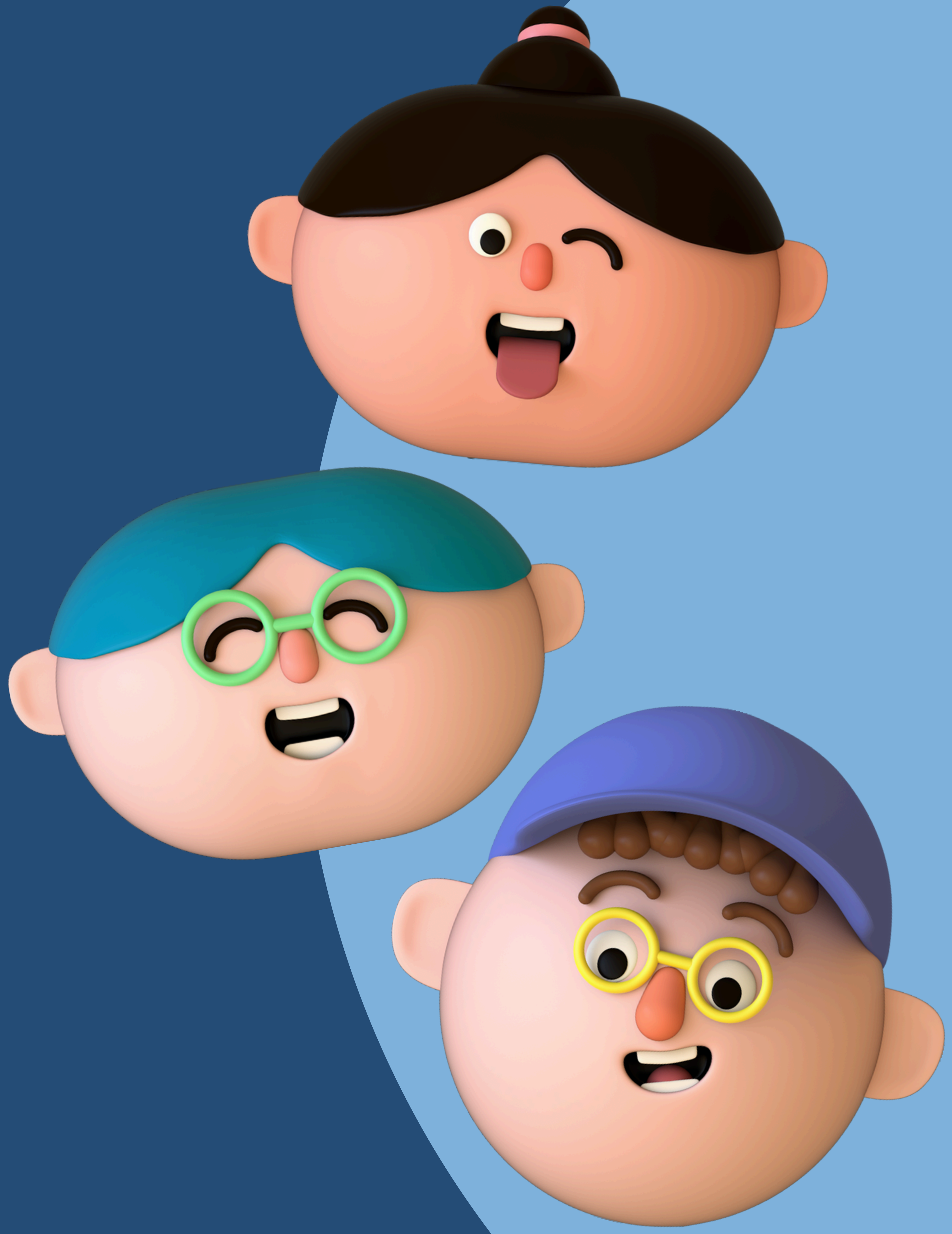
CARTOONIZE WITH CYCLE GAN

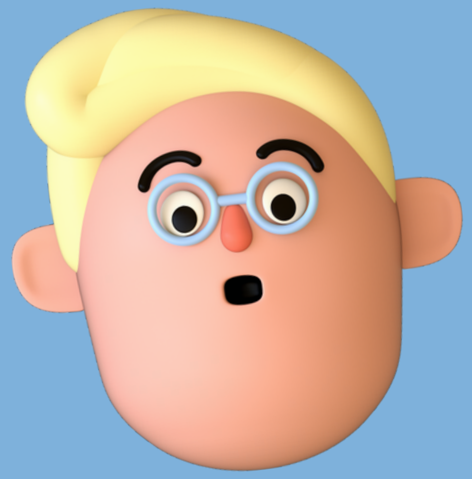


INTRODUCTION

- Utilizing Cycle Generative Adversarial Networks (CycleGAN) for unpaired image translation.
- Transforming celebrity photographs into charming cartoon renditions.
- Employs innovative techniques like cycle consistency loss for training without paired data.

