

# Image “Outpainting” and Hole Filling: Final Report

CS 5787 Deep Learning Final Project Report

Wentao Ye  
wy335@cornell.edu  
Cornell University  
New York, New York, USA

Mitchell Krieger  
mak483@cornell.edu  
Cornell University  
New York, New York, USA

Sebastian Jay  
srj63@cornell.edu  
Cornell University  
New York, New York, USA

## ACM Reference Format:

Wentao Ye, Mitchell Krieger, and Sebastian Jay. 2024. Image “Outpainting” and Hole Filling: Final Report: CS 5787 Deep Learning Final Project Report. In . ACM, New York, NY, USA, 8 pages. <https://doi.org/10.1145/nnnnnnn.nnnnnnn>

## TEAM MEMBERS

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

## INTRODUCTION

This paper explores the relationship between image inpainting and outpainting by training a model capable of interpolating between two disparate images, blending them seamlessly into a single coherent scene. Inpainting, or image interpolation, is a computer vision task that aims to fill in missing or removed sections of an image, ensuring the completed area integrates smoothly with the existing content. Outpainting, or image extrapolation, generates extensions of an image beyond its original borders.

Drawing inspiration from both inpainting and outpainting, our objective is to generate a transitional region between two images. Traditional inpainting requires an understanding of the context and semantics surrounding the missing area to blend edges seamlessly into the original image. Outpainting, while sharing these challenges, has less context to infer from since it involves extending the image into an unknown space. Additionally, outpainting must handle long-range semantic dependencies, ensuring the generated extensions remain consistent with the original image no matter how far the extrapolation extends. Our task incorporates these challenges and introduces an added complexity: the need for the generated region to be semantically consistent with both images.

## RELATED WORK

To address the challenges of our task, we leverage deep learning architectures capable of capturing both the semantics at the edges of each image and the relationships between regions across the two images. Historically, Convolutional Neural Networks (CNNs) have been widely used in vision tasks due to their efficiency and ability to learn spatial relationships [7, 8]. However, CNNs have inherent limitations stemming from their design. Their architecture emphasizes locality, with weight sharing across the entire input, which makes them less adept at capturing long-range dependencies or relationships between non-local regions—a key requirement for complex tasks like ours.

Transformer-based architectures, particularly those employing attention mechanisms, have shown effectiveness in both inpainting and outpainting due to their capacity to model long-range dependencies. For example, [5] demonstrated that incorporating a contextual attention layer significantly improved inpainting performance. Vision Transformers (ViTs) introduced by [2] expanded on this by applying self-attention mechanisms to computer vision tasks, enabling the modeling of global relationships in an image. Further advancements, such as Swin Transformers [9], refined the transformer architecture for vision tasks using hierarchical computation and shifted windows to improve efficiency and adaptability. These approaches, however, remain computationally expensive and require large datasets for effective training [1].

Yang et al. (2019) proposed a U-Net GAN architecture to perform very long outpainting of a scene in one direction. They used Skip Horizontal Connections to connect each layer of the encoder and decoder in the U-Net and an LSTM-based Recurrent Transfer Network to transfer the encoded sequences to the decoder. Using this method and generating in multiple steps, they were able to demonstrate long outpainting. Lu et al. (2021) expanded on this work by combining inpainting, outpainting, and image blending to fill in a scene between two images horizontally. They introduced a similar U-Net GAN architecture to Yang et al. but also incorporated contextual attention and a Bidirectional Content Transfer module, which used LSTMs as a bottleneck to ensure spatial and semantic consistency across two images. Our work generalizes this approach by extending it to painting between two images in multiple directions.

---

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

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

© 2024 Copyright held by the owner/author(s). Publication rights licensed to ACM.

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.

## METHODS

Our primary objective is to create a system that can generate a coherent transitional region between two images, effectively producing a seamless blend. The solution we propose combines insights from both inpainting and outpainting tasks. In particular, we aim to generalize the notion of completing missing image regions (inpainting) to a setting where the "hole" is defined by two non-overlapping image segments: one at the top-right corner and another at the bottom-left corner. By doing so, we effectively create a scenario where the model must understand and synthesize a large intermediate area that harmonizes with both input segments.

