FakeCheck - Detecting and Classifying Whether an Image of Human Face Is Real or Fake

Praveen Kumar Sridhar, Kapil Sahu, Meghaana Tummapudi

April 15, 2023

1 Problem Statement

The widespread use of images on social media makes it easy to create and share fake images quickly, which poses a threat to the credibility of news and social communication. Social media id-theft is also on the rise, with scammers using fake profiles to deceive people. Image forgery/manipulation techniques contribute to unrealistic beauty ideals, leading to negative effects on mental health. Hence, monitoring and detecting fake facial images is important to maintain the truthfulness of the content.

2 Dataset

2.1 140K Real and Fake Faces

The dataset consists of all 70k REAL faces from the Flickr dataset collected by Nvidia, as well as 70k FAKE faces generated by StyleGAN. It is a balanced dataset. The images are of dimensions 256x256x3 representing the coloured images. The dataset is split into train, validation and test set. An example of real and fake images is in fig 7.

2.2 DFFD

The Diverse Fake Face Database (DFFD) dataset consists of GAN generated images with greater diversity in FAKE images. It consists of FAKE images generated by StyleGAN, PGGAN and StarGAN with balanced demographics like 48% Male subjects and 52% of Female subjects. The age range is from 21-50 years and the image dimensions are 256 x 256 x 3 pixels. Examples can be seen in fig 8.

2.3 CelebA Dataset

CelebFaces Attributes Dataset (CelebA) is a large-scale face attributes dataset with more than 200K celebrity images, each with 40 attribute annotations. The images in this dataset cover large pose variations and background clutter. We have used this dataset to extract additional REAL images. Examples are in fig 9.

3 Low-Risk Methodology & Results

As a part of low-risk goals, we have evaluated our phase-1 model's performance on the newly curated dataset and retrained the best-performing models to detect more diverse FAKE images. We decided to retrain them on the new dataset fig 10, so that it can learn from the FAKE images generated by StyleGAN, StarGAN and PGGAN and improve their performances. We retrained the VGG16 and VGG19 models (Model architecture is attached in fig 11) on the newly curated dataset and it resulted in the performance boost of the models to detect FAKE images more accurately. The results and metrics can be found in the results section fig 4.

4 Medium-Risk Methodology & Results

As a medium-risk objective, we aimed to comprehend the production process of fake images, and thus conducted experiments with DCGAN. By generating synthetic images through DCGAN, we could create additional fake images independently in the event of retraining without relying on external sources.

4.1 Deep Convolutional Generative Adversarial Networks (DCGAN)

For our project, we utilized a dataset containing 140K real and fake faces, and trained a DCGAN using the entire set of real images, which amounted to 50K. The purpose of this was to generate fake images via DCGAN. DCGANs are made of two distinct models, a generator and a discriminator refer fig 12. The generator creates fake images that resemble the real ones, and the discriminator decides if an image is real or fake. The goal is for the generator to improve its fakes while the discriminator improves its accuracy. The ideal outcome is when the generator produces perfect fakes, and the discriminator can only guess randomly. Let D(x) be the descriminator, and G(z) be the generator. D and G play a minimax game in which D tries to maximize the probability it correctly classifies reals and fakes ($logD(x)$), and G tries to minimize the probability that D will predict its outputs are fake ($log(1-D(G(z)))$). The GAN loss used here can be seen in fig 1. The Arch for the generator is shown in detail in fig 13. The fake images generated by our DCGAN can be found in fig 14.

$$\min_{G}\max_{D} V(D,G) = \mathbb{E}_{x\sim p_{data}(x)}\left[logD(x)\right] + \mathbb{E}_{z\sim p_z(z)}\left[log(1-D(G(z)))\right]$$

Figure 1: DCGAN Loss

5 High-Risk Methodology & Results

As a part the high-risk goals we tested our retrained model on the custom GAN-generated images. Researched modeling complex GAN like PGGAN and StarGAN to generate fake images. However, this task could not be fully accomplished as it demands high-end computing resources and we concluded it as a part of an excellent failure in our project. A big positive for us was to design a UI 2 and an API 3 using 'Streamlit' open-source library and then deploy it on the Google Cloud Platform (GCP). This ensures access to our API over the internet to detect any FAKE human face image.

