

Enhancing Inpainting Performance on Ocean Scenes through Fine-Tuning

2018013309 김승희, 2014010919 이준홍

1 Introduction

1.1 Inpainting 기술 개요

Inpainting은 이미지에서 누락된 영역을 주변 문맥 정보를 활용하여 자연스럽게 복원하거나 특정한 방식으로 채우는 기술이다. 이 과정은 손상된 예술작품의 복원, 사진에서 불필요한 객체 제거, 영상 편집 등 다양한 용도로 사용된다. 전통적으로는 픽셀 기반의 필터링 기법이나 텍스처 합성 기법이 사용되었지만, 최근에는 딥러닝 모델을 기반으로 더 복잡하고 세밀한 작업이 가능해졌다.

1.2 동작 원리

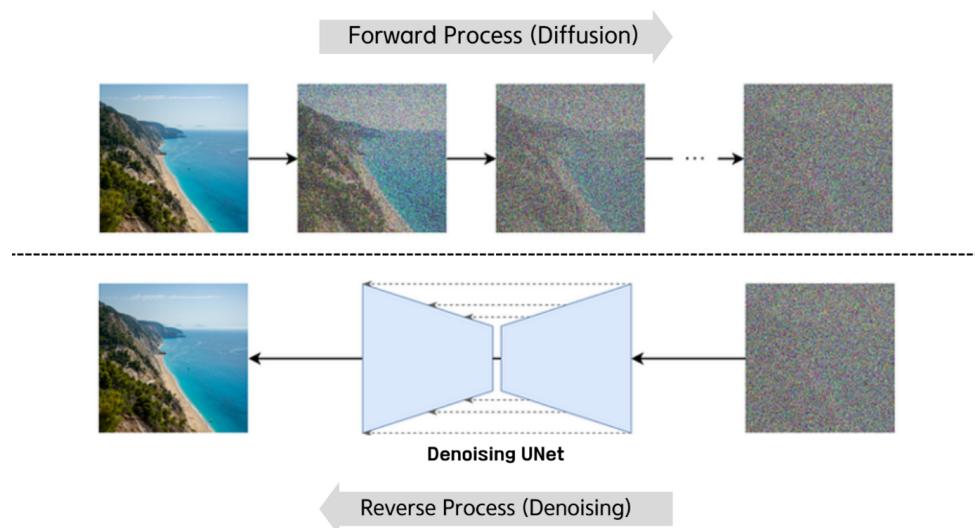


Figure 1: diffusion 모델 동작 원리

Inpainting 기술은 **Diffusion Model**을 포함한 다양한 딥러닝 모델을 활용하여 수행된다. Diffusion Model은 노이즈를 점진적으로 제거하는 과정을 통해 이미지를 생성하는 방식으로, 누락된 영역을 복원하는 데 강력한 성능을 발휘한다. Diffusion 모델의 학습 과정은 크게 두 단계로 나누어지는데, Forward Process(Diffusion, 데이터에 점진적으로 노이즈를 추가하여 정규 분포(가우시안 노이즈)로 만드는 과정)와 Reverse Process(Denoising, 노이즈 데이터를 다시 원래 데이터로 복원하는 과정)를 거쳐 이미지의 구조적 및 문맥적 정보를 학습하게 된다. 최종적으로 입력 이미지와 유사한 텍스처 및 패턴을 생성하여 복원 이미지를 제공하게 된다. Diffusion Model은 복잡한 구조와 패턴을 가진 이미지에서도 높은

품질의 복원을 제공하고, 동일한 입력 조건에서 다중 해를 제공하며, 학습이 안정적이고 모드 붕괴(Mode Collapse)와 같은 문제를 겪지 않는다는 장점이 있다. 하지만 모델의 설계가 복잡하고 연산량이 높으며, 대규모의 학습 데이터가 필요하다는 문제점도 있다. 이를 해결하기 위한 여러 변형 모델들이 제안되고 있다. Latent Diffusion Model은 Diffusion Model의 계산 효율성을 개선하기 위해 고안된 모델이다. 이 접근법은 원본 이미지 공간이 아닌 임베딩 공간(Latent Space)에서 Diffusion 과정을 수행함으로써 연산량을 크게 줄인다. Dynamic Diffusion은 손상된 영역의 크기와 위치에 따라 복원 과정을 동적으로 조정하는 방법이다. 이를 통해 복잡한 구조에서도 효율적이고 세밀한 복원이 가능하게 된다.

