A Style-Based Generator Architecture for Generative Adversarial Network

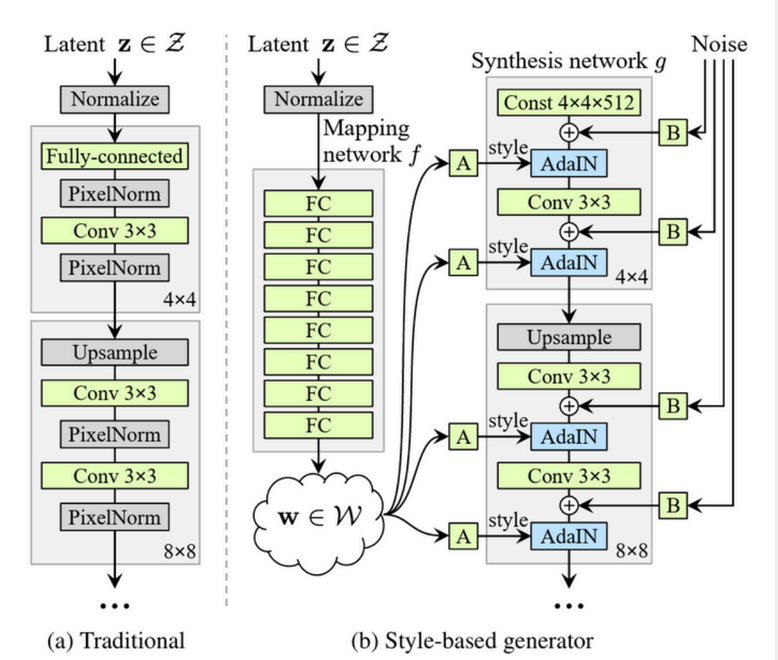
ABSTRACT

StyleGAN is a novel architecture proposed by NVIDIA that significantly improves the quality and control of generative models for image synthesis. By introducing a style-based generator and key components such as adaptive instance normalization (AdaIN), StyleGAN allows fine-grained control over image features at various scales. In this project, we studied the StyleGAN architecture and implemented core components using publicly available resources. We generated synthetic images using pre-trained models and explored modifications to latent vectors to observe style mixing and variation.

INTRODUCTION

Generative Adversarial Networks (GANs) have revolutionized the field of synthetic image generation. However, traditional GANs often struggle with control over image features and stability during training. StyleGAN, introduced by Karras et al. (2019), overcomes these limitations by separating high-level attributes from stochastic variation and providing intuitive control over image styles.

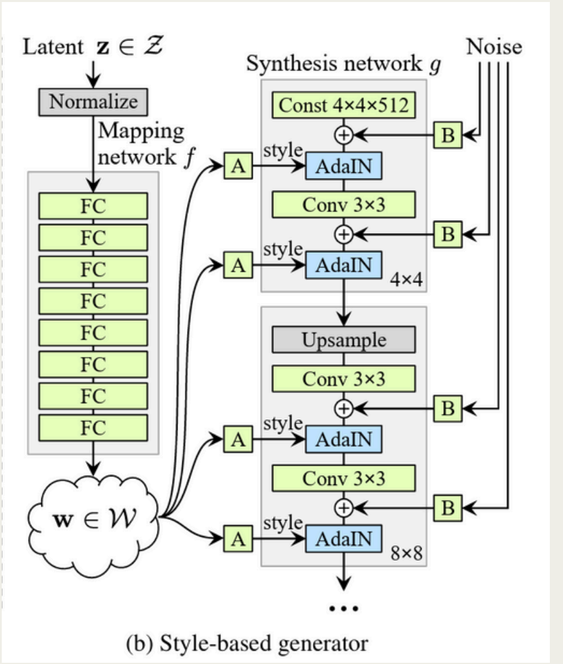
StyleGAN has had a significant impact in the field of image synthesis, particularly in applications like face generation, art, and deepfake technology. This project aims to study the architecture of StyleGAN and implement parts of the model to generate high-quality, realistic images.



STYLE-BASED GENERATOR ARCHITECTURE

The authors propose a new architecture that separates high-level attributes from stochastic variations, leveraging principles from style transfer literature.

