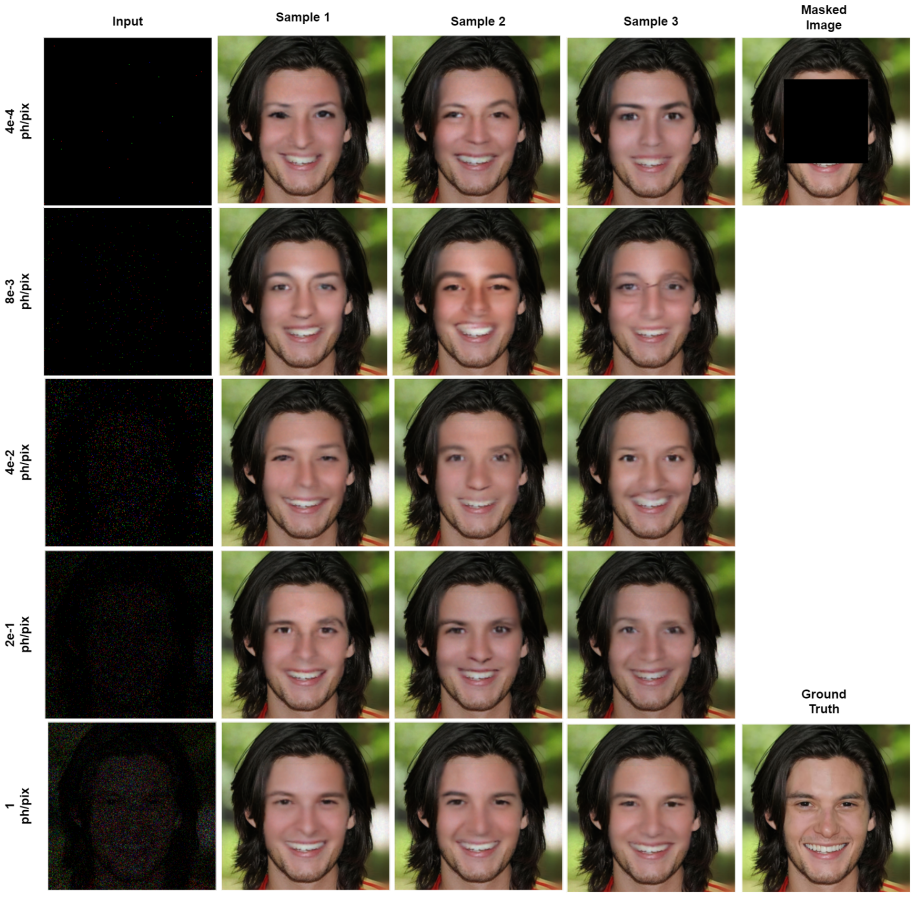

Figure 7.4: Diversity Denoising in Inpainting with Cascaded Conditional GAP. Here, we demonstrate the diversity denoising capabilities of the Cascaded Conditional GAP model across various shot noise levels. Each row represents a different input image degraded with Poisson noise at varying intensities (photons per pixel), ranging from extremely low (4e-4) to moderate (1). For each noisy input, the model generates three distinct plausible denoised outputs (Samples 1-3). The masked image (top right) indicates the inpainting region, while the Ground Truth (bottom right) is the original, noise-free image for reference.

