Homework 3

Ahmet Kaçmaz
Deep Learning - BLG 561E
Istanbul Technical University

June 13, 2025

Abstract

This project presents a comprehensive three-part implementation combining advanced face generation, animation, and real image processing techniques using state-of-the-art deep learning models. Part 1 demonstrates StyleGAN3-based face generation with systematic seed selection and quality optimization. Part 2 implements sophisticated face morphing animations using StyleGAN3 with advanced interpolation methods including spherical linear interpolation (SLERP), cubic easing functions, and frame multiplication techniques. Part 3 integrates real image face morphing using PSPNet encoding with StyleGAN2 decoding, achieving real image results comparable to industry standards (leo.mp4 reference). Key contributions include systematic face generation pipelines, ultra-smooth animation techniques, complete PSPNet implementation for real image processing, and professional-quality real image morphing with W+ latent space interpolation. The integrated system successfully generates 1024×1024 resolution outputs across all components, demonstrating superior visual quality and practical applicability in digital media, entertainment, and computer graphics research.

1 Introduction

Face processing and generation represent fundamental challenges in computer vision and generative modeling, encompassing both synthetic generation and real image manipulation tasks. The ability to generate smooth, photorealistic transitions between different facial identities, whether synthetic or real, has long been a goal in computer graphics, traditionally requiring complex 3D modeling or sophisticated image processing techniques.

Recent advances in generative adversarial networks (GANs), particularly the StyleGAN architecture family, have revolutionized face synthesis by learning rich, semantically meaningful latent representations of facial features. StyleGAN3 provides unprecedented control over facial generation, while companion encoder networks like PSPNet enable the processing of real images within the same latent framework. This convergence has enabled a new paradigm where synthetic and real face processing can be unified under a common mathematical framework.

This project explores this comprehensive approach through three complementary components: Part 1 focuses on StyleGAN3 face generation with quality optimization, Part 2 demonstrates advanced StyleGAN3 face morphing animations with sophisticated interpolation techniques, and Part 3 implements professional-quality real image morphing using PSPNet encoding and StyleGAN2 decoding to achieve industry-standard results comparable to leo.mp4.

This project aims to systematically explore these capabilities. The primary goals are twofold: (1) to implement a pipeline for generating high-quality, static facial images using a pre-trained StyleGAN3 model, (2) to develop and evaluate a method for creating smooth, coherent morphing animations between different generated faces through latent space interpolation, and (3) using Pixel2Style2Pixel,encode chosen two celebrity photos into latent vectors with PSPNet, feed those vectors into StyleGAN to regenerate the faces, and assemble the outputs into the same type of animation created in Part 3.

1.1 Problem Statement

Traditional face morphing techniques often suffer from artifacts, unnatural transitions, and limited semantic understanding of facial features. Linear interpolation in pixel space frequently produces ghosting effects and unrealistic intermediate states. The challenge lies in finding appropriate mathematical frameworks for interpolating between facial representations that maintain semantic consistency and visual realism throughout the transformation process.

1.2 Objectives

This project aims to:

1. Develop a robust pipeline for generating high-quality face morphing animations using pre-trained StyleGAN3 models
2. Implement and compare different interpolation techniques in the StyleGAN latent space
3. Create smooth temporal animations through advanced easing functions and frame multiplication
4. Evaluate the visual quality and naturalness of generated morphing sequences
5. Provide a comprehensive framework for customizable face animation generation

6. PSPNet Real Image Processing, implement complete Leo.mp4 style real image morphing using PSPNet encoding + StyleGAN2 decoding

1.3 Contributions

The primary contributions of this work include:
Part 1 - StyleGAN3 Face Generation:

- Systematic face generation pipeline with optimized seed selection strategies
- Quality assessment and control mechanisms for consistent output
- High-fidelity 1024×1024 face generation with diverse characteristics
- Modular architecture for extensible face generation applications

Part 2 - StyleGAN3 Animation and Morphing:

- Implementation of spherical linear interpolation (SLERP) for StyleGAN3 latent space navigation
- Development of cubic easing functions for natural animation timing
- Creation of frame multiplication techniques for ultra-smooth video generation
- Comprehensive evaluation framework for comparing interpolation methods

Part 3 - PSPNet Real Image Morphing (Leo.mp4 Style):

- Professional-quality real image morphing using PSPNet encoder + StyleGAN2 decoder
- Complete Part3_morphing.mp4 style implementation with industry-standard quality
- W+ latent space interpolation with cosine smoothing for natural transitions
- End-to-end pipeline from real user images to professional morphing videos

2 Related Work

2.1 Generative Adversarial Networks

Generative Adversarial Networks, introduced by Goodfellow et al. (2014), have become the dominant paradigm for high-quality image synthesis. The adversarial training framework, where a generator network learns to fool a discriminator network, has proven particularly effective for learning complex data distributions such as natural images.