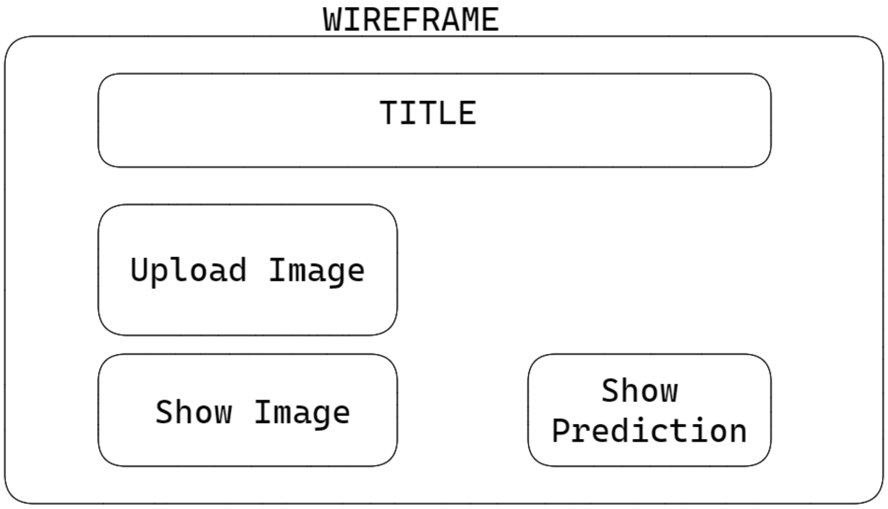


Figure 2: UI Wireframe

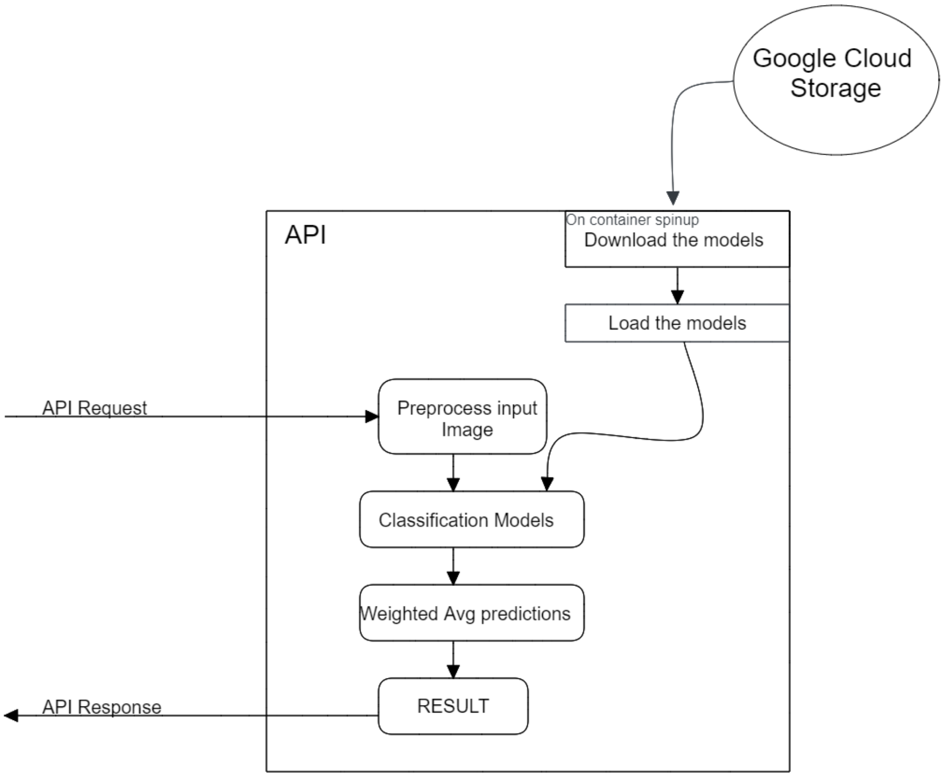
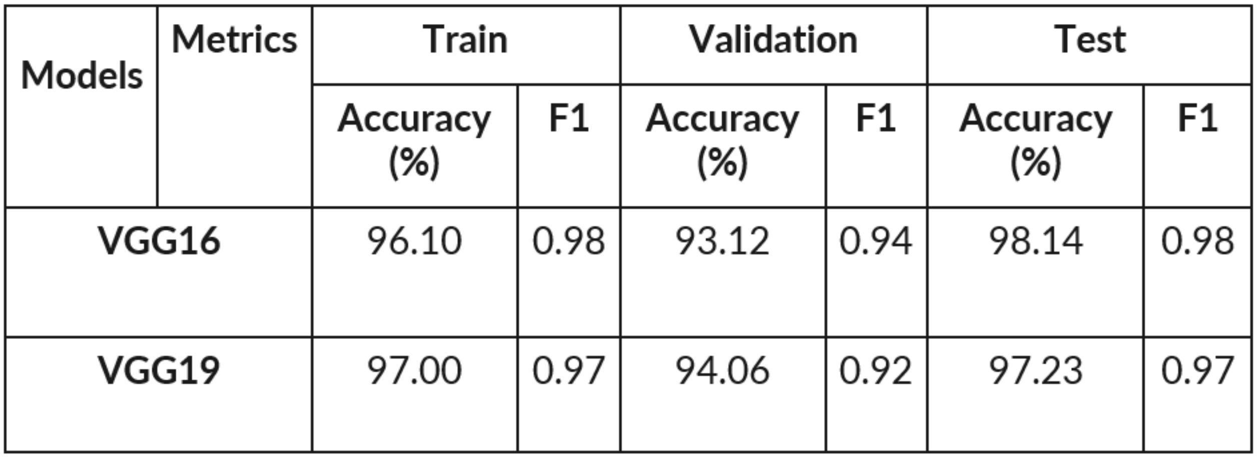


Figure 3: API Design

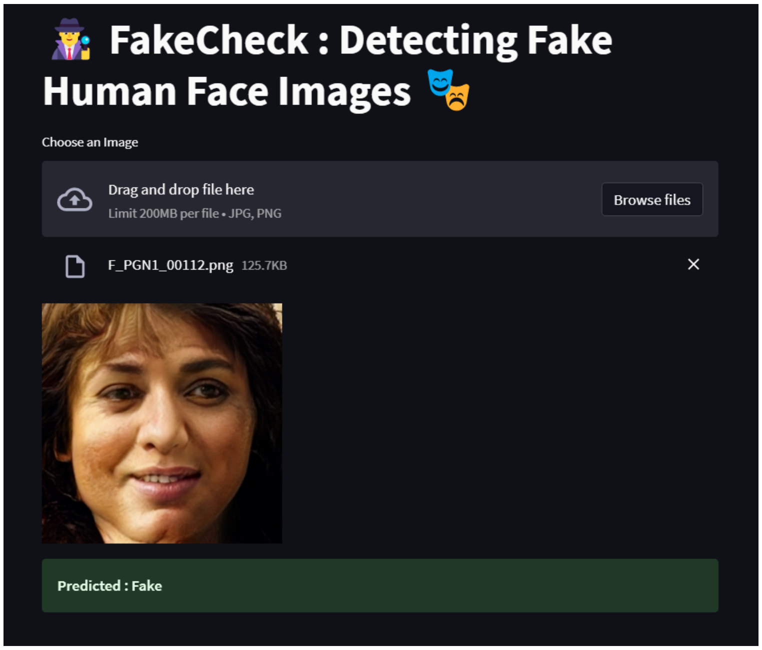
6 Results

The chosen evaluation metrics for comparing the performance of the models are accuracy and f1 score. These metrics were chosen because the dataset is balanced and they provide a comprehensive view of the model's ability to correctly classify both real and fake images. The accuracy of VGG-16 and VGG-19 models after retraining is listed in the table. Please refer to the following Evaluation Metrics table 4 for performance metrics and the final user interface 5 to access the API:



