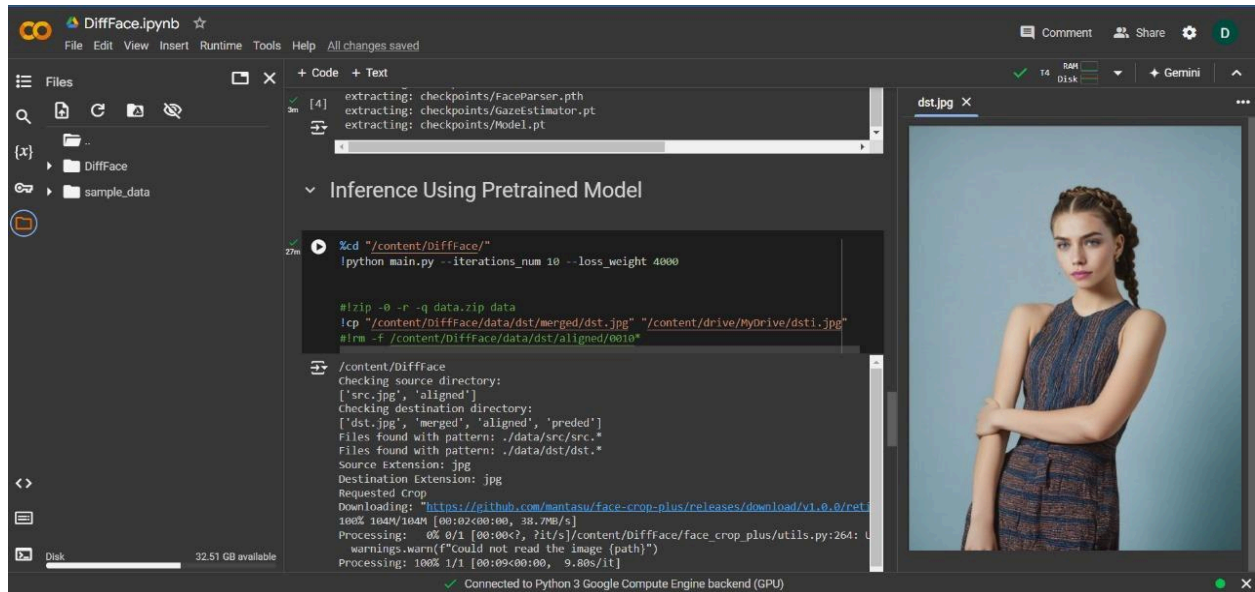


Project Report: Face Replacement Using Diffusion Model

Prepared By: Deyan Shah



Objective:

The primary objective of this project was to replace the face in a provided image with a different one using a diffusion model, ensuring the final image appeared natural and seamless. This task required a combination of advanced machine learning techniques and careful parameter tuning to achieve a visually appealing result.

Project Overview:

1. Initial Approach: Stable Diffusion with DiffFace

The project began with the implementation of Stable Diffusion, a diffusion model known for generating high-quality images. To enhance the face-swapping capabilities, DiffFace—a model specifically designed for face-swapping—was applied in conjunction with Stable Diffusion.

Challenges Encountered:

- **Blending Issues:** The initial attempts, even with the integration of DiffFace and Stable Diffusion, did not yield satisfactory results. The new face was not blending naturally into the target image, leading to a final product that looked artificial and unrealistic. This issue highlighted the need for careful parameter tuning and further refinement of the process.

2. Refining the Approach with DiffFace

To overcome the challenges faced during the initial attempts, the focus shifted towards fine-tuning DiffFace's parameters while continuing to use it alongside Stable Diffusion. This refinement process was essential to achieve a natural integration of the new face onto the original body.

Why DiffFace?

- **Specialization:** DiffFace is optimized for face-swapping tasks, offering better integration of facial features into the target image.
- **Compatibility:** It works effectively with Stable Diffusion, enhancing the natural appearance of the swapped face, making it indistinguishable from the original.

Implementation Details:

1. Setup and Configuration

- **Cloning the Repository:**
 - The DiffFace repository was cloned from GitHub to gain access to the necessary scripts and tools required for the project.
- **Library Installation:**
 - All required libraries were installed, ensuring that the environment was fully equipped to run the face-swapping process.