This section first outlines our overall framework and then discusses the detailed architecture of the generator and discriminator, our training procedure, and the loss functions used to ensure high-quality, contextually coherent image synthesis.

### Overall Framework

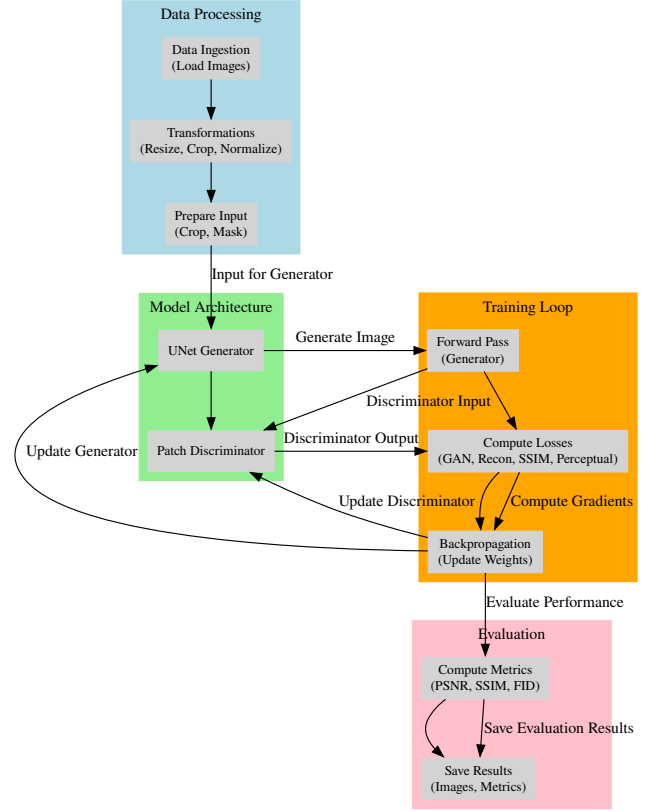
We frame our problem as learning a conditional generative model capable of taking two (possibly disparate) image patches as input and producing the missing middle region that stitches them together. While traditional inpainting models focus on filling small masked areas within a single image, we require the ability to fill a large gap that is spatially extended and semantically complex—one that must maintain coherence across potentially very different input segments.

In our approach, we rely on a Generative Adversarial Network (GAN) framework to produce plausible completions. The key intuition is that the discriminator will guide the generator to not only produce visually appealing local textures, but also maintain global consistency. We refine this idea by using a PatchGAN discriminator, which encourages detailed local realism, and a U-Net-based generator with skip connections to retain crucial spatial information.

Figure 1 shows the general high-level process of our method. Given two partial images—one cropped from the upper-right quadrant and another from the lower-left quadrant—we concatenate them along a predetermined boundary and feed the resulting partial image (with a large missing region in the center) into our generator. The generator synthesizes the missing portion, producing a final, coherent image. The discriminator then evaluates local patches of this completed image to ensure fidelity and realism.

### Generator Architecture

**U-Net-based Generator:** Our generator is inspired by the U-Net architecture, originally introduced for biomedical image segmentation [15]. The U-Net design is well-suited to our task because it uses a symmetric encoder-decoder structure with



**Figure 1: Overview of the general process.** Two separate image patches (A and B) are combined with a large missing region in the center. Our system synthesizes a realistic, coherent image that bridges the gap between these two input patches.

skip connections that maintain spatial details. The encoder progressively extracts increasingly abstract and semantically rich features, while the decoder gradually reconstructs the image, guided by skip connections that help preserve local structure and high-frequency details lost in downsampling.

Unlike traditional U-Nets used in segmentation, our generator must handle a complex conditional input: a partially completed scene with missing sections. We adapt the network to take in our concatenated partial images and to output a completed image of the same size. The final layer of the generator uses a Tanh activation, ensuring that output values lie in the range  $-1, 1$  to match the normalized input image range.