Models	Metrics	Train		Validation		Test	
		Accuracy (%)	F1	Accuracy (%)	F1	Accuracy (%)	F1
VGG16		96.10	0.98	93.12	0.94	98.14	0.98
VGG19		97.00	0.97	94.06	0.92	97.23	0.97

Figure 4: Evaluation Metrics



🕵️ FakeCheck : Detecting Fake Human Face Images 🎭

Choose an Image

Drag and drop file here

Limit 200MB per file • JPG, PNG

F_PGN1_00112.png 125.7KB

Predicted : Fake

Figure 5: API Snapshot

7 Future Work

Based on this phase, we can confidently conclude that our model exhibits strong performance across all three sets: train, validation, and test. For future improvements, it would be beneficial to acquire additional computing resources in order to generate our own fake images, thereby enhancing the robustness of our current pipeline.

8 References

1. Dataset: 140k Real and Fake Faces
2. DFFD dataset

3. Celeb Dataset

4. VGG: Very Deep Convolutional Networks For Large-Scale Image Recognition

5. DCGAN

6. StreamLit

7. Project Source Code

8. FakeCheck API link

9 Appendix

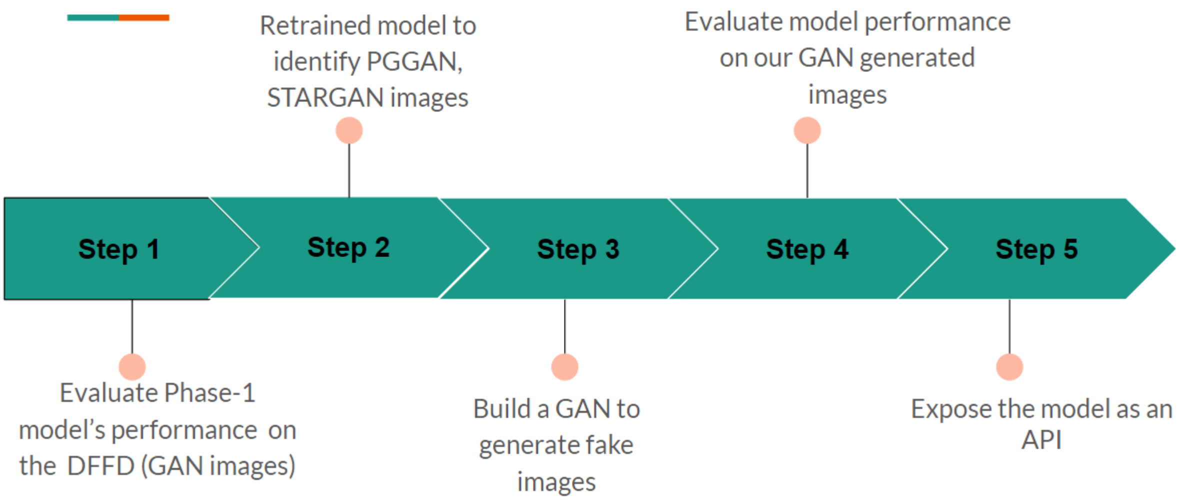

Retrained model to identify PGGAN, STARGAN images
Evaluate model performance on our GAN generated images
Step 1
Step 2
Step 3
Step 4
Step 5
Evaluate Phase-1 model's performance on the DFFD (GAN images)
Build a GAN to generate fake images
Expose the model as an API

