Classical Artwork Generation using Generative Adversarial Networks
CS-GY 6953 / ECE-GY 7123 Deep Learning Project Report — Spring 2024

Giorgi Merabishvili[1], Aaftab Mohammad[2], Mohd Faizaan Khan[3]

[1]New York University, Tandon School of Engineering, Department of Electrical and Computer Engineering (ECE)
[2]New York University, Tandon School of Engineering, Department of Electrical and Computer Engineering (ECE)
[3]New York University, Tandon School of Engineering, Department of Computer Science and Engineering (CSE)
Final Project GitHub Repository:
https://github.com/giorgix121/Classical-Artwork-Generation-using-Generative-Adversarial-Networks

Abstract
This paper presents a Generative Adversarial Network (GAN) designed to generate high-quality artwork images based on the "Best Artworks of All Time" dataset. Our GAN model features a generator-discriminator architecture optimized through spectral normalization and dropout techniques to enhance image quality and stability. The generator comprises initial layers focusing on broader features and detail layers refining finer details, while the discriminator incorporates adaptive average pooling and Gaussian noise augmentation to combat overfitting. Over 500 epochs, we iteratively adjusted learning rates based on discriminator performance feedback, achieving notable improvements in the fidelity and diversity of generated images. Experimental results demonstrate the efficacy of our approach in producing realistic and diverse artwork images, advancing the state-of-the-art in GAN-based art generation.

Introduction
Generative Adversarial Networks (GANs) have revolutionized image synthesis, enabling the creation of realistic images across various domains. This paper proposes a GAN architecture to generate artwork images using the "Best Artworks of All Time" dataset. To address common challenges like mode collapse and instability, we incorporate spectral normalization and dropout layers. The generator is designed with initial layers capturing broad features and detail layers refining these features for high-quality images. The discriminator is augmented with Gaussian noise to improve robustness.
We employed a feedback mechanism to dynamically adjust the learning rates of the generator's layers based on discriminator performance. This adaptive strategy enhanced training stability and output quality. Over 500 epochs, our model consistently produced diverse and high-quality artwork images, demonstrating its potential in GAN-based image synthesis in the artistic domain.

Literature Survey
GANs like StyleGAN and BigGAN have pushed boundaries of art generation, producing complex and artistically diverse images. While some studies like Elgammal et al. (2017) have shown how AI can learn to imitate artistic styles, others (Kazemi 2019) have investigated the possibility of using these models to produce completely original works. Building on this foundation, our project focused on classical art to explore historical artistic expressions.

Dataset
The "Best Artworks of All Time" dataset, sourced from Kaggle, serves as the foundation for our GAN-based art generation model. This dataset comprises a diverse collection of 8683 images featuring renowned paintings from various artists and art movements. Each image in the dataset has been resized to 128x128 pixels to standardize input dimensions for our model. We applied data augmentation techniques, including random horizontal flips and rotations, to increase the variability of the training set and improve the model's generalization capabilities. Additionally, we calculated the dataset's mean and standard deviation to normalize the images, ensuring consistent training conditions. This dataset's rich diversity and high-quality images provide an excellent basis for training our GAN to generate realistic and varied artwork images.

Architecture
I – Generator
Architecture:
The generator is designed with a two-stage approach. The initial layers capture broad features using ConvTranspose2d layers with dimensions of 1024 and 512 feature maps, followed by ReLU activations and BatchNorm2d layers for stabilization. The detail layers refine these features using ConvTranspose2d layers with smaller

dimensions (256, 128, 64) and similar activation and normalization layers, concluding with a final ConvTranspose2d layer outputting a 3-channel image with a Tanh activation function.

Reason for Use:

This architectural design allows the generator to progressively build up an image from a low-resolution representation to a high-resolution output, effectively capturing both broad structural features and fine details. The use of BatchNorm2d layers helps stabilize the training process by normalizing intermediate feature maps. The inclusion of adaptive learning rates for the initial and detail layers based on discriminator feedback further enhances the training dynamics.