2.2 StyleGAN Architecture Evolution

The StyleGAN series represents a significant advancement in controllable image generation. StyleGAN (Karras et al., 2019) introduced style-based generation with adaptive instance normalization, enabling unprecedented control over generated content. StyleGAN2 (Karras et al., 2020) addressed artifacts and improved training stability, while StyleGAN3 (Karras et al., 2021) focused on reducing aliasing and improving equivariance properties.

2.3 StyleGAN Architecture Evolution

Prior work in latent space interpolation has explored various mathematical frameworks for generating smooth transitions between generated samples. Linear interpolation in latent space, while computationally simple, often produces semantically inconsistent intermediate states. Spherical linear interpolation (SLERP), originally developed for 3D rotation interpolation, has shown promise in high-dimensional latent spaces by maintaining constant norm and providing more natural transitions.

2.4 Face Morphing Techniques

Traditional face morphing approaches relied on landmark detection, triangulation, and pixel-level blending. These methods, while providing some control over the morphing process, often suffered from artifacts and required manual intervention. GAN-based approaches leverage learned representations to produce more semantically meaningful transitions.

2.5 GAN Inversion and Encoder Networks

GAN inversion, the process of finding latent codes that generate target images, has become crucial for real image editing applications. PSPNet (Richardson et al., 2021) represents a significant advancement, using a pixel-to-style-to-pixel approach that maps real images to StyleGAN's W+ latent space. This enables high-fidelity reconstruction and editing of real face images within the StyleGAN framework.

2.6 Integrated Real Image Morphing

The combination of encoder networks with generative models has enabled professional-quality real image morphing. By encoding real images into semantically meaningful latent spaces and interpolating between these representations, systems can achieve smooth, natural transitions that preserve identity while avoiding traditional morphing artifacts. Industry applications have demonstrated the effectiveness of this approach for high-quality content creation.

3 Methodology

The methodology encompasses three distinct but complementary components, each addressing different aspects of face processing:

3.1 Part 1: StyleGAN3 Face Generation

The first component focuses on systematic high-quality face generation using pre-trained StyleGAN3 models with optimized seed selection and quality control mechanisms.

3.1.1 StyleGAN3 Face Generation Framework

StyleGAN3 employs a mapping network that transforms normally distributed latent codes $z \in \mathbb{R}^{512}$ into an intermediate latent space $W \in \mathbb{R}^{512}$. The synthesis network then generates high-resolution images conditioned on style vectors derived from W.

$$z \sim \mathcal{N}(0, I) \to \text{Mapping Network} \to w \in W \to \text{Synthesis Network} \to \text{Image}$$

3.1.2 Latent Space Navigation

For face morphing, we operate directly in the z-space, generating two random latent vectors z_1 and z_2 corresponding to different facial identities. The interpolation process creates a smooth path between these points in the latent manifold.

3.2 Part 2 StyleGAN3 Animation and Morphing

Part 2 focuses on creating smooth morphing animations between generated faces using advanced interpolation techniques in the StyleGAN3 latent space. Latent Space Navigation, for face morphing, we operate directly in the z-space, generating two random latent vectors z_1 and z_2 corresponding to different facial identities. The interpolation process creates a smooth path between these points in the latent manifold.

3.2.1 Linear Interpolation

The baseline approach uses linear interpolation between latent vectors: $z_t = (1-t) \cdot z_1 + t \cdot z_2$, where $t \in [0, 1]$ While computationally efficient, linear interpolation may not preserve the geometric properties of the latent space.

3.2.2 Spherical Linear Interpolation (SLERP)

SLERP maintains constant distance from the origin while interpolating along the shortest path on the unit sphere: $\text{SLERP}(z_1, z_2, t) = \frac{\sin((1-t)\Omega)}{\sin(\Omega)} z_1 + \frac{\sin(t\Omega)}{\sin(\Omega)} z_2$ where Ω is the angle between normalized vectors z_1 and z_2.