1.3 활용 분야

Inpainting 기술은 다양한 응용 사례에서 중요한 역할을 한다:

- 이미지 복원: 역사적 자료나 손상된 예술작품 복원을 위한 기술로 사용된다. 예를 들어, 훼손된 그림의 빈 부분을 복구하거나 오래된 사진에서 손상된 영역을 재구성할 수 있다.
- 불필요한 객체 제거: 사진에서 원치 않는 물체나 사람을 제거하는 데 활용된다. 예를 들어, 여행 사진에서 지나가는 행인이나 방해 요소를 제거하여 더 깨끗한 이미지를 만들 수 있다.
- 영화 및 영상 편집: 영상의 특정 장면에서 객체를 제거하거나 추가하는 작업에 활용된다. 이러한 기술은 특히 영화 제작 및 광고 편집에서 필수적인 도구로 자리 잡을 수 있다.
- 예술 및 창의적 활용: 인공지능을 활용하여 새로운 스타일의 이미지를 생성하거나, 기존 이미지에 창의적인 변화를 줄 수 있다. 예를 들어, 새로운 배경을 추가하는 등의 작업이 가능하다.

이 외에도 의료 영상 처리, 위성 이미지 분석, 증강현실(AR) 및 가상현실(VR) 콘텐츠 제작 등 다양한 분야에서도 inpainting 기술이 활용되고 있다.

2 Problem Statement

2.1 주요 Inpainting 모델의 한계점

우리는 Inpainting 모델의 성능 한계점을 분석하기 위해 다양한 사례를 실험적으로 분석하였다. 특히, huggingface Inpainting 코드의 baseline인 kandinsky-2-2-decoder-inpaint 모델과, 또 다른 최신 모델인 stable-diffusion-2-inpainting 모델을 기반으로 한계점을 분석했다. 분석 결과, 현존하는 Inpainting 모델은 특히 바다를 배경으로 한 이미지에서 사람을 제거하는 작업이 제대로 수행되지 않는다는 점을 확인하였다. 모델은 마스킹된 영역에 대해서 인물을 잘 제거하지 못하고 다른 인물로 대체하는 모습을 보였으며, 제거를 하더라도 바다 특유의 복잡한 텍스처를 자연스럽게 복원하지 못했다. 따라서 우리는 모델의 파인튜닝을 통해 이 한계점을 해결하고자 했다.

2.2 문제 예시

Figure 2은 현재 주요 Inpainting 모델의 한계점을 보여주는 예시이다. 두 모델의 결과를 비교해 보면, 각 모델이 가진 한계점이 명확히 드러난다. kandinsky-2-2-decoder-inpaint 모델은 사람을 제거하는 데 어느 정도 성공적인 모습을 보였으나, 제거된 영역에서 바다의