Figure 2 illustrates the architecture of our U-Net generator. On the left, we show the downsampling path, which encodes the input into a deep latent representation. On the right, the corresponding upsampling path decodes this latent representation back into a full-resolution image, with skip connections bridging corresponding layers.

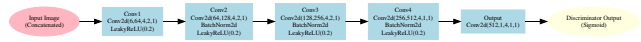


**Figure 2: The U-Net-based generator architecture.** The input is fed into a series of convolutional layers that progressively downsample and extract features, and then the decoded representations are combined with earlier features through skip connections to produce the final completed image.

## Discriminator Architecture

**PatchGAN Discriminator:** A traditional image-level discriminator that outputs a single scalar value for real/fake judgment can sometimes ignore subtle local details. To encourage the model to produce high-quality textures and local patterns, we adopt a PatchGAN discriminator [4]. Instead of a single output, the PatchGAN outputs an  $N \times N$  grid of values, each corresponding to the realism of a local patch in the image. This design is known to better capture local style and texture, and it prevents the generator from focusing only on global structure at the expense of local detail.

Our PatchGAN discriminator takes the generated or real completed images as input and classifies each patch as real or fake. This encourages the generator to produce realistic local details throughout the image, not just to fool a global classifier. Figure 3 shows the schematic of our PatchGAN discriminator. It progressively downsamples the input image through convolutional layers until the spatial resolution matches that of the patch-based output. Each output value corresponds to the authenticity of a small local region of the image.



**Figure 3: The PatchGAN discriminator architecture.** It classifies overlapping patches of the image, encouraging the generator to produce realistic local textures and details.

## Training Procedure

**Adversarial Training:** We train our model following the adversarial learning paradigm. During training, the generator ( $G$ ) receives two partial images and attempts to complete the missing region. The discriminator ( $D$ ) then tries to distinguish the real, fully combined images from the synthetic outputs. We adopt a least-squares GAN (LSGAN) formulation [12] for stable training. The generator’s goal is to produce samples that the discriminator classifies as real, while the discriminator aims to differentiate between real and generated samples accurately.

Our training alternates between optimizing  $D$  and  $G$ . By iteratively improving both models, we steer the generator toward producing outputs that appear increasingly like the ground-truth combined images, balancing local detail with global coherence.

**Implementation Details:** We implement our model using PyTorch. The input images are normalized to  $-1, 1$ . The optimizer used is Adam [6] with a learning rate of  $2 \times 10^{-4}$ ,  $\beta_1 = 0.5$ , and  $\beta_2 = 0.999$ . We train our models on a single GPU, and regularly monitor training by evaluating intermediate outputs to ensure visual plausibility. Model checkpoints are saved periodically.

## Loss Functions

**Adversarial Loss:** For the adversarial part, we use the least-squares objective:

$$\mathcal{L}_{\text{GAN}} = \mathbb{E}Dx - 1^2 \mathbb{E}Dx^2,$$

where  $x$  denotes the real completed image and  $Gx$  is the generator’s output given partial input.

**Reconstruction Loss (L1 Loss):** To ensure that the generated regions closely approximate the ground truth, we use an L1 reconstruction loss on the masked area:

$$\mathcal{L}_{\text{recon}} = \|x - Gx\|_1.$$

**Structural Similarity (SSIM) Loss:** Beyond pixel-level accuracy, we incorporate an SSIM loss to encourage structural fidelity:

$$\mathcal{L}_{\text{SSIM}} = 1 - \text{SSIM}(x, Gx),$$

where  $\text{SSIM}(\cdot, \cdot)$  measures perceptual similarity.

**Perceptual Loss:** We also use a perceptual loss, computed from features of a pre-trained VGG-19 network [?], to ensure that the synthesized regions align with the high-level semantics of the original scenes:

$$\mathcal{L}_{\text{perc}} = \sum_l \|\phi_l x - \phi_l Gx\|_2,$$

where  $\phi_l$  is the feature map at layer  $l$  of VGG-19.