Figure 6: Workflow

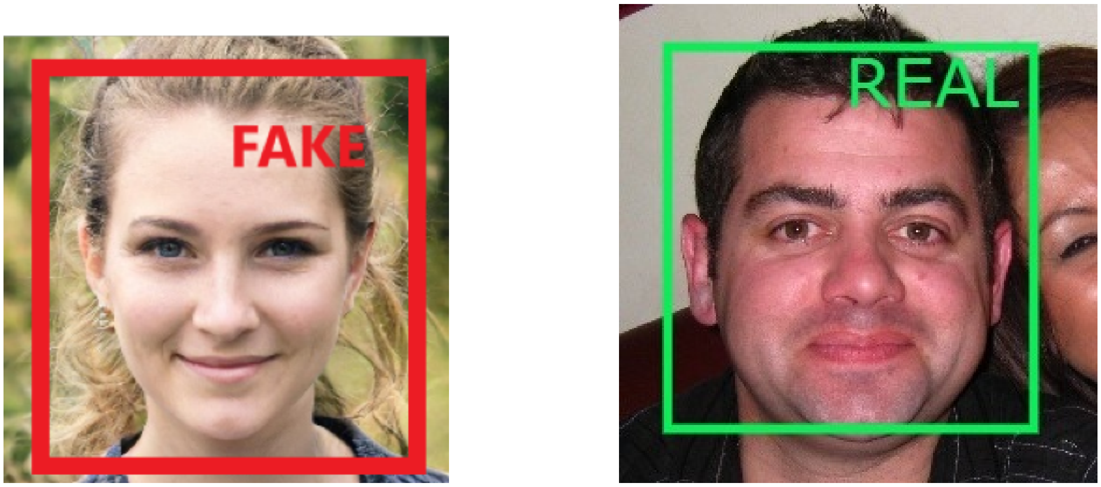


Figure 7: An example of Fake and Real images, indistinguishable to human eyes

Figure 8: Couple of examples from DFFD dataset

Figure 9: Couple of examples from CelebA dataset