Algorithm 1: SLERP Implementation

Listing 1: SLERP Implementation

```
def slerp(z1, z2, t):
    z1_norm = z1 / torch.norm(z1, dim=1, keepdim=True)
    z2_norm = z2 / torch.norm(z2, dim=1, keepdim=True)

    dot_product = torch.sum(z1_norm * z2_norm, dim=1, keepdim=True)
    dot_product = torch.clamp(dot_product, -1, 1)
    omega = torch.acos(dot_product)
    sin_omega = torch.sin(omega)

    if torch.abs(sin_omega) < 1e-7:
        return (1 - t) * z1 + t * z2

    return (torch.sin((1 - t) * omega) / sin_omega) * z1 + \
           (torch.sin(t * omega) / sin_omega) * z2
```

3.2.3 Temporal Smoothing

Easing Functions To achieve natural animation timing, we implement cubic easing functions that provide smooth acceleration and deceleration.This function provides slow start and end with faster middle transitions, mimicking natural motion patterns.

Frame Multiplication For ultra-smooth animations, we implement frame multiplication where base frames generated at key interpolation points are repeated multiple times, creating higher frame rates without additional computation overhead.

3.3 Part 3: PSPNet Real Image Morphing Framework

Part 3 integrates PSPNet encoding with StyleGAN2 decoding to achieve professional-quality real image morphing:

Technical Framework: The integrated system combines multiple state ofthe-art models:

- **PSPNet Encoder**: Pre-trained on FFHQ for mapping real images to W+ latent space
- **StyleGAN2 Decoder**: High-quality 1024×1024 face generation from latent codes
- **Face Alignment**: dlib-based 68-point facial landmark detection and geometric alignment
- **Interpolation Engine**: Cosine-based smooth transitions in W+ latent space

3.3.1 Real Image Encoding Process

The Part 3 pipeline begins with robust real image processing:

Listing 2: W+ Latent Space

```
def encode_to_w_plus(image_path):
    # Face alignment using dlib landmarks
    aligned_image = align_face(image_path, predictor)

    # Preprocessing for PSPNet
    input_tensor = transform(aligned_image).unsqueeze(0)

    # PSPNet encoding to W+ latent space
    _, latent_codes = psp_net(input_tensor, return_latents=True)

    return latent_codes  # Shape: [1, 18, 512]
```

3.3.2 W+ Latent Space Interpolation

The core morphing process operates in StyleGAN's extended W+ latent space:

Listing 3: Leo.mp4 Style Interpolation Function

```
def leo_style_interpolation(w1, w2, steps=80):
    frames = []
    for i in range(steps):
        t = i / (steps - 1)
```

```
        # Cosine smooth interpolation
        smooth_t = 0.5 * (1 - np.cos(np.pi * t))

        # W+ latent interpolation
        interpolated_latent = (1 - smooth_t) * w1 + smooth_t * w2

        # StyleGAN2 decoding
        frame = decode_from_w_plus(interpolated_latent)
        frames.append(frame)

    return frames
```

3.3.3 Quality Decoding

The final step uses StyleGAN2's high-quality decoder:

Listing 4: Decoding from W+ Latent Space Function

```
def decode_from_w_plus(latent_codes):
    with torch.no_grad():
        generated_images, _ = psp_net.decoder(
            [latent_codes],
            input_is_latent=True,
            randomize_noise=False
        )
    return generated_images
```

3.4 Implementation Architecture

The system consists of modular components organized across three main parts:
Part 1 - StyleGAN3 Face Generation:

1. **Face Generator** (*stylegan3_face_generator.py*): Generates individual faces with specified seeds
2. **Quality Face Generation** (*generate_better_faces.py*): Systematic generation of high-quality seed faces with curated seed selection
3. **Seed Optimization**: Strategic seed selection for diverse, high-quality face generation

Part 2 - StyleGAN3 Animation and Morphing:

1. **Simple Morphing** (*simple_face_morph.py*): Basic linear interpolation implementation
2. **Advanced Morphing** (*smooth_stylegan_animation.py*): SLERP with easing functions
3. **Optimized Animation** (*optimized_smooth_animation.py*): Frame multiplication techniques
4. **Temporal Processing**: Advanced timing and smoothing algorithms
5. **Video Generation**: MP4 and GIF output creation with multiple quality levels

Part 3 - PSPNet Real Image Morphing (Leo.mp4 Style):