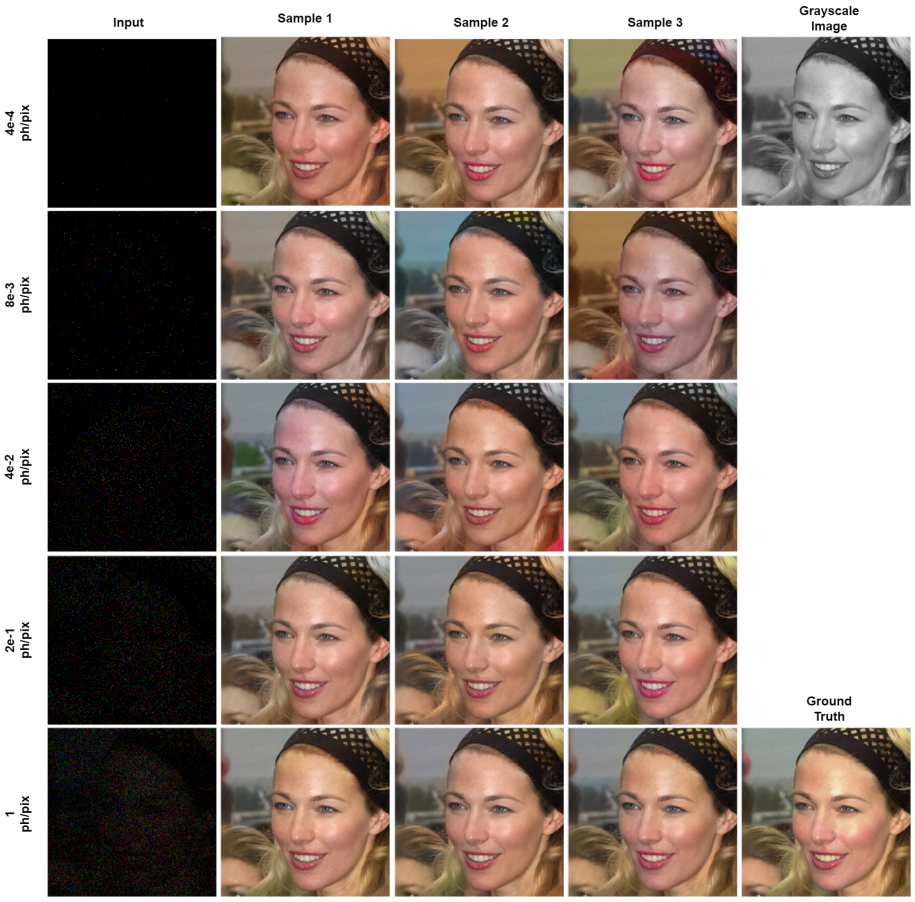

Figure 7.5: Diversity Denoising in Colorization with Cascaded Conditional GAP. This figure showcases the diversity in denoising and colorization achieved by the Cascaded Conditional GAP model under varying Shot noise levels. The grayscale image (top right) indicates the colorization condition, while the Ground Truth (bottom right) is the original image for reference.

CHAPTER 8

Discussion

In this chapter, we present our findings from experimenting with the Conditional GAP model for inverse problems. We begin by discussing the model's advantages and limitations. This is followed by an ablation study focusing on the impact of the denoising parameter. Finally, we conclude with suggestions for future research directions.

8.1 Analysis of Model Performance

8.1.1 Successes

Inpainting

The Conditional GAP model was implemented to tackle the challenging inverse problem of inpainting large masked regions, specifically masking out 25% of the image. Despite the complexity of this task, which is inherently more difficult than colorization or super-resolution, the model produced respectable results and the resultant outputs were visually convincing. The model also effectively generated diverse denoising solutions that progressively matched the ground truth as the noise level decreased.

Colorization

For the colorization task, the Conditional GAP model successfully captured realistic colors when tested on the CelebA-HQ dataset. Even though the dataset primarily contains images of faces, the model was able to generate vibrant and lifelike colors, enriching the scenes in the images. The ill-posed nature of the colorization task became apparent in the Diversity Denoising results, where the model could output multiple plausible solutions for a single grayscale input, showcasing its ability to handle uncertainty in color interpretation.

Super Resolution

In the case of 4× Super Resolution, the Conditional GAP model demonstrated strong performance by effectively reconstructing textures and fine-grained details from low-resolution inputs. The model managed to achieve solid evaluation scores while producing high-quality high-resolution outputs. These results highlight the model's capacity to recover intricate details and enhance low-resolution images.

The success in achieving these results paves a whole new area of generative modeling based on Poisson Noise and the capability of the Conditional GAP Model.

8.1.2 Limitations

Comparative Analysis

The base Conditional GAP model underperforms significantly, both quantitatively and qualitatively, across all modules when compared to the Cascaded Conditional GAP. This can be attributed to the fact that the base model is trained over the entire noise range of [-40, 30] on the FFHQ 256x256 dataset. The complexity of this dataset, combined with the wide range of shot noise, makes it difficult for the base Conditional GAP to generalize effectively. As a result, the model often makes suboptimal initial estimates for the generative process 5.3, which hinders its ability to denoise effectively. The Cascaded Conditional GAP addresses this issue by distributing the noise range across multiple models, each specializing in a smaller noise range, leading to superior performance.

