A Generative Adversial Network for Human Face Completion

Traditional GANs

Generative Adversarial Networks (GANs) were first introduced by Ian Goodfellow and his colleagues in 2014. The original GAN architecture consists of two neural networks: a generator and a discriminator. The generator's objective is to generate data that mimics real data, while the discriminator's task is to distinguish between real data and data generated by the generator.[6]

Traditional GANs operate by having these two networks play a zero-sum game. The generator tries to fool the discriminator by producing more realistic data over time, while the discriminator becomes better at detecting fake data. The training process continues until the generator creates data that is indistinguishable from the real data according to the discriminator.

Types of GANs

Over the years, several variations of GANs have been developed to address specific challenges or to enhance their capabilities:

- **Deep Convolutional GANs (DCGANs)**: This type of GAN incorporates convolutional layers in both the generator and discriminator, making them particularly effective for image generation tasks. DCGANs are designed to capture spatial hierarchies in images, which allows them to generate high-quality, realistic images.[1]
- **Conditional GANs (cGANs)**: In cGANs, both the generator and discriminator are conditioned on additional information, such as class labels. This allows the generator to produce images of specific categories, providing more control over the generated output.[2]
- **CycleGANs**: These are used for image-to-image translation tasks where the goal is to transform images from one domain to another (e.g., converting paintings to photographs). CycleGANs do not require paired datasets, making them highly flexible.[3]
- **Wasserstein GANs (WGANs)**: WGANs address the issue of mode collapse in traditional GANs by using the Wasserstein distance as a loss function. This results in more stable training and often leads to higher-quality generated data.[4]
- **StyleGANs**: StyleGANs introduce a new architecture that allows for control over different aspects of the generated images, such as facial features in face generation tasks. These GANs are particularly famous for generating highly detailed and realistic.[5] images.

Project

GitHub Project Repository Link

The goal of our project is to develop a GAN-based model for face completion. Face completion involves generating the missing parts of a face given an incomplete image. We have implemented a Deep Convolutional GAN (DCGAN) architecture, which is particularly well-suited for this task due to its ability to capture spatial hierarchies and generate high-resolution images.

Project Objectives

- Develop a robust GAN architecture capable of performing face completion tasks.
- Enhance the generated image quality by integrating perceptual loss, which ensures the generated images not only resemble real images in pixel space but also in high-level feature space.
- Log training metrics and generated images to Weights and Biases (WandB) for monitoring and analysis.

Project Plan

The project is structured into the following phases:

1. **Phase 1: Data Preparation**
   Prepare the dataset by dividing images into masked and unmasked pairs for training the GAN. Part of the image (either left or right) is randomly masked and fed as input to the model.

2. **Phase 2: Model Development**
   Implement the Generator and Discriminator models using the DCGAN architecture.

3. **Phase 3: Training and Evaluation**
   Train the GAN model on the dataset, logging metrics and generated images to W and B. Evaluate the model's performance using validation data. The loss used is Binary Cross-Entropy with Logits Loss (BCE-WithLogitsLoss) for both the Generator and Discriminator

4. **Phase 4: Fine-Tuning and Optimization**
   Adjust model hyperparameters and refine the architecture to achieve the best possible image completion results. A learning rate of 0.0001 and batch size of 64 using the Adam optimizer was tried out.

Model Architecture

1. The generator is designed to take in masked images (images with missing regions) and output completed images. It uses a series of convolutional and transposed convolutional layers to transform the input image through various feature spaces. The network starts with a 3-channel input image and processes it through several convolutional layers, each followed by a LeakyReLU activation and batch normalization, increasing the number of filters (from 64 up to 512). The transposed convolutional layers then progressively upsample the image back to its original resolution. The final layer uses a Tanh activation function to output a 3-channel image, representing the completed image.