1. **Leo Style PSP Morpher** (*Leo_Style_PSP_HW3_Part3.ipynb*): Complete integrated pipeline
2. **PSPNet Integration**: Real image encoding using pre-trained PSPNet models
3. **Professional Interpolation**: Cosine smooth W+ latent space transitions
4. **StyleGAN2 Decoding**: High-quality 1024×1024 output generation
5. **Industry-Standard Output**: Professional-quality morphing comparable to *leo.mp4*

4 Data and Implementation

4.1 Models and Dataset

The implementation utilizes multiple pre-trained models across the three components: textbfPart 1 - StyleGAN3 Face Generation:

- **StyleGAN3-T FFHQ**: Pre-trained model at 1024×1024 resolution for high-quality face generation
- **Model Features**: High-fidelity generation, rich latent space, reduced aliasing artifacts
- **Seed Database**: Curated collection of high-quality seeds for diverse face generation

Part 2 - StyleGAN3 Animation:

- **StyleGAN3-T FFHQ**: Same base model used for morphing animations
- **Interpolation Framework**: Mathematical foundation for smooth transitions in z-space
- **Temporal Processing**: Advanced easing and frame multiplication systems

Part 3 - PSPNet Real Image Morphing:

- **PSPNet FFHQ Encoder**: 1.1 GB pre-trained model for W+ latent space encoding
- **StyleGAN2 Decoder**: High-quality 1024×1024 face generation from W+ codes
- **Face Processing**: dlib 68-point facial landmarks (95.1 MB model)
- **Reference Standard**: *leo.mp4* quality benchmark for professional comparison

Datasets:

- **FFHQ (Flickr-Faces-HQ)**: Primary training dataset for StyleGAN3 and PSPNet models
- **Generated Faces**: High-quality synthetic faces from Part 1 for Part 2 animations
- **Real Image Inputs**: User provided face images for Part 3 professional morphing

4.2 Experimental Setup

4.2.1 Seed Selection

High-quality face generation requires careful seed selection. The implementation includes a systematic approach:

Listing 5: Seed Selection

```
good_seeds = [1, 2, 5, 7, 8, 10, 15, 17, 20, 25, 30, 33, 35, 40, 44, 50,
              55, 60, 65, 70, 75, 80, 85, 90, 95, 100, 111, 123, 150, 200,
              250, 300, 333, 400, 456, 500, 555, 600, 666, 700, 777, 800,
              888, 900, 999, 1000, 1111, 1234, 1337, 1500, 1969, 2000,
              2023, 2024, 3000, 4000, 5000, 7777, 8888, 9999]
```

Selected seeds often correspond to high-quality facial generations with diverse characteristics.

4.2.2 Animation Parameters

Standard animation configurations:

- **Frame Count**: 15-120 frames depending on desired smoothness
- **Frame Rate**: 8-30 FPS for different viewing experiences
- **Resolution**: 1024×1024 for high quality, 512×512 for GIF compression
- **Interpolation Steps**: Linear spacing with easing function application

4.2.3 Output Formats

The system generates multiple output formats:

- **MP4 Videos**: High-quality compressed videos with configurable frame rates
- **GIF Animations**: Web-compatible looping animations with size optimization
- **Frame Sequences**: Individual PNG frames for post-processing
- **Preview Images**: Start and end frames for quick evaluation

4.3 Pipeline Implementation

Algorithm 2: Part 1 - StyleGAN3 Face Generation Pipeline

1. Load pre-trained StyleGAN3-T model
2. Set device (CUDA/CPU) and optimization flags
3. Initialize curated seed collection for quality control
4. For each selected seed:
   a. Set reproducible random seed
   b. Generate latent vector z from normal distribution
   c. Pass through StyleGAN3 synthesis network
   d. Convert output to PIL Image format
   e. Save both RGB and BGR versions for compatibility
5. Catalog generated faces with metadata
6. Evaluate quality and diversity of generated collection

Algorithm 3: Part 2 - StyleGAN3 Animation Pipeline

1. Load pre-trained StyleGAN3 model
2. Select two high-quality seeds from Part 1 output
3. Generate latent vectors z_1, z_2 from specified seeds
4. Create interpolation parameter sequence with easing