- Architecture Design:
- Separates high-level attributes from stochastic variations.
  - AdaIN operations control semantic attributes.
  - Explicit noise inputs introduce stochastic variations.
  - Inspired by style transfer techniques.
- Intermediate Latent Space ($W \neq Z$):
  - A learned space W\mathcal{W}W, mapped from Z\mathcal{Z}Z, achieves better disentanglement.
- Disentanglement Metrics:
  - Perceptual Path Length: Measures interpolation smoothness.
  - Linear Separability: Assesses attribute separation via linear classifiers.
  - No encoder required.
- FFHQ Dataset:
  - High-quality human face dataset with wide diversity (age, ethnicity, lighting, etc.).



(b) Style-based generator

STYLEGAN: CORE ARCHITECTURAL COMPONENTS

1. Mapping Network:

- Applies a non-linear transformation $f: Z \nrightarrow W$ using an 8-layer MLP.
- Ensures that intermediate latent space W is disentangled from input space Z.
- Mathematical Form: $w=f(z), z \in Z, w \in W$

2. Constant Input:

- Synthesis begins from a learned constant tensor of shape $4 \times 4 \times 512$.
- All variation is driven by style (AdaIN) and noise inputs.

STYLEGAN: CORE ARCHITECTURAL COMPONENTS

3. AdaIN Operations:

- Applies Adaptive Instance Normalization after each convolution:

$$\text{AdaIN}(x_i, y) = y_{s,i} \cdot \frac{x_i - \mu(x_i)}{\sigma(x_i)} + y_{b,i}$$

- xi: feature map, μ(xi),σ(xi)\mu(x_i) : its mean & std.
- ys,i,yb,i: scale and bias from style vector **w**.

4. Noise Inputs:

- Injects per-pixel Gaussian noise n N(0,1) after each convolution after each convolution.
- Scaled by learnable weights to produce stochastic variation.

STYLE MIXING AND REGULARIZATION IN STYLEGAN

- Purpose: Prevent adjacent layers from assuming fixed correlations by randomly switching styles during training.
- Method: Two latent codes (z1, z2) ↠ mapped to styles (w1, w2) ↠ applied at different layers of the generator.
- Effect: Localizes style control, improves disentanglement, and enhances variation.

```
class GenBlock(nn.Module):
    def forward(self, x, w):
        x = self.adain1(self.leaky(self.inject_noise1(self.conv1(x))), w)
        x = self.adain2(self.leaky(self.inject_noise2(self.conv2(x))), w)
        return x
```

- adain: Applies style via Adaptive Instance Normalization
- inject_noise: Adds per-pixel noise for fine detail
- w: Style vector used across the block
- Note: While not explicitly implemented, progressive growing already encourages style separation. Future versions may integrate explicit style mixing

PROPERTIES OF THE STYLE-BASED GENERATOR (STYLEGAN)

1. Separation of Global Effects from Stochasticity

- Key Insight: StyleGAN separates global features (e.g., clothing type, color) from fine-grained stochastic details (e.g., fabric texture).
- How?
  - Latent vector w (from mapping network) controls global structure.
  - Per-pixel noise injection adds randomness to fine textures.
  - AdaIN (Adaptive Instance Normalization) applies w to control style across generator layers.
- Implementation Note:
  Global features are driven by w, while noise input introduces localized texture variation.

PROPERTIES OF THE STYLE-BASED GENERATOR (STYLEGAN)

2. Truncation Trick in W Space

- Goal: Improve image quality at the cost of sample diversity.
- Mechanism:
  - Compute average latent vector: $\bar{w}$ (mean of all w samples).
  - During inference:
  - $w' = \bar{w} + \psi(w - \bar{w})$
  - where $\psi < 1$ pulls w closer to the average — reducing variance.
- Result: Generates more realistic images with fewer artifacts.

Implementation Note:

Although not included yet, truncation can be added by computing $\bar{w}$ and scaling latent vectors during inference.

OUR EXPERIMENTAL RESULTS

Dataset Used

- FFHQ (Flickr-Faces-HQ): 70,000 1024 x 1024 diverse face images are used in original paper
- **Our Dataset**: Women's Clothing Images from Kaggle.
- Images were resized progressively for training.
- This dataset supports resolutions up to 128 x 128, sufficient for clothing details.