Inpainting

Although Conditional GAP performs decently in inpainting, the model struggles with generating high-frequency details in the masked regions. This can be attributed to two factors: the noise range of the cascaded model and the nature of the generative process. The cascaded CGAP has a noise range of [-40, 10], which is narrower than the optimal range of [-40, 40] used in the original GAP model. Increasing the range by adding more models leads to overfitting, particularly in the case of inpainting. As shown in Figure 8.1, Diversity Denoising results on the CelebA-HQ dataset reveal a decline in performance when increasing the number of cascaded models. The downward trend in the graph indicates that the higher range model prefers real-shot noise images over the denoised-shot noise image. The trend becomes increasingly unstable with seven cascaded models compared to five. For stability, we limit the inpainting task to five cascaded models.

Additionally, the generative nature of Conditional GAP tends to favor low-frequency reconstructions. In Figure 8.1, the photon accumulation process is visualized for each inverse problem, where the shot noise input and the predicted photon distribution are compared. During the generative process We sample photons from the output and add them to the input shot noise image. However, during inpainting, the Conditional GAP makes a low-frequency estimate for the masked region and a high-frequency estimate for the unmasked region, which leads to the unmasked region getting sampled more often than the masked

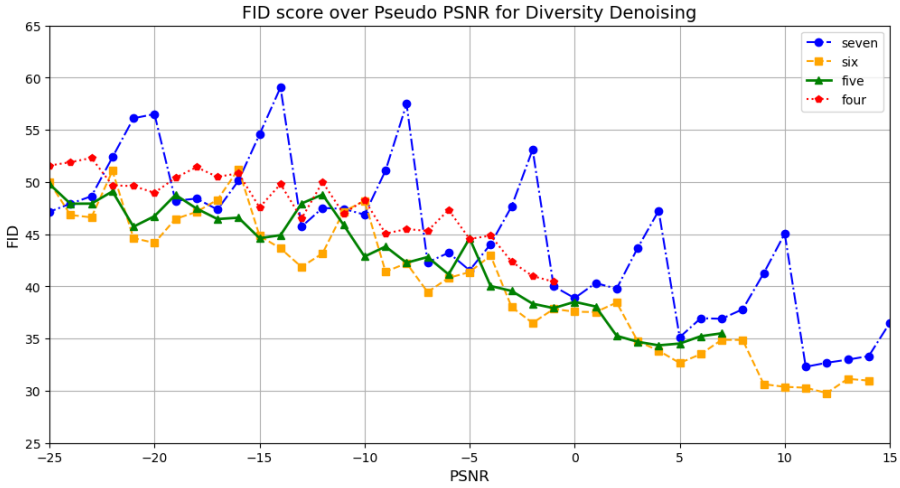

Figure 8.1: Comparison of FID Scores for Diversity Denoising for Different amounts of Cascades in the CGAP. The graph compares the FID scores of four different Cascaded CGAP models, "four," "five," "six," and "seven", denoting the number of models in the Cascade.

region. This imbalance leads to a significant shift in the shot noise image, deviating from the original training distribution.

Colorization

Colorization, when applied to the FFHQ dataset, presents a relatively more straightforward inverse problem due to the dataset primarily consisting of aligned face images. As a result, the Conditional GAP model tends to overfit to this specific task. While the model produces realistic and vibrant colors, its performance should be further evaluated on more diverse and complex datasets [24] to avoid overfitting and to test its generalization capabilities in colorization tasks.

Super Resolution

In the case of 4× Super Resolution, the Conditional GAP model struggles to reconstruct high-frequency details effectively. This limitation is particularly noticeable when recovering intricate details from low-resolution inputs. The model's ability to generate high-frequency data remains a challenge that needs further refinement.

Resources

One of the main limitations of the project was the computational budget to conduct longer experiments. To reduce the computational cost, several compromises were made regarding the model's architecture and training procedure. For instance, lower-resolution images were used for specific experiments and models were trained on an increased noise range to accommodate fewer models within the memory. This limited the exploration of the model's potential capability.