**Combined Objective:** Our final generator objective combines these terms:

$$\mathcal{L}_G = \mathcal{L}_{\text{GAN}}G, D\lambda_{\text{recon}}\mathcal{L}_{\text{recon}}G\lambda_{\text{SSIM}}\mathcal{L}_{\text{SSIM}}G\lambda_{\text{perc}}\mathcal{L}_{\text{perc}}G,$$

with weights  $\lambda_{\text{recon}}$ ,  $\lambda_{\text{SSIM}}$ , and  $\lambda_{\text{perc}}$  controlling the importance of each component.

## Alternative Approaches Considered

We also considered transformer-based methods and diffusion-based inpainting models, inspired by recent work in long-range dependency modeling. However, these approaches typically require extensive computational resources and large training datasets to achieve high-quality results. For instance, Vision Transformers (ViTs) or transformer-based architectures excel at capturing global dependencies, but training them from scratch on our dataset proved challenging and computationally expensive.

Likewise, diffusion-based models are known for their high-quality samples, but training a diffusion inpainting model from scratch is computationally intensive and can struggle without massive training resources. By contrast, our proposed GAN-based approach with a U-Net generator and PatchGAN discriminator yields plausible results more efficiently, making it a practical solution within our computational constraints.

## Justification of Approach

Our implementation uses PyTorch, and we normalize inputs to the  $-1, 1$  range. The models are trained using the Adam optimizer (Kingma & Ba, 2015) with a learning rate of  $2 \times 10^{-4}$  and  $\beta_1 = 0.5$ ,  $\beta_2 = 0.999$ . We periodically save model checkpoints and evaluate intermediate outputs to ensure that the generator’s outputs become progressively more realistic and semantically meaningful.

Our chosen approach strikes a balance between complexity, computational feasibility, and output quality. The U-Net generator ensures that spatial details and global coherence are preserved, while the PatchGAN discriminator encourages realistic local texture synthesis. The combination of reconstruction, SSIM, and perceptual losses helps produce both structurally consistent and perceptually convincing completions. Although state-of-the-art pretrained diffusion or transformer-based models could, in theory, produce even more photorealistic results, our method achieves a strong baseline performance with significantly fewer computational resources, making it suitable for a wide range of applications and research settings.

## EXPERIMENTS

In this section, we present a comprehensive set of experiments aimed at evaluating our primary approach—a GAN-based pipeline using a U-Net generator and a PatchGAN discriminator—against two alternative diffusion-based strategies. Specifically, we compare: (1) our final GAN model, (2) a state-of-the-art pretrained Stable Diffusion model, and (3) a non-pretrained diffusion model trained from scratch. We employ both qualitative visual comparisons and quantitative metrics such as PSNR, SSIM, and FID to assess the realism, structural coherence, and overall quality of the generated images.

### Experimental Setup

For all experiments, we use images from the NS-Outpainting dataset. This dataset consists of wide, high-resolution panoramic images of natural scenes, which is suitable for the outpainting and image completion tasks we consider. Each image is processed by removing specific quadrants (e.g., the top-right and bottom-left portions) to create challenging input scenarios for our models. The models are then tasked with reconstructing the missing regions, effectively stitching together disparate image segments into one seamless panorama.

We report three primary experiments that correspond to different steps of complexity and difficulty in image completion:

- (1) **Same-scene completion:** The model receives two quadrants from the *same* scene as input, and must fill in the missing central region. This tests the ability to preserve global semantics and produce a consistent image with limited context.
- (2) **Cross-scene completion:** The model receives quadrants sourced from *two distinct scenes*, creating an even more challenging scenario. The model must handle semantic

disparity and blend two unrelated inputs into a visually coherent transition.

- (3) **Sparse-quadrant completion:** We increase the difficulty further by reducing the provided information, giving only partial portions of the respective quadrants (e.g., only a sub-region of the upper-right quadrant and a sub-region of the lower-left quadrant), thus creating a larger masked area in the center. This tests the robustness and generalization ability of the model under minimal input constraints.

At each stage, we compare the outputs produced by our three modeling approaches.

