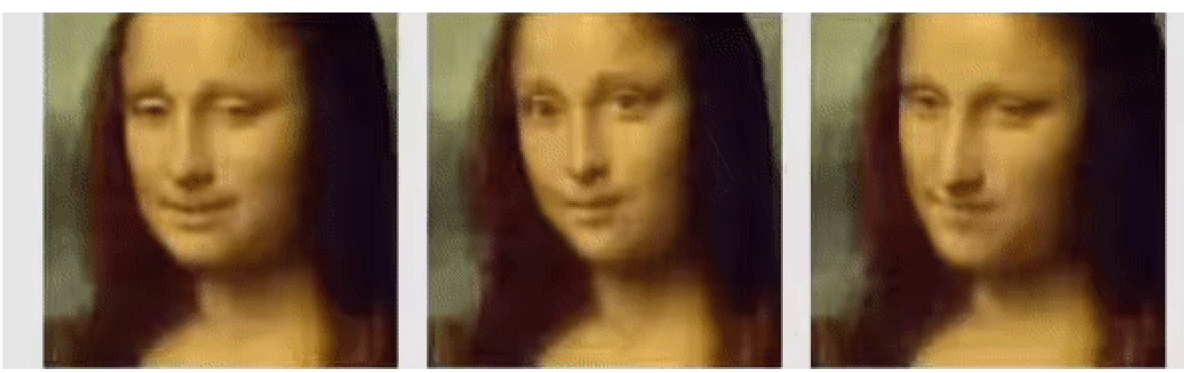
Deep Learning
Homework 3

Snir Lugassy (206312506)
Ofri Hefetz (209028067)

Index:



GitHub repository: https://github.com/snirlugassy/CelebA-GAN

Data set

CelebFaces Attributes Dataset, or “CelebA” for short, is an image dataset that identifies celebrity face attributes. It contains 202,599 face images across five landmark locations, with 40 binary attribute annotations for each image. It currently includes data relating to over 10,000 celebrities.

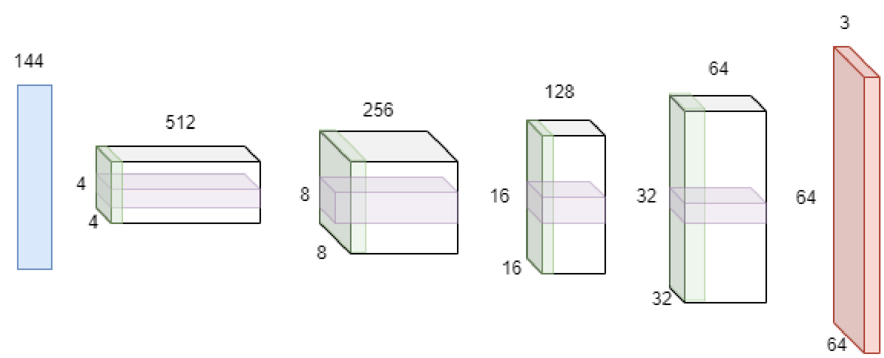
“CelebA” includes many diverse images covering various poses and background variations. It provides rich annotations for the images useful for training machine learning and computer vision models.

Model Architecture:

We used a DCGAN, a direct extension of the GAN except that it explicitly uses convolutional and convolutional-transpose layers in the discriminator and generator, respectively.