8.2 Albation Study

We investigate the importance of the denoising parameter β that affects both the generative process and the diversity denoising. The β controls the number of photons added to the input shot noise image in every iteration; a smaller β indicates a slower generative process, and a larger β indicates a faster generative process.

8.2.1 Denoising Scheduler β

We evaluate the Cascaded Conditional GAP model for varying β values starting from $1e-3$ and increasing by an order of 10.

8.2.2 Inpainting

We provide the quantitative results in the Table 8.1 and qualitative results at 8.2. We observe that the model favors a slower β in terms of perceptual quality favoring diversity over the average rough estimates.

Method	**PSNR ↑**	**SSIM ↑**	**FID ↓**	**LPIPS ↓**
CGAP $\beta = 10^2$	<u>23.82 ± 2.57</u>	<u>0.86 ± 0.03</u>	28.734	0.12 ± 0.04
CGAP $\beta = 10$	23.11 ± 2.77	**0.87±0.03**	28.546	0.13 ± 0.04
CGAP $\beta = 1$	**23.84±2.51**	0.85 ± 0.03	26.835	<u>0.11 ± 0.03</u>
CGAP $\beta = 1e^-1$	23.40 ± 2.52	0.85 ± 0.03	<u>26.630</u>	**0.10±0.03**
CGAP $\beta = 1e^-2$	22.74 ± 2.58	0.84 ± 0.04	**25.918**	**0.10±0.03**
CGAP $\beta = 1e^-3$	22.39 ± 2.51	0.84 ± 0.04	28.820	**0.10±0.03**

Table 8.1: Quantitative evaluation (PSNR, SSIM, FID, LPIPS) on CelebA-HQ 256×256-1k validation dataset for the Cascaded CGAP. **Bold**: best, <u>underline</u>: second best.

8.2.3 Colorization

We provide the quantitative results in the Table 8.2 and qualitative results at 8.2. For Colorization we observe that the model prefers larger β for the CelebA-HQ dataset, favoring average estimates over diversity.

Method	PSNR ↑	SSIM ↑	FID ↓	LPIPS ↓
CGAP $\beta = 10^2$	**24.61±3.69**	**0.95±0.02**	26.268	**0.08±0.05**
CGAP $\beta = 10$	24.26 ± 3.70	0.94 ± 0.02	**25.644**	**0.08±0.05**
CGAP $\beta = 1$	24.16 ± 3.51	0.94 ± 0.02	27.325	0.09 ± 0.04
CGAP $\beta = 1e^-1$	23.83 ± 3.76	0.94 ± 0.03	26.685	0.09 ± 0.04
CGAP $\beta = 1e^-2$	22.50 ± 3.40	0.92 ± 0.03	29.649	0.10 ± 0.05
CGAP $\beta = 1e^-3$	21.98 ± 3.56	0.91 ± 0.03	30.271	0.11 ± 0.05

Table 8.2: Quantitative evaluation (PSNR, SSIM, FID, LPIPS) on CelebA-HQ 256×256-1k validation dataset for the Cascaded CGAP. **Bold**: best, underline: second best.

8.2.4 Super Resolution

We provide the quantitative results in the Table 8.3 and qualitative results at 8.2. Here, we notice the model preferring β as 1 which is in between the two extremes.

Method	PSNR ↑	SSIM ↑	FID ↓	LPIPS ↓
CGAP $\beta = 10^2$	**27.87±2.83**	0.83 ± 0.04	36.769	0.15 ± 0.04
CGAP $\beta = 10$	27.60 ± 3.23	**0.84±0.04**	35.712	0.14 ± 0.04
CGAP $\beta = 1$	27.49 ± 3.06	0.83 ± 0.04	**35.554**	0.13 ± 0.04
CGAP $\beta = 1e^-1$	26.65 ± 3.20	0.81 ± 0.04	37.531	0.13 ± 0.04
CGAP $\beta = 1e^-2$	25.84 ± 3.17	0.78 ± 0.05	42.917	**0.12±0.03**
CGAP $\beta = 1e^-3$	25.60 ± 2.99	0.77 ± 0.05	45.296	**0.12±0.03**

Table 8.3: Quantitative evaluation (PSNR, SSIM, FID, LPIPS) on CelebA-HQ 256×256-1k validation dataset for the Cascaded CGAP. **Bold**: best, underline: second best.

