Fine-Tuning StyleGAN for CryptoPunk-Style Image Generation: A Study in Style Transfer and Generative Adversarial Networks

Seong Woo AHN
Graduate School of AI
chrisahn99@kaist.ac.kr

Alexandre Petit
Graduate School of AI
alexandrepetit@kaist.ac.kr

Victor Osso
Graduate School of AI
osso.victor@orange.fr

Abstract

This report explores the application of StyleGAN and StyleGAN2 architectures to develop a CryptoPunk-style image generator, blending artistic creativity with cutting-edge generative adversarial networks (GANs). The project aimed to fine-tune StyleGAN models on a dataset of 10,000 pixelated CryptoPunk images, leveraging transfer learning to create personalized cartoon versions of user-provided inputs. Despite challenges such as low-resolution training data and limited computational resources, our experiments successfully demonstrated the feasibility of generating CryptoPunk-style outputs by fine-tuning a pre-trained StyleGAN model. Through trials of layer swapping and style mixing, the project also provided insights into the mechanisms behind style transfer and advanced GAN techniques. While the final implementation faced limitations, the study highlights the potential for further optimization and application of StyleGAN architectures in creative AI projects.

1. Introduction

In recent years, the intersection of artificial intelligence (AI) and art has become an area of growing interest. With the emergence of novel algorithmic techniques, generative art has risen in prominence. Among these techniques, style blending stands out as one of the most popular approaches. This technique enables the combination of two images: one serving as the content image and the other as the style image. The algorithm produces a new image that retains the coarse structural features of the content image while incorporating the stylistic attributes (e.g., Impressionism or Cubism) of the style image, as illustrated in Figure 1.

Despite the creative potential of generative art, ensuring certification and authenticity remains a significant challenge, particularly in digital spaces. This limitation has led to the rise of blockchain-based projects that address these concerns. Non-Fungible Tokens (NFTs), for instance, enable the certification of ownership and uniqueness for digital assets. A notable example is CryptoPunks, a collection of 10,000 unique 24x24 pixel art characters stored on the Ethereum blockchain. These digital collectibles can be traded or purchased, with the total estimated value of the collection surpassing 1,600,000 ETH (over $5 billion USD as of February 2022). Figures 2 and 3 depict the largest recorded CryptoPunk sale and an example of a CryptoPunk profile page, respectively.

The CryptoPunk style presents a unique artistic approach

Figure 1. Example of style transfer: the content image (left) and the style image (right) are combined to produce the output.

Figure 2. Largest CryptoPunk sale recorded to date.

Figure 3. Attributes of a specific CryptoPunk. The profile page details its features and ownership status.

that applies pixelated aesthetics to human faces. This style has broad appeal, as it allows individuals to generate pixel art versions of their own images, which can serve as personalized profile pictures. Furthermore, this concept has significant potential for applications in social media filters, widely used on platforms like Instagram and TikTok. A CryptoPunk filter could provide distinctive and creative features for casual users while enabling influencers to create and monetize stylized versions of their appearances. Additionally, the integration of NFTs ensures that users can own and trade their personalized CryptoPunk images securely.

In 2018, NVIDIA introduced StyleGAN, a state-of-the-art model capable of generating highly realistic human faces. This model was later refined with the release of StyleGAN2, which achieved even greater performance and control over generative outputs.

2. Definition of the Task

The primary objective of this project is to develop an algorithm capable of generating a CryptoPunk-style version of a given input image. Leveraging transfer learning techniques, the algorithm would apply the distinctive CryptoPunk style to an input image, such as a photograph of a person. The CryptoPunk style was chosen for its unique and creative pixel-art aesthetic, which enables users to generate personalized cartoon versions of themselves. An example of a manually created CryptoPunk-style profile image is shown in Figure 5.

Figure 4. Original input image.

In an ideal scenario, with sufficient computational resources and time, the project could be extended to include animation. This additional feature would involve generating CryptoPunk-style animations from video inputs, such as GIFs, by mimicking head, mouth, and eye movements from the source video. However, we acknowledge the increased complexity of this task and thus consider it an optional extension rather than the core focus of the project.