OBJECTIVE

This project utilizes CycleGAN to translate images between celebrity photos and cartoons. It trains generators and discriminators to minimize adversarial and cycle-consistency losses, aiming to generate accurate representations of each domain. The goal is to create high-quality image translations while preserving the distinct characteristics of celebrities and cartoons.



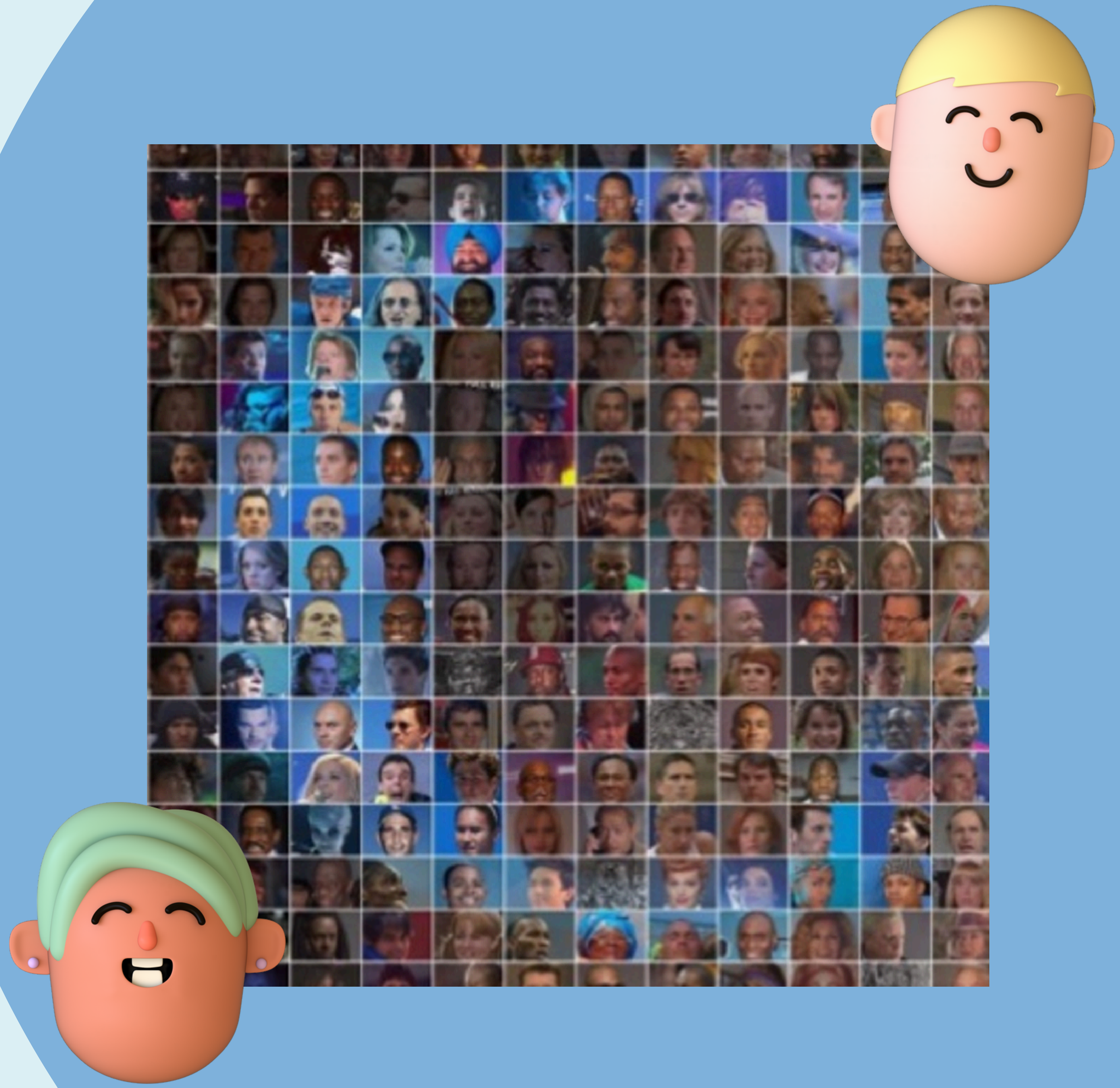
☰ ABOUT DATASET

CelebFaces Attributes (CelebA) Dataset

- 202,599 number of face images of various celebrities

cartoonset10k

- This dataset has 10,000 images



WORKFLOW

1. DATA LOADING AND PRE-PROCESSING

images are resized to a common size (64x64 pixels) and normalized before being saved in compressed NumPy arrays.

