GAN-Based Face Rotation for Creative Portraits

Team:
Harika Velagala (hxv220012)
Namratha Reddy Karedyy (nxk220058)

Problem Summary

The objective here involves creating a GAN-driven model capable of rotating facial orientations within artistic portraits across different angles. The primary aim is to create a dataset comprising artistic portraits through the integration of a 3D face model, thereby facilitating GAN training that not only rotates facial features but also retains the distinct artistic styles inherent in these portraits. In this process, we try to solve below challenges:

- Artistic Style Preservation: Maintaining the unique artistic style within a portrait while rotating faces poses a challenge that traditional face rotation models might not address adequately.
- Diversity in Artistic Portraits: Artistic portraits exhibit diverse forms like oil paint, watercolor, and banknote renditions, each possessing its characteristic traits. Adapting a model to encompass these diverse styles becomes pivotal.
- Quality of Rotated Faces: Generating compelling and top-tier rotated facial features within artistic portraits is paramount for applications spanning art restoration to the creation of novel artworks.

This problem aims to bridge the gap in existing face rotation models by specifically addressing the complexities of artistic portraits, preserving their unique styles, and producing accurate and convincing rotated faces at various angles.

Introduction

This project's main goal is to use GANs' capacity to create intriguing and distinctive artistic portraits. Two neural networks, the discriminator and the generator, are involved in a high-stakes game to form GANs. Whereas the discriminator aspires to be a perceptive art critic who can tell the genuine from the fake, the generator aims to produce images that are not distinguishable from authentic portraiture.

The route that this project takes starts with preparing the data and continues with designing and training a GAN model. The generator's capacity to create portraits is honed during the training process. Our GAN develops into a virtual artist as we iterate through this adversarial process, being able to create portraits. We explore optional fine-tuning approaches to reach the maximum levels of creativity and visual attractiveness.

Description of Dataset

The "Art Portraits" dataset, which is accessible on Kaggle, is a set of picture data meant to be used in Generative Adversarial Network (GAN) model creation. The majority of the photographs in the collection are portraits and are usually creative depictions of people's faces. These portraits may cover a broad spectrum of historical periods, artistic movements, and media. The dataset is appropriate for a range of deep learning and machine learning problems, such as:

1. GAN Instruction: It can be applied to GAN model training in order to produce new portrait photos. AI generated portrait photos can be produced by using this dataset to train a GAN.
2. Category of Images: The dataset can be used to create image classification models that tag or classify portraits according to various parameters, such as historical periods or creative styles.
3. Compositing Images: The photos can be utilized for image processing operations like artistic filter application on portrait photos, picture enhancement, or style transfer.
4. Creative Evaluation: This dataset can be helpful for researchers and art fans to examine various portrait art trends, techniques, and artistic styles.

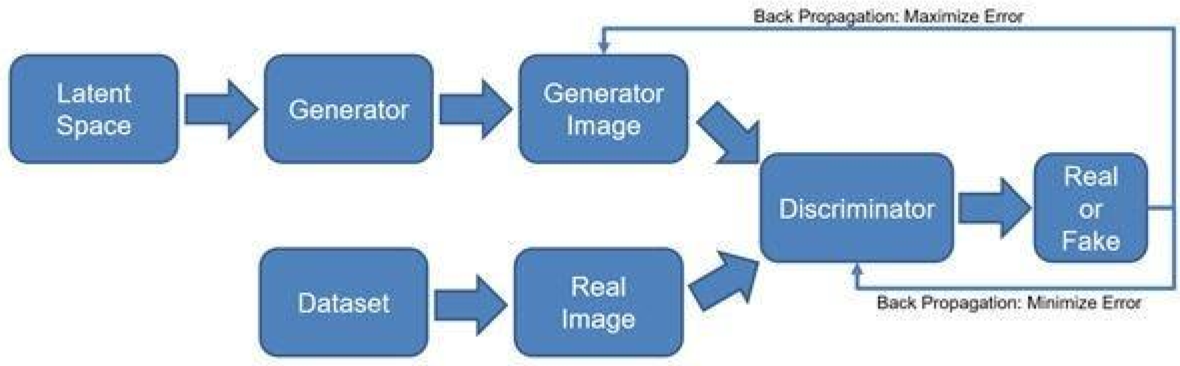
Implementation

Code :
https://colab.research.google.com/drive/1rBUSYBp8LGB6K-yW-vf9VgOzOy0hyvSO?usp=sharing

Step 1 : Data Collection
Data sourced from Kaggle is a portrait based image dataset as explained above.
Link to dataset: https://www.kaggle.com/datasets/karnikakapoor/art-portraits/

Step 2: Data Preprocessing
Data Preprocessing: Common preprocessing steps include resizing images to a consistent size, normalizing pixel values, and possibly augmenting the data to increase its diversity.

