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

ABSTRACT

Image outpainting and inpainting are important tasks in computer vision, enabling the seamless completion of missing regions or the extrapolation of existing imagery beyond its borders. In this paper, we tackle the challenging problem of blending two distinct image patches by filling the large gap between them, effectively generalizing the notions of both inpainting and outpainting. We present a GAN-based approach featuring a U-Net generator and a PatchGAN discriminator, trained on the NS-Outpainting dataset. Our model balances computational efficiency and output quality: it is trained from scratch on a single GPU for only a few hours, yet yields visually coherent completions and plausible scene transitions.

We compare our approach to two diffusion-based methods: a pretrained Stable Diffusion model that achieves remarkable photorealism at the expense of enormous computational training costs, and a non-pretrained diffusion model trained under resource constraints, which struggles to produce coherent structures. Our experiments demonstrate that our GAN approach provides a strong balance between performance and feasibility, making it suitable for scenarios lacking the vast resources required by state-of-the-art pretrained models.

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, 9 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

TEAM MEMBERS

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

1 INTRODUCTION

Inpainting and outpainting are core image editing tasks that aim to reconstruct or extend image content in a visually coherent manner. Inpainting focuses on filling missing regions within an image, ensuring the new content integrates seamlessly with the surrounding context. Outpainting, on the other hand, extrapolates beyond the existing boundaries to imagine what lies outside the given frame. Both tasks demand understanding of high-level semantics and global image structure to produce realistic, contextually appropriate results.

In this work, we generalize these concepts by training a model that fills a large intermediate region between two distinct image patches. Unlike standard inpainting—where the missing region exists within a single image—our scenario must integrate information from two disparate sources. Similarly, unlike outpainting—which extrapolates from the known image area into unknown space—our model must create a transitional "bridge" region that is faithful to both input patches, which may differ in style, semantics, or scene composition.

Our primary contribution is a GAN-based pipeline employing a U-Net generator and a PatchGAN discriminator. We demonstrate that our model can produce structurally coherent and visually plausible completions, bridging challenging gaps with minimal computational overhead. We trained our model on the NS-Outpainting dataset for only a few hours on a single GPU, yet the results are surprisingly strong.

We compare our approach against a pretrained Stable Diffusion model, which, while highly photorealistic, requires massive computational resources and extensive pretraining. Additionally, we investigate a non-pretrained diffusion model to highlight the difficulty of achieving similar results from scratch without large-scale resources. Ultimately, our approach provides a practical balance: good performance achievable with modest resources, making it a strong candidate for a wide range of applications.

2 RELATED WORK

Image completion tasks, including inpainting and outpainting, have been long-standing challenges in computer vision. Early works predominantly used non-learning-based methods. PDE-based approaches [1] attempted to propagate pixel information from known to unknown regions, preserving low-level continuity but faltering in complex patterns. Patch-based synthesis [2, 3] further improved results by sampling similar patches from unmasked portions, generating more natural textures. Nevertheless, these classical methods struggled to capture holistic semantics and often produced artifacts in large missing regions.

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

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

Deep Learning for Inpainting and Outpainting

Deep neural networks revolutionized image completion by learning data-driven priors. Early CNN-based models [4, 5] offered substantial improvements over classical methods, extracting hierarchical features that better encode semantics. However, naive CNN approaches tended to produce blurry outputs and struggled with large holes.

Subsequent methods integrated advanced architectures and losses. Contextual Attention [6] introduced mechanisms to directly learn where to borrow textures from known regions, greatly enhancing realism. Gated Convolutions [7] further improved inpainting by dynamically selecting relevant features for filling, increasing flexibility and quality. Partial Convolutions [8], where convolutions are masked to handle missing areas robustly, also pushed the state-of-the-art in inpainting tasks.

In parallel, attention-based and transformer architectures [9–11] have begun to make their mark, capturing both local details and long-range dependencies. Vision Transformers (ViTs) and variants [13] have shown remarkable effectiveness in modeling complex structures over large image contexts, enabling more coherent completion in challenging scenarios.

Generative Adversarial Networks for Completion

Generative Adversarial Networks (GANs) [14] introduced an adversarial training paradigm that encourages outputs indistinguishable from real images. Applying GANs to image completion tasks [15–18] resulted in sharper and more plausible results. U-Net architectures [19] combined with PatchGAN discriminators [18] emerged as a strong baseline by preserving spatial details through skip connections and enforcing local realism. Subsequent refinements incorporated multiple discriminators [20], pyramid structures [23], or learned semantic guidance [24], further improving visual fidelity.