8.3 Future Work

Several future directions can be explored to address the limitations faced in this project and improve the performance of Conditional GAP models for image inverse problems. Architectural enhancements can focus on designing networks that are better equipped to handle high-frequency details, which were challenging in tasks such as inpainting and super-resolution. This could involve incorporating frequency-aware architectures, which are specifically designed to process high-frequency information, or integrating wavelet-based CNNs to capture image details across multiple frequency scales [30]. Additionally, Attention mechanism [29, 39] popularized by the Transformers [54], can be used to dynamically weigh over different parts of the input data to represent richer features within the network [41] that can potentially lead to better reconstruction.

In order to manage varying range of noise levels we can explore curriculum learning [32]. Instead of training separate models for different noise levels, a

progressive noise training strategy could be employed, where the model starts with low noise levels and gradually introduces higher noise levels as training progresses. This strategy could result in a more robust and generalized understanding of noise.

Finally, a unified framework for handling multiple inverse problems could be developed akin to the Palette model [47]. Multi-task learning can enable a single Conditional GAP model to simultaneously understand and address multiple image reconstructiuon tasks.

These future directions directly address the current limitations of the model, particularly in high-frequency detail reconstruction and noise handling. The proposed methods have the potential to significantly enhance the performance and applicability of Conditional GAP models in solving a broader range of complex inverse problems.

Figure 8.2: Impact of Denoising Scheduler Parameter (β) on Qualitative Results. Each row corresponds to a specific task: inpainting, colorization, and super-resolution. The leftmost column displays the deformed input. The subsequent columns showcase model outputs generated with varying β values, ranging from $1e-3$ to 10^2. The rightmost column provides the ground truth image for reference.