Figure 5. Handcrafted CryptoPunk-style design based on the original input.

3. Dataset

The training process requires a dataset of CryptoPunk images. The original collection of 10,000 CryptoPunks is readily available on platforms such as Kaggle, making it our primary dataset for this project. This dataset comprises 10,000 PNG images, each with a resolution of 24x24 pixels. The low resolution of these images presents challenges, which will be discussed in later sections. These images will serve as the training data, allowing the algorithm to learn the distinctive style of CryptoPunks.

Several factors motivated our choice of the CryptoPunk dataset. First, CryptoPunks represent one of the most widely recognized NFT collections in the crypto marketplace, making their style both relevant and desirable. Second, the dataset's simplicity (low resolution and pixelized design) intuitively suggested that the training process might be less computationally intensive, reducing the time required for training. Finally, the availability of abundant data further solidified our decision to use this dataset for fine-tuning our model.

4. Related Work

This project draws inspiration from several existing works. Articles such as "Online Tools to Create Mind-Blowing AI Art" [5] and "What is Generative Art?" [3] provided an overview of various forms of AI-generated art, broadening our understanding of the creative potential of these technologies.

Figure 6. Example of style blending using StyleGAN.

Figure 7. Style blending example combining human faces with Japanese art.

Figure 8. Examples of CryptoPunks generated using DGGAN.

A key influence was the article "StyleGAN Network Blending" [6], which introduced the concept of "layer swapping" in StyleGAN models. This approach facilitates the combination of real facial photographs with stylized Japanese artwork, as depicted in Figure 6. The method builds upon foundational research conducted by NVIDIA [4].

The authors demonstrated how swapping layers in StyleGAN2 models generates photorealistic images infused with artistic styles, as illustrated in Figure 7.

Similarly, articles such as "Blending StyleGAN2 Models to Turn Faces into Cats" [7] and "Fine-Tuning StyleGAN2 for Cartoon Face Generation" [1] applied layer swapping techniques to blend human faces with cartoons or animal styles.

In addition, prior work on CryptoPunk generation, such as "Generate NFT CryptoPunks with DGGAN" [2], utilized a DGGAN model to create CryptoPunk-like images. However, DGGAN predates StyleGAN2 and lacks some of its advanced capabilities. Figure 8 presents examples of CryptoPunks generated using DGGAN.

These works collectively inspired the methodology of our project, which leverages StyleGAN2 to create personalized CryptoPunk-style NFTs through style blending.

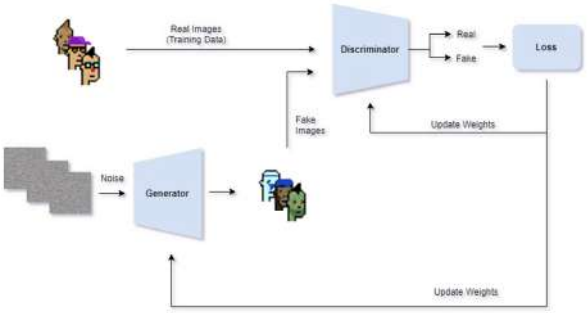

Figure 9. Architecture of a GAN. The generator and discriminator are trained in an adversarial loop.

5. Method

To implement the proposed style blending for creating CryptoPunk-style NFTs, we utilize the StyleGAN2 architecture. Before delving into StyleGAN2-specific details, it is essential to review the foundational principles of *Generative Adversarial Networks* (GANs).

5.1. Overview of GANs

GANs consist of two neural networks: a *generator* and a *discriminator*, trained in an adversarial manner. The generator aims to create realistic images, while the discriminator seeks to distinguish between real and synthetic images.

The training process alternates between the two networks:

1. The discriminator is trained to classify real images (from the dataset) and fake images (generated by the untrained generator).

2. The generator is then trained to produce images that "fool" the discriminator by minimizing the difference between its outputs and real images.

3. This cycle repeats, iteratively improving both the generator and discriminator until the generator produces highly realistic images.