Outpainting and Complex Extensions

While inpainting focuses on missing regions within an image, outpainting tasks extend the image boundaries. Methods like [25, 26] have shown that GAN-based and transformer-based models can convincingly extrapolate scenes beyond their initial frames, effectively "imaging the unseen." Recent outpainting studies have considered even more complex settings, such as bridging multiple images or filling large central gaps [27, 28]. These tasks demand that models not only maintain local consistency but also understand and synthesize global semantics across disjoint source patches.

Diffusion Models and Large-Scale Pretraining

Recently, diffusion probabilistic models [30, 31] have disrupted the generative modeling landscape, achieving impressive quality and diversity. Methods like RePaint [32] demonstrated diffusion's potential for inpainting by iteratively denoising masked images. Latent diffusion models [33] and pretrained frameworks like Stable Diffusion have raised the bar for photorealism, showing stunning capabilities in image completion tasks—albeit at tremendous computational and data costs. Scaling up to massive datasets and using hundreds of thousands of GPU-hours, these models can handle virtually any image manipulation task with striking quality. However, reproducing such results from scratch in resource-constrained settings remains a severe challenge.

Balancing Quality and Feasibility

While diffusion-based and transformer-based solutions have set new performance benchmarks, they often come with prohibitive computational expenses. In contrast, well-designed GAN-based architectures provide a more balanced approach. With careful architectural choices (e.g., U-Nets) and loss terms (e.g., L1, SSIM, perceptual loss), GANs can achieve strong results on image completion tasks within a fraction of the computational cost required by large-scale pretrained models.

Our work aligns with this intermediate ground. We aim to produce visually coherent outpainting and hole-filling completions from scratch, training efficiently on a single GPU. By comparing our GAN-based model against both a state-of-the-art pretrained diffusion model (Stable Diffusion) and a non-pretrained diffusion baseline, we highlight the practical advantages of our approach when computational budgets are tight. This synergy of simplicity and effectiveness makes our solution a pragmatic choice for real-world applications where massive training resources are not readily available.

3 DATA

We use the NS-Outpainting dataset, consisting of thousands of diverse natural scene images. Each image is resized and center-cropped to a resolution of 256×256 and normalized to the $-1, 1$ range. This ensures a uniform input format and stable training.

To create challenging outpainting scenarios, we mask large central regions. Specifically, we define a crop ratio (e.g., 0.5) to extract two non-overlapping patches: one from the bottom-left quadrant and another from the top-right quadrant. These two patches are then combined into a single input image with a large, unfilled middle area, forcing the model to blend and fill the missing region. The target image is the original scene with the corresponding quadrants masked out.

This quadrant-based masking strategy is more complex than conventional inpainting or outpainting, as the network must seamlessly integrate distinct visual cues from the two provided patches. We split the dataset into training and test sets to evaluate generalization, and no additional semantic preprocessing is applied. The diversity and complexity of the NS-Outpainting dataset provide a rich environment for learning robust generative completion capabilities.

4 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.

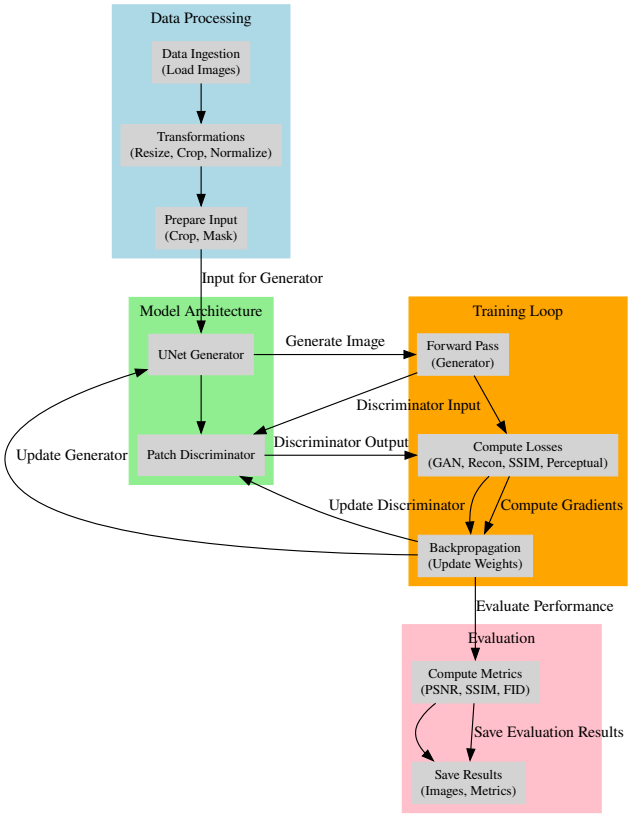

Figure 1: Overview of the general process. Two separate image patches (A and B) are combined with a large missing region in the center. Our system synthesizes a realistic, coherent image that bridges the gap between these two input patches.