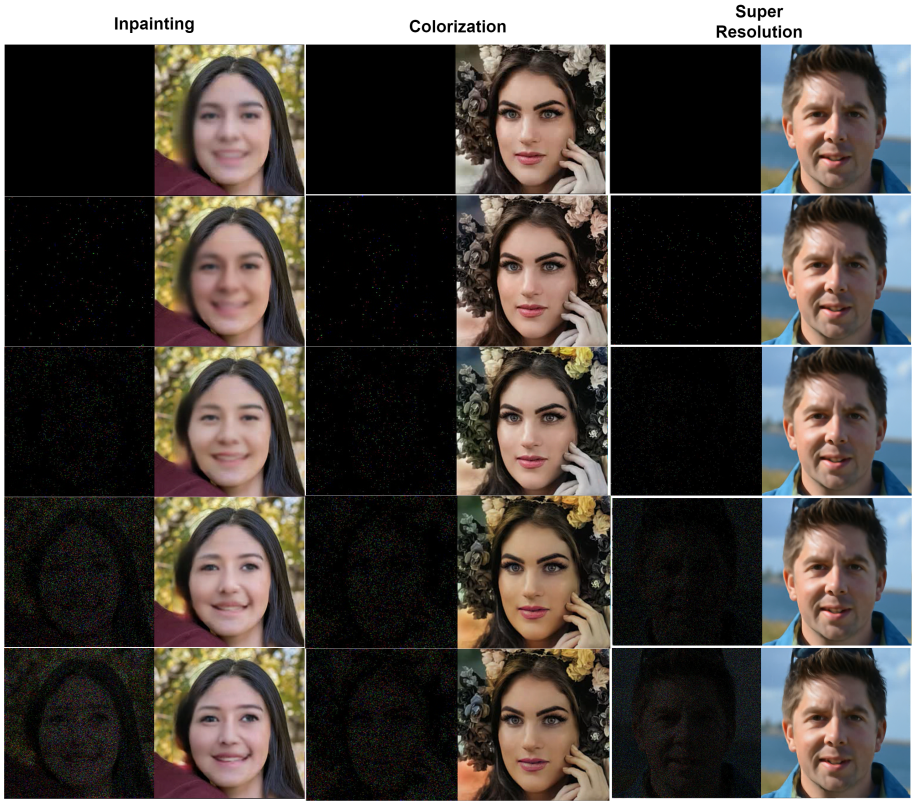

Figure 8.3: Accumulation of Photon in Cascaded Conditional GAP. This figure visualizes the photon accumulation process within our Cascaded Conditional GAP model for different inverse problems: inpainting (left column), colorization (middle column), and super-resolution (right column). Each row represents a different stage of photon accumulation, starting from an extremely low photon count. The figure demonstrates how the model gradually refines the generated image as more photons are incorporated, leading to a plausible solution for each inverse problem.

CHAPTER 9

Conclusion

In this project we present a comprehensive framework for solving image inverse problems specifically, inpainting, colorization, and super-resolution through the proposed Conditional GAP model. By introducing conditional inputs we demonstrated its capability to generate plausible and diverse solutions while addressing the challenges posed by the Invserse problems and the Poisson Noise. We experimentally proved that Poisson noise can be adapted into strong Generative Models and showcased the capability of the Generative Accumulation of Photons to solve complex inverse problems in the form of Conditional GAP model.

However, the study also highlighted some limitations, particularly in tasks requiring high-frequency detail reconstruction, such as inpainting and super-resolution, where the models faced challenges in capturing finer details. Despite these limitations, the results provide a promising foundation for further refinement and exploration of generative modeling approaches, particularly in dealing with Poisson noise and complex inverse problems. These findings make the way for future research in improving Poisson based generative models performance in diverse and challenging image restoration tasks.

Bibliography

[1] J. Ansel, E. Yang, H. He, N. Gimelshein, A. Jain, M. Voznesensky, B. Bao, P. Bell, D. Berard, E. Burovski, G. Chauhan, A. Chourdia, W. Constable, A. Desmaison, Z. DeVito, E. Ellison, W. Feng, J. Gong, M. Gschwind, B. Hirsh, S. Huang, K. Kalambarkar, L. Kirsch, M. Lazos, M. Lezcano, Y. Liang, J. Liang, Y. Lu, C. Luk, B. Maher, Y. Pan, C. Puhrsch, M. Reso, M. Saroufim, M. Y. Siraichi, H. Suk, M. Suo, P. Tillet, E. Wang, X. Wang, W. Wen, S. Zhang, X. Zhao, K. Zhou, R. Zou, A. Mathews, G. Chanan, P. Wu, and S. Chintala. PyTorch 2: Faster Machine Learning Through Dynamic Python Bytecode Transformation and Graph Compilation. In *29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2 (ASPLOS '24)*. ACM, Apr. 2024.

[2] C. Barnes, E. Shechtman, A. Finkelstein, and D. B. Goldman. Patchmatch: a randomized correspondence algorithm for structural image editing. *ACM Trans. Graph.*, 28(3), jul 2009.

[3] G. Batzolis, J. Stanczuk, C.-B. Schönlieb, and C. Etmann. Conditional image generation with score-based diffusion models.

[4] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester. Image inpainting. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '00, pages 417–424. ACM Press/Addison-Wesley Publishing Co.

[5] N. Bose, H. Kim, and H. Valenzuela. Recursive implementation of total least squares algorithm for image reconstruction from noisy, undersampled multiframes. In *1993 IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 5, pages 269–272 vol.5. ISSN: 1520-6149.

[6] H. Chung, J. Kim, M. T. Mccann, M. L. Klasky, and J. C. Ye. Diffusion posterior sampling for general noisy inverse problems.

[7] A. Deshpande, J. Lu, M.-C. Yeh, M. Jin Chong, and D. Forsyth. Learning diverse image colorization. pages 6837–6845.

[8] W. Falcon and The PyTorch Lightning team. PyTorch Lightning, Mar. 2019.

[9] P. Felzenszwalb, D. McAllester, and D. Ramanan. A discriminatively trained, multiscale, deformable part model. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. ISSN: 1063-6919.

[10] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial networks.

[11] S. W. Hasinoff. Photon, poisson noise. In K. Ikeuchi, editor, *Computer Vision: A Reference Guide*, pages 608–610. Springer US.