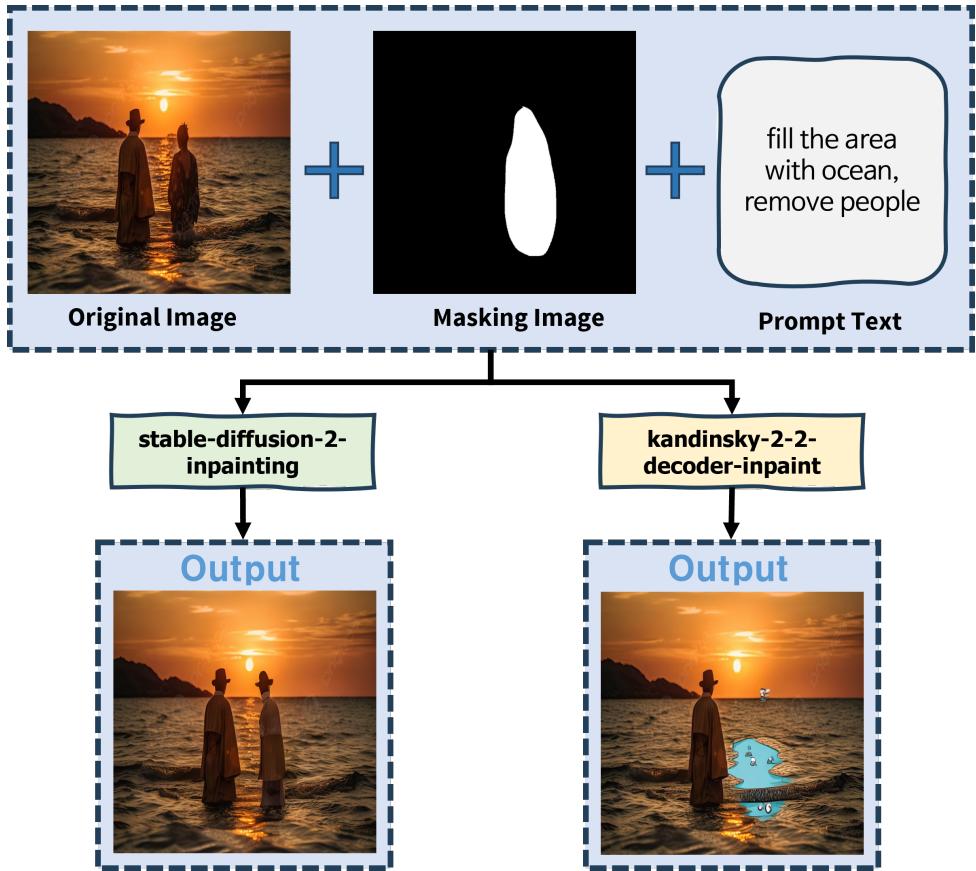


Figure 2: 바다 배경에서 Inpainting 모델의 결과 비교

텍스처를 자연스럽게 복원하지 못하고 부자연스러운 왜곡 현상을 보였다. stable-diffusion-2-inpainting 모델은 사람을 제거하는 작업 자체에 자주 실패하며, 제거된 영역이 다른 사람이나 사물로 대체되는 경향을 보였다. 하지만 이 모델은 주변 바다 텍스처를 비교적 자연스럽게 복원하는 데 강점을 보였다. 따라서 우리는 학습 대상 모델로 stable-diffusion-2-inpainting을 선정하였다. 이는 해당 모델이 사람을 제거하는 작업은 실패했으나, 바다와 같은 복잡한 배경 텍스처를 복원하는 데 있어 상대적으로 우수한 결과를 보였기 때문이다. 이를 기반으로, 사람 제거 성능과 텍스처 복원 성능을 동시에 향상시키기 위한 Fine-Tuning 전략을 적용하고자 한다.

3 Methods

3.1 학습 데이터셋 구축

우리는 바다에 특화된 파인튜닝을 위해 바다 배경 이미지를 포함하는 데이터셋을 수집하였다. 이를 위해 Hugging Face에 존재하는 `louiscklaw/beach_512` 데이터셋을 사용하였다. 해당 데이터셋은 8,361장의 바다 이미지를 포함하고 있어, 학습 데이터의 소스로 적합하다고 판단하였다. 다만, 이 데이터셋은 이미지 복원 작업에 필요한 마스킹 이미지를 제공하지 않고 바다 이미지 파일만 포함하고 있었다. 이를 해결하기 위해, 우리는 랜덤 마스킹 생성 알고리즘을 구현하여 각 이미지에 마스킹을 추가하는 방식을 사용하였다. 구체적으로, 각 이미지에 무작위로 위치와 크기가 다른 마스킹 영역을 생성하여 학습에 필요한 입력-출력 쌍을 구축하였다.