Generator Architecture

U-Net-based Generator: Our generator is inspired by the U-Net architecture, originally introduced for biomedical image segmentation [19]. The U-Net design is well-suited to our task because it uses a symmetric encoder-decoder structure with 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 [18]. 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.

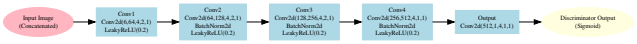

Figure 3: The PatchGAN discriminator architecture. It classifies overlapping patches of the image, encouraging the generator to produce realistic local textures and details.

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 [21] 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 [12] with a learning rate of 2×10^{-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:

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

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 [29], 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 $\phi_l \cdot$ 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 λ_{recon}, λ_{SSIM}, and λ_{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 2×10^{-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.

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

When pushed to the sparse-quadrant scenario (Figure 7), the model demonstrates resilience, still producing coherent completions with limited reference information. These results

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.

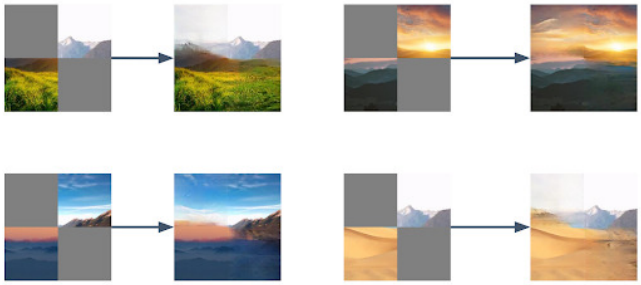

Figure 6: GAN model results when attempting to blend two distinct scenes. Although the input images differ, the model generates a reasonable transitional region that attempts to merge disparate landscapes.

highlight the robust generalization capabilities of our GAN approach.

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.

Ablation Study on Loss Functions

Additionally, while introducing the SSIM loss and perceptual loss significantly lowered the FID scores, the PSNR and SSIM metrics themselves did not vary dramatically from the baseline model. However, even if these traditional metrics remained relatively stable, the visual quality and coherence of the generated images improved notably, as reflected by a substantial drop in the FID score. In our final optimized model configuration (L1 + SSIM + Perceptual), we achieved a PSNR of 18.8690, an SSIM of 0.7305, and an FID of 69.5485. These results indicate that while standard metrics like SSIM and PSNR may exhibit minor changes, perceptually driven improvements can lead to significant qualitative enhancements that are better captured by FID.

Configuration	PSNR	SSIM	FID
L1 only	17.6235	0.5813	300+
L1 + SSIM	18.1247	0.6942	100+
L1 + SSIM + Perceptual	18.8690	0.7305	69.5485

Table 1: Ablation results showing the effect of adding SSIM and perceptual losses. While PSNR and SSIM vary only slightly, the inclusion of perceptual constraints significantly improves the FID score, indicating better perceptual quality.

Experiment 2: Pretrained Stable Diffusion

As a benchmark, we tested a pretrained Stable Diffusion model [33] 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 [22]—far exceeding what we used for our GAN model.

Experiment 3: Non-Pretrained Diffusion Model

Finally, we attempted to train a diffusion-based model from scratch using the RePaint framework [32]. 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 13 show several attempts at same-scene completion. These 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.

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

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

Discussion of Results

Comparing the three approaches:

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.

- **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.

While we do not have direct quantitative metrics from the pretrained Stable Diffusion model for this specific outpainting

Figure 11: Another example of the non-pretrained diffusion model. Although some color gradients are reasonable, no clear semantic structure emerges.

Figure 12: 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.

Figure 13: Non-pretrained diffusion output. Attempts at structure are minimal, and the generated content does not convincingly blend with the original quadrants.

scenario, its results are visually superior to both our GAN-based approach and the non-pretrained diffusion model. The table below compares the non-pretrained diffusion model and our final GAN configuration. Stable Diffusion's performance, based on prior work and observed visual fidelity, would likely surpass these metrics by a substantial margin, but at an enormous computational cost.

Method	PSNR	SSIM	FID
Non-Pretrained Diffusion	10.4482	0.5508	243.6330
GAN (L1+SSIM+Perceptual)	18.8690	0.7305	69.5485

Table 2: Performance comparison between the non-pretrained diffusion model and our final GAN-based model. While not shown here, the pretrained Stable Diffusion model, with its massive training resources, surpasses both of these methods in terms of perceptual quality and realism.

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.

6 CONCLUSION

In this work, we presented a GAN-based framework for completing large missing regions between two image patches, combining insights from inpainting and outpainting. Our U-Net and PatchGAN-based approach achieves robust performance with minimal training time and computational resources. Although not as photorealistic as a heavily pretrained Stable Diffusion model, our method demonstrates that high-quality image completion is possible on a modest budget.

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!