The architecture of a standard GAN is illustrated in Figure 9.

5.2. StyleGAN Architecture

StyleGAN, introduced by NVIDIA in 2018, extends traditional GANs by introducing control over the style attributes of generated images, such as pose, facial structure, lighting, and texture. This is achieved by disentangling the latent space through an additional *mapping network*, which transforms the input noise vector into an intermediate latent vector. This vector is then used to modulate different layers of the generator network, enabling fine-grained control over image attributes (Figure 10).

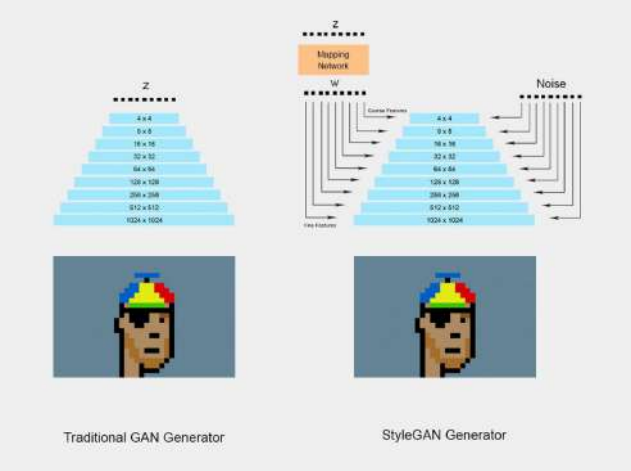

Figure 10. StyleGAN generator architecture, showcasing the disentanglement of style attributes.

5.3. Style Blending Through Layer Swapping

The StyleGAN architecture facilitates style blending by allowing layers of one trained model to be swapped with those of another. This enables the combination of features from two styles. For instance, an image can retain the coarse features (e.g., facial structure, posture) of one model while adopting the fine-grained attributes (e.g., texture, shading, colors) of another.

In this project, we aim to generate CryptoPunk-style images by swapping layers between a StyleGAN model trained on the CryptoPunk dataset and a pre-trained StyleGAN model trained on the Flickr-Faces-HQ (FFHQ) dataset [4]. Figures 11 and 12 illustrates the effect of layer swapping at different resolutions:

- Swapping low-resolution layers affects global features such as facial structure and hairstyle.
- Swapping high-resolution layers modifies fine details such as texture and color.

5.4. Implementation Details

To implement this, we fine-tune a pre-trained StyleGAN model on the CryptoPunk dataset. For style mixing, we use the following steps:

1. Train a StyleGAN2 network on the CryptoPunk dataset.
2. Swap selected layers of this trained model with layers from the pre-trained FFHQ model.
3. Generate new images by combining global features from the FFHQ model with CryptoPunk-style attributes.

Figure 11. Impact of low-resolution layer swapping on global image features.

Figure 12. Impact of high-resolution layer swapping on fine details.

Notable differences between StyleGAN and StyleGAN2 [4] include:

- Simplified processing of the constant input layer.
- Removal of mean normalization in feature maps.
- Relocation of noise injection modules outside the style block.

An example of images from the FFHQ dataset, which serves as the pre-trained model, is shown in Figure 13.

6. Experiment

6.1. Environment

All technical work for this project was conducted in a Google Colab environment. By utilizing a Colab Pro account, we benefited from the following hardware specifications:

Figure 13. Examples from the FFHQ dataset, used as the pre-trained model for style blending.

- 1 NVIDIA Tesla T4 GPU with 16GB of GDDR6 RAM,
- 24 hours of continuous runtime per session.

The decision to use Google Colab was motivated by its ease of use, compatibility with Google Drive for storage, and prior experience of team members with this platform. Google Drive was used to store the CryptoPunks dataset, along with supplementary files such as pre-trained network weights and model checkpoints.

6.2. Understanding the Concepts

6.2.1 Experimenting with Style Mixing

The initial objective was to mix human facial features with artistic styles. Two approaches were considered: generating synthetic faces or using existing images. The latter approach required preprocessing steps to ensure compatibility with the GAN framework.