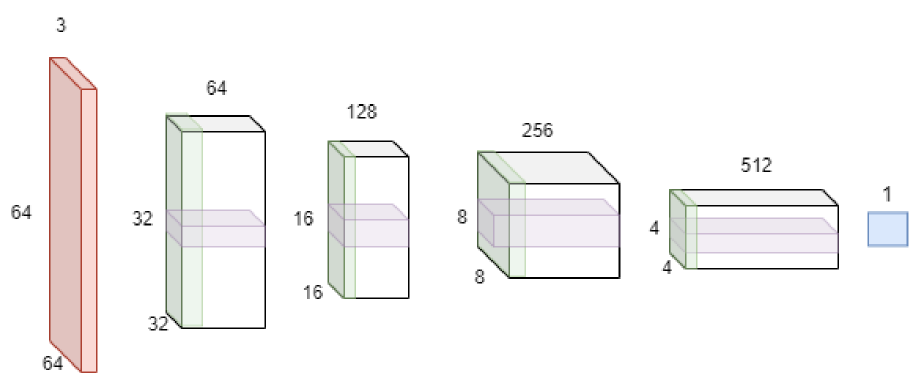
Generator

In the generators we've used 2D transpose convolution (“ConvTranspose2d”) to upscale from the latent space into the output space (generated image space). After each ConvTranspose2d layer, we've used batch normalization and ReLU non-linear activation.



Discriminator

In the discriminator we've used 2D convolutional layers, followed by batch normalization and “LeakyReLU” with a negative slope of 0.2.



Transformations

We used the following transformations for the “CelebA” dataset:

1. Resize to 78x64
2. Center crop 64x64
3. PIL to torch Tensor
4. Normalize channels

We have tried different sizes for example 3x128x128 in particular. This size slowed down the training process and was more challenging for the generator to compete with the discriminator.

Hyperparameters

- Batch size: 128
- Learning rate: 0.00009
- Number of epochs: 30
- Regularization: L2 regularization using Adam’s “weight_decay” parameter
- Weight init:
  - Convolutional layers: N(0, 0.02)
  - Batch normalization: N(1, 0.02)
- Optimization: Adam

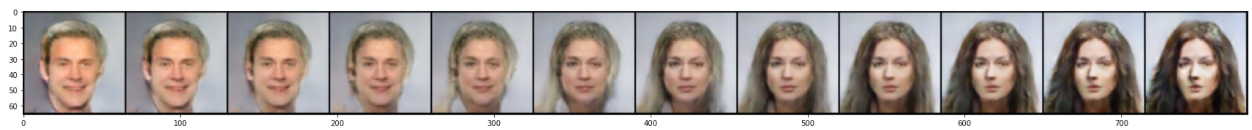
Latent Space Exploration

Our latent space consists of 16 discrete dimensions, and 128 continuous dimensions (a total of 144 dimensions). An important feature of GAN latent space is the continuity of the generator. Slight changes in the latent vector will cause slight changes in the output image.

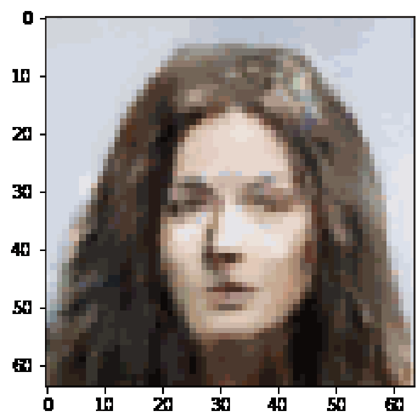
The 16 discrete dimensions were sampled uniformly at random from [0,9]
The 128 continuous dimensions were sampled uniformly at random from [-2,2]

To explore this continuity we interpolated 2 latent vectors:

$$\alpha \cdot z_1 + (1 - \alpha) \cdot z_2$$



Animated interpolations: [(Link to gif)]



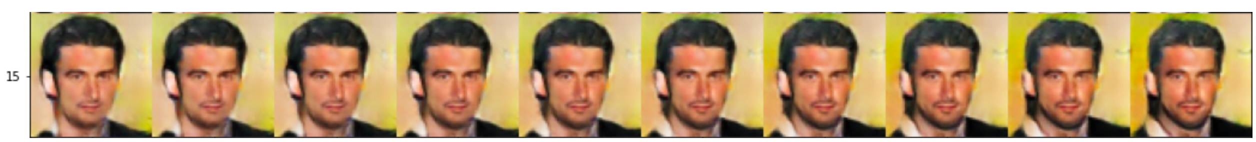
Exploring the discrete dimensions

(Appendix A includes the full image grids)