Advantages:

- High-Quality Image Generation: The progressive refinement approach enables the generation of detailed and high-quality images.
- Stabilized Training: Batch normalization layers help in maintaining stable training dynamics.
- Efficient Feature Learning: The two-stage architecture ensures efficient learning of both global structures and local details.
- Adaptive Training: Adjusting learning rates based on discriminator feedback optimizes training and prevents mode collapse.

```
==========
Total params: 13,245,312
Trainable params: 13,245,312
Non-trainable params: 0
----------
```

Figure 1: Number of parameters of Generator

II - Discriminator

Architecture:

The discriminator employs a series of spectral normalized Conv2d layers, starting with 64 feature maps and progressively increasing to 512. Each Conv2d layer is followed by a LeakyReLU activation and a dropout layer to prevent overfitting. Gaussian noise is added to the input images to increase robustness. The final layers include an adaptive average pooling layer, flattening, and a Sigmoid activation to output a single probability score.

Reason for Use:

Spectral normalization in the Conv2d layers constrains the spectral norm of the weight matrices, ensuring more stable training. Dropout layers provide regularization, reducing the risk of overfitting. Adding Gaussian noise to the input images improves the discriminator's robustness, enabling it to handle variations and noise more effectively.

Advantages:

- Training Stability: Spectral normalization helps maintain stable and balanced training dynamics between the generator and discriminator.
- Regularization: Dropout layers help prevent overfitting, allowing the discriminator to generalize better.
- Enhanced Robustness: Adding Gaussian noise to the input images improves the discriminator's ability to handle variations and noise, leading to more robust performance.
- Improved Feedback: The robust discriminator provides better feedback to the generator, facilitating more realistic image generation.

```
==========
Total params: 2,763,776
Trainable params: 2,763,776
Non-trainable params: 0
----------
```

Figure 2: Number of parameters of Discriminator

Methodology

I - Loss Function

We utilized the Binary Cross-Entropy (BCE) loss for both the generator and discriminator. The discriminator aims to maximize the log-likelihood of correctly classifying real and fake images, while the generator aims to minimize the log-likelihood of the discriminator correctly identifying its outputs as fake. This adversarial loss function ensures that the generator produces increasingly realistic images over time.

Real Images Loss:

$L_{D_{real}} = -E_{\mathbf{x}\sim p_{data}(\mathbf{x})}[\log D(\mathbf{x})]$

$D(\mathbf{x})$ is the discriminator's output for real images x, and $p_{data}(x)$ is the data distribution.

Fake Images Loss:

$L_{D_{fake}} = -E_{\mathbf{z}\sim p_{\mathbf{z}(\mathbf{z})}}[\log(1-D(G(\mathbf{z})))]$

$G(\mathbf{z})$ is the generator's output for the input noise z, and $p_z(z)$ is the noise distribution.

Total Discriminator Loss:

$L_D = L_{D_{real}} + L_{D_{fake}}$

Generator Loss:

$L_G = -E_{\mathbf{z}\sim p_{\mathbf{z}(\mathbf{z})}}[\log D(G(\mathbf{z}))]$

II - Training Procedure

The training process involves alternating updates between the generator and the discriminator. For each iteration, we perform multiple updates to the discriminator using both real and generated images, followed by an update to the generator. The discriminator's training includes adding Gaussian noise to the real images to enhance robustness, while the generator's training involves backpropagating the adversarial loss to improve image generation quality.

III - Hyperparameters

- The learning rates for the generator's initial and detail layers are set to 0.0002 and 0.00005, respectively.
- The Adam optimizer with beta values of 0.5 and 0.999 is used for both the generator and discriminator.
- A batch size of 64 and a total of 500 epochs are employed.
- The latent vector size for the generator's input is set to 128.

IV - Data Augmentation

To improve the generalization capability of the model, we applied several data augmentation techniques, including random horizontal flips and random rotations up to 5 degrees. These augmentations increase the variability in the training data, helping the model learn more robust features.

V - Learning Rate Schedulers

A dynamic learning rate adjustment strategy was implemented based on the performance feedback of the discriminator. If the discriminator's performance exceeded a predefined threshold, the learning rates for both the generator's initial and detail layers were adjusted to maintain balanced training dynamics. This adaptive learning rate mechanism prevents the discriminator from overpowering the generator or becoming too weak.

