Generative Inpainting: A Survey Study

Kien Tran Manoj Parmar Palash Choudhary Tanmay Kenjale
Georgia Institute of Technology
Atlanta, GA 30332
ktran332@gatech.edu, manoj.parmar@gatech.edu, palash.choudhary@gatech.edu, tkenjale3@gatech.edu

Abstract

Image inpainting is one of the image restoration techniques that involves filling in missing or damaged pixels in an image. It has several applications such as object removal, image restoration, etc. Generative inpainting is a class of techniques that use generative models to fill in the missing parts. It is superior to classical approaches due to the ability to "hallucinate" content from the learned distribution and to do it in a way that is consistent with the context surrounding the missing patch. Recent years have seen significant advancements in the field, especially the shift from GAN-based models to Diffusion-based models with tools like DALLE-2 and Stable diffusion. For this reason, we are motivated to conduct an academic survey study of the generative inpainting techniques. This paper aims to understand the basic principles, differences, strengths, and weaknesses of different inpainting techniques, especially DCGAN [10], Inpainting with contextual attention [16], and diffusion-based models [11].

We have implemented and evaluated these approaches on the Places205 dataset. We found that the diffusion-based model with pre-trained weights performs the best, while DCGAN and Contextual Attention achieve similar level of results. We also found that training these models was highly time-consuming and computationally costly and that using pre-trained weights is the best option.

1. Introduction/Background/Motivation

Image inpainting is a challenging computer vision task that involves filling in missing or corrupted parts of an image in a visually consistent manner. The goal of image inpainting is to develop algorithms that can learn to understand the underlying structure and context of an image and fill new content that seamlessly blends with the existing content.

The initial days saw the utilization of patch-based techniques and texture synthesis. A patch-based approach typically synthesizes missing contents by copying and pasting similar patches from known image contexts. These approaches can work well, especially in stationary background inpainting with repeating patterns. However, these approaches can fall short in completing large missing regions of complex scenes for semantic inpainting. This is because patch-based approaches heavily rely on patch-wise matching by low-level features. Such a technology is unable to synthesize contents for which similar patches do not exist in known image contexts.

The introduction of the Generative Adversarial Network (GAN) framework in 2014 led to deep learning based generative solutions to inpainting. Generative models are superior to classical approaches due to the ability to "hallucinate" content from the learned distribution and to do it in a way that is consistent with the context surrounding the missing patch.

A study in 2018 used a Coarse-to-Fine architecture with GAN losses for both the local output and global output [16]. Coarse-to-Fine involves one stage that produces an intermediate inpainting result and a second stage that produces the final result. This approach was limited because it only worked well on rectangular cutouts for inpainting. A study in 2019 built upon the previous approach but extended it to work with free-form cutouts, and it achieved this by using gated convolution and sketch input masks [15]. The use of gated convolution adds computational complexity, and so an approach in 2021 avoided this module and built a custom block called Aggregated COntextual-Transformation (AOT) [17] that was able to outperform gated convolution. Another benefit of the AOT architecture is that it consists of only one stage instead of two.

More recently, diffusion-based models have outperformed GAN-based models such as the ones discussed above. The diffusion model works by gradually injecting noise into the image and training a model, often a U-Net, to learn the denoising process. Diffusion can alleviate GAN's issues of vanishing gradients and the difficulty of balancing the generator and discriminator. The approach in [11] shows improvements over previous GAN models for inpainting.

Inpainting has wide-ranging applications in various domains, such as image editing, restoration, and completion. One such famous implementation of inpainting is the magic eraser feature of Google Pixel smartphones. It helps remove unwanted objects from the images. Further, inpainting can also be used to enhance old pictures and documents.