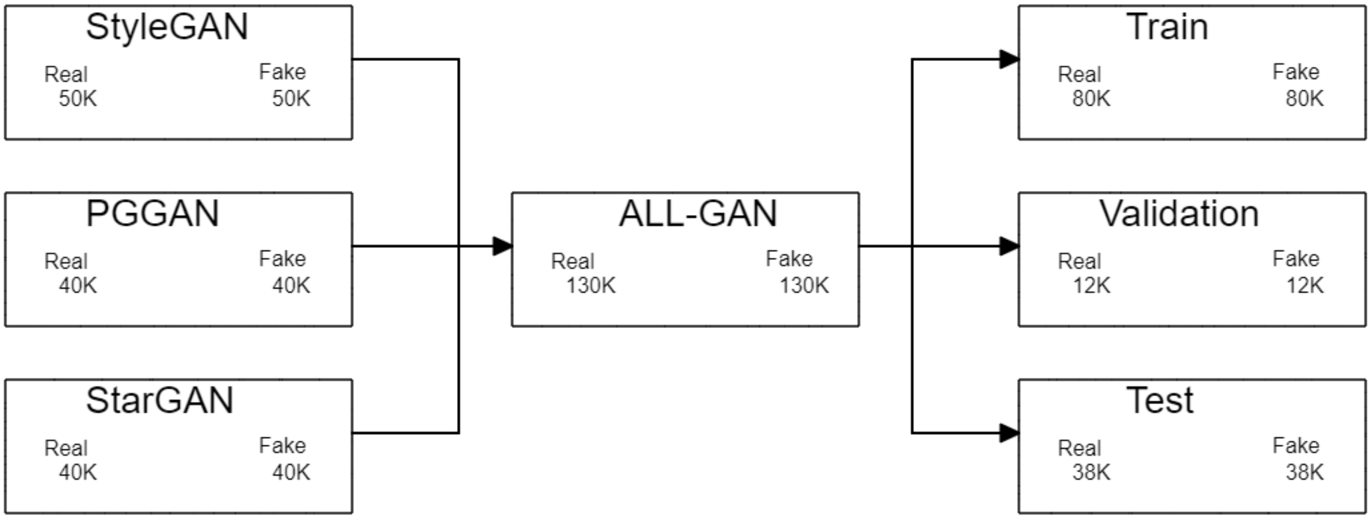


Figure 10: Curated GAN Data

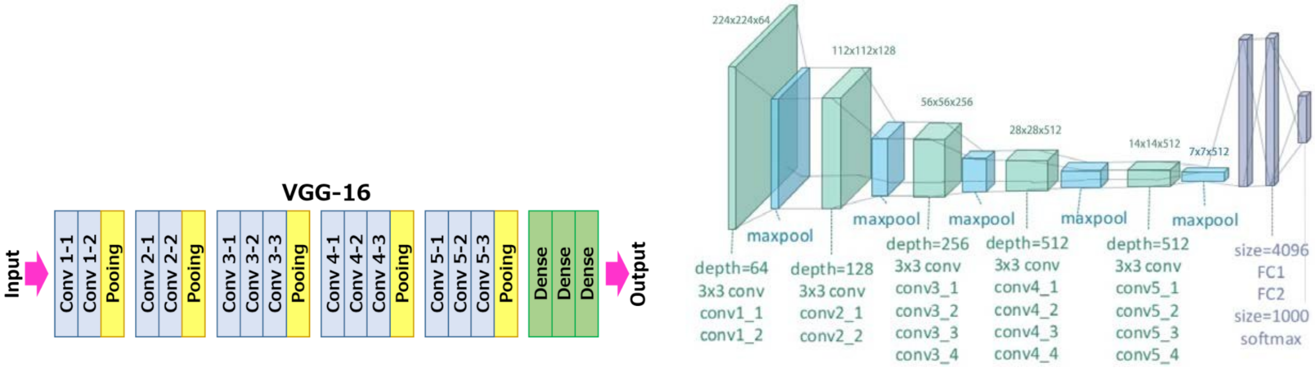


Figure 11: VGG16 and VGG19 Architecture

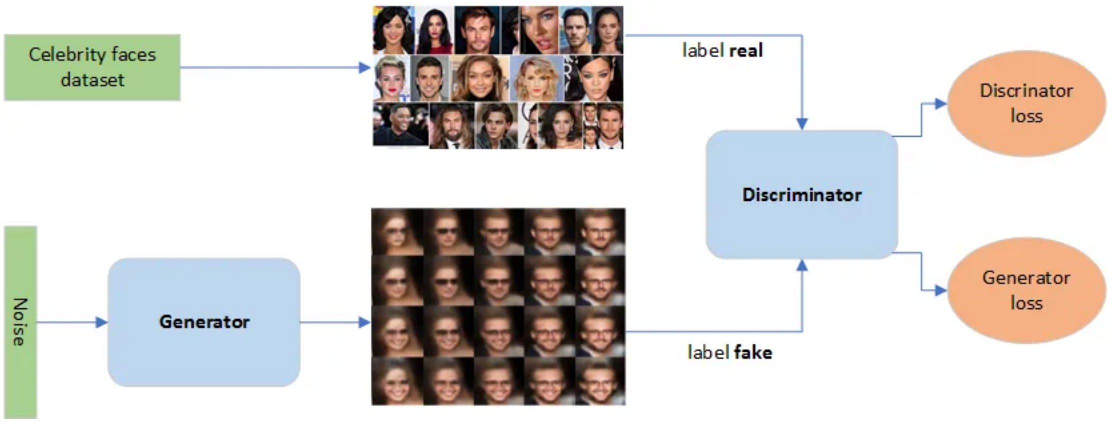


Figure 12: A simple DCGAN Arch.

