# Final Project Progress Report

Nik Bhatt UNI: nb3097

## Overview

My midterm paper opened my eyes to the state of art for inpainting GANs, particularly for image synthesis. That said, I wonder about the general applicability of these newer algorithms.

During my literature review, I read a paper by Xiang, et al. [1] that tested many different algorithms with the same data sets in an attempt to provide a consistent basis of comparison. I would like to take that idea further, to see how well the latest algorithms perform in the object removal use case, using a completely unseen dataset.

## Opportunities for novelty

I identified four main areas that are advancements over the Xiang paper:

| | Xiang | Mine | Benefit |
|---|---|---|---|
| **Datasets** | ImageNet, CelebA, CelebA-HQ Places, and Paris StreetView. (These are the same datasets that the models are trained against.) | OpenImages v7 (https://storage.googleapis.com/openimages/web/download_v7.html) | OpenImages is a completely novel dataset to these models, eliminating the chance of contamination or overfitting from the training process. |
| **Models** | A wide range of models up to 2021 | Newer models up to 2023 | Only one model (the "control") is analyzed in the other paper. |
| **Masks** | NVidia's Irregular Mask set, which is a random mask generator. | Object Detection masks generated by Facebook's Detectron2. | These masks represent what end users would want to use an inpainting algorithm for. |
| **Reusability** | Xiang does not provide any code. | I will provide a reusable library on GitHub | Researchers can use my library to evaluate their design and others without reinventing the wheel. |

I would compute three metrics: Fréchet Inception Distance (FID), Learned Perceptual Image Patch Similarity (LPIPS), and Structural Similarity (SSIM). These are the primary metrics used in current inpainting research. I would start by replicating the findings in the papers, using their pre-trained models. Then I would switch to the new dataset and masks.

## Changes since the proposal:

1. My original plan was predicated on an incorrect assumption. I assumed that I could easily reuse models from various papers without requiring much, if any, code from the researchers. I have found that the models are tightly coupled with the implementations in python. I have spent many fruitless hours trying to detach models in a way that would let me use them outside of the researchers' repos. As a result, I will not be converting them to CoreML. In fact, I will be leaving them in situ.
2. I have also had a lot of difficulty getting some of the GANs to run at all on my Mac, primarily due to assumptions about CUDA. For example, one model is half-float and only runs on CUDA, with no CPU fallback. This may limit the number of GANs that I can evaluate. I have found three so far: HiFill, MI-GAN, and CoModGAN. This should still provide enough variety to answer the research questions.

3. Based on my experience with HiFill, I thought I could run these GANs at different resolutions, but it appears that some of them are locked to specific (small) input and output sizes. So, I will only be able to run the models the way they are trained.
4. That said, the basics of my project and novelty are unchanged. I will be using a different data set and masks. I will be using newer GANs, and I will try to use different resolutions when possible. I also added something: a way to make my work much more reusable by other researchers.

**Research Questions**

1. Are these GANs able to remove objects with good quality results, or has that been diminished in the race for image synthesis?
2. Can I reproduce the results (FID, LPIPS, and SSIM) as described in the papers using the data sets that the researchers use?
3. How different are the results from their data sets vs. the results from using the OpenImages dataset?
4. What does the difference tell us? If the OpenImages results are worse, does that imply overfitting / overtraining? Is there an explanation if the OpenImages results are better?
5. Which recent algorithms are best at object removal with an unseen dataset, both in terms of quality and runtime? If possible, can they handle larger images on a standard laptop without retraining?

**Value to the User Community**

As I have worked on this project, I have identified a core problem that inpainting researchers have. Namely, the ability to analyze and compare the results from other papers. There are two steps to that. First, is getting the other GANs to run at all. Second, is getting output and verifying the metrics. The first problem I cannot solve because most of the issues are related to system configurations and the lack of reusability. However, the second problem is solvable. So, my target users are the inpainting GAN researchers. I will be providing a python project with the following features:

- Easy configurability. This is important because each GAN uses different ways of storing originals, masks, and generated images. For example, some GANs use masks that are bitwise-inverted from others.
- In addition to flexible storage options, I am employing an adapter pattern for finding and processing images. A researcher writes a simple adapter subclass in python, and my library uses it to process the output from the target GAN. By testing a few different GANs, I will be able to determine the necessary adapter methods.
- The ability to calculate FID, LPIPS, and SSIM, along with medians.
- Automatic generation of a PNG that is the original image with the mask translucently composited, like this:

- Generation of an animated GIF that shows the original, the above PNG, and the generated image to make it easier to visualize the inpainting result.
- Some set of charts and/or graphs for the LPIPS and SSIM.
- Output from the library for the GANs I have analyzed. I have not determined what that output will be yet.

**Why would users use your project?**
This project will make it easier for researchers to evaluate their GANs against existing designs. They can even use it to help with visualizing and analyzing their own design. The use of a separate dataset (OpenImages) also provides an alternative to the few datasets commonly in use.

**Demo:**
Since this mostly runs via the command line, there won't be a fun demo. I plan to show some examples of the output from my library - both the animated GIFs and some of the charts. I also will provide the results of my analysis of a few GANs - how similar my results are to theirs (both with their datasets and mine). I anticipate that there will some deviation between my results and theirs. I might also describe how my library works and can be used by researchers.

**Delivery:**
I will be delivering the library via GitHub repo. I plan to provide a read me that describes to use it. Datasets will not be included, as they are downloaded from other locations on the internet.

**References:**
[1] Xiang, et al. Deep learning for image inpainting: A survey. In *Pattern Recognition, Volume 134, Feb 2023.* https://www.sciencedirect.com/science/article/abs/pii/S003132032200526X

[2] Sargsyan, et al. MI-GAN: A Simple Baseline for Image Inpainting on Mobile Devices. In *International Conference on Computer Vision* (ICCV), 2023. https://openaccess.thecvf.com/content/ICCV2023/papers/Sargsyan_MI-GAN_A_Simple_Baseline_for_Image_Inpainting_on_Mobile_Devices_ICCV_2023_paper.pdf