### Experiment 1: GAN-Based Model

Our GAN model, described in the Methods section, utilizes a U-Net-based generator and a PatchGAN discriminator. We train the model for approximately 5 hours on a single NVIDIA RTX 2070 GPU. Figure 4 shows representative results from the simpler same-scene completion scenario. Despite the challenging masked region, the GAN outputs structurally coherent images that blend smoothly across the boundaries. While not always perfect, the model preserves global context and generates plausible textures, vegetation, and lighting conditions.

Quantitatively, as shown in Figure 5, our GAN model achieves a PSNR and SSIM that indicate good structural integrity and perceptual quality. Although it may not achieve the absolute fidelity of more computationally expensive methods, its results are visually appealing and achieved with modest computational resources.



**Figure 4: GAN model outputs for same-scene quadrant completion. The model successfully reconstructs missing regions, preserving semantics and producing coherent textures.**

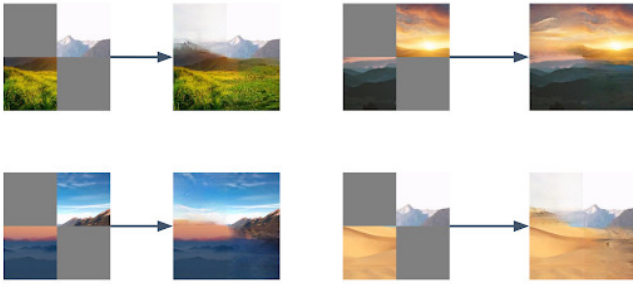
Figure 6 demonstrates the GAN’s performance when blending two distinct scenes. The model still attempts to create a plausible transitional region, though subtle artifacts can



Metric	Value
Peak Signal-to-Noise Ratio (PSNR)	18.8690
Structural Similarity Index Measure (SSIM)	0.7305
Fréchet inception distance (FID)	69.5485

**Figure 5: Performance metrics of our GAN model on the test set (same-scene completion):** PSNR, SSIM, and FID indicate the model’s ability to generate structurally sound and perceptually consistent outputs.

appear due to stark semantic differences. Even so, it significantly outperforms random guessing and often yields a visually coherent hybrid panorama.



**Figure 6: GAN model results when attempting to blend two distinct scenes.** Although the input images differ, the model generates a reasonable transitional region that attempts to merge disparate landscapes.

When pushed to the sparse-quadrant scenario (Figure 7), the model demonstrates resilience, still producing coherent completions with limited reference information. These results highlight the robust generalization capabilities of our GAN approach.

### Experiment 2: Pretrained Stable Diffusion

As a benchmark, we tested a pretrained Stable Diffusion model [14] obtained from HuggingFace. This model was extensively trained on vast image corpora using hundreds of thousands of GPU-hours, resulting in state-of-the-art generative capabilities. Figure 8 and Figure 9 show outputs from the same-scene completion scenario. The Stable Diffusion model’s results are often photorealistic, with high-quality textures and smooth transitions.

When tackling the sparse-quadrant scenario, as shown in Figure 10, Stable Diffusion excels at creating visually convincing completions even with minimal contextual information. However, one must note the extensive compute and training resources underlying this performance—over \$600,000 in training costs [13]—far exceeding what we used for our GAN model.



**Figure 7: GAN model output in the sparse-quadrant scenario,** where only partial quadrant information is provided from two distinct images. The model continues to produce plausible transitions and maintain scene coherence.



**Figure 8: Stable Diffusion output for same-scene completion.** The pretrained model produces high-quality, realistic reconstructions with rich detail and coherent lighting.

### Experiment 3: Non-Pretrained Diffusion Model

Finally, we attempted to train a diffusion-based model from scratch using the RePaint framework [11]. Unlike Stable Diffusion, which benefits from massive pretraining, our scratch-trained diffusion model is limited to the NS-Outpainting dataset and a modest training budget. Figures 11 through 15 show several attempts at same-scene completion. These



Figure 9: Another example of Stable Diffusion on same-scene completion. The output seamlessly integrates the missing regions with minimal artifacts.

results are significantly weaker, with the model struggling to produce coherent global structures or realistic textures. While some color patterns and vague shapes emerge, the outputs lack semantic consistency and visually pleasing features.