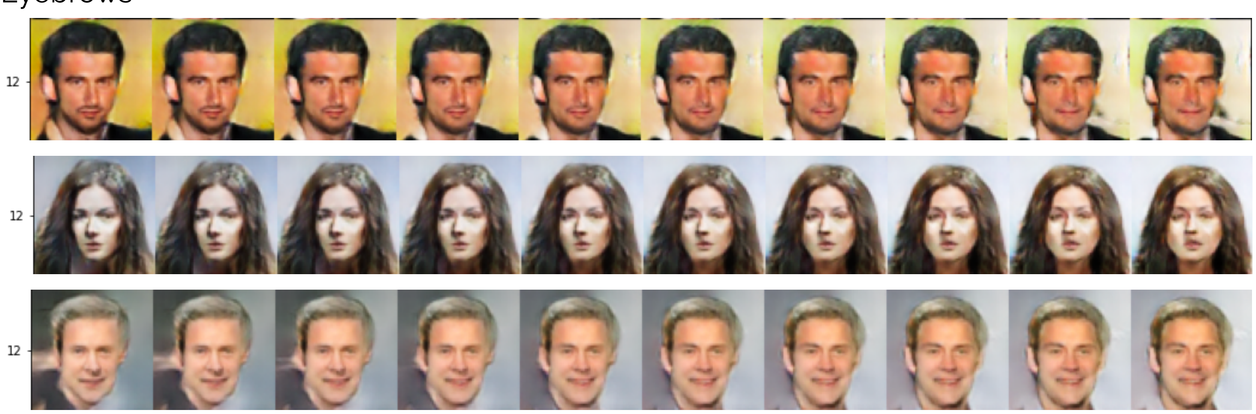
Gender

Aging + white hair

Facial hair

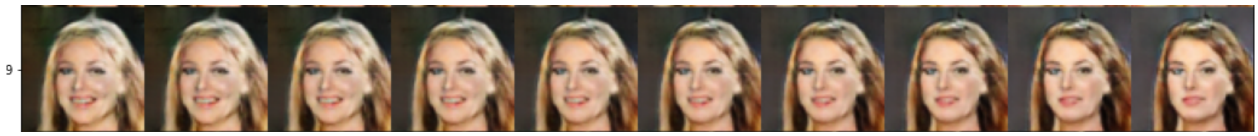


Eyebrows

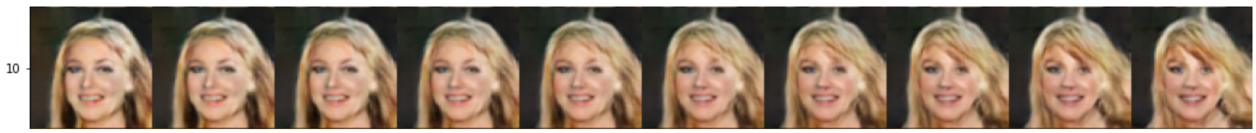


Hair color + natural face expression

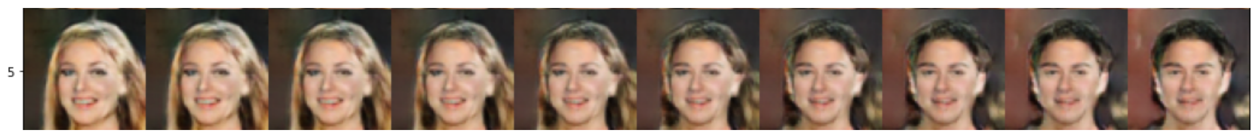
Smile + hair color



Ponytail + angle + eye open



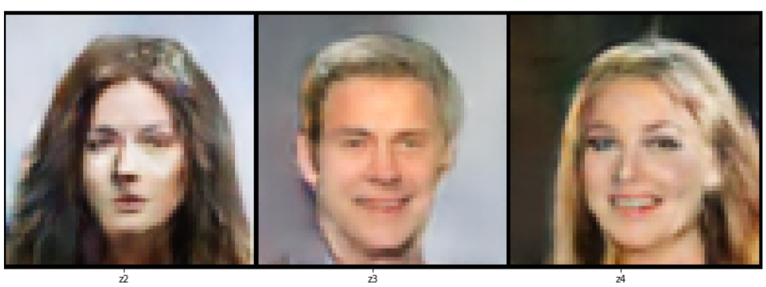
Gender + hair color



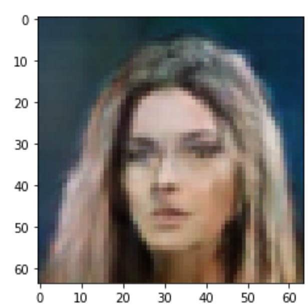
<u>Arithmetic operations in the latent space</u>

In some cases, we could use arithmetics in the latent space, and “add” features to the generated face.