For the purpose of our project, we trained the model on a subset of the Places205 dataset [1]. The Places205 dataset is a large-scale scene-centric dataset with 205 common scene categories. The training dataset contains around 2.5M images from these categories. In the training set, each scene category has a minimum of 5,000 and a maximum of 15,000 images. The subset chosen for our training data is 10,000 images for DCGAN and 1000 images for Contextual Attention approach. For diffusion-based approach, we are using a pre-trained model. Further, we compare the 3 models on an evaluation set of randomly selected 500 images from the same dataset. The dataset has images of size 256x256, and we utilized both rectangular and free-form masks with mask areas between 10%-25%.

2. Approach

2.1. DCGAN

We first implemented a deep convolutional generative adversarial network (DCGAN) as a baseline approach to image inpainting [10]. This approach extends the traditional GAN [6] by using Convolution and Transposed Convolution layers instead of fully-connected layers.

Our DCGAN consists of a generator model and a discriminator model. The generator model inputs a masked image and outputs the inpainted image. The architecture of the generator is inspired by a U-Net [12]. The U-Net applies 6 layers of contraction blocks followed by 6 layers of expansion blocks. The contraction block consists of a 2D Convolution with a kernel size of 4 and stride of 2, a Batch Normalization layer, and LeakyReLU activation. Each contraction halves the size of the image and doubles the number of channels. An expansion block consists of a 2D Transposed Convolution with a kernel size of 4 and stride of 2, a Batch Normalization layer, and ReLU activation. The final expansion block has a Tanh activation. Each expansion doubles the size of the image and halves the number of channels. We also employ skip connections for better gradient flow.

The discriminator model inputs training images and the outputs of the generator and outputs a probability that the input is a real image as opposed to a generated image. The architecture consists of 6 contraction blocks with a 2D Convolution and Sigmoid activation at the end.

The discriminator was trained with binary cross-entropy loss and the generator was trained with a combination of the discriminator loss and an L1 loss. Both models utilized the Adam optimizer with a learning rate of 0.0002. The general GAN implementation was inspired by the PyTorch DCGAN tutorial [8] and the architecture was inspired by an implementation by [5].

Since DCGAN is a relatively old technique and one that is not specifically designed for image inpainting, we expected the results to be poor. We also expected that optimizing the model to be difficult because of the difficulties involved in balancing the generator and discriminator. In practice, our expectations were correct. With the default parameters from our research, we found that the model produced pixelated images with grid-like artifacts. We also found that the discriminator performed too well, causing the generator's performance to deteriorate. We will go over our problem-solving process and fine-tuning in the next section.

2.2. Inpainting with Contextual Attention

The second approach that we used was generative inpainting using contextual attention [16]. Earlier approaches often created distorted structures or blurry textures inconsistent with surrounding areas. This is due to the ineffectiveness of convolution neural networks in explicitly borrowing or copying information from distant spatial locations.

The network consists of two stages as shown in Figure 1. The first stage is a "coarse" network which makes an initial prediction for the image. The second stage is a "refinement" network that takes as input this coarse image to generate better results with contextual attention integrated into this stage.

The general architecture for the coarse network follows an encoder-decoder structure with a total of 17 layers. The initial layers are contraction blocks of 2D convolution layers to decrease the image size and increase the number of channels. 4 dilated convolution layers are also used to have a wider range of receptive fields and extract information from far apart regions. The decoder architecture uses deconvolution layers to restore the image to its original size.

The second stage of the process integrates contextual attention, which uses the features of previously known patches as filters to process the generated patches. This is done by matching generated patches with known contextual patches using convolution, weighing relevant patches using channel-wise softmax, and reconstructing the generated patches with contextual patches using deconvolution. A spatial propagation layer is also included to encourage spatial coherence. To enable the network to hallucinate new content, a parallel convolutional pathway is used alongside the contextual attention pathway. These two pathways are combined and fed into a single decoder to generate the final output.

Finally, the discriminator model is trained on input images and the generated image from the refinement network to output a probability of a real image vs a fake image.