Step 3: GAN Architecture
Generator Design: The generator's role is to create realistic portrait images from random noise. It typically consists of convolutional and transposed convolutional layers.
Discriminator Design: The discriminator's task is to distinguish between real and generated images. It often comprises convolutional layers followed by fully connected layers.



Step 4: Loss Functions and Optimizers
Loss Functions: Define the loss functions for the generator and discriminator. These loss functions guide the training process. Binary cross-entropy loss is a common choice for GANs.
Optimizers: Select optimizers for updating the model's weights during training. Adam is a popular optimizer for GANs.

Step 5: Training Loop
Training Loop: Implement the training loop for the GAN. This loop consists of several iterations (epochs). During each iteration:

- Random noise is provided to the generator.
- Real and fake images are presented to the discriminator.
- Discriminator and generator losses are computed.
- Gradient descent is used to update the network weights.

Epochs: Train the GAN over multiple epochs to allow the model to learn and improve. The number of epochs depends on the complexity of the dataset and desired quality of generated images.

Step 6: Generate Portraits

Generate Portraits: After successful training, you can use the trained generator to produce new portrait images. These generated images are a creative output of the GAN model.

Step 7: Evaluation and Tuning

Evaluation: Assess the quality of the generated portraits. You can employ metrics like Inception Score or conduct user surveys to gather feedback and measure the performance of your GAN.

Tuning: If the quality of the generated images is not satisfactory, you can fine-tune the GAN by adjusting hyperparameters, architecture, or data preprocessing to enhance the results.

Step 8: Save the Model

Model Saving: Save the trained generator and discriminator models. This allows you to use them for future generation or transfer learning tasks without having to retrain the GAN.

Challenges we faced:

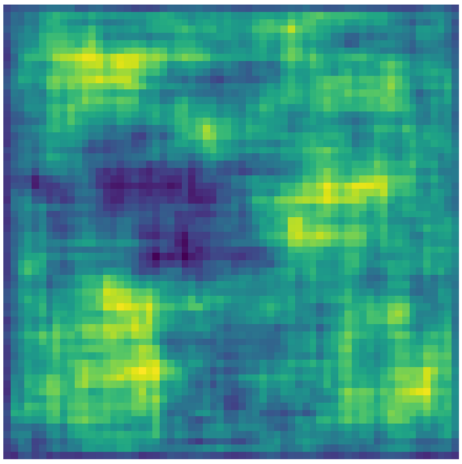
- Evaluation Metrics: Being able to decide on metrics like error functions, activation functions was time consuming and involved a lot of research.
- Computational Resources: GANs demand significant computational power and memory, especially when handling high-resolution images. We faced many roadblocks due to resource availability that could process non binary images. We tried to execute the same code in GPU, Linux and even Google Colab Pro version. After rigorous efforts, we tried to optimize the code and used images in their binary form for execution.
- Quality of Rotated Faces: Although we observed rotation in our results at higher epoch values, it was difficult to assess the quality and accuracy of the final results. As time taken to run these programs were high, it was difficult to generate clear and quality driven images in our computer.

GAN Architecture

1. Generator:

Random noise is intended to be transformed into creatively rotating the portraits by the generator. It starts with a dense layer that receives as input a noise vector of 100 dimensions. The outcome is reshaped into a 10x10x1 feature map, which is used as a basis for creating images with a higher resolution. There are two transposed convolutional layers with 128 and 64 filters in each. These layers preserve spatial information while upsampling the feature map. To stabilize and expedite the training process, batch normalization layers are introduced. Nonlinearity is introduced through the use of leaky ReLU activation functions.
Three channels are produced by the final convolutional layer, which represent the RGB color channels in the output image. The activation function 'tanh' guarantees that the values of the output pixels are between [1, 1], which makes it appropriate for creating images.

