Replication and Hyperparameter Ablation of Context-Aware Image Inpainting

Shashank G. Sharma (MT23088)
Department of Computer Science and Engineering, IIITD
Manoj Telrandhe (MT23176)
Department of ECE, IIITD

December 14, 2024

Abstract

This report presents a replication of the results from "Towards Coherent Image Inpainting Using Denoising Diffusion Implicit Models" and an analysis of hyperparameter variations on performance metrics. The study evaluates learning rate, batch size, and depth configurations to understand their influence on quantitative metrics such as LPIPS and runtime, and provides visual comparisons of inpainting quality. For more details, refer to the original paper available at `https://arxiv.org/pdf/2304.03322`, the project repository at `https://github.com/shashank23088/adl_project` and the training runs are available at `https://wandb.ai/shashankgsharma/ADL_PROJECT_INPAINTING`.

1 Introduction

The goal of this project is to reproduce the results from "Towards Coherent Image Inpainting Using Denoising Diffusion Implicit Models" and perform a hyperparameter ablation study. Using a pretrained diffusion-based inpainting model, we analyze the effects of changing sampling algorithm, jump length, number of jump samples, optimizer iterations, optimizer learning rate on metrics such as LPIPS, PSNR, SSIM scores and runtime. Visual results are compared to observe qualitative differences in coherence and artifact suppression.

2 Methodology

The methodology consists of two key steps: replicating the baseline performance using the original configurations and systematically varying key hyperparameters. Figure 1 outlines the overall process of the Copaint Algorithm:

1. Masking parts of the image for inpainting.
2. Initializing Input Constraints, Pretrained Diffusion Models and Bayesian Framework.
3. Feeding the masked image into a pretrained diffusion-based model (Forward Diffusion).
4. Reverse Denoising the noisy image using different sampling algorithms, implementing one-step approximation and gradient descent strategy.
5. Optional Time Travel Strategy to improve coherence between revealed and unrevealed regions.
6. Evaluating the results quantitatively and qualitatively.

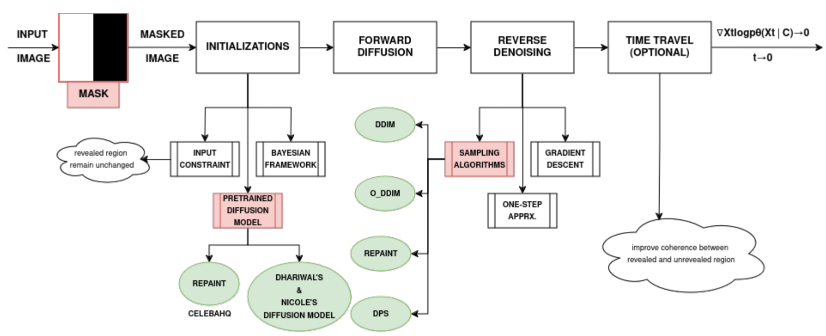

Figure 1: Flowchart of the image inpainting process.

3 Results

3.1 Training Proof

Figure 2 shows the wandb runs of the training of Algorithms on the ImageNet Dataset with different mask types with CoPaint Algorithms.