The first preprocessing step involved aligning and cropping the face in the existing image to 1024x1024 pixels. This step was essential, as StyleGAN models such as FFHQ were trained on centered and standardized images of this resolution. Figure 14 illustrates the alignment and cropping process.

The next step involved projecting the preprocessed image into the latent space of the pretrained StyleGAN2 model (FFHQ). This projection yielded a latent vector representation, enabling the generator to approximate the input image.

Finally, the layer-swapping technique was applied. Layers from two pre-trained models (e.g., FFHQ and a Disney-style StyleGAN2 model) were swapped to blend stylistic attributes. Results of mixing human faces with Disney-style features are shown in Figure 15.

6.2.2 Understanding the Impact of Layer Swapping

The second objective was to study the impact of swapping specific layers. Experiments involved swapping layers between an FFHQ-trained StyleGAN model and a Naver Webtoon StyleGAN. Results, shown in Figure 16, revealed that swapping higher-resolution layers (e.g., layers 5 or 6) primarily affected fine details such as texture and color. Conversely, swapping lower-resolution layers modified broader structural features, such as facial shape.

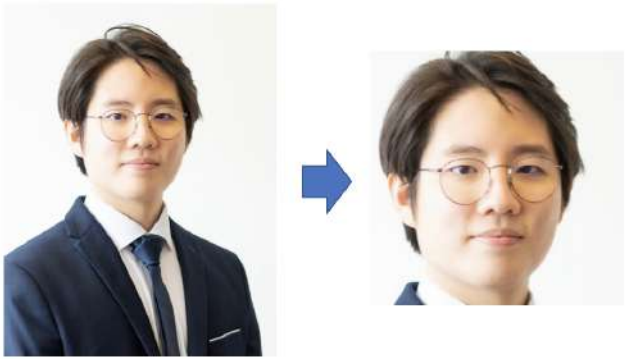

Figure 14. Preprocessing: Centering and cropping the face to 1024x1024 pixels.

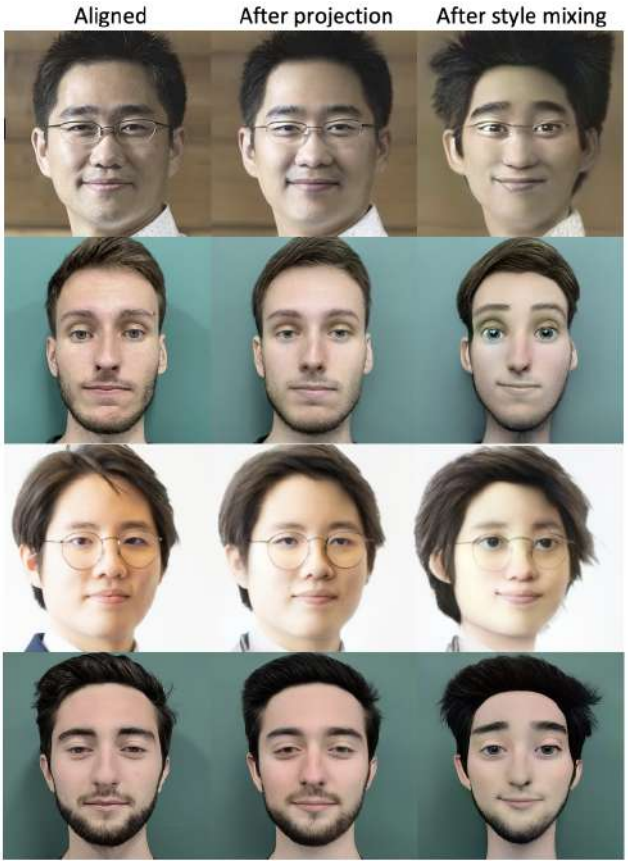

Figure 15. Style blending: Mixing human faces with Disney-style attributes.