In the following example, $z_4 - z_3$ it captures the feature of blonde hair, and is added to z_2



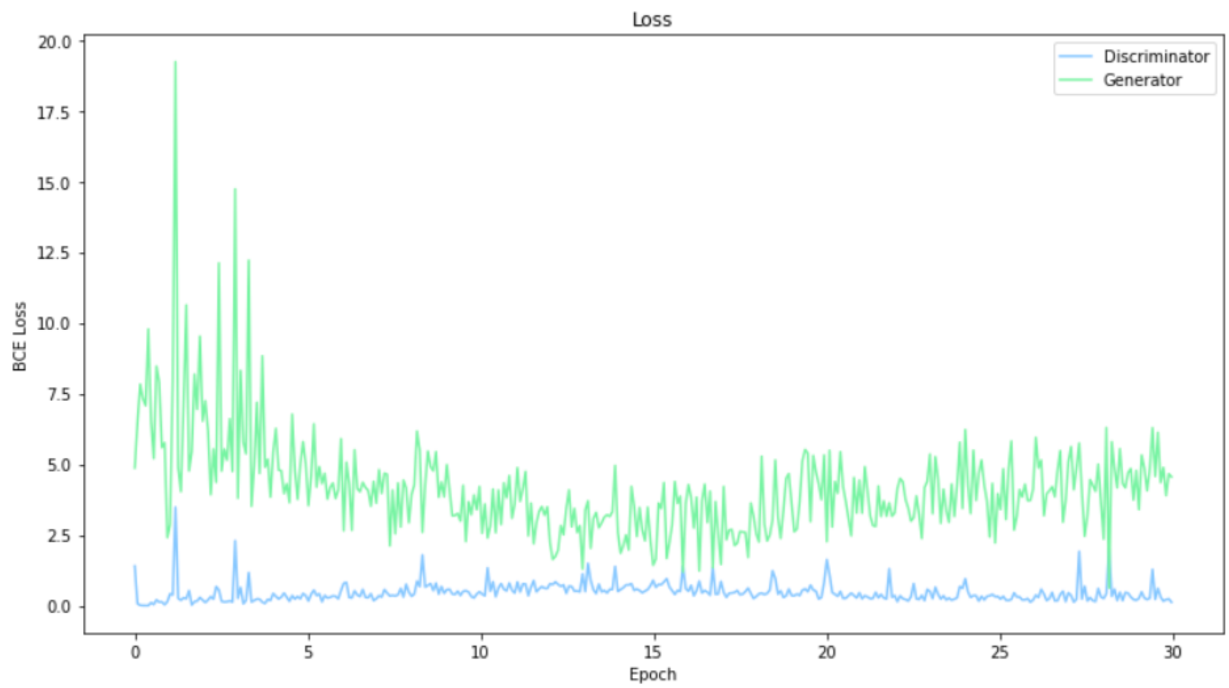
$$z_2 + z_4 - z_3$$



Continuos Latent Space

While exploring the continuos latent dimensions we observed that each dimension captures a very subtle feature, unlike the discrete dimension in which each feature was "responsible" for multiple outcomes.

Training convergence plots as a function of training time



Summary

a. Training a GAN is not a stable process and requires a lot of trial and error, there are general guidelines from recent papers, however, they still don't guarantee improved results (especially with 1 GPU and limited time).