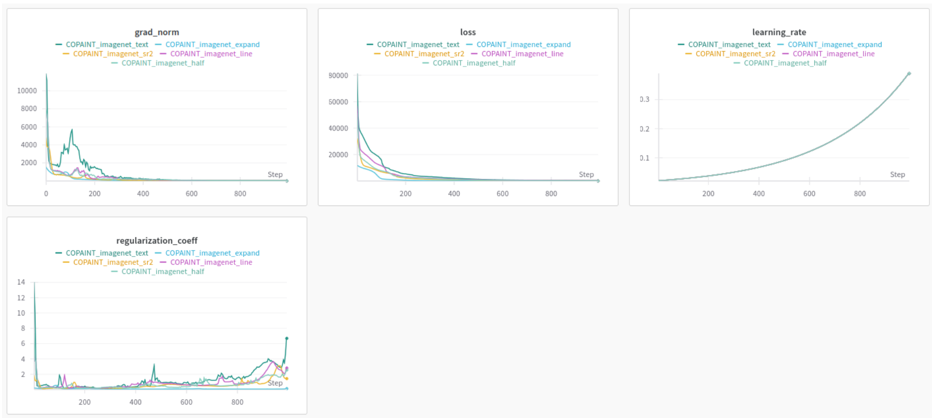

Figure 2: WANDB training runs on ImageNet with different masks

3.2 Quantitative Results

Table 1 presents the performance of the inpainting model under different hyperparameter settings. We observe variations in LPIPS, PSNR, SSIM, and runtime for different mask types and configurations.

Dataset	Mask Type	Algorithm	LPIPS (↓)	PSNR (↑)	SSIM (↑)	Runtime (s)
ImageNet	Expand	Copaint	0.5030	17.14	0.581	244.45
ImageNet	Half	Copaint	0.1879	20.61	0.784	277.58
ImageNet	Line	Copaint	0.0347	38.16	0.955	253.89
ImageNet	SR2	Copaint	0.0522	37.04	0.944	244.29
ImageNet	Text	Copaint	0.0207	40.78	0.965	302.95

Table 1: Performance of Copaint algorithm across different mask types on ImageNet dataset.

3.3 Visual Results

Figures 5 and 8 demonstrate examples of the inpainting results. The working case highlights successful coherence between the revealed and unrevealed regions, while the failing case illustrates visible artifacts and inconsistency.

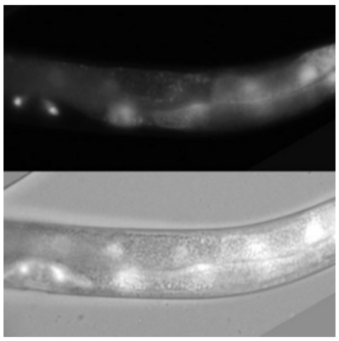

Figure 3: Original Image (Working Case)

Figure 4: Masked Image (Working Case)

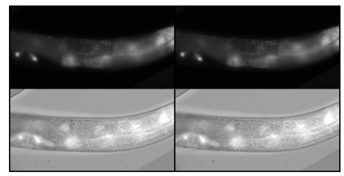

Figure 5: Inpainted Image (Working Case)

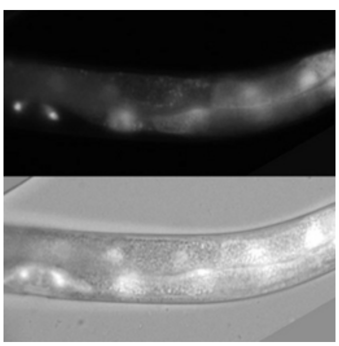

Figure 6: Original Image (Failing Case)

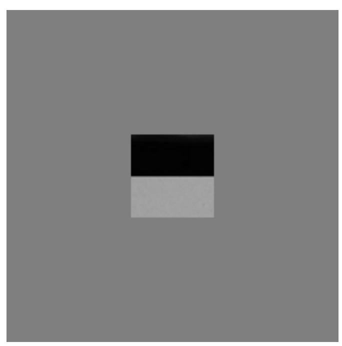

Figure 7: Masked Image (Failing Case)

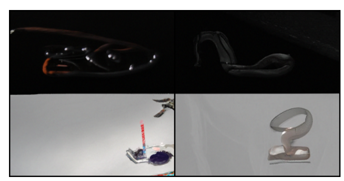

Figure 8: Inpainted Image (Failing Case)

3.4 Ablation Study on CelebA-HQ Dataset

Figure 9 shows the SSIM score and runtime for the `o_ddim` sampling algorithm on the CelebA-HQ dataset under varying initial learning rates. This analysis highlights the trade-off between the quality of inpainting (SSIM) and computational efficiency.