2. Model Preparation

- **Pretrained Model Weights:**
 - Pretrained model weights, essential for the DiffFace model to function, were imported from Google Drive. These weights provided the foundational knowledge required for the model to perform face-swapping tasks effectively.

```
[4] # Copy 'model_generic_xseg' directory from Google Drive to 'DiffFace/checkpoints'
# lcp -r "/content/drive/MyDrive/model_generic_xseg" "/content/DiffFace/checkpoints"

Requirement already satisfied: gdown in /usr/local/lib/python3.10/dist-packages (5.1.0)
Requirement already satisfied: beautifulsoup4 in /usr/local/lib/python3.10/dist-packages (from gdown) (4.12.3)
Requirement already satisfied: filelock in /usr/local/lib/python3.10/dist-packages (from gdown) (3.13.1)
Requirement already satisfied: requests[socks] in /usr/local/lib/python3.10/dist-packages (from gdown) (2.31.0)
Requirement already satisfied: tqdm in /usr/local/lib/python3.10/dist-packages (from gdown) (4.66.1)
Requirement already satisfied: soupsieve>1.2 in /usr/local/lib/python3.10/dist-packages (from beautifulsoup4) (2.5)
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.10/dist-packages (from requests[socks]) (3.3.2)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages (from requests[socks]) (3.6)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.10/dist-packages (from requests[socks]) (2.2.2)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/dist-packages (from requests[socks]) (2024.7.4)
Requirement already satisfied: PySocks!=1.5.7,>=1.5.6 in /usr/local/lib/python3.10/dist-packages (from requests[socks]) (1.7.1)
Downloading...
From (original): https://drive.google.com/uc?id=1z7u38LsVaPV2ew\_Ci\_Ylf48i-Cge8po
From (redirected): https://drive.google.com/uc?id=1z7u38LsVaPV2ew\_Ci\_Ylf48i-Cge8po&confirm=t&usp=sharing
To: /content/DiffFace/checkpoints.zip
100% |██████████| 3.07G/3.07G [00:32<00:00, 94.7MB/s]
/content/DiffFace
Archive: checkpoints.zip
replace checkpoints/Arcface.tar? [y]es, [n]o, [A]ll, [N]one, [r]ename: n
replace checkpoints/FaceParser.pth? [y]es, [n]o, [A]ll, [N]one, [r]ename: n
replace checkpoints/GazeEstimator.pt? [y]es, [n]o, [A]ll, [N]one, [r]ename: n
replace checkpoints/Model.pt? [y]es, [n]o, [A]ll, [N]one, [r]ename: n
```

- **Initial Model Training:**
 - The model was initially trained to understand the input images (source and destination). This involved running a setup script that ensured no replacement was made during the initial phase.

```
[ ] %cd "/content/DiffFace/"
!python main.py --iterations_num 10 --loss_weight 4000

#!zip -0 -r -q data.zip data
!cp "/content/DiffFace/data/dst/merged/dst.jpg" "/content/drive/MyDrive/dsti.jpg"
#!rm -f /content/DiffFace/data/dst/aligned/0010*

/content/DiffFace
```

3. Face-Swapping Execution

- **Main Script Execution:**
 - The **main.py** script from the DiffFace repository was executed to perform the face swap. This script was the core of the project, responsible for handling the image processing and face replacement tasks.
- **Parameter Tuning:**
 - Two critical parameters were adjusted during the execution:

- Iterations Number (**iterations_num**): This parameter controls the number of times the model processes the image, impacting the quality and refinement of the final output.
- Loss Weight (**loss_weight**): This parameter influences the balance between the preservation of facial features and the integration of the new face into the target image. Fine-tuning this value was crucial to achieving a natural look.