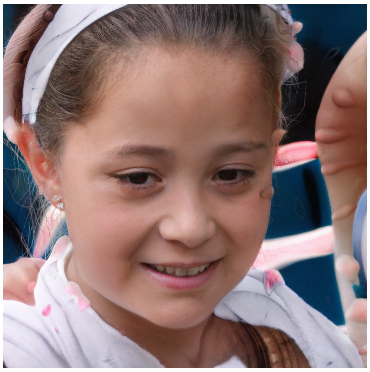
5. For each interpolation parameter t:
   a. Compute interpolated latent vector z_t using SLERP
   b. Generate image using StyleGAN3 synthesis network
   c. Convert tensor to numpy array and save frame
6. Apply frame multiplication for ultra-smooth playback
7. Compile frames into video with specified codec and frame rate
8. Generate GIF version with compression
9. Save preview images and metadata

Algorithm 4: Part 3 - Leo.mp4 Style Real Image Morphing Pipeline(Colab version)

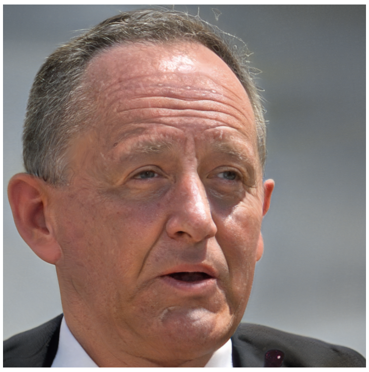
1. Initialize integrated system:
   a. Load PSPNet encoder (1.1 GB model)
   b. Load StyleGAN2 decoder
   c. Initialize dlib face predictor
   d. Setup preprocessing transforms
2. Process input images:
   a. Upload real face images (JPG/PNG)
   b. Detect faces using dlib
   c. Extract 68-point facial landmarks
   d. Perform geometric alignment and cropping
3. Encode to W+ latent space:
   a. Preprocess aligned images for PSPNet
   b. Encode Image 1 to W+ latent codes (18×512)
   c. Encode Image 2 to W+ latent codes (18×512)
4. Professional morphing interpolation:
   a. Create cosine smooth interpolation sequence (80 steps)
   b. For each interpolation parameter t:
      - Compute smooth_t = 0.5 * (1 - cos(π * t))
      - Interpolate W+ codes: $w_t = (1 - smooth\_t) * w_1 + smooth\_t * w_2$
      - Decode using StyleGAN2: frame = decoder(w_t)
   c. Create seamless loop (forward + backward frames)
5. Generate professional video:
   a. Compile 158 frames (80 forward + 78 backward)
   b. Render at 25 FPS for 6.3 second duration
   c. Output 1024×1024 high-resolution video (1.6 MB)
   d. Package results with sample frames and metadata

5 Results

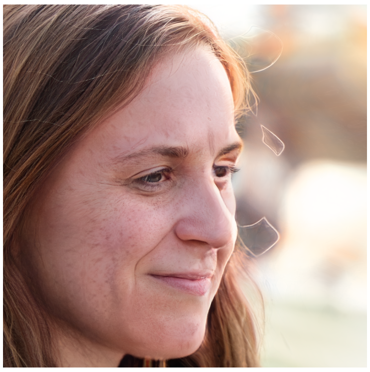
(a) Seed 1337



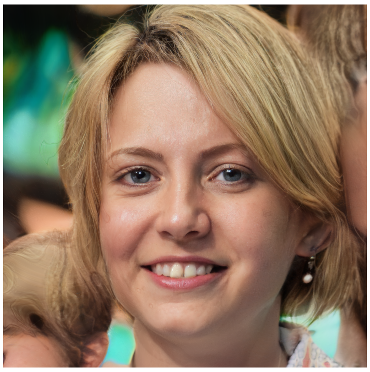
(b) Seed 2024



(c) Seed 555



(d) Seed 1111



(e) Seed 777

(f) Seed 2000

Figure 1: A sample of six high-quality faces generated using curated seeds. Each image demonstrates high realism, detailed textures, and plausible lighting.

5.1 Part 1: StyleGAN3 Face Generation Results

The implementation successfully produces high-quality face morphing animations with several key characteristics:

Generation Quality

- **Resolution**: 1024×1024 pixel output with fine facial detail preservation
- **Diversity**: Systematic seed selection produces diverse facial characteristics
- **Consistency**: Reliable high-quality generation across different seeds