## Discussion of Results

Comparing the three approaches:

- **GAN Model:** Efficient and stable training leads to decent-quality outputs with modest computational resources. The results, especially for same-scene completion, are coherent and visually plausible. Although the GAN struggles slightly when blending very disparate scenes or with extremely sparse quadrant information, it maintains a reasonable level of realism and structural consistency.
- **Pretrained Stable Diffusion:** Achieves superior visual fidelity, realism, and detail, even in challenging scenarios. However, its success relies on massive pretraining and significantly greater computational resources. For practical scenarios without such resources, replicating Stable Diffusion’s performance from scratch is infeasible.
- **Non-Pretrained Diffusion:** Without large-scale pretraining and extensive compute, a diffusion model trained from scratch underperforms our GAN approach. The outputs are generally incoherent and lack semantic structure, highlighting the difficulty of training diffusion models in data-constrained and resource-limited settings.



Figure 10: Stable Diffusion results for the cross-scene scenario, with two distinct scenes as input. Despite severely limited context, the pretrained model generates plausible scenes, showcasing its strong generalization and realism. Notably, unlike our model, it does not generate a single cohesive scene from the two input images and instead generates two separate scenes in its outputted montage image.

Overall, our experiments show that while pretrained diffusion models can achieve remarkable realism, their resource requirements are prohibitive for many applications. In contrast, our GAN-based approach provides a strong balance between performance quality and computational feasibility, making it a practical solution for image outpainting and hole-filling tasks.

## CONCLUSION

As can be seen from the images above, the best results that we achieved from a model that we developed were obtained from our final GAN model, which integrates a U-Net-based generator and a PatchGAN-based discriminator.

Notably, the state of the art pretrained Stable Diffusion model “*was trained using 256 Nvidia A100 GPUs on Amazon Web Services for a total of 150,000 GPU-hours, at a cost of \$600,000.*” By contrast, we achieved the results shown above with our GAN model after training it from scratch for just 5 hours on a single RTX 2070 GPU. We’d love to see what our final GAN model can achieve with a training budget of \$600,000!

## APPENDIX

All of our code can be found in [our repository](#) on Github.



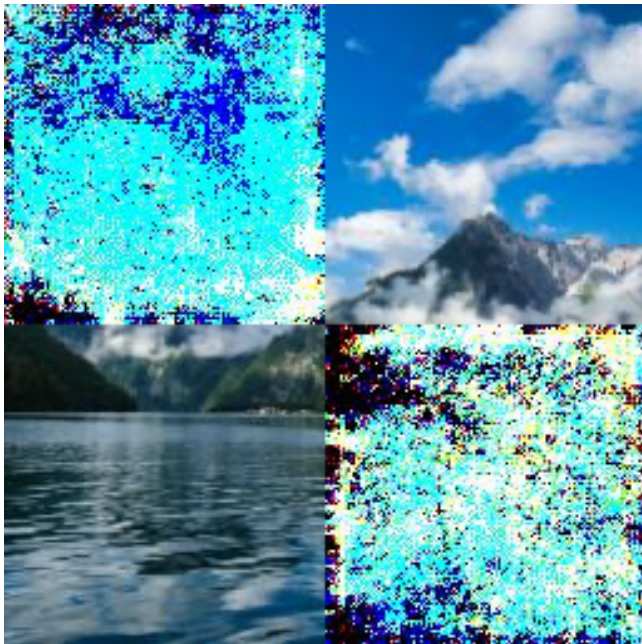


Figure 11: Non-pretrained diffusion model output for same-scene completion. The generated region appears blurry and lacks coherent structure.