2. DISCRIMINATOR MODEL

Two discriminator models (one for each-celebrity , cartoon) are defined using convolutional layers with Instance Normalization instead of Batch Normalization.

3. GENERATOR MODEL

Two generator models are defined using convolutional layers and ResNet blocks. These models aim to transform images from one domain to another.

4. COMPOSITE MODEL

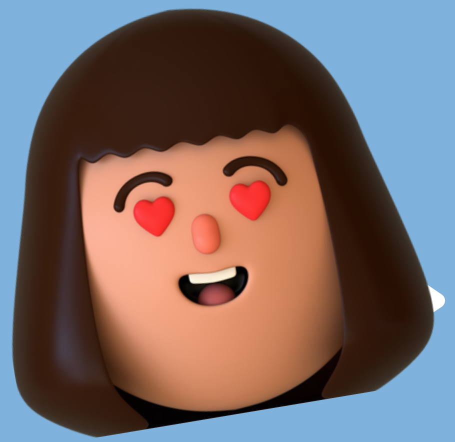
Composite models are defined to combine the generators and discriminators for training. These models are responsible for adversarial training and cycle consistency.

5. LOAD REAL SAMPLES

Defines functions for loading, generating, saving models, updating an image pool, and displaying predictions.

6. TRAIN CYCLE GAN

Trains a CycleGAN model for image-to-image translation, updating generators and discriminators iteratively using real and fake samples, and utilizes an image pool for stabilization