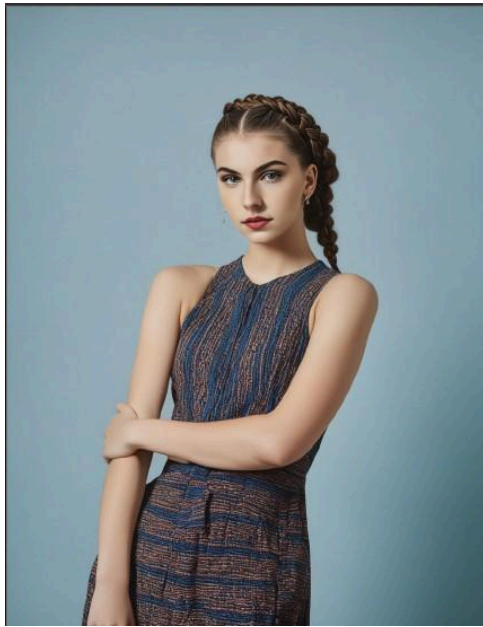
4. Image Upload and Processing

- Source and Destination Images:
 - The source image (**src.jpg**) was uploaded to the **diffFace/data/src** directory, and the destination image (**dst.jpg**) was placed in the **diffFace/data/dst** directory. These images served as the inputs for the face-swapping process.
- Model Testing:
 - After setting the parameters, the face-swapping process was re-executed to test the new input images. The model processed these images and saved the output in the **DiffFace/data/merged/dst.jpg** folder, where the final merged image could be reviewed.

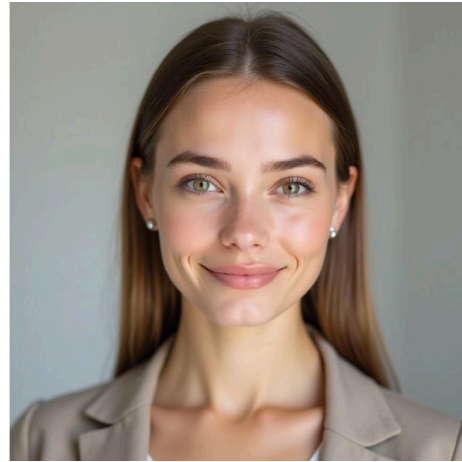
Outcome:

The project successfully achieved a face replacement that looked natural and met the project's requirements. The use of DiffFace alongside Stable Diffusion proved effective, and the refined process, along with careful parameter tuning, allowed for a seamless integration of the new face onto the original body, producing a final image that was both realistic and visually coherent.

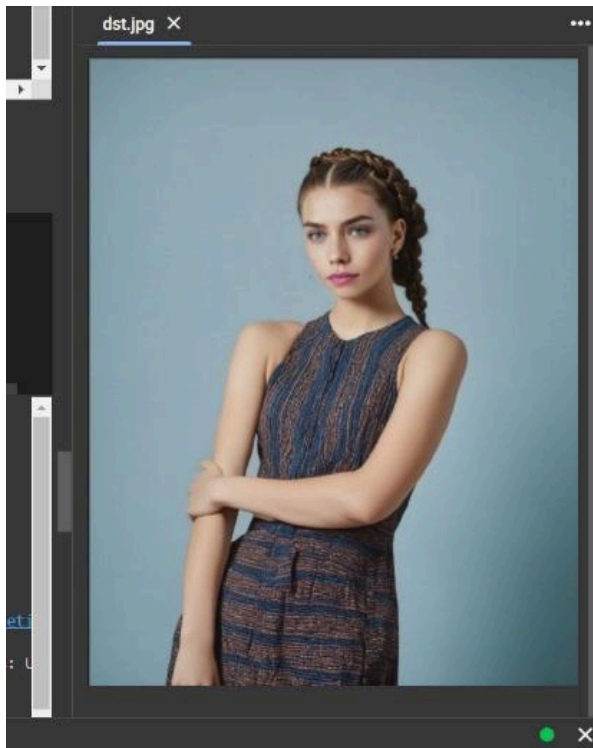
Required Screen shots



+



After merging these 2 picture i got the following result



After Face Replace



Original

Methodology for Reproduction:

1. **Repository Setup:**
 - Clone the DiffFace GitHub repository.
 - Import the necessary pretrained weights and download the required checkpoints.
2. **Initial Model Training:**
 - Train the model by running the initial setup script, ensuring no replacements are made at this stage.
3. **Image Upload:**
 - Upload the source image to **diffFace/data/src**.
 - Upload the destination image to **diffFace/data/dst**.
4. **Execution:**
 - Re-run the main script to process the new inputs, with parameters such as **iterations_num** and **loss_weight** adjusted as needed. The final output image will be saved in the **DiffFace/data/merged/dst.jpg** folder.