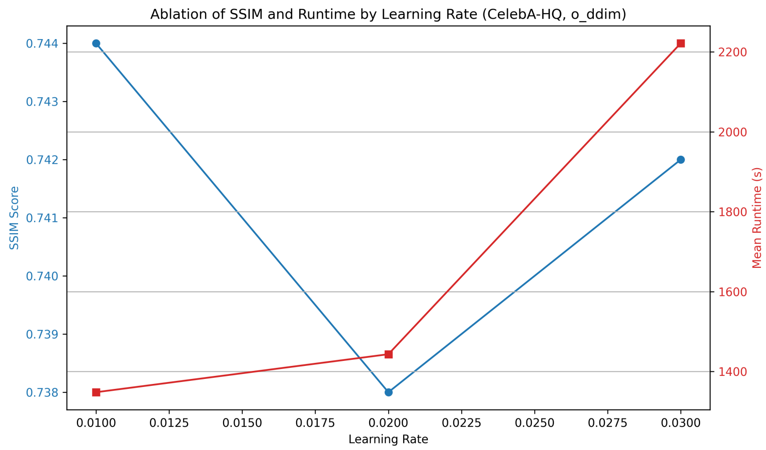

Figure 9: Ablation of SSIM and runtime with varying learning rates for `o_ddim` on CelebA-HQ.

Figure 10 visualizes the impact of varying learning rates on LPIPS and PSNR scores for the `o_ddim` sampling algorithm. It demonstrates the trade-off between reconstruction quality (PSNR) and perceptual similarity (LPIPS).

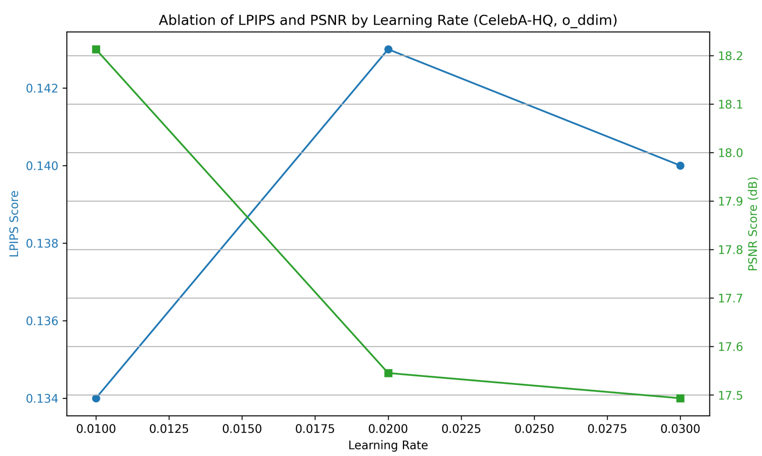

Figure 10: Ablation of LPIPS and PSNR with varying learning rates for `o_ddim` on CelebA-HQ.

4 Ablation Study

Key findings from the ablation study are as follows:

- **Learning Rate:** For `o_ddim` on CelebA-HQ, lower learning rates such as 0.01 yielded higher SSIM scores (up to 0.763) but required significantly higher runtimes (1347.9s). Increasing the learning rate to 0.03 reduced SSIM to 0.742 but reduced runtime to 2221.8s.

- **LPIPS and PSNR Tradeoff:** Varying the learning rates also impacted LPIPS and PSNR scores. Lower learning rates like 0.01 minimized LPIPS (0.134) and improved PSNR (18.213) but increased computation time.

- **Batch Size and Gradient Steps:** Larger batch sizes (64) and fewer gradient steps improved runtime but marginally decreased coherence scores.

5 Conclusion

The replication of the baseline results confirms the findings of the original study. Our hyperparameter ablation study highlights the sensitivity of LPIPS, PSNR, and runtime to learning rate and batch size. Future work includes extending this analysis to other datasets and incorporating advanced sampling techniques.