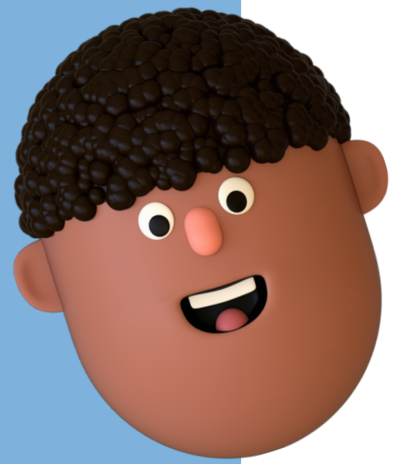


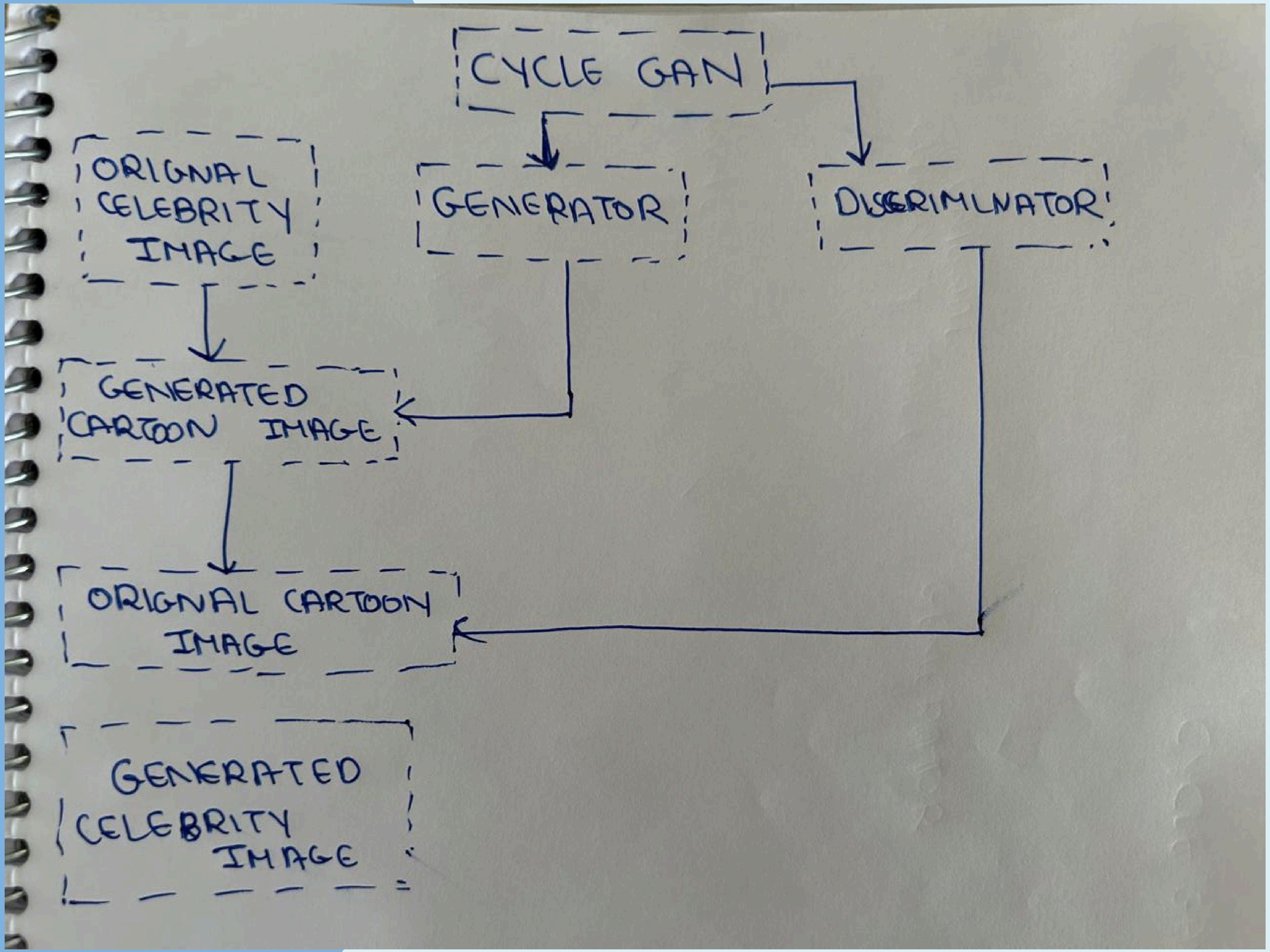
After loading real samples, the model follows these key steps:

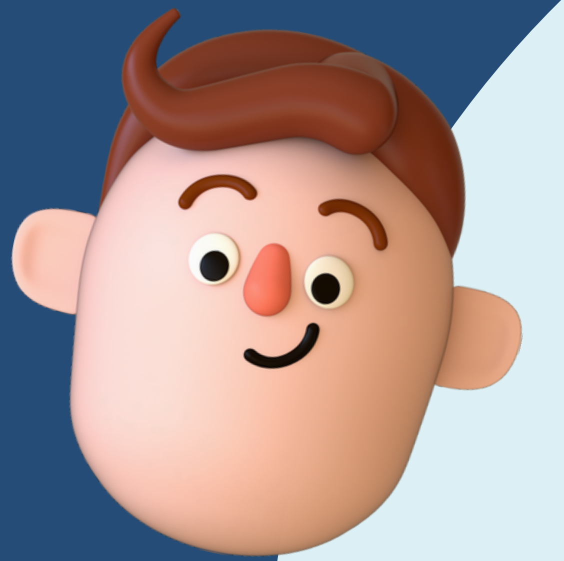
- **Batch Generation:** Real samples are selected in batches from each domain.
- **Fake Image Generation:** Generators produce fake images for each domain.
- **Image Pool Update:** Fake images are updated in an image pool to stabilize training.

Model Training:

- **Initialization**: It initializes parameters like the number of epochs, batch size, and the shape of the discriminator output.
- **Data Preparation**: The training dataset is unpacked into sets for the two domains being translated.
- **Image Pools**: Pools for fake images from each domain are initialized.
- **Training Loop**: It iterates over epochs and within each epoch over the batches of data.
- **Sample Generation**: Real and fake samples for both domains are generated.
- **Image Pool Update**: Fake samples are added to their respective image pools.
- **Generator Training**: The generators are trained using adversarial and cycle loss functions.
- **Discriminator Training**: The discriminators are trained on real and fake samples.
- **Performance Monitoring**: Performance metrics are printed for each batch.
- **Visualization and Model Saving**: Every few epochs, predictions are shown and the models are saved.