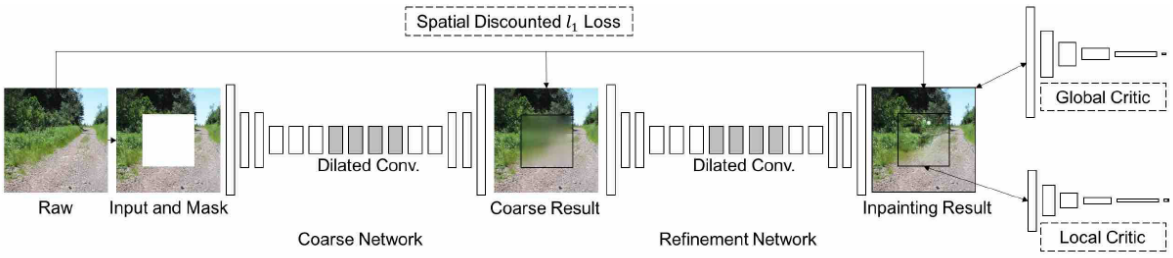

Figure 1. Two-Stage Contextual Attention Architecture

There are two discriminators - a global and a local discriminator. The global discriminator assesses if the completed image is coherent as a whole, while the local discriminator focuses on a small area centered at the generated region to enforce local consistency. Both the discriminators have 4 convolution layers to generate high-level features and a final dense layer to output the probabilities.

The generator network uses the ELUs as activation functions instead of ReLU while the discriminator uses Leaky ReLU as activation with alpha = 0.2. The coarse network is trained with the reconstruction loss explicitly (L1 loss), while the refinement network is trained with the reconstruction as well as WGAN-GP losses (including gradient penalty) [2]. We used Adam optimiser for generator and discriminator models with a learning rate of 0.0001. We further talk about the results in detail in the experiments section. The final code is available here. https://github.com/Manoj98/generative-inpainting. The following code was used as a baseline for the above code. [4]

2.3. Diffusion-based Model

The third approach that we used was diffusion. The diffusion model works by gradually injecting noise into the image and training a model, often an Unet, to learn the denoising process. The denoising model can be easily conditioned by the unmasked area of the image or external information such as text description. This characteristic makes the diffusion approach more versatile. In addition, inference in the diffusion-based in-painting model starts with complete noise in the masked area and then gradually removes the noise conditioned on the unmasked area, which allows the model to create complex and refined outputs. This inference process, however, will be time-consuming due to the loop through multiple time steps.

For the implementation of the diffusion model, we initially planned to replicate Saharia et al.[13] [9] and train the model ourselves on the Places205 dataset. However, we soon found out that the training of the model was too slow and computationally expensive to create meaningful results in a short amount of time. For that reason, we decided to leverage the Stable Diffusion model[11] with pre-trained weights and codes by Platen et al.[14]. We adapted the model for the in-painting task by writing the transformation pipeline for the input and output of the pre-trained diffusion model. We then run inference on the shared test set and compared the results with other models that we have implemented.

3. Experiments and Results

3.1. Experiment setup

We randomly selected a 500-image subset of the Places205 dataset for evaluation purposes. Each image will be cropped by either a rectangular mask or a free-form mask and then fed as input to the inpainting models. The prediction results are then evaluated using 3 metrics: Frechet Inception Distance (FID)[7] - code implementation by Brownlee[3] from Machine Learning Mastery, Mean Squared Error (MSE), and Mean Absolute Error (MAE). In addition, we also randomly selected samples and make qualitative assessments about the strengths and weaknesses of different methods.

Due to a limitation of the implementation, the Contextual Attention model works with rectangular crops only whereas the other 2 models work with free-form crops. For that reason, we decided to limit the total masked area to 10% -25% of the original image for both masking methods, thus making the performance on the in-painting task comparable.