Image Quality

- StyleGAN achieves a lower FID (4.40 vs. 5.25 for Progressive GAN) on FFHQ .
- In our model we use a WGAN-GP loss with gradient penalty
- Trained progressively from 4x4 to 128x128, our model is able to generate realistic and convincing clothing images

OUR EXPERIMENTAL RESULTS

Disentanglement

- Perceptual Path Length: Measures how smooth transitions are in the latent space.
- Linear Separability: Evaluates how well features like color and pattern can be separated.
- Through our experimentation ,the mapping network and progressive training likely enhance W space linearity, as seen in StyleGAN. The separation of style and noise suggests potential disentanglement of clothing attributes

Figure 1: Generated Clothing Samples

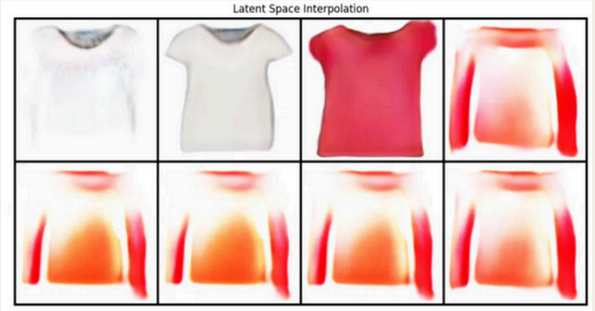


Figure 2: Latent Space Interpolation

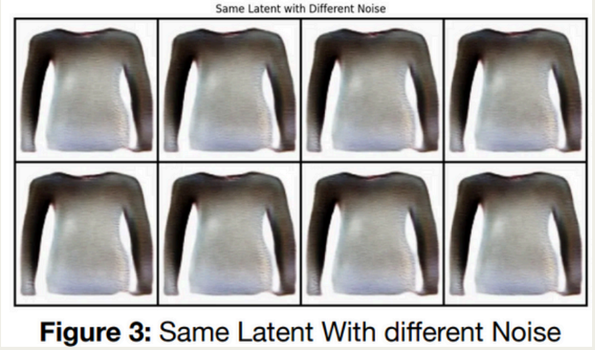


Figure 3: Same Latent With different Noise

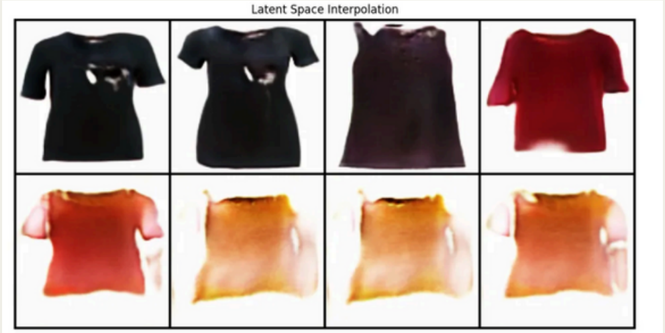


Figure 4: Latent Space Interpolation

OUR EXPERIMENTAL RESULTS

Our StyleGAN model

It has been trained for 200 epochs and sucessfully generates realistic and convincing clothing items capturing both the general shape and finer details like patterns , necklines and fabric texture .

Smooth interpolation :

Interpolations is about retaining structure and not just blending images. This is clearly seen in our model seen in the transition between clothing styles (e.g., white blouse ↠ red t-shirt in fFig 2) is smooth. It is a direct result of the mapping network that transforms the initial latent code z into the intermediate representation w.

OUR EXPERIMENTAL RESULTS

Controlling Small Variations :

By fixing the latent code and changing only noise we get multiple versions of the same clothing with tiny variations (grey long sleeved shit Fig 3). The core high level features remain the same highlighting that noise injection on the pixel level does not hinder the models ability to separate identity from stochastic features.

Style Mixing & Feature Control :

Our experiments show that the generator network controls features at different scales. Early layers define the overall shape and type of clothing, while middle layers handle design elements. The later layers add fine details like fabric texture, allowing precise editing and better control over the generated images. This hierarchical control makes it easier to change specific parts of the image and allows advanced editing that older GAN models couldn’t do.

CONCLUSION

StyleGAN advances GANs by controlling synthesis through styles, improving quality and interpretability
Our adapted model is successfully trained for women’s clothing. Generated images are high-quality and controllable with progressive growing and style-based synthesis.

Future work:
Add explicit disentanglement metrics.
Train at even higher resolutions.

Thank you!