CYCLE GAN
ORIGNAL CELEBRITY IMAGE
GENERATOR
DISCRIMINATOR
GENERATED CARTOON IMAGE
ORIGNAL CARTOON IMAGE
GENERATED CELEBRITY IMAGE



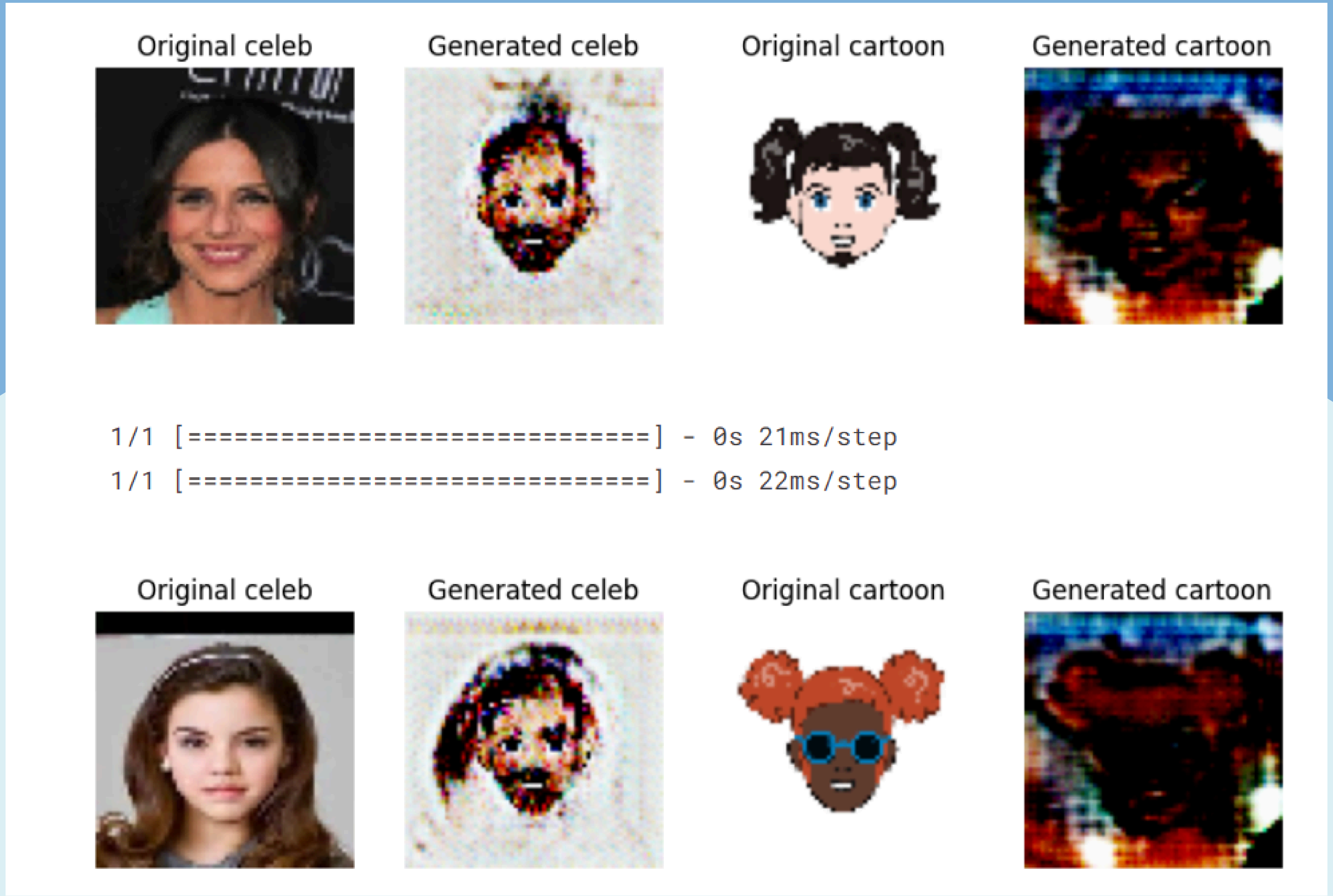
OpenCV

TOOLS USED

Keras

A deep learning library

RESULTS


Original celeb
Generated celeb
Original cartoon
Generated cartoon
1/1 [==============================] - 0s 21ms/step
1/1 [==============================] - 0s 22ms/step
Original celeb
Generated celeb
Original cartoon
Generated cartoon