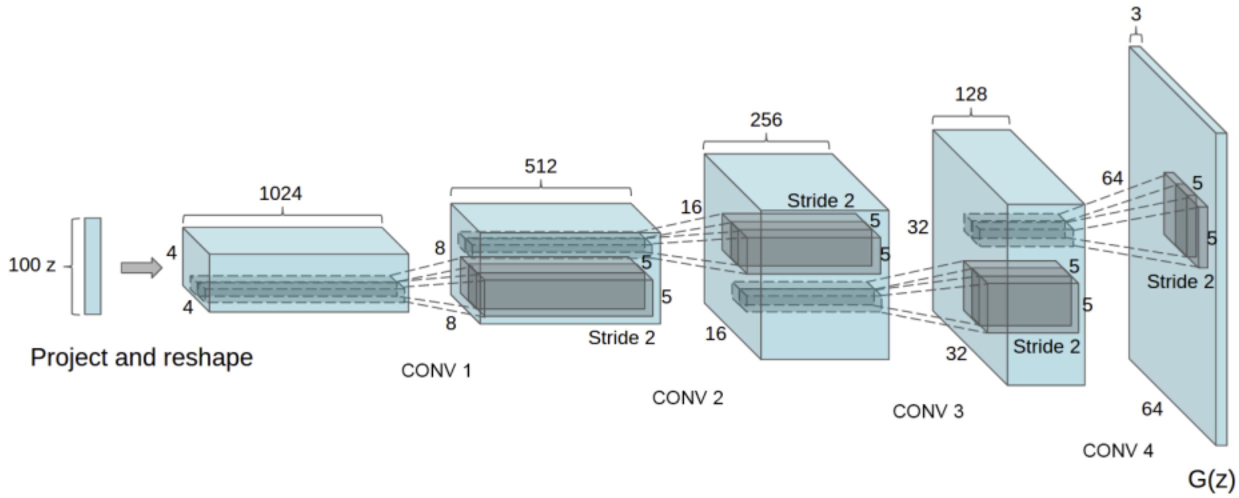


Figure 13: Generator Arch.

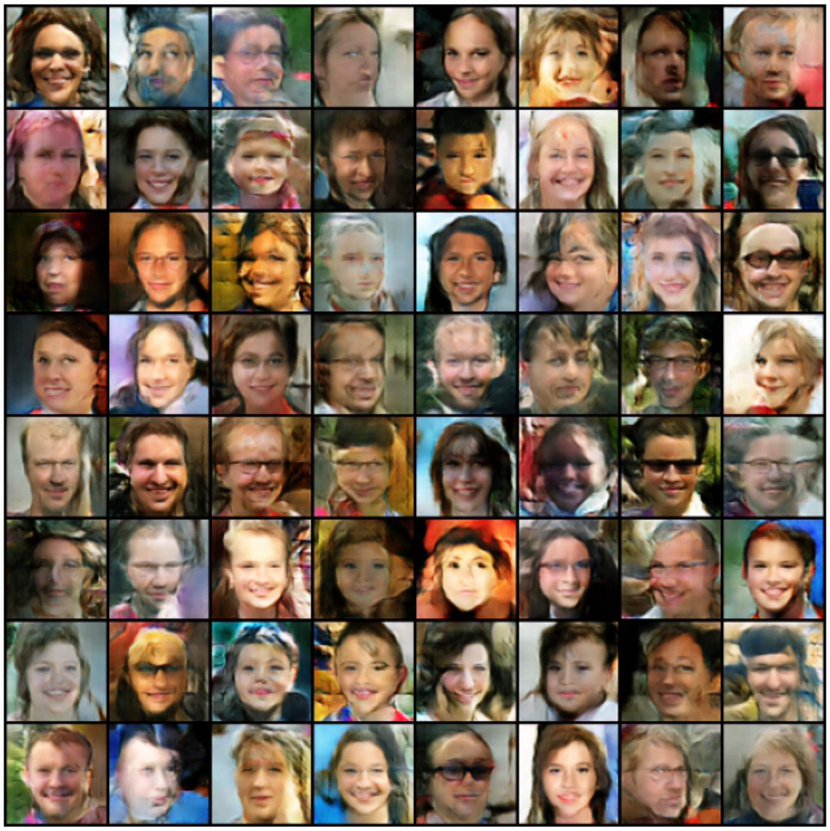


Figure 14: DCGAN generated images