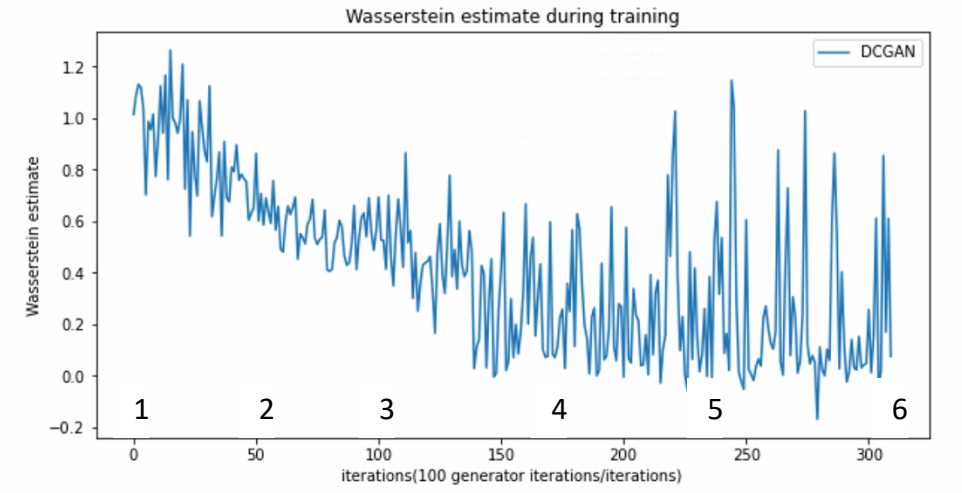
WGAN Implementation Results Bella Godiva 20214805

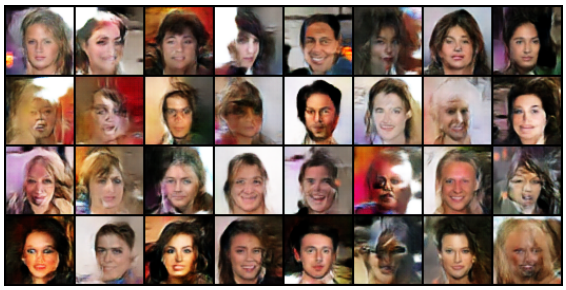
1. WGAN (Generator: DCGAN, Discriminator: DCGAN without last sigmoid layer)



1

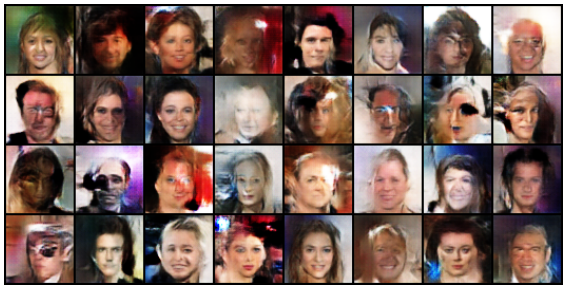
2

3



4

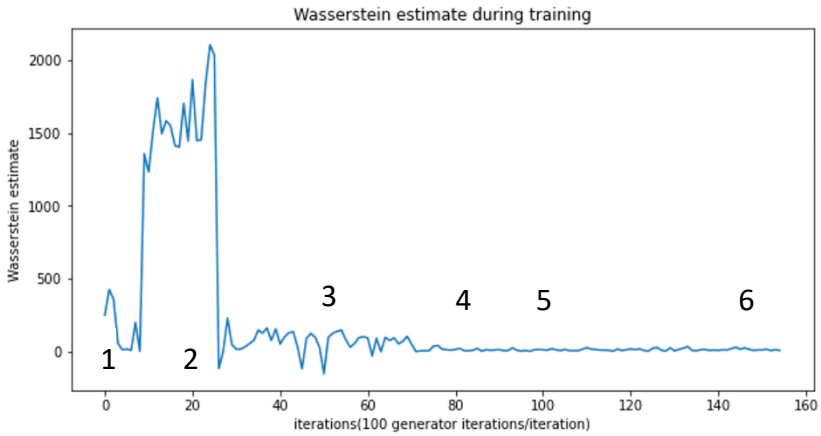
5

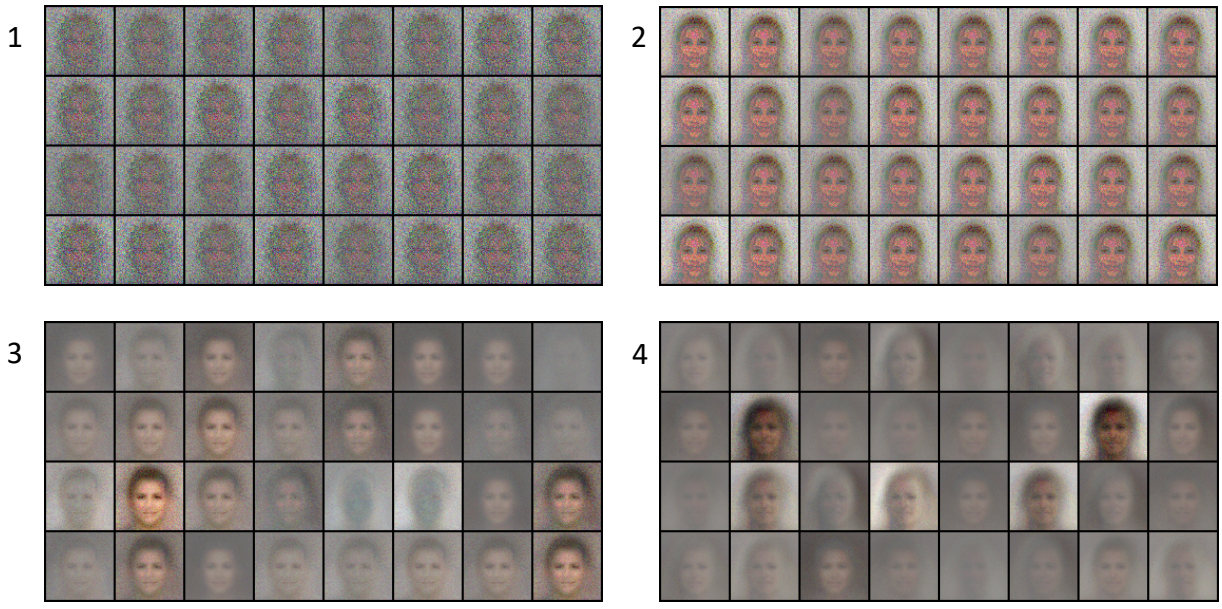


6

Wasserstein