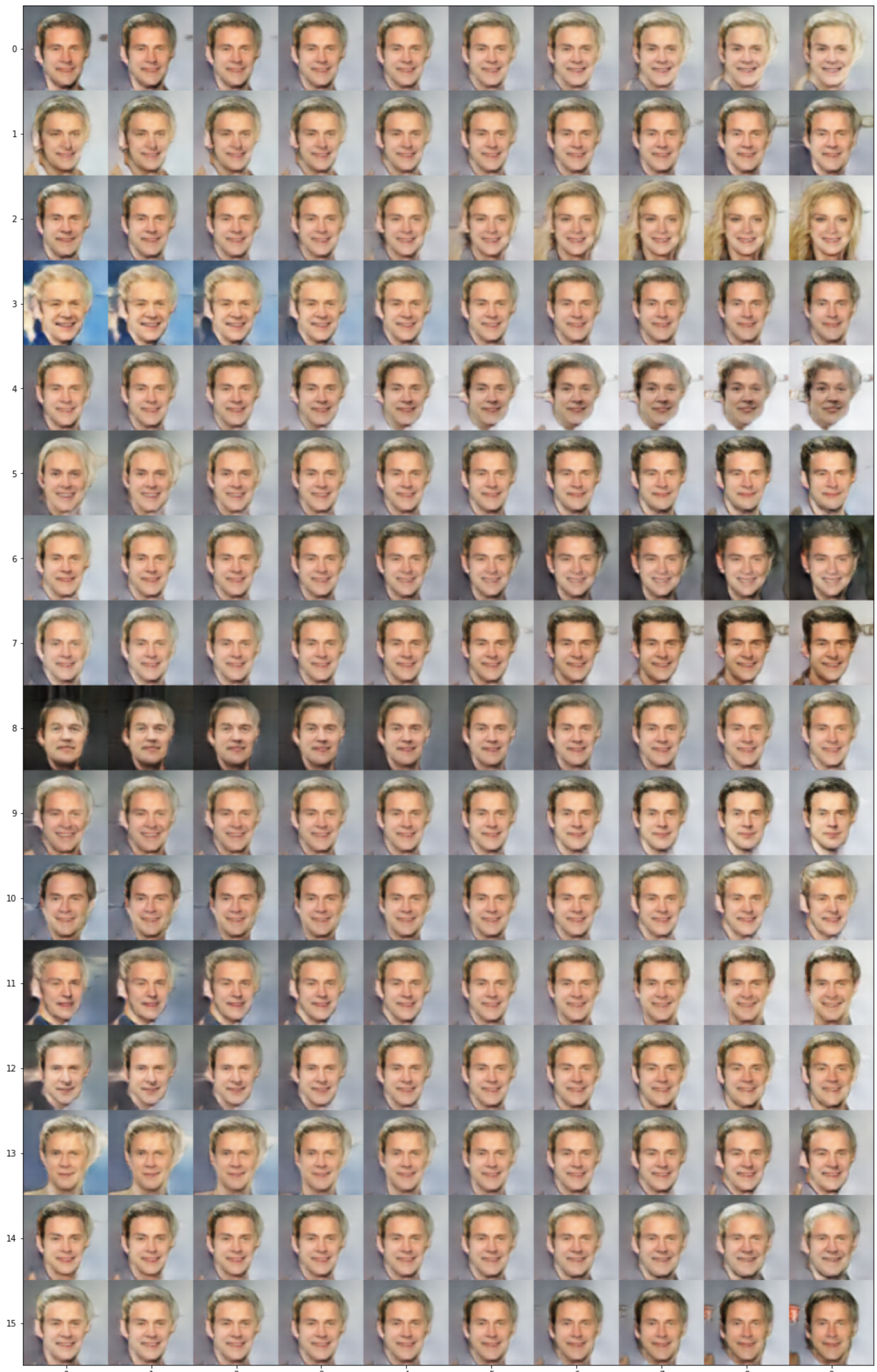
b. In vanilla DCGAN the features in the latent space are entangled, and each dimension is responsible for multiple features (considering mostly the discrete features).