VI - Other Training Details

Throughout the training process, we used the Adam optimizer for both the generator and discriminator with different learning rates for the initial and detail layers of the generator (0.0002 and 0.00005 respectively). We employed spectral normalization in the discriminator to stabilize training and prevent exploding gradients. Models were saved at regular intervals to allow for monitoring progress and future fine-tuning. Additionally, we used TensorBoard for visualizing training metrics and generated images to assess the quality and diversity of the outputs over time. A fixed latent vector was used to generate images at the end of each epoch for consistent evaluation and comparison. This comprehensive approach ensures the production of high-quality realistic artwork images showcasing the potential of our GAN architecture.

Results

The results of our GAN-based artwork generation model demonstrate significant improvements in the quality and diversity of the generated images over the training period. The model was trained for 500 epochs, during which we observed that the adaptive learning rate mechanism effectively balanced the training dynamics between the generator and discriminator, preventing issues such as mode collapse.

The generated images exhibited a high level of detail and artistic style closely resembling the artworks in the training dataset. Notable improvements were observed after incorporating spectral normalization and dropout layers in the discriminator, which contributed to more stable training and higher quality

outputs. The use of Gaussian noise in the discriminator's input further enhanced the robustness of the model, allowing it to generate more realistic and diverse images.

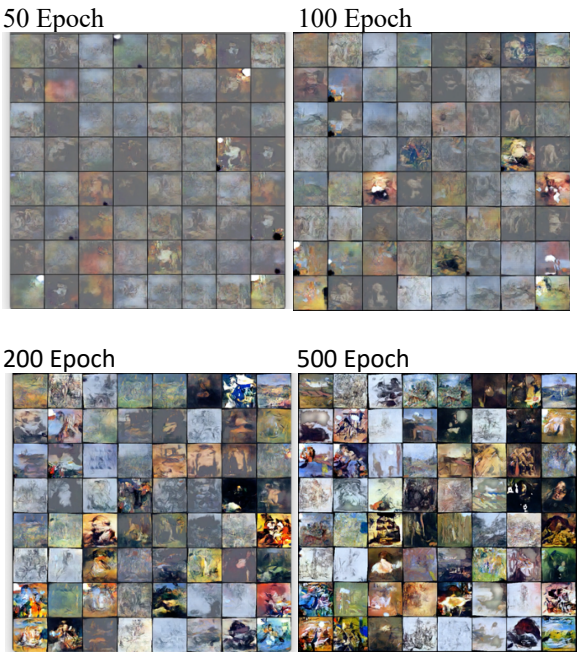

Figure 3: Images generated at different epochs

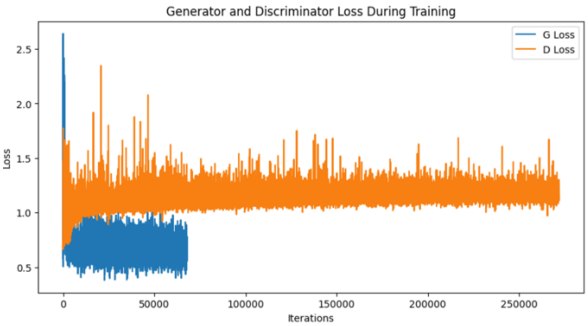

Figure 4: Generator and Discriminator Loss During Training

System Specifications:

```
CPU:  Intel(R) Xeon(R) CPU @ 2.00GHz

GPU: Tesla V100-SXM2-16GB
System Memory: 52217.5 MB
Python Version: 3.10.12 (main, Nov 20 2023, 15:14:05) [GCC
CUDA Version: 12.1
Torch Version: 2.2.1+cu121
```

Figure 5: System Specifications

References:

- Elgammal, A., Liu, B., Elhoseiny, M., & Mazzone, M. (2017). CAN: Creative Adversarial Networks, Generating "Art" by Learning About Styles and Deviating from Style Norms. arXiv preprint arXiv:1706.07068.
- Baker, J. (2023). ARTEMIS: Using GANs with Multiple Discriminators to Generate Art. arXiv. https://arxiv.org/abs/2311.08278
- Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2014). Generative Adversarial Nets. Advances in Neural Information Processing Systems, 27.