[12] J. Hays and A. A. Efros. Scene completion using millions of photographs. *ACM Trans. Graph.*, 26(3):4–es, jul 2007.

[13] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick. Masked autoencoders are scalable vision learners.

[14] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition.

[15] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter. GANs trained by a two time-scale update rule converge to a local nash equilibrium.

[16] J. Ho, A. Jain, and P. Abbeel. Denoising diffusion probabilistic models.

[17] J. Ho, C. Saharia, W. Chan, D. J. Fleet, M. Norouzi, and T. Salimans. Cascaded diffusion models for high fidelity image generation.

[18] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks.

[19] A. Jolicoeur-Martineau. The relativistic discriminator: a key element missing from standard GAN.

[20] T. Karras, T. Aila, S. Laine, and J. Lehtinen. Progressive growing of GANs for improved quality, stability, and variation.

[21] T. Karras, S. Laine, and T. Aila. A style-based generator architecture for generative adversarial networks.

[22] B. Kawar, M. Elad, S. Ermon, and J. Song. Denoising diffusion restoration models.

[23] D. P. Kingma and M. Welling. An introduction to variational autoencoders. 12(4):307–392.

[24] A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc.

[25] A. Krull, H. Basevi, B. Salmon, A. Zeug, F. Müller, S. Tonks, L. Muppala, and A. Leonardis. Image denoising and the generative accumulation of photons.

[26] A. Krull, T.-O. Buchholz, and F. Jug. Noise2void - learning denoising from single noisy images.

[27] J. Lehtinen, J. Munkberg, J. Hasselgren, S. Laine, T. Karras, M. Aittala, and T. Aila. Noise2noise: Learning image restoration without clean data.

[28] A. Levin, D. Lischinski, and Y. Weiss. Colorization using optimization. *ACM Trans. Graph.*, 23(3):689–694, aug 2004.

[29] B. Liu and I. Lane. Attention-based recurrent neural network models for joint intent detection and slot filling.

[30] P. Liu, H. Zhang, W. Lian, and W. Zuo. Multi-level wavelet convolutional neural networks.

[31] Z. Liu, P. Luo, X. Wang, and X. Tang. Deep learning face attributes in the wild.

[32] W. Lotter, G. Sorensen, and D. Cox. A multi-scale CNN and curriculum learning strategy for mammogram classification. In M. J. Cardoso, T. Arbel, G. Carneiro, T. Syeda-Mahmood, J. M. R. Tavares, M. Moradi, A. Bradley, H. Greenspan, J. P. Papa, A. Madabhushi, J. C. Nascimento, J. S. Cardoso, V. Belagiannis, and Z. Lu, editors, *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*, pages 169–177. Springer International Publishing.

[33] Q. Luan, F. Wen, D. Cohen-Or, L. Liang, Y.-Q. Xu, and H.-Y. Shum. Natural image colorization. In *Proceedings of the 18th Eurographics Conference on Rendering Techniques*, EGSR'07, page 309–320, Goslar, DEU, 2007. Eurographics Association.

[34] A. Lugmayr, M. Danelljan, A. Romero, F. Yu, R. Timofte, and L. Van Gool. RePaint: Inpainting using denoising diffusion probabilistic models.

[35] T. maintainers and contributors. TorchVision: PyTorch's Computer Vision library, Nov. 2016.

[36] S. Marcel and Y. Rodriguez. Torchvision the machine-vision package of torch. In *Proceedings of the 18th ACM International Conference on Multimedia*, MM '10, page 1485–1488, New York, NY, USA, 2010. Association for Computing Machinery.

[37] F. Murtagh. Multilayer perceptrons for classification and regression. 2(5):183–197.

[38] K. Nazeri, E. Ng, and M. Ebrahimi. Image colorization with generative adversarial networks. volume 10945, pages 85–94.

[39] Z. Niu, G. Zhong, and H. Yu. A review on the attention mechanism of deep learning. 452:48–62.

[40] K. O'Shea and R. Nash. An introduction to convolutional neural networks.

[41] X. Pan, C. Ge, R. Lu, S. Song, G. Chen, Z. Huang, and G. Huang. On the integration of self-attention and convolution.