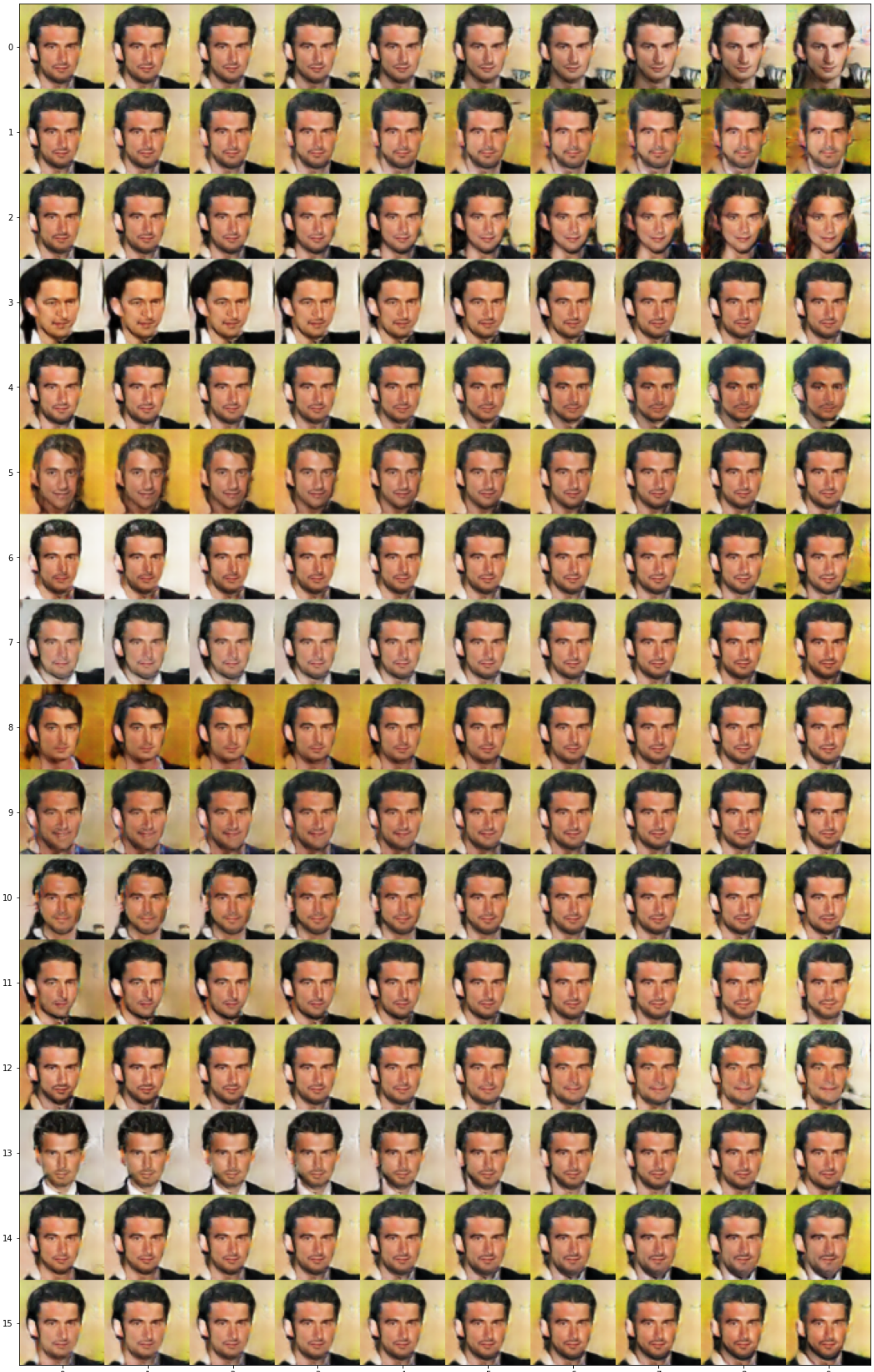
c. The task of the discriminator is much easier, and the generator tends to struggle.