Below is one sample fake portrait created by generator:



2.Discriminator :

It is the discriminator's responsibility to discern between created and actual images. The 'same' padding, stride of 2, and 64 filters make up the first convolutional layer. It handles three color channels and 80x80 pictures. Leaky ReLU activation functions help in the learning of pertinent characteristics by introducing nonlinearity. Next is a second convolutional layer with 128 filters, stride 2, and'same' padding. The discriminator's output indicates whether the input image is real or fake after the features are flattened and run through a dense layer with a single unit.

Loss Functions

1. Generator Loss:

The generator's loss function attempts to quantify the degree to which it can generate images that deceive the discriminator into believing they are real.It makes use of the binary cross entropy loss, which measures the discrepancy between the goal label of "real" (i.e., ones) and the discriminator's categorization of false images.

2. Discriminator Loss:

The discriminator loss function assesses how well the discriminator separates authentic images from phoney ones. It is made up of two parts:

Real Loss: Indicates how successfully the discriminator identifies authentic photos. This label is the target: "real.

Phoney Loss: Indicates how well a discriminator can identify Phoney photos that have been created. The intended designation is "fake."

Since both factors are essential to the discriminator's performance, the total loss is the sum of the real and false losses.

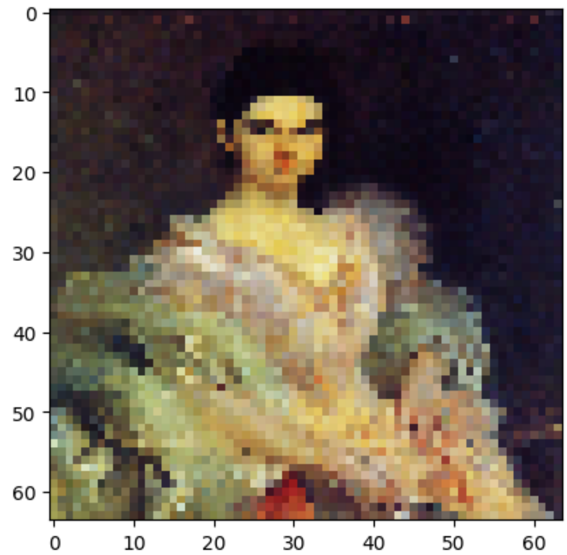
Training Loop

The two neural networks compete during the training process, which uses adversarial training. The discriminator aims to improve their ability to discern authentic from fraudulent photos, while the generator tries to produce better forgeries to trick the discriminator. They are alternately updated during the training loop's several epochs.

Using the Adam optimizer, which modifies model parameters to minimize specified loss functions, the models are optimized throughout the training phase.

Data Preprocessing

Before training, real image data is preprocessed to match the architecture of the model. Images are shrunk to a consistent size of 80 by 80 pixels. To stabilize training and reduce the possibility of gradient problems, pixel values are normalized to the range [0, 1].