[42] R. Pascanu, T. Mikolov, and Y. Bengio. On the difficulty of training recurrent neural networks.

[43] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros. Context encoders: Feature learning by inpainting.

[44] M. V. Perera and V. M. Patel. Analyzing bias in diffusion-based face generation models.

[45] M. Prakash, M. Delbracio, P. Milanfar, and F. Jug. Interpretable unsupervised diversity denoising and artefact removal.

[46] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation.

[47] C. Saharia, W. Chan, H. Chang, C. A. Lee, J. Ho, T. Salimans, D. J. Fleet, and M. Norouzi. Palette: Image-to-image diffusion models.

[48] C. Saharia, J. Ho, W. Chan, T. Salimans, D. J. Fleet, and M. Norouzi. Image super-resolution via iterative refinement.

[49] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition.

[50] N. Singh, S. S. Rathore, and S. Kumar. Towards a super-resolution based approach for improved face recognition in low resolution environment. 81(27):38887–38919.

[51] J. Song, C. Meng, and S. Ermon. Denoising diffusion implicit models.

[52] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2818–2826. IEEE.

[53] M. Tancik, P. P. Srinivasan, B. Mildenhall, S. Fridovich-Keil, N. Raghavan, U. Singhal, R. Ramamoorthi, J. T. Barron, and R. Ng. Fourier features let networks learn high frequency functions in low dimensional domains.

[54] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need.

[55] J. Wang, H. Lu, Z. Liang, D. Eremina, G. Zhang, S. Wang, J. Chen, and J. Manzione. An experimental study on the noise properties of x-ray CT sinogram data in radon space. 53(12):3327–3341.

[56] X. Wang, K. Yu, S. Wu, J. Gu, Y. Liu, C. Dong, C. C. Loy, Y. Qiao, and X. Tang. ESRGAN: Enhanced super-resolution generative adversarial networks.

[57] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli. Image quality assessment: from error visibility to structural similarity. 13(4):600–612. Conference Name: IEEE Transactions on Image Processing.

[58] M. Weigert, U. Schmidt, T. Boothe, A. Müller, A. Dibrov, A. Jain, B. Wilhelm, D. Schmidt, C. Broaddus, S. Culley, M. Rocha-Martins, F. Segovia-Miranda, C. Norden, R. Henriques, M. Zerial, M. Solimena, J. Rink, P. Tomancak, L. Royer, F. Jug, and E. W. Myers. Content-aware image restoration: pushing the limits of fluorescence microscopy. 15(12):1090–1097.

[59] Z. Wojna, V. Ferrari, S. Guadarrama, N. Silberman, L.-C. Chen, A. Fathi, and J. Uijlings. The devil is in the decoder: Classification, regression and GANs.

[60] Y. Xie, M. Yuan, B. Dong, and Q. Li. Diffusion model for generative image denoising.

[61] J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu, and T. Huang. Free-form image inpainting with gated convolution.

[62] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang. The unreasonable effectiveness of deep features as a perceptual metric.

[63] Y. Zhang, Y. Zhu, E. Nichols, Q. Wang, S. Zhang, C. Smith, and S. Howard. A poisson-gaussian denoising dataset with real fluorescence microscopy images. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11702–11710. ISSN: 2575-7075.

[64] M. Özbey, O. Dalmaz, S. U. Dar, H. A. Bedel, Özturk, A. Güngör, and T. Çukur. Unsupervised medical image translation with adversarial diffusion models.

Appendix

Github repository

The code used in this thesis is publicly accessible in a GitLab repository located at https://git.cs.bham.ac.uk/projects-2023-24/axa2274. The repository is structured to organize code by the specific inverse problem being addressed: colorization, inpainting, and super-resolution. Each task-specific directory contains Python files defining the dataset class, the conditional GAP model architecture, training scripts, testing scripts, and inference scripts for both standard and cascaded models. Additionally, a tasks.py file contains utility functions for the inpainting task, and a comprehensive README.md file provides an overview of the project and contains explanations on how to run the code. The repository utilizes Python 3.5 and Due to the computational demands of deep learning models, access to a GPU is highly recommended for efficient training and evaluation.