Image "Outpainting" and Hole Filling: Progress Report

CS 5787 Deep Learning Final Project Milestone

Wentao Ye
wy335@cornell.edu
Cornell University

Mitchell Krieger
mak483@cornell.edu
Cornell University

Sebastian Jay
srj63@cornell.edu
Cornell University

ACM Reference Format:
Wentao Ye, Mitchell Krieger, and Sebastian Jay. 2024. Image "Outpainting" and Hole Filling: Progress Report: CS 5787 Deep Learning Final Project Milestone. In . ACM, New York, NY, USA, **??** pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

TEAM MEMBERS

Wentao Ye (wy335), Mitchell Krieger (mak483), Sebastian Jay (srj63)

PROPOSAL

Traditionally, image interpolation or inpainting tries to fill in a removed section of an image. There are a variety of methodologies for image interpolation, but our hope is to attempt a new approach where we can interpolate a scene between two disparate images by using image outpainting or image extrapolation. Outpainting or image extrapolation is the process of taking an image and generating an extension of the image beyond the original border. In general, outpainting is a difficult task because it requires the model to expand beyond a known region to generate scenes that look real. Our idea is to take two images and see if we can train a GAN-like architecture to outpaint these two images towards one another to complete a scene. We will be basing this project on the work done in Generalised Image Outpainting with U-Transformer by Gao et. al (2022), which leveraged a transformer-based GAN they call a U-transformer. In the past, image outpainting only focused in the horizontal direction. However, in this paper, Gao et al, were able to outpaint horizontally and vertically. We hope to use this architecture to train a new GAN-based approach that can interpolate diagonally adjacent image gaps. This is different from interpolation in the past which was able to use the context of the surrounding image to fill in a removed section. Instead, our focus is on seeing if the model can understand how to connect different scenes in a realistic way. If successful, it means that the model understands relationships between disparate images and can model the visual world in a more precise way.

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM
https://doi.org/10.1145/nnnnnnn.nnnnnnn

DATASETS

We used the NS-Outpainting dataset to train our models. It is used in the original U-Transformer paper. We also tested our trained model on additional images sourced from the internet.

PROGRESS

1. Reproduction of the paper

Below is an example of what we were able to achieve within 20 epochs, to fill in an image using the original U-Transformer model.

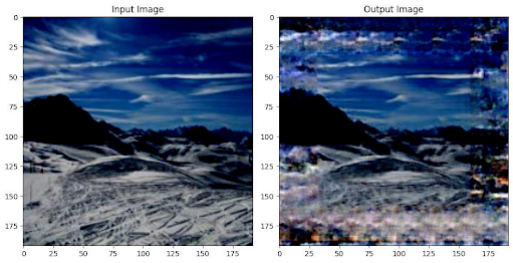

Figure 1: Image generated using U-Transformer after 20 epochs.

The U-transformer has the following shortcomings:

- Fixed size 192
- Expensive, requires an NVIDIA A100 GPU in Google Colab to train.
- Slow, using an RTX 2070 takes 3 hours per epoch, i.e. 1500 hours for 500 epochs.
- Hard code in the Swintransformer model with the center logic so we have to update everything inside the model if we want to crop with different regions.

2. New GAN Solution

The U-Transformer is based on Swin-transformer, which is quite slow, so we want to make it faster and cheaper.

Specifically, we use a GAN framework featuring a U-Net GAN generator and a PatchGAN discriminator for image outpainting and scene interpolation. The generator's encoder-decoder with skip connections creates missing regions, while the discriminator evaluates patch realism. Training optimizes adversarial and reconstruction losses using the Adam optimizer, enabling seamless scene integration. Its benefits are as follows:

- Fast, low memory usage, can run on a single RTX 2070
- It achieved the results shown below after 20 epochs, training for 3 hours
- 256×256 size

- Weights
- Code (Google Drive)

(a) Generated (b) Input (c) Original

Figure 2: Results from the new GAN model (L-R: Generated, Input, Original).

(a) Generated (b) Input (c) Original

Figure 3: Results from the new GAN model (L-R: Generated, Input, Original).

(a) Generated (b) Input (c) Original

Figure 4: Results from the new GAN model (L-R: Generated, Input, Original).

3. New Diffusion Model

In addition, we looked into additional papers that use diffusion models for tasks like this instead of GANs such as "RePaint: Inpainting using Denoising Diffusion Probabilistic Models" by Lugmayr et al (2022). Using this research we built an initial diffusion model that inpaints the NS-outpainiting dataset using diffusion. This model was trained on pairs of images and varying masks to represent unknown regions. In the training process, noise is gradually added to only the area where the masks are and MSE is used to attempt to predict this noise over 1000 timesteps. Then the diffusion model attempts to denoise only the pixels that are in the mask. After training for 14 epochs, the diffusion approach shows promising results.

- Trained on NVIDIA A100 in Google Colab, training took 2.66 hours for 14 epochs
- 320 × 512 Image size
- Weights
- Code (Google Drive)

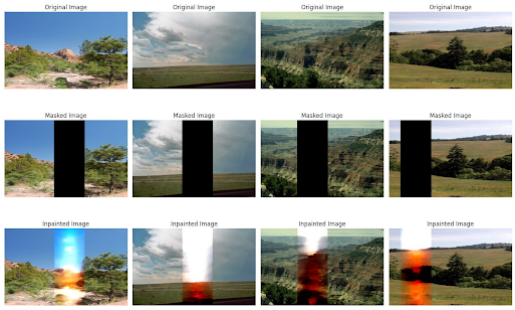

Figure 5: Original, masked, and inpainted images using diffusion model.

NEXT STEPS

We would like to train our model on more data, including on images from the RealEstate10K dataset. We'd like to figure out how to make our model predict only the missing parts of the image, rather than regenerating the entire input image including its intact portions.

We will use the following metrics to evaluate our results:

- **FID (Frechet Inception Distance)**: Assesses the similarity between the distributions of generated and real images, measuring both quality and diversity.
- **PSNR (Peak Signal-to-Noise Ratio)**: Evaluates the reconstruction quality by quantifying the pixel-level differences between generated and original images.
- **SSIM (Structural Similarity Index)**: Measures the structural similarity and perceptual quality between generated and ground truth images.
- **IS (Inception Score)**: Evaluates the clarity and diversity of generated images using a pre-trained Inception model.

Additional TODOs for GAN:

- Randomize location and size of cropped areas in training images (hard based on GAN)
- Bigger image size
- Smaller crop size
- Generate only the missing part, instead of using mask (hard based on GAN)
- Use two different images as each input sample
- Update loss function to optimize the performance

Additional TODOs for Diffusion:

- Train for additional epochs
- More masks of varying shapes
- Use Wasserstein Loss to better anchor generated inpaints to the original image, and penalize excess RGB color divergence in output image from input

- Better tune Gaussian Blur to make the boundary between the inpaint and the original model less stark.