3.2 학습 데이터셋 예시



Figure 3: 랜덤 마스킹 알고리즘을 적용시킨 학습 데이터셋 샘플

3.3 모델 파인튜닝을 위한 GPU 실험 환경 구축

stable-diffusion-2-inpainting 모델을 파인튜닝하기 위해 Colab GPU 환경을 우선적으로 검토하였으나, 메모리 부족 문제로 인해 파인튜닝 작업을 수행할 수 없음을 확인하였다. 이에 따라, 보다 높은 성능의 GPU를 사용할 수 있는 대안을 모색하였다. 조사 결과, 클라우드를 통해서 GPU를 온라인으로 대여할 수 있다는 것을 찾게 되었고, vast.ai 플랫폼을 활용하는 방안을 선택하였다. vast.ai는 사용자에게 GPU 서버 인스턴스를 시간 단위로 대여할 수 있는 서비스를 제공하며, A100 GPU를 시간당 약 0.5달러로 이용할 수 있다는 점에서 비용 효율성이 뛰어난 것으로 판단되었다. 이를 활용하여 파인튜닝 작업을 진행할 수 있는 실험 환경을 구축하였다.

3.4 모델 파인튜닝 진행

수집한 데이터셋을 활용하여 기존 stable-diffusion-2-inpainting 모델을 파인튜닝하여 바다 배경의 마스킹 복원 성능을 개선하였다.

3.4.1 실험 환경 설정

모델 파인튜닝은 PyTorch 프레임워크와 Hugging Face의 Diffusers 라이브러리를 기반으로 수행되었다. GPU 가속을 위해 NVIDIA A100 GPU 환경을 사용하였으며, 주요 실험 환경 설정은 아래의 Table 1과 같다:

Table 1: 실험 환경 설정

항목	설명
모델	stabilityai/stable-diffusion-2-inpainting
이미지 크기	512 × 512 픽셀로 고정
배치 크기	9
학습률	5×10^{-7}
학습 에포크	3
마스킹 증강 횟수	10배 증강 (각 이미지당 마스킹 버전 10개 생성)
데이터셋 분할	학습 80%, 검증 10%, 테스트 10%

3.4.2 학습 데이터셋 및 마스킹 생성

8,361장의 바다 이미지 중, 학습 속도와 자원 효율성을 고려하여 500장의 이미지를 무작위로 선택하였다. 모델 입력 데이터로 사용하기 위해 이미지에 랜덤 마스킹을 적용하였으며, 다양한 도형(타원, 사각형, 다각형)을 포함한 마스킹 이미지를 생성하였다. 마스킹의 개수는 각 이미지당 최대 5개의 랜덤 도형을 포함하도록 설정하였다. 마스킹 증강 횟수가 10이기 때문에 총 5,000장의 이미지를 이용하게 된다.

3.4.3 데이터 분할 및 평가 전략

데이터셋은 학습(train), 검증(validation), 테스트(test) 세 가지로 분할하여 데이터 로더를 생성하였다. 각 데이터셋의 분할 비율은 80% (학습), 10% (검증), 10% (테스트)로 설정하였다. 테스트 데이터셋은 모델의 최종 성능 평가에 사용하였다.

3.4.4 모델 구성 및 학습 프로세스

파인튜닝에는 Hugging Face의 StableDiffusionInpaintPipeline을 사용하였다. 모델의 주요 구성 요소는 다음과 같다:

- UNet: 노이즈 제거를 위한 주요 아키텍처
- VAE (Variational Autoencoder): 이미지의 잠재 표현을 생성
- CLIP Text Encoder: 텍스트 임베딩을 생성하여 컨텍스트 정보를 제공

각 구성 요소의 파라미터는 FP32로 유지하며, AdamW 최적화 알고리즘을 사용하였다.

3.4.5 학습 손실 함수

손실 함수로는 예측된 노이즈와 실제 노이즈 간의 평균제곱오차(MSE)를 사용하였다. 학습 과정은 GradScaler를 활용한 Mixed Precision Training으로 수행하여 GPU 메모리 사용량을 최적화하였다.

3.4.6 결과 및 평가

학습 데이터셋으로 모델을 학습한 후, 검증 데이터셋에서 평균 손실 값을 기반으로 성능을 평가하였으며 테스트 데이터셋을 통해 모델의 최종 성능 평가를 진행했다.