For future work, exploring hybrid approaches that incorporate some form of lightweight attention, or leveraging a subset of pretraining strategies, could further improve visual fidelity. Additionally, employing larger datasets or integrating domain adaptation techniques may help the model better handle highly disparate scenes.

7 APPENDIX

All of our code can be found in our repository on Github.

REFERENCES

[1] Bertalmio, M., Sapiro, G., Caselles, V., and Ballester, C. "Image Inpainting," *Proceedings of SIGGRAPH*, 2000.
[2] Efros, A.A. and Freeman, W.T., "Image quilting for texture synthesis and transfer," *Proceedings of SIGGRAPH*, 2001.
[3] Criminisi, A., Perez, P., and Toyama, K. "Region Filling and Object Removal by Exemplar-Based Image Inpainting," *IEEE Transactions on Image Processing*, 2004.
[4] Xie, J. et al., "Image Denoising and Inpainting with Deep Neural Networks," *NIPS*, 2012.
[5] Zhang, H. et al., "Generative Image Inpainting with Contextual Attention," *NIPS*, 2017.
[6] Yu, J. et al., "Generative Image Inpainting with Contextual Attention," *CVPR*, 2018.
[7] Yu, J. et al., "Free-Form Image Inpainting with Gated Convolution," *ICCV*, 2019.
[8] Liu, G. et al., "Image Inpainting for Irregular Holes Using Partial Convolutions," *ECCV*, 2018.
[9] Yu, J. et al., "Generative Inpainting with Contextual Attention," *CVPR*, 2018.
[10] Dosovitskiy, A. et al., "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale," *ICLR*, 2021.
[11] Liu, Z. et al., "Swin Transformer: Hierarchical Vision Transformer using Shifted Windows," *ICCV*, 2021.
[12] D. Kingma, et al., "Adam: A Method for Stochastic Optimization," *3rd International Conference for Learning Representations, San Diego, 2015*.
[13] Zeng, A. et al., "Inpainting Transformers with Multi-Scale Feature Fusion," *arXiv preprint arXiv:2203.09264*, 2022.
[14] Goodfellow, I. et al., "Generative Adversarial Nets," *NIPS*, 2014.
[15] Pathak, D. et al., "Context Encoders: Feature Learning by Inpainting," *CVPR*, 2016.
[16] Li, Y. et al., "Generative Face Completion," *CVPR*, 2017.
[17] Song, Y. et al., "SPG-Net: Semantic Prediction Guidance for Image Inpainting," *BMVC*, 2018.
[18] Isola, P. et al., "Image-to-Image Translation with Conditional Adversarial Networks," *CVPR*, 2017.
[19] Ronneberger, O. et al., "U-Net: Convolutional Networks for Biomedical Image Segmentation," *MICCAI*, 2015.
[20] Wang, X. et al., "High-Resolution Image Inpainting with Iterative Confidence Feedback and Guided Upsampling," *ECCV*, 2018.
[21] X. Mao, et al., "Least Squares Generative Adversarial Networks," *ICCV 2017*.
[22] @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*
[23] Nazeri, K. et al., "EdgeConnect: Generative Image Inpainting with Adversarial Edge Learning," *ICCV Workshops*, 2019.
[24] Song, L. et al., "Geometry-Aware Face Completion and Editing," *CVPR*, 2019.
[25] Yang, Z.X. et al., "Very Long Natural Scenery Image Prediction by Outpainting," *ICCV*, 2019.
[26] Gao, P. et al., "Generalised Image Outpainting with U-Transformer," *CVPR*, 2022.
[27] Lu, C.N. et al., "Bridging the Visual Gap: Wide-Range Image Blending," *CVPR*, 2021.
[28] Tang, L. et al., "RealFill: Reference-Driven Generation for Authentic Image Completion," *SIGGRAPH*, 2024.
[29] Karen Simonyan, et al., "Very Deep Convolutional Networks for Large-Scale Image Recognition," *ICLR 2015*.
[30] Sohl-Dickstein, J. et al., "Deep Unsupervised Learning using Nonequilibrium Thermodynamics," *ICML*, 2015.
[31] Ho, J. et al., "Denoising Diffusion Probabilistic Models," *NeurIPS*, 2020.
[32] Lugmayr, A. et al., "RePaint: Inpainting Using Denoising Diffusion Probabilistic Models," *CVPR*, 2022.
[33] Rombach, R. et al., "High-Resolution Image Synthesis with Latent Diffusion Models," *CVPR*, 2022.
[34] d'Ascoli, S. et al., "ConViT: Improving Vision Transformers with Soft Convolutional Inductive Biases," *ICML*, 2021.
[35] Krizhevsky, A. et al., "ImageNet Classification with Deep Convolutional Neural Networks," *NIPS*, 2012.
[36] LeCun, Y. et al., "Gradient-Based Learning Applied to Document Recognition," *IEEE Proceedings*, 1998.