3.2. DCGAN Experiments

We trained the DCGAN model in PyTorch on 10000 images of size 256x256 from the Places205 dataset. Each image was paired with a randomly selected free-form crop. We trained the models for multiple epochs until the generator's validation loss was no longer improving significantly,

indicating that maximum performance was reached. This typically occurred around 22 epochs.

For validation purposes, we looked at both the loss comparing the generator's entire image output to the ground truth image and the loss comparing the generator's output for the inpainting region compared to the same region in the ground truth. The former gives an idea of how well the model can reconstruct the overall image and the latter gives an idea of how well the model can inpaint.

The main parameters we chose to tune were the generator loss function and the hidden dimension size of generator and discriminator. We initially started with a generator loss function that was purely based on adversarial loss from the discriminator. We hypothesized that adding a reconstruction loss to the generator that compared the generator outputs to the ground truth images would help train the generator to produce good inpainting results. We tried both L1 and L2 reconstruction losses, and we found that L1 produced visually better results whereas L2 produced many artifacts.

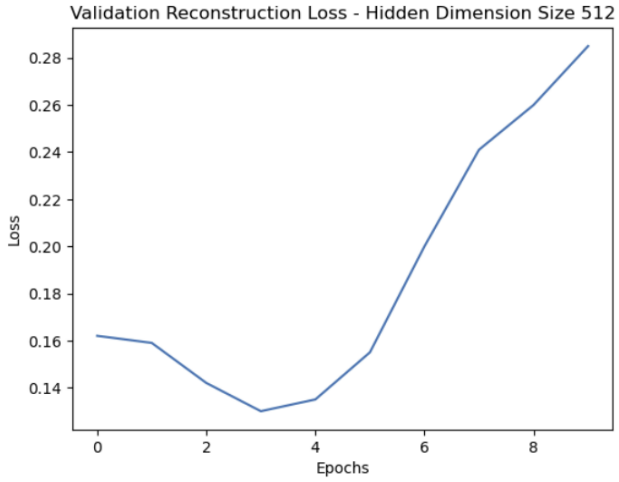

Figure 2. Diverging reconstruction loss when discriminator is stronger than the generator

As for hidden dimension sizes, we started with size 512 for both the generator and discriminator. While this worked well for a few epochs of training, we found that after some time the discriminator loss would converge close to zero and the generator loss would increase rapidly as seen in Figure 2. We hypothesized this was because the discriminator was learning too quickly and was not allowing the generator to learn.

To solve this problem, we decreased the discriminator's hidden size progressively until we reached a desirable level of generator performance. We found that a discriminator hidden size of 64 worked the best. We also found that increasing the hidden size of the generator had little effect on performance while greatly increasing computation cost, but decreasing the hidden size of the generator severely reduced performance.

In general, the model's performance during training and validation was similar, indicating that there was little overfitting present. We believe that the DCGAN architecture is not sufficient for good generative inpainting, and therefore the model underfits. From our qualitative observations, we see that DCGAN does a reasonable job at capturing the correct colors and shapes of masked regions but fails to hallucinate fine details and results in blurry areas. For such a relatively simple and general-purpose model, the results exceeded our expectations.

3.3. Contextual Attention Experiments

The whole network was trained in PyTorch on 1000 sample images from Places205 data. For training, we used images of resolution 256x256 while the rectangular masks were created at random with the largest hole size of 128x128. The batch size selected for training was 16. The model in the original paper was trained for 50K iterations to converge. However, due to resource constraints, we were able to train it for only 7500 iterations due to which the results haven't converged yet which can be seen in Figure 3.

The model was trained on Nvidia A100 GPUs using Google Colab Pro which took around 5 hours to train for 1000 images. We have summarized the qualitative and quantitative results in the results comparison section. Specifically, the reconstruction error (L1 error) goes down from 0.092340 to 0.064443 over a span of 7500 iterations.