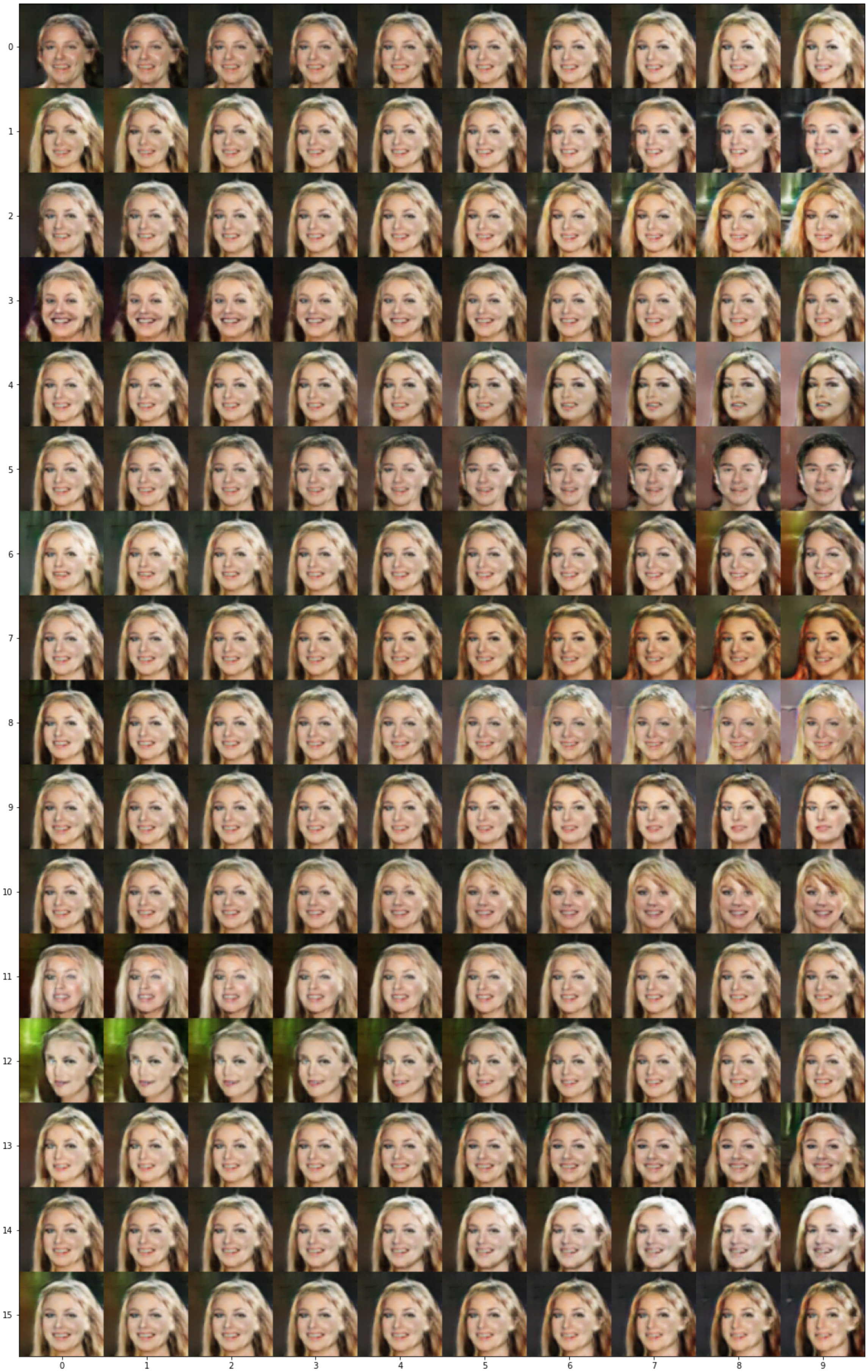
d. Following the previous observation, in DCGAN more epochs will not necessarily cause better results, the generator tends to diverge if the discriminator becomes very good in the discrimination.