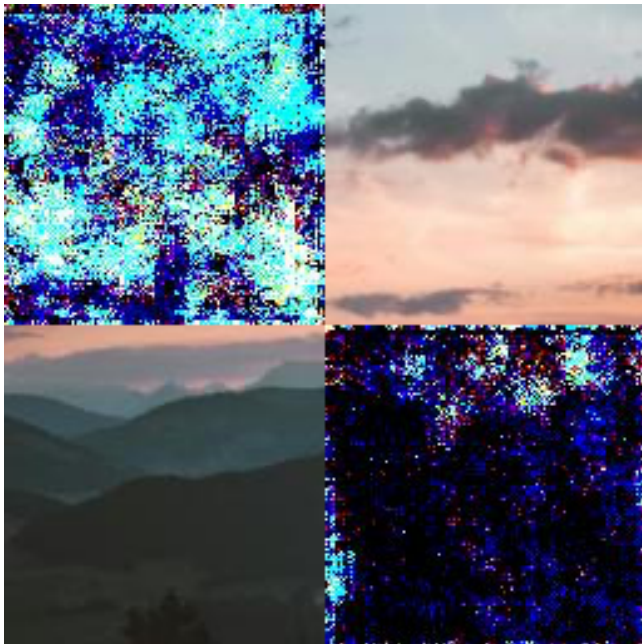


Figure 12: Non-pretrained diffusion model output for same-scene completion. The model fails to reconstruct meaningful details or textures and produces amorphous shapes.

## REFERENCES

- [1] Stéphane d'Ascoli et al., "ConViT: Improving Vision Transformers with Soft Convolutional Inductive Biases," *CVPR 2021*, 2021.



Figure 13: Another example of the non-pretrained diffusion model. Although some color gradients are reasonable, no clear semantic structure emerges.



Figure 14: Non-pretrained diffusion output. The image completion remains incoherent, suggesting that extensive data and compute are required for diffusion methods to match GAN-level performance.

- [2] A. Dosovitskiy, et al., "Discriminative Unsupervised Feature Learning with Exemplar Convolutional Neural Networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.



Figure 15: Non-pretrained diffusion output. Attempts at structure are minimal, and the generated content does not convincingly blend with the original quadrants.

- [3] Penglei Gao et al., "Generalised Image Outpainting with U-Transformer," *CVPR 2022*.
- [4] P. Isola, et al., "Image-to-Image Translation with Conditional Adversarial Networks," *CVPR 2017*.
- [5] J. Yang, et al., "Contextual Attention for Image Inpainting," *CVPR 2018*.
- [6] D. Kingma, et al., "Adam: A Method for Stochastic Optimization," *3rd International Conference for Learning Representations, San Diego, 2015*.
- [7] A. Krizhevsky, et al., "ImageNet Classification with Deep Convolutional Neural Networks," *NIPS 2012*.
- [8] Y. LeCun, et al., "Gradient-Based Learning Applied to Document Recognition," *Proceedings of the IEEE*.
- [9] Z. Liu, et al., "Swin Transformer: Hierarchical Vision Transformer Using Shifted Windows," *ICCV 2021*.
- [10] Chia-Ni Liu, et al., "Bridging the Visual Gap: Wide-Range Image Blending," *CVPR 2021*.
- [11] Andreas Lugmayr, et al., "RePaint: Inpainting Using Denoising Diffusion Probabilistic Models," *CVPR 2022*.
- [12] X. Mao, et al., "Least Squares Generative Adversarial Networks," *ICCV 2017*.
- [13] @EMostaque on Twitter, "The Stable Diffusion model was trained using 256 Nvidia A100 GPUs on Amazon Web Services for a total of 150,000 GPU-hours, at a cost of \$600,000." <https://twitter.com/EMostaque/status/1509780730730734592>
- [14] Robin Rombach, et al., "High-Resolution Image Synthesis with Latent Diffusion Models," *CVPR 2022*.
- [15] Olaf Ronneberger, et al., "U-Net: Convolutional Networks for Biomedical Image Segmentation," *MICCAI 2015*.
- [16] Luming Tang, et al., "RealFill: Reference-Driven Generation for Authentic Image Completion," *SIGGRAPH 2024*.
- [17] Zongxin Yang, et al., "Very Long Natural Scenery Image Prediction by Outpainting," *ICCV 2019*.
- [18] Jiahui Yu et al., "Generative Image Inpainting with Contextual Attention," *CVPR 2018*.