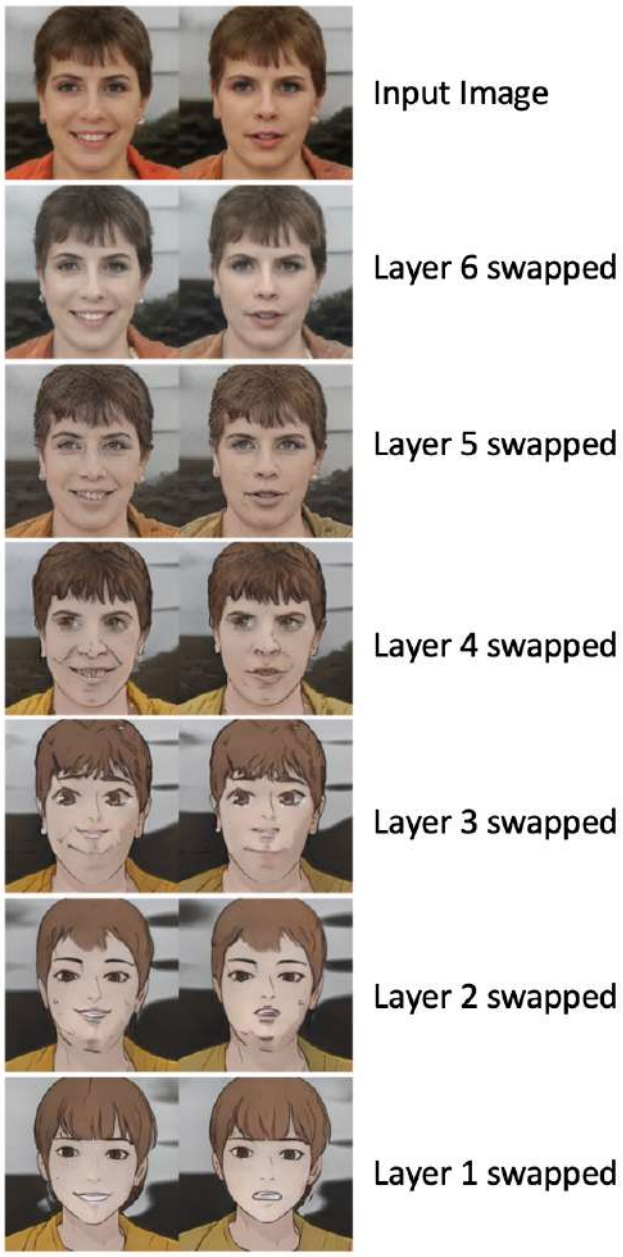

Figure 16. Impact of layer swapping on style blending.

6.3. Project Journey

Following the conceptual experiments, the project transitioned to implementing a StyleGAN2 network fine-tuned on the CryptoPunks dataset. This phase faced numerous challenges, including difficulty finding compatible repositories and computational constraints.

Repositories utilized included:

- *Cartoon-StyleGAN: Fine-Tuning StyleGAN2 for Cartoon Face Generation* by **happy-jihye**,

Figure 17. Estimated training times for StyleGAN2 on limited hardware.

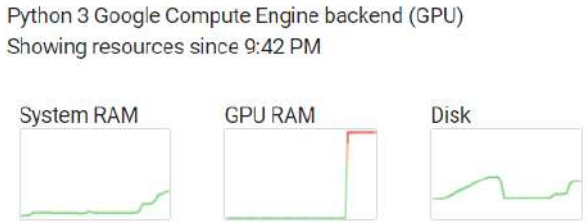

Figure 18. Resource limitations during fine-tuning with StyleGAN2.

Repository name	*StyleGAN2 for practice* by **eps69**	*Cartoon-StyleGAN: Fine-tuning StyleGAN2 for Cartoon Face Generation* by **happy-jihye**	*StyleGAN - Official Tensorflow Implementation* by **NVLabs**
Backend type	Tensorflow	Pytorch	Tensorflow
Encountered difficulties	required training time too long (from scratch ~**14** days) insufficient GPU	insufficient GPU to launch training	initial difficulties with creating the TFRecords
Successful implementation?	no	no	yes

Figure 19. Summary of repositories explored during the implementation phase.

