Smart Face Age Editing with GANs - Engineering Track

Warade Atharv Abhijit
2022582
warade22582@iiitd.ac.in

Note

This report reflects a change in project topic from the originally proposed *Ocular Disease Detection* to the current topic *Smart Face Age Editing with GANs*. This is because I have already made a project on classification and wanted to explore new things in the field of Computer Vision

1. Problem Statement

We propose an AI-powered system that enables realistic and controllable aging or de-aging of human facial images. The key novelty is the ability to transform a face into any target age range selected by the user from predefined bins (e.g., 0–10, 11–20, ..., 91–100). This functionality is built using StarGAN, a multi-domain image-to-image translation model capable of learning transformations across age groups within a unified framework.

Input: A facial image and a target age bin (e.g., 31–40 years).
Output: The same face transformed to appear as though it belongs to the selected age group.
Users: General public, VFX studios, digital content creators, forensic investigators, and privacy-conscious individuals.
Interface: A web app where users upload an image and select a target age group. The system then displays the generated face image for that age bin next to the original.

Why this problem? Most existing models offer binary age editing (young to and fro old). This project pushes forward by allowing nuanced control over the apparent age. It has applications in storytelling, missing person reconstruction, and social media avatars.

Challenges include:
- Preserving identity while changing age
- Generalizing across lighting/pose variations
- Avoiding bias or hallucination due to unbalanced data

2. User Interface and Progress

The interface is being developed using React.js for the frontend and FastAPI for backend inference. The user uploads an image, selects a target age bin (e.g., 41–50), and receives a transformed face image.

Completed:
- Web UI with upload and age group selection dropdown

In Progress:
- Preprocessing pipeline to match age binning with UI selections
- Frontend layout using React.js and TailwindCSS
- Backend inference API integration with UI
- Return and display of transformed image after model inference
- UI validation and responsiveness testing

Tools used: React.js, TailwindCSS, FastAPI, VSCode, GPT-4o.

3. Related Work

1. StarGAN v1 (Yunjey Choi et al., CVPR 2018) – StarGAN v1 enables multi-domain image-to-image translation using a single generator and discriminator. It uses a domain label as a one-hot vector to guide transformations, such as young → old or male → female, while preserving identity. The model combines adversarial loss, domain classification loss, and cycle consistency loss to produce realistic and consistent outputs. StarGAN v1 is known for its simplicity, efficiency, and support for reversible transformations. It works well with structured domain labels (like age bins), making it suitable for controlled facial editing with limited computational resources.

2. StarGAN v2 (CVPR 2020) – StarGAN v2 improves visual quality and supports diverse outputs per domain using style-based control. It introduces a style encoder and mapping network to allow both latent-guided and reference-guided translation. Unlike v1, it does not use cycle consistency, focusing instead on perceptual and style losses. The architecture produces high-quality and varied results but is more complex and computationally demanding. It requires more VRAM and training time, making it better suited for commercial applications than lightweight setups. Style diversity is its strength, but identity preservation may vary across different styles and domains.

BASELINE: StarGAN v1 is selected due to its domain control via one-hot encoding, lower compute needs, and built-in cycle consistency, which preserves identity—essential for realistic age transformations.

4. Dataset and Evaluation Metrics

Dataset: IMDB-WIKI dataset, preprocessed and filtered to form a balanced subset (which has relatively equal images for all age groups) containing roughly equal samples per age bin.

Each image is aligned using MTCNN and resized to 128×128 resolution. One-hot vectors represent the target domain during training.

Evaluation Metrics:

- Frechet Inception Distance (FID) – Measures the similarity between real and generated images in terms of feature distributions; lower FID indicates higher realism.
- Classification accuracy of age group using a pre-trained classifier (optional) – Assesses whether the generated image appears to belong to the target age group. We'll optionally use a pre-trained age classifier to check if the model correctly transforms input images into the desired age group.
- Identity preservation via cosine similarity on facial embeddings – Measures how well the identity of a person is preserved in the generated image by comparing feature vectors.
- Consistency loss (cycle consistency) – Evaluates how well the model can reconstruct the original image when an aged image is de-aged back to the original domain. A lower consistency loss indicates stronger identity and structure retention across transformations.
- Identity loss-based consistency – During evaluation, we check if the generator makes minimal changes when transforming an image to its original domain. This reflects whether the model has learned to avoid unnecessary alterations. A low difference score indicates high identity retention when no transformation is expected.

Age Bin	Number of Images
0–10	2000
11–20	3000
21–30	3000
31–40	3000
41–50	2000
51–60	3000
61–70	2000
71–80	1500
81–90	1000
91–100	246

Table 1. Balanced Age-Binned Dataset Stats

5. Analysis of Results

Currently in training phase. We will evaluate:

- Training FID score per epoch
- Qualitative inspection across bins (e.g., 18 → 40, 60 → 20)
- Identity preservation with facial recognition model (ArcFace)
- Inference latency for end-user experience (target less than 1s)

6. Compute Requirements

- **Local Machine:** NVIDIA RTX 3050 (4GB VRAM), 16GB RAM - Used for testing and low-resolution inference.
- **Training Platform:** Google Colab (Free tier) - equipped with Tesla T4 (16GB VRAM) or similar GPUs.
- **Storage:** Local (up to 200GB) and/or Google Drive (up to 100GB) used for saving dataset, model checkpoints, and logs.
- **Training Resolution:** 128x128 (chosen to balance quality and memory). 256x256 is possible with reduced batch sizes.
- **Inference:** Can be performed locally on the 3050 GPU or on Colab using saved models.
- **Limitations:**
  - Google Colab's 12-hour session limit requires periodic checkpoint saving and restart handling.
  - Training at 512x512 resolution is not feasible with current resources.
  - Batch size restricted to 1–2 during training due to GPU memory constraints.

7. Individual Tasks

Component	**Responsible**
Data Preprocessing	Warade Atharv Abhijit
UI Development	Warade Atharv Abhijit
StarGAN Training	Warade Atharv Abhijit
API Integration	Warade Atharv Abhijit
Evaluation	Warade Atharv Abhijit
Deployment	Warade Atharv Abhijit

8. Next Steps

- Train StarGAN with age-bin labels from balanced dataset csv file
- Create inference pipeline using FastAPI and PyTorch
- Connect frontend to backend and enable user testing
- (Optional) Add age-estimation model for optional output verification