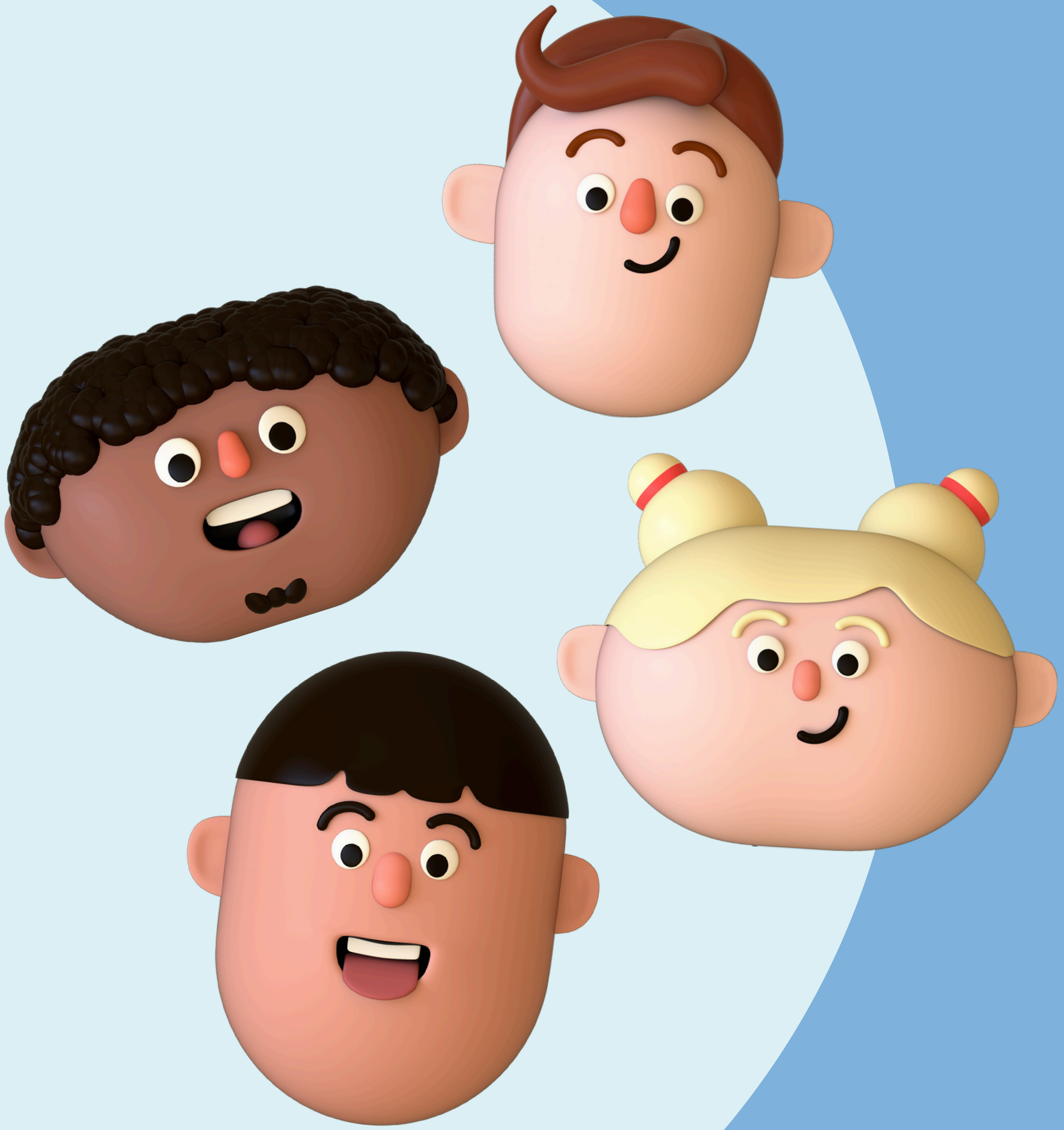
CONCLUSION

Our project successfully translated celebrity photos into cartoon depictions, though with room for improvement. By refining our generator and discriminator models, we can achieve higher-quality outputs. Strategies like adjusting network architecture and optimizing training parameters offer avenues for enhancement. Additionally, diversifying the dataset can enrich the model's understanding.

REFERENCES

- https://github.com/FilipAndersson245/cartoon-gan
- https://www.irjmets.com/uploadedfiles/paper/volume3/issue_1_january_2021/5620/1628083227.pdf
- https://towardsdatascience.com/avatargan-generate-cartoon-images-using-gan-1ffe7d33cfbb

- https://towardsdatascience.com/avatargan-generate-cartoon-images-using-gan-1ffe7d33cfbb



THANK YOU

PRESENTED BY:

- GARGI JOSHI (21070126029)
- JANHAVI PATIL (21070126040)
- HEMAKSHI BHATNAGAR (21070126036)