Jojo GAN Face Stylization Real Time

Project Report

Instructor: Pawut Satitsuksanoh

Members: Thu Ya Hlaing (6511238)

Ye Myat Moe (6511233)

Table of Contents

1. Introduction

The JoJoGAN Face Stylization Real-Time project transforms a webcam video feed into an anime-style image. The project leverages deep learning techniques to create a high-quality, real-time face stylization effect, specifically inspired by the anime style.

Key Features:

- Real-time processing
- Simple and user-friendly interface
- High-quality stylization using neural networks

2. Methodology

The project is designed with the following components:

2.1 Data Processing

- **Input:** Accepts webcam feed or uploaded images.
- **Preprocessing:** Includes normalization, resizing, and format conversion to ensure compatibility with the model.

2.2 User Interface

- Built with **Tkinter** to provide an interactive GUI.

- The interface includes buttons for uploading images and controlling real-time video processing.

2.3 Stylization Process

- **Model Inference:** The input image is passed through a custom deep learning model to apply the anime style.
- **Post-processing:** The stylized image is denormalized and displayed in real-time.

3. Model

3.1 Architecture

The model is based on a **Convolutional Neural Network (CNN)** with the following features:

- **Base Network:** Incorporates Residual Blocks to retain essential image features.
- **Downsampling:** Utilizes DownBlock layers to reduce the image resolution while increasing feature depth.
- **Upsampling:** UpBlock layers restore the resolution to the original size, applying the stylization effect.
- **Residual Blocks:** Enhance the quality of stylization by preserving key features.

3.2 Training

- The model is trained on a dataset of images paired with their stylized counterparts.
- **Loss Function:** Uses Mean Squared Error (MSE) to minimize the difference between the stylized output and the target style.

3.3 Implementation

- The model is implemented using PyTorch for flexibility and efficiency.
- Pretrained weights are used to facilitate fast and reliable inference.

4. Code Overview

4.1 Main Components

- **Model Definition:** Includes classes like ResBlock, DownBlock, UpBlock, and SimpleGenerator.
- **Stylization Logic:** Handles the forward pass through the model to generate stylized images.
- **User Interface:** A Tkinter-based application manages image input, display, and real-time processing.

4.2 Key Functions

- **load_model:** Loads pretrained weights for the model.
- **stylize_image:** Handles image preprocessing, model inference, and post-processing.
- **update_frame:** Continuously captures webcam feed and applies the stylization in real time.

5. Results

The model successfully achieves one-shot real-time stylization, transforming real-time webcam feeds into an anime-style appearance with high fidelity.

GitHub Repository: https://github.com/YeMyat144/face-style

6. References

- https://github.com/mchong6/JoJoGAN
- https://blog.paperspace.com/one-shot-face-stylization-with-jojogan/
- https://research.google/blog/mediapipe-facestylizer-on-device-real-time-few-shot-face-stylization/