3.4. Results comparison

Finally, we compare the results of different implementations both quantitatively and qualitatively. Table 1 below shows the quantitative evaluation of a DCGAN model, a Contextual Attention model, and a pre-trained diffusion model on the inpainting task. It is clear that the Stable Diffusion model vastly outperformed the other two models on FID, highlighting the effectiveness of the diffusion method. Regarding MSE and MAE, however, the 3 models have quite comparable results. Figure 3 shows the qualitative output of the model for a few random images. The output from the Stable Diffusion model is quite natural, only show weakness when zooming into the details. The Stable Diffusion model can also "hallucinate" objects that were not there to fill in the blank. However, this dataset consists mostly of places without many people or other objects,

Table 1. Performance comparison

Models	FID	MSE	MAE
DCGAN	486.39	**0.086**	**0.140**
Contextual Attention	290.13	0.162	**0.141**
Stable Diffusion (Pre-trained)	**121.24**	0.158	0.209

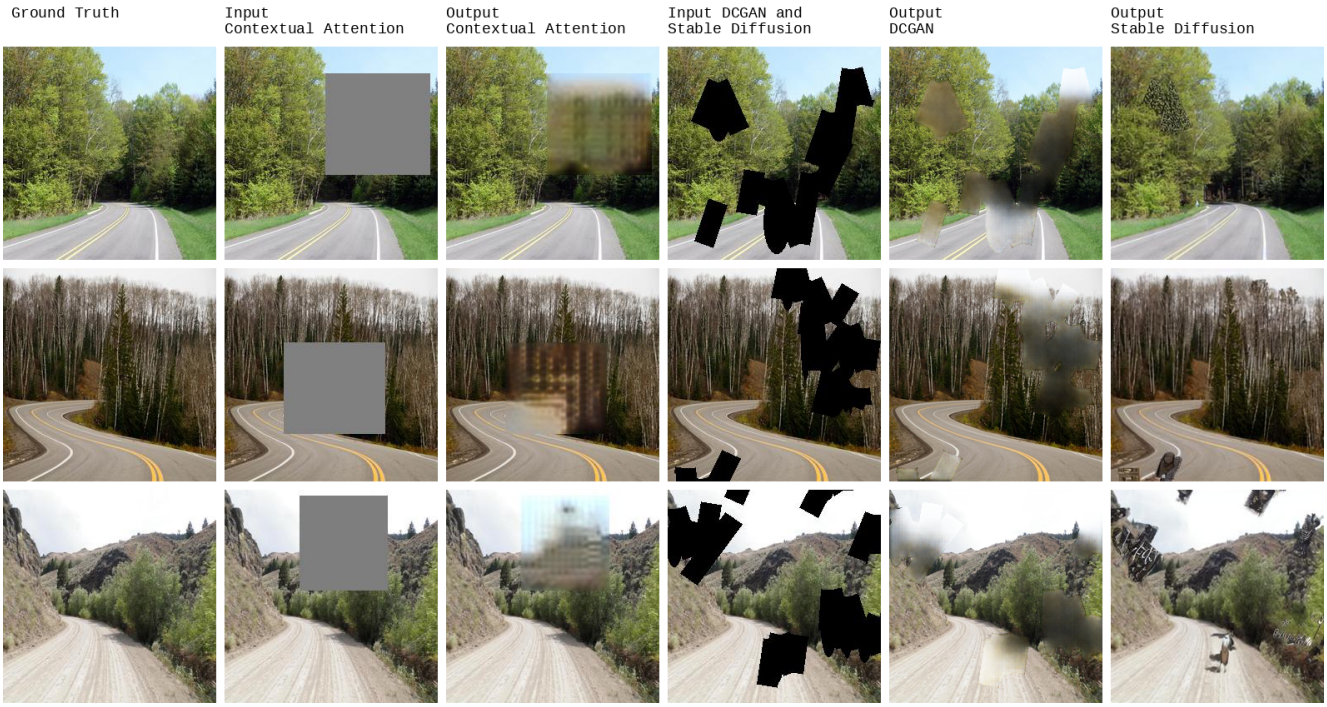

Figure 3. Inpainting results of 3 different methods

which means hallucinations can be a weakness if comparing the ground truth and the generated image directly using MSE or MAE. The Contextual Attention model demonstrates a good understanding of the macrostructure of the image but its patches are pixelated in nature and lack details, which can be an easy red flag to the human eye. The DCGAN model achieves the best MSE and MAE, but has the worst FID. The low FID performance shows clearly in Figure 3, where its patches are foggy and too smooth.