4 Experimental Results

4.1 Train Loss, Validation Loss

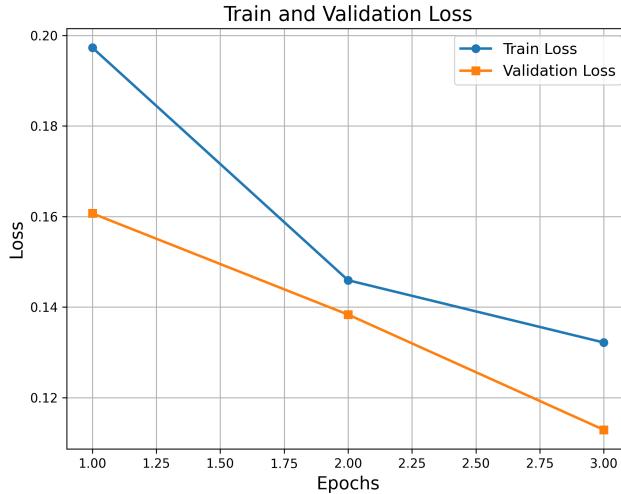


Figure 4: 파인튜닝 과정의 Train Loss, Validation Loss Chart

Figure 4와 같이 학습 과정에서 Train Loss와 Validation Loss가 에포크가 진행됨에 따라 적절히 감소하는 양상을 보였다. 이는 모델이 안정적으로 학습되었음을 시사한다. 특히, Train Loss와 Validation Loss 모두 일정하게 감소하며, 학습이 진행되는 동안 두 값이 급격히 증가하거나 불안정한 변화를 보이지 않아 Overfitting이 발생하지 않았음을 확인할 수 있었다. 이러한 결과는 파인튜닝 과정이 성공적으로 수행되었음을 보여준다.

4.2 결과 비교

7page의 Figure 5과 Figure 6은 파인튜닝이 되지 않은 바닐라 모델과 파인튜닝된 모델의 Inpainting 결과를 비교한 샘플들이다. 바닐라 모델의 경우, Inpainting 작업에서 사람을 제거하는 데 실패하거나, 제거된 영역이 부자연스럽게 표현되는 사례가 빈번히 관찰되었다. 반면, 파인튜닝된 모델은 사람을 효과적으로 제거함과 동시에 바다의 텍스처를 자연스럽게 복원하는 결과를 보였다. 특히, 파인튜닝된 모델은 바다의 복잡한 텍스처를 보다 일관성 있게 재현하며, 바닐라 모델 대비 높은 품질의 결과물을 생성하였다. 이는 파인튜닝 과정을 통해 모델이 바다 배경의 텍스처와 패턴을 더 잘 학습했음을 시사한다. 다만, 여전히 사람의 형상이 미세하게 남아있는 샘플이 일부 존재하며, 완벽한 학습이 이루어졌다고 평가하기에는 한계가 있다. 그럼에도 불구하고, 파인튜닝된 모델이 바닐라 모델에 비해 개선된 성능을 보였다는 점에서 위 실험 결과는 의미가 있다고 할 수 있다.

4.3 Evaluation Metric

Inpainting 결과를 단순히 사람의 눈으로 판단하는 것은 주관적이며 기준이 애매한 경우가 많다. 따라서, 모델이 얼마나 잘 Inpainting을 수행했는지 평가하기 위해 정량적 기준을 설정하는 것이 중요하다. 이를 위해 Inpainting 관련 연구 논문을 조사한 결과, SSIM (Structural Similarity Index Measure)과 LPIPS (Learned Perceptual Image Patch Similarity)가 대표적인 평가 지표로 사용된다는 것을 확인하였다.