- *StyleGAN2 for Practice* by **eps696**,
- *StyleGAN - Official TensorFlow Implementation* by **NVLabs**.

Initial attempts to train a StyleGAN2 network from scratch proved infeasible due to excessive training times, as illustrated in Figure 17. Training a model of this complexity requires significantly greater computational resources than were available.

Fine-tuning a pre-trained FFHQ model was pursued as an alternative, but this too encountered GPU resource limitations, resulting in errors such as *CUBLAS_STATUS_ALLOC_FAILED* and *CUDNN_STATUS_EXECUTION_FAILED* (Figure 18).

Ultimately, the project reverted to an earlier version of the architecture (StyleGAN) using NVLabs' official TensorFlow implementation. Table 19 summarizes the repositories explored during the project.

6.4. Training Parameters

The StyleGAN model was fine-tuned using the following parameters:

Figure 20. Resized CryptoPunk image (1024x1024).

- **Iterations:** 3500 kimg (thousands of images shown to the discriminator),
- **Optimizer:** Adam, with $\beta_1 = 0.0$, $\beta_2 = 0.99$, $\epsilon = 10^{-8}$,
- **Loss Function:** Non-saturating GAN loss for the generator and logistic simplegp loss for the discriminator,
- **Learning Rate:** Custom decaying schedule (0.001–0.003),
- **Batch Size:** Decayed from 128 to 32 over training iterations,
- **Evaluation Metric:** Frechet Inception Distance (FID) for image quality assessment.

7. Results

7.1. Low-Resolution Dataset Challenges

The CryptoPunks dataset, comprising 24x24 pixel images, posed challenges for training due to its low resolution. To match the input size of the pre-trained model, these images were resized to 1024x1024 pixels, which introduced significant blur (Figure 20). This affected the quality and sharpness of the generated outputs.

7.2. Fine-Tuning Results

The fine-tuning process was carried out using the CryptoPunks dataset and a pre-trained FFHQ model. The training ran for approximately 12 hours on a single NVIDIA Tesla T4 GPU. While additional trials were considered to extend the training time, computational constraints and technical challenges limited the total training duration to 12 hours. Figures 21 to 24 illustrate the progression of the model during training, showcasing the generator's outputs at various stages.

Figure 21 depicts the outputs of the baseline pre-trained FFHQ model before fine-tuning, which generated realistic human faces with no resemblance to the CryptoPunk style. After 280 kimg (thousands of images seen by the discriminator), the generator begins to learn basic features of the CryptoPunk dataset, as seen in Figure 22. At this stage, the outputs show early signs of pixelated patterns and deviations from photorealism.

Figure 21. Baseline: Original FFHQ model outputs (before fine-tuning).

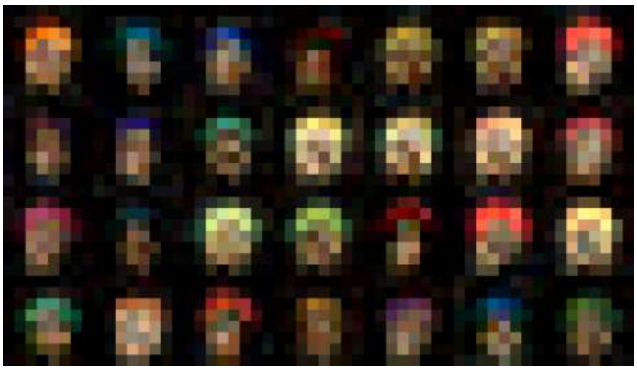

Figure 22. Early training phase (kimg=280).

As training progresses to the mid-phase (1864 kimg), the generator captures increasingly distinct features of the CryptoPunk dataset, such as simplified geometry and vivid color palettes, as shown in Figure 23. While the outputs remain blurry and less defined than the original dataset, they begin to exhibit stylistic attributes characteristic of CryptoPunks.

By the end of training (3500 kimg), the generator outputs images that closely approximate the pixelated aesthetic of CryptoPunks, as illustrated in Figure 24. While the outputs are still not as sharp as those in the original dataset (Figure 25), they demonstrate the generator's ability to learn and reproduce the essential characteristics of the target style.