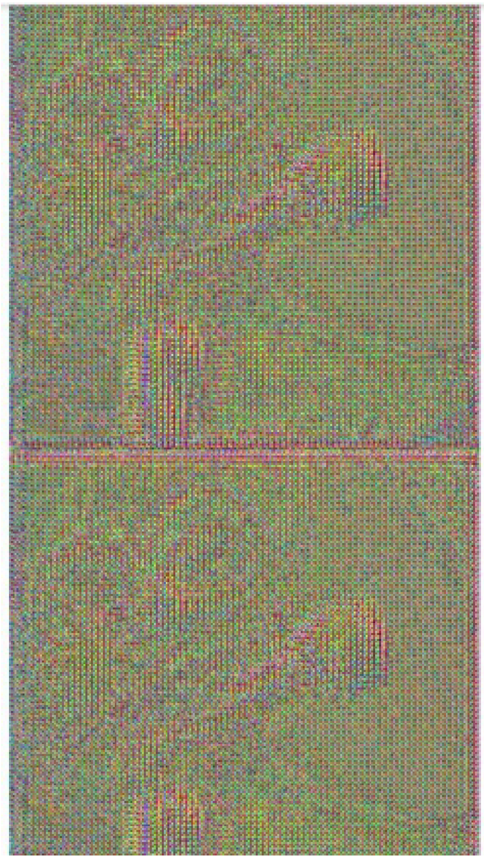


Generated Images

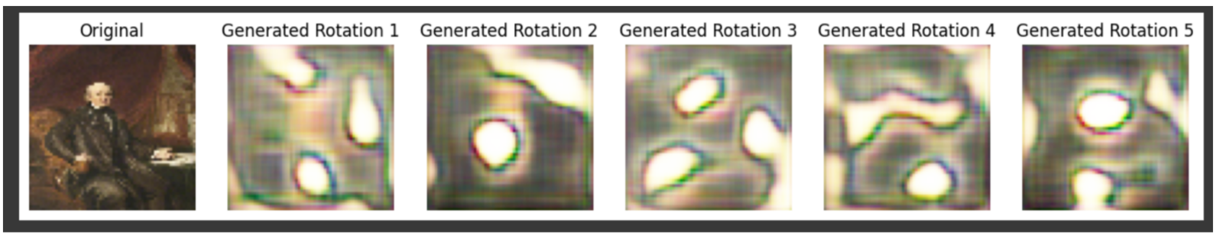
The trained GAN model is able to produce creatively rotated portrait photos. The generator adds artistic expression to the original photographs by generating inventive variations of the input face using random noise as input.

Results

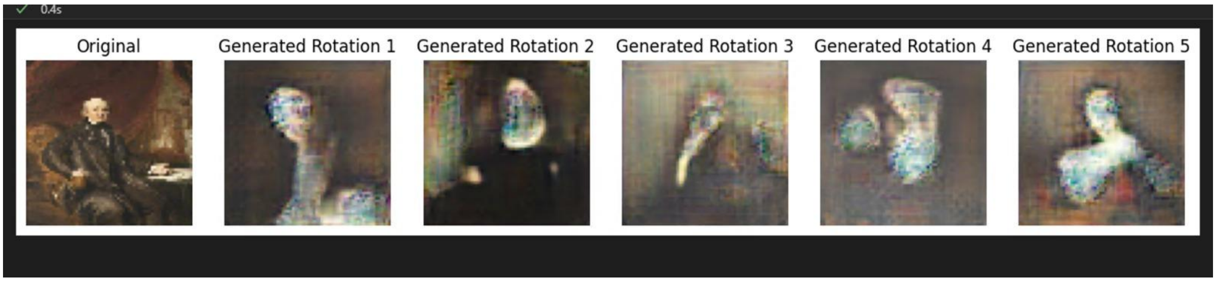
The code saves the trained generator model after running the training loop for a predetermined number of epochs. It also stores generated images during training so that they can be assessed and used to track the model's development. The trained generator model can be used to create creatively rotated faces once training is complete. We ran the model over a sample image and this is the rotated image generated.



Result for 20 epochs:



Result for same portrait for 150 epochs: (Processing time = 341 mins)



We observe the image quality improves with increasing epochs. Above two images show rotations for 5 different angles.

Analysis:

Goals achieved:

- We were able to successfully observe face rotation which was our primary goal
- The dataset included diverse paintings like oil painting, watercolor painting. Our algorithm was successfully able to rotate the images
- Explored the need for different technical components. For example, we understood that batch normalization aids in stabilizing training and maintaining image quality. This took place when we compared our image results before and after implementing batch normalization

Future Research and Applications

Subsequent research endeavors could entail augmenting the visual aesthetics of the produced imagery and investigating alternative GAN structures. The project's output has applications in graphic design, digital art, photography, and the creation of artistic content, among other creative fields.

Conclusion

• The research opens up new possibilities for imaginative and creative digital art and design by showcasing the ability of GANs to produce artfully rotating portrait photos.

• This section's code demonstrates the model architecture and loss functions that form the foundation of this fascinating project.

References

Generative Adversarial Networks (GANs):

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... & Bengio, Y. (2014). Generative adversarial nets. Advances in neural information processing systems (NeurIPS).

Portrait Generation and Manipulation:

Brock, A., Donahue, J., & Simonyan, K. (2019). Large scale GAN training for high fidelity natural image synthesis. In the International Conference on Learning Representations (ICLR).

Artistic Image Generation with GANs:

Gatys, L. A., Ecker, A. S., & Bethge, M. (2015). A neural algorithm of artistic style. arXiv preprint arXiv:1508.06576.
Chen, Q., Xu, Z., & Weinshall, D. (2017). Infogan-crf: An unsupervised generative model with controllable representation. In Proceedings of the IEEE International Conference on Computer Vision (ICCV).

Face Rotation and Manipulation:

Zhang, Z., Song, Y., Qi, H., Huang, H., & Zhang, S. (2019). Adversarial Autoencoders for Compact Representations of 3D Point Clouds. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).