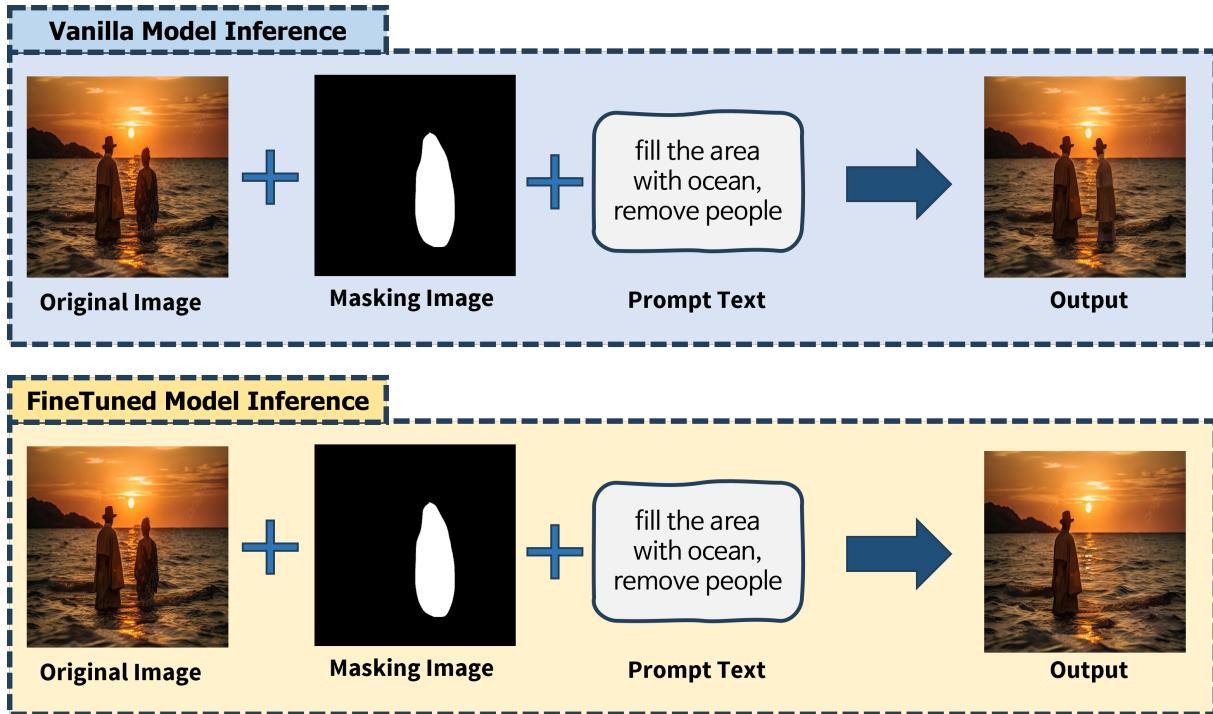


Figure 5: 바닐라 모델과 파인튜닝된 모델의 결과 비교

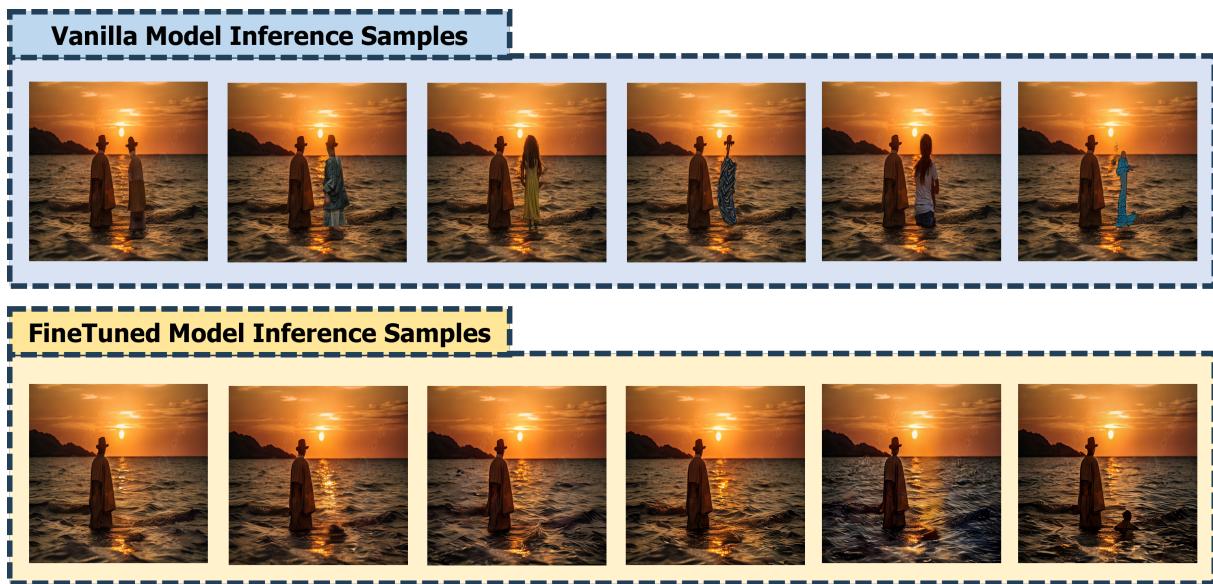


Figure 6: 바닐라 모델과 파인튜닝된 모델의 결과 샘플

SSIM (Structural Similarity Index Measure) SSIM은 두 이미지 간의 구조적 유사성을 측정하는 지표로, 밝기, 대비, 구조의 세 가지 요소를 고려하여 계산된다. 이 지표는 Inpainting 결과가 원본 이미지와 얼마나 비슷한지를 평가하는 데 자주 사용된다. SSIM 값은 $[0, 1]$ 범위를 가지며, 값이 1에 가까울수록 두 이미지가 구조적으로 유사하다는 것을 의미한다.

LPIPS (Learned Perceptual Image Patch Similarity) LPIPS는 사람의 시각적 인식을 기반으로 두 이미지 간의 유사성을 측정하는 지표이다. 딥러닝 모델을 활용하여 이미지의 고차원 특징을 비교하며, SSIM보다 인간의 주관적 판단과 더 잘 맞는 결과를 제

공하는 것으로 알려져 있다. LPIPS 값은 $[0, 1]$ 범위를 가지며, 값이 낮을수록 두 이미지가 더 유사하다는 것을 의미한다. 이 지표는 Inpainting 결과가 시각적으로 얼마나 자연스러운지를 평가하는 데 유용하다.

보완적 활용 SSIM과 LPIPS는 각각 구조적 유사성과 시각적 자연스러움을 측정하기 때문에 Inpainting 결과의 품질을 평가할 때 서로 보완적으로 사용될 수 있다. 이러한 지표를 활용하여 Test Dataset에 대한 Evaluation을 진행하였으며, 이를 기반으로 파인튜닝 코드를 수정하여 모델 성능을 개선하였다.