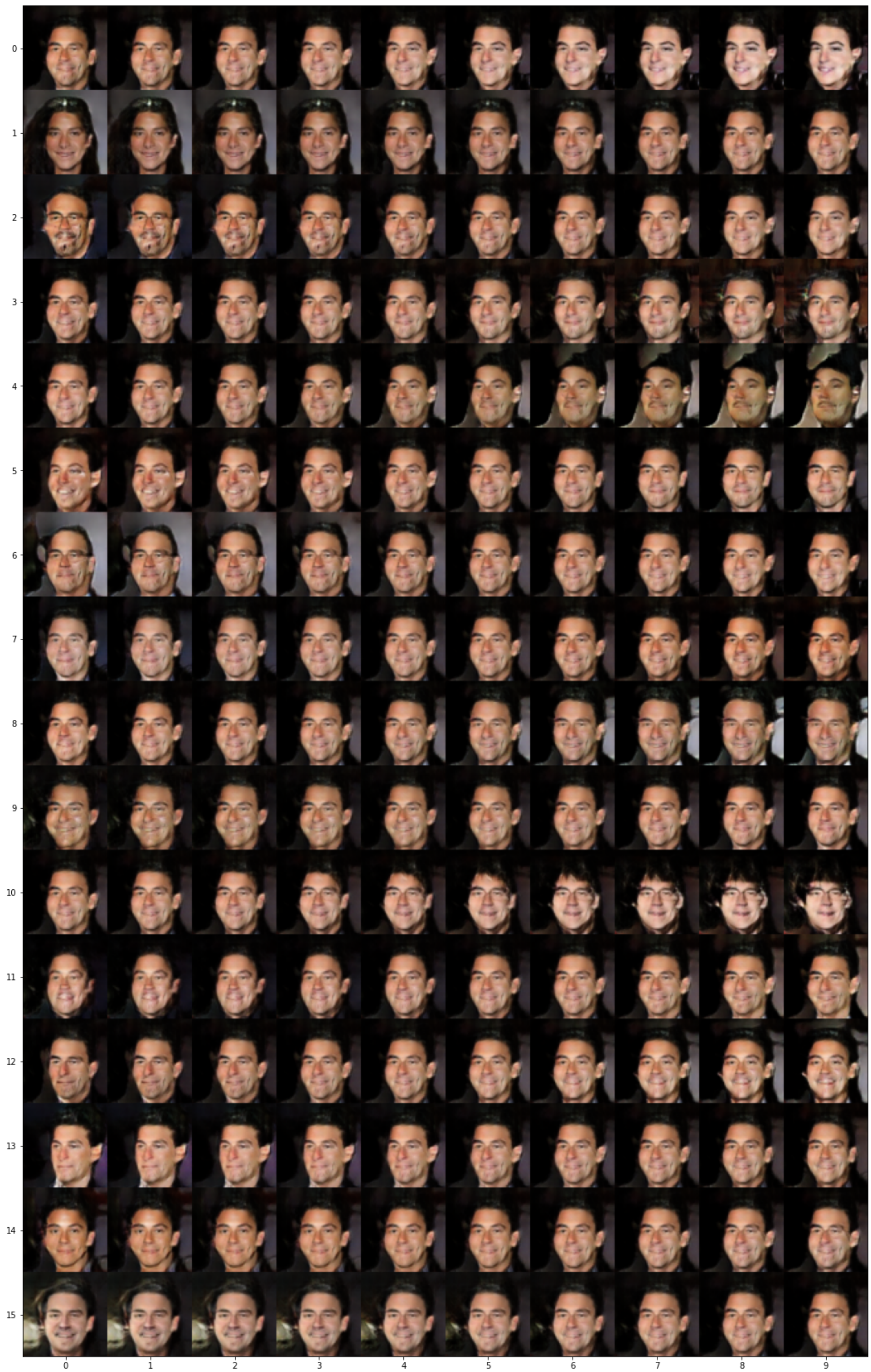
Appendix A-results

Examples of dimensional change (X-axis = value, Y-axis= dimension)


0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
0
1
2
3
4
5
6
7
8
9


0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
0
1
2
3
4
5
6
7
8
9


0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
0 1 2 3 4 5 6 7 8 9


0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
0
1
2
3
4
5
6
7
8
9