Overall, the fine-tuning process successfully enabled the model to learn a representation of the CryptoPunk dataset, achieving a stylistic transformation from the original FFHQ model. Extending the training time would likely have improved the sharpness and fidelity of the generated outputs.

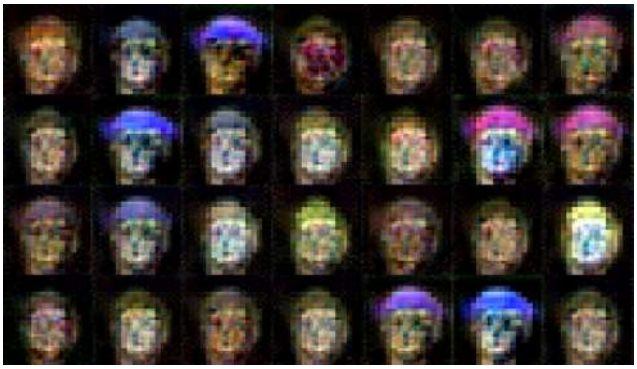

Figure 23. Mid-training phase (kimg=1864).

Figure 24. Final training phase (kimg=3500).

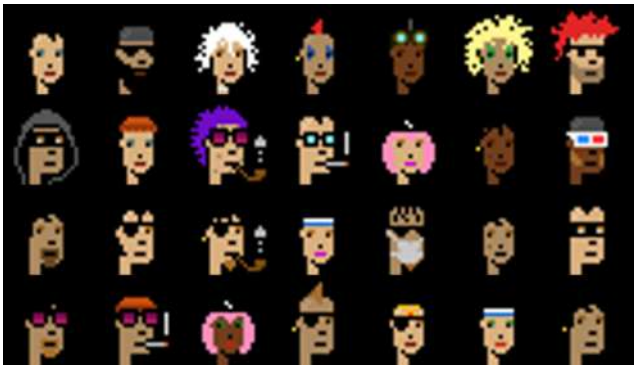

Figure 25. Original CryptoPunk training images.

Additionally, the successful adaptation of the pre-trained FFHQ network indicates that the initial layers of the model remain compatible, which theoretically supports the application of layer swapping for style mixing.

While the final results do not reach the level of sharpness achieved in previous works such as [2], it is important to note that our implementation benefits from the advanced representational capabilities of StyleGAN, which enables greater parameterization and control over style features compared to older architectures like DGGAN.

8. Conclusion

Although the original goal of achieving style blending through layer swapping between the CryptoPunk fine-tuned model and the FFHQ pre-trained model was not realized, the project provided valuable insights into the workings of StyleGAN and its advanced capabilities. Experiments with layer swapping between FFHQ and Naver Webtoon StyleGAN models demonstrated the mechanisms of style mixing, laying the groundwork for future extensions.

Despite limitations in computational resources and time, the project successfully fine-tuned a StyleGAN model on the CryptoPunks dataset. This achievement underscores the feasibility of adapting state-of-the-art GAN architectures to novel artistic domains, even with constrained resources.

Future work could build upon these findings by extending training durations, optimizing hyperparameters, and performing ablation studies to analyze the influence of specific network components. Furthermore, more advanced GPU resources could facilitate the exploration of real-world applications, including personalized style blending for creative industries and NFT marketplaces. With these enhancements, a more comprehensive and impactful study of CryptoPunk-style blending could be achieved.

References

[1] Jihye Back. Fine-tuning stylegan2 for cartoon face generation, 2021. 3

[2] Bao Tram Duong. Generate nft cryptopunks with dggan, 2021. 3, 8

[3] Amy Goodchild. What is generative art?, 2022. 2

[4] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4401–4410, 2019. 3, 4

[5] Poornima Nataraj. Online tools to create mind-blowing ai art, 2022. 2

[6] Justin Pinkney. Stylegan network blending, 2020. 3

[7] Tarang Shah. Blending stylegan2 models to turn faces into cats, 2021. 3