5 Failure Case

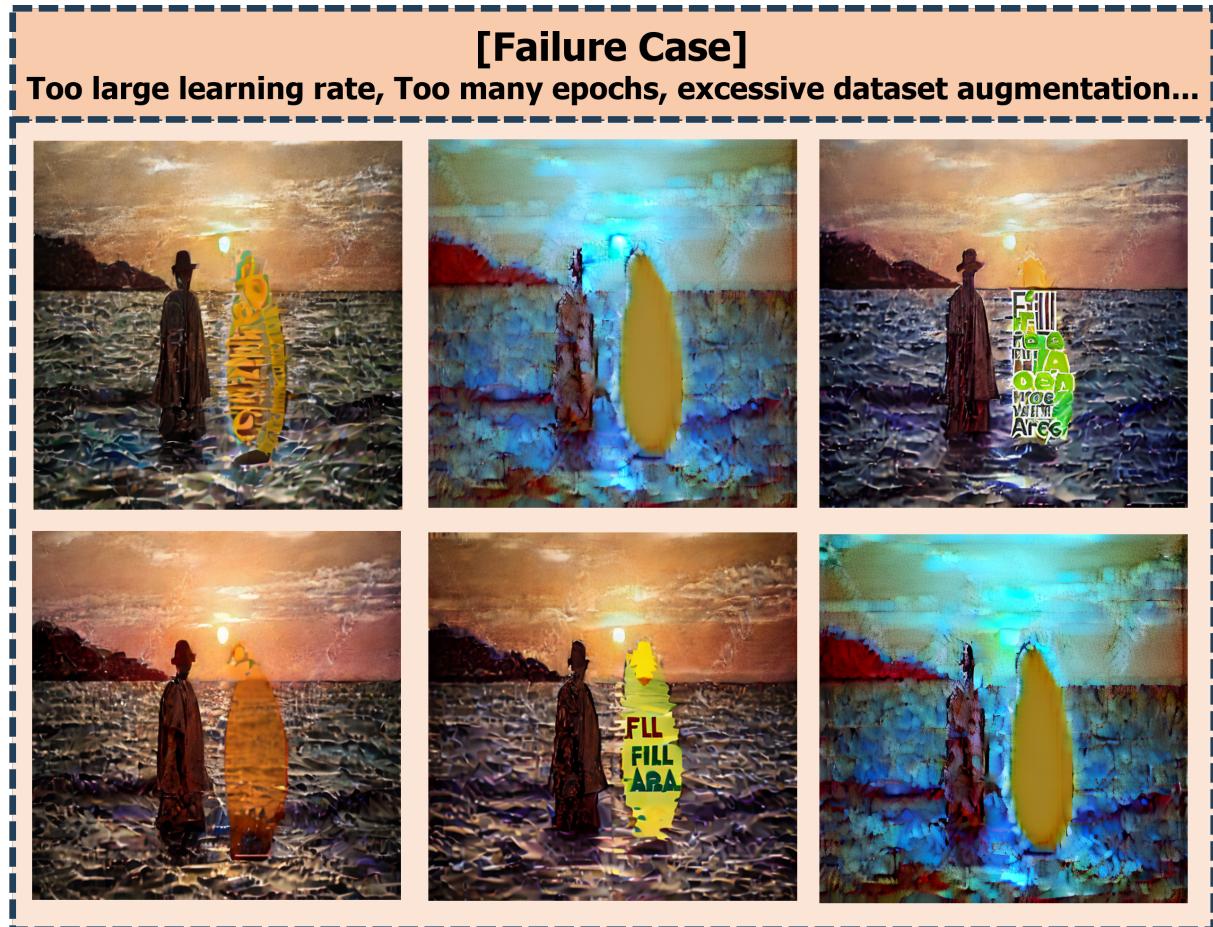


Figure 7: Failure Cases

위 Figure 7는 파인튜닝 과정에서 관찰된 실패 사례를 나타낸다. 이러한 실패 사례는 주로 지나치게 큰 학습률, 과도한 학습 에포크 수, 그리고 데이터 증강의 과도한 적용과 같은 설정으로 인해 발생하였다. 이로 인해 모델이 본연의 성능을 잃고, Inpainting 작업을 적절히 수행하지 못하는 결과를 보였다. 이러한 실패 사례를 통해, 학습률, 에포크 수, 데이터 증강의 수준 등 다양한 하이퍼파라미터를 조정하며 최적의 파인튜닝 설정(Configuration)을 탐색할 수 있었다. 이러한 Trial-and-Error 과정은 모델의 성능을 안정적으로 향상시키기 위한 필수적인 단계임을 알 수 있었다.

6 Conclusion

본 프로젝트에서는 기존 Inpainting 모델의 한계를 극복하고, 바다 배경 이미지를 포함한 복잡한 텍스처 복원 성능을 개선하기 위해 데이터셋 구축 및 모델 파인튜닝을 수행하였다. Hugging Face의 [louiscklaw/beach_512](#) 데이터셋을 활용하여 랜덤 마스킹 증강 알고리즘을 구축하여 학습 데이터를 생성하였으며, stable-diffusion-2-inpainting 모델을 기반으로 성능을 개선하고자 하였다. 실험 결과, 파인튜닝된 모델은 기존 바닐라 모델 대비 사람 제거와 바다 텍스처 복원 작업에서 우수한 성능을 보였다. 특히, 바다와 같은 복잡한 배경에서도 인물 제거 기능과 함께 자연스러운 텍스처 복원을 수행하였다. 다만, 일부 사례에서는 사람의 형상이 완전히 제거되지 않는 등 여전히 개선의 여지가 있음을 확인하였다. 이러한 결과는 추가적인 데이터 증강, 하이퍼파라미터 최적화 등을 통한 학습이 필요함을 보인다.

7 Team Role Distribution

- 2018013309 김승희: Inpainting 모델의 한계점 분석, 바다 이미지 데이터셋 수집, 데이터셋 랜덤 마스킹 알고리즘 코드 구현, 모델 파인튜닝 코드 구현, 실험, Failure Case 분석, 보고서 작성, PPT 제작, 발표
- 2014010919 이준홍: Inpainting 모델 작동 원리 분석, Inpainting 실사용 분야 조사, 온라인 GPU 대여 서비스 조사, Inpainting Evaluation Metric 조사, SSIM, LPIPS Metric 코드 구현