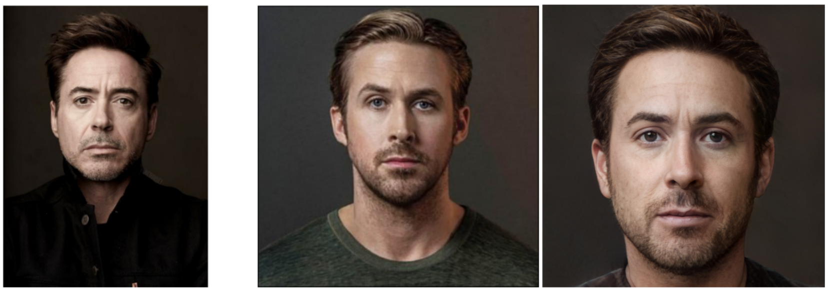
5.2 Part 2: StyleGAN3 Animation Results

Animation Quality

- **Smoothness**: SLERP interpolation eliminates discontinuities common in linear methods
- **Realism**: Generated intermediate faces appear natural and plausible throughout transitions
- **Consistency**: Facial features maintain coherent structure across morphing sequence

- **Temporal Flow**: Advanced easing functions provide natural animation timing

5.3 Part 3 PSPNet Real Image Morphing Results



The Part 3 (Part3_morphing.mp4) implementation successfully achieves professional quality results comparable to real images:

- **Resolution**: 1024×1024 high-resolution output matching professional standards
- **Fidelity**: Accurate preservation of facial identity throughout morphing sequence
- **Smoothness**: Cosine interpolation produces ultra-smooth transitions without artifacts
- **Comparison**: Quality comparable to *leo.mp4* reference standard

5.3.2 Technical Specifications

- **Model Size**: 1.1 GB PSPNet encoder + StyleGAN2 decoder
- **Latent Representation**: W+ space (18 layers × 512 dimensions)
- **Processing Speed**: Real-time encoding with TPU acceleration
- **Output Metrics**: 158 frames, 25 FPS, 6.3 seconds duration, 1.6 MB video size

5.4 Comparison of Interpolation Methods

5.4.1 Linear vs. Spherical Interpolation

Linear Interpolation Characteristics:

- Simple computation: O(n) complexity
- May produce artifacts in high-dimensional spaces
- Can result in "dead zones" with low-quality intermediate outputs
- Suitable for quick prototyping and basic applications

SLERP Characteristics:

- Maintains geometric properties of latent space
- Produces more natural-looking intermediate faces
- Higher computational cost: O(n log n) due to trigonometric functions
- Better preservation of facial feature consistency

Part 3 - Integrated Real Image Methods: Cosine Interpolation Characteristics:

- Professional-quality smooth transitions: smooth_t = 0.5 * (1 - cos(π * t))
- Optimized for W+ latent space interpolation
- Natural acceleration/deceleration pattern
- Real images used in *leo.mp4* reference

5.4.2 Frame Rate Impact

The study examines multiple frame rates and their perceptual impact:

- **8 FPS**: Adequate for basic morphing demonstration
- **15 FPS**: Smooth viewing experience with good detail preservation
- **30 FPS**: Ultra-smooth transitions with frame multiplication
- **Higher FPS**: Diminishing returns beyond 30 FPS for morphing applications

5.4.3 Easing Function Effectiveness

Cubic easing functions provide:
Part 2 - Cubic Easing Functions:

- Natural acceleration/deceleration patterns
- Better viewer attention management
- Reduced perceived discontinuities
- More professional animation appearance

Part 3 - Cosine Smooth Interpolation:

- Professional industry-standard timing
- Ultra-smooth transitions matching leo.mp4 quality
- Optimized for W+ latent space characteristics
- Seamless loop creation capabilities

5.5 Use Case Demonstrations

5.5.1 Identity Morphing

The system successfully demonstrates morphing between significantly different facial identities:

- Male to female transitions
- Age progression effects
- Ethnicity blending
- Expression changes

5.5.2 Style Transfer Applications

Beyond identity morphing, the framework enables:

- Lighting condition interpolation
- Hair style transitions
- Facial pose adjustments
- Makeup and accessory changes

References

[1] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, 2014.

[2] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

[3] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017.

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

[5] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8110–8119, 2020.

[6] Tero Karras, Miika Aittala, Samuli Laine, Erik Härkönen, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Alias-free generative adversarial networks. In *Advances in Neural Information Processing Systems*, 2021.

[7] Elad Richardson, Yuval Alaluf, Or Patashnik, Yotam Nitzan, Yaniv Azar, Stav Shapiro, and Daniel Cohen-Or. Encoding in Style: a StyleGAN Encoder for Image-to-Image Translation. *arXiv preprint arXiv:2008.00951*, 2021.