Comparing the quantitative and the qualitative evaluation, FID seems to align better with human perception when it comes to evaluating the in-painting task. According to this metric, stable diffusion is the best model overall.

4. Work Division

Table 2 presents the contributions of each team member in the project. All the team members have contributed equally to the project. Among the four of us, we have developed 2 models, DCGAN and Contextual Attention, adapted a pre-trained Diffusion-based model, and evaluated them on a shared test set. The heavy lifting for DCGAN was done by Tanmay Kenjale, Contextual Attention was done by Manoj Parmar and Palash Choudhary, and Diffusion-based model and evaluation were done by Kien Tran. All the other work including report writing was divided equally.

References

[1] Places205 Dataset. https://datasets.activeloop.ai/docs/ml/datasets/places205-dataset/. 2

[2] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein gan, 2017. 3

[3] Jason Brownlee. How to implement the frechet inception distance (fid) for evaluating gans, 2019. https://machinelearningmastery.com/how-to-implement-the-frechet-inception-distance-fid-from-scratch/. 3

[4] daa233. Generative inpainting with contextual attention in pytorch. https://github.com/daa233/generative-inpainting-pytorch, 2021. 3

[5] Qiwen Fu, Yuxin Yang, and You Guan. Image inpainting and object removal with deep convolutional gan, 2018. 2

[6] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks, 2014. 2

[7] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. 2017. 3

[8] Nathan Inkawhich. Dcgan tutorial. 2

[9] Liangwei Jiang. Janspiry/palette-image-to-image-diffusion-models: Unofficial implementation of palette: Image-to-image diffusion models by pytorch, 2022. 3

[10] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks, 2016. 1, 2

Student Name	Contributed Aspects	Details
Kien Tran	Data Loader, Diffusion-based model, Evaluation	Extract free-form masks and create data loader; Research, adapt, and run diffusion-based model; Evaluate results across all methods, Create visualization and write report
Manoj Parmar	Rectangular mask creation, Contextual Attention model, Report	Research on different approaches, Extract rectangular masks; Create generator network; Model trainer, Create utils for the code, Write report
Palash Choudhary	Random mask creation, Contextual Attention model, Report	AOT Paper Survey, Create randomized masks; Create discriminator network; Model validation, Write report
Tanmay Kenjale	DCGAN model, Report	Researched different DCGAN implementations and adapted different techniques to inpainting; Model finetuning and validation; Write report

Table 2. Contributions of team members

[11] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2021. 1, 3

[12] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation, 2015. 2

[13] Chitwan Saharia, William Chan, Huiwen Chang, Chris A. Lee, Jonathan Ho, Tim Salimans, David J. Fleet, and Mohammad Norouzi. Palette: Image-to-image diffusion models, 2021. 3

[14] Patrick von Platen, Suraj Patil, Anton Lozhkov, Pedro Cuenca, Nathan Lambert, Kashif Rasul, Mishig Davaadorj, and Thomas Wolf. Diffusers: State-of-the-art diffusion models. https://github.com/huggingface/diffusers, 2022. 3

[15] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas Huang. Free-form image inpainting with gated convolution, 2019. 1

[16] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S. Huang. Generative image inpainting with contextual attention, 2018. 1, 2

[17] Yanhong Zeng, Jianlong Fu, Hongyang Chao, and Baining Guo. Aggregated contextual transformations for high-resolution image inpainting, 2021. 1