2. The discriminator acts as a binary classifier, distinguishing between real and generated (fake) images. It takes a 3-channel image as input and passes it through a series of convolutional layers. Similar to the generator, these layers also utilize LeakyReLU activations and batch normalization, but the number of filters increases only up to 512. The final output of the discriminator is a single value per image, obtained through a 1x1 convolution, which represents the probability of the input being a real image.

Model Training

The Model was trained for 100 epochs and after every epoch the image generated were logged into wandb (Weights and Biases). Below are the training graphs for the same.The images shown below are at 15 epochs.

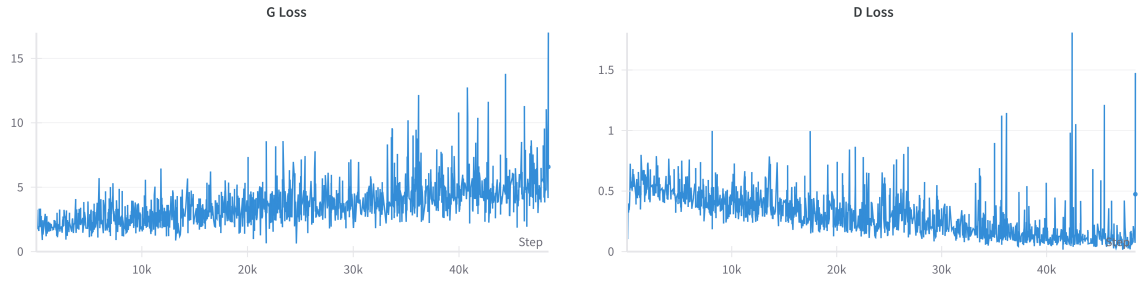

Figure 1: Training Losses for Generator (left) and Discriminator (right).

Results

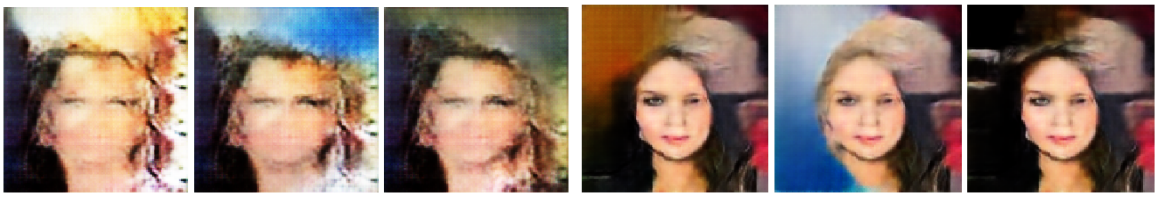

Figure 2: Comparison of Images after epochs.

Future Works

In the future, several enhancements to the current GAN model are planned:

- **Increase Kernel Size:** One potential improvement is to increase the kernel size in the convolutional layers of the generator and discriminator. This could capture larger spatial features and improve the quality of the generated images.[7]
- **Modify the Masking Strategy:** Instead of randomly masking parts of the input images, we are considering using a more structured masking approach. This might help the GAN learn more effectively and produce higher quality completions.
- **Text-to-Image GAN:** Another exciting direction is to develop a GAN model that can generate images based on textual descriptions. This would involve integrating a text encoder and training the model over an extended period, possibly up to 200 epochs, to improve the quality and relevance of the generated images.
- **Longer Training Duration:** Increasing the number of training epochs to 200 may allow the model to learn more complex features and improve the overall image generation quality. This extended training will also be accompanied by more sophisticated loss functions to enhance convergence and stability.

References

1. https://arxiv.org/abs/1406.2661
2. https://arxiv.org/abs/1411.1784
3. https://arxiv.org/abs/1511.06434
4. https://arxiv.org/abs/1701.07875
5. https://arxiv.org/abs/1611.04076
6. https://arxiv.org/abs/1703.10593
7. Ding, X., Zhang, X., Han, J. and Ding, G., 2022. Scaling up your kernels to 31x31: Revisiting large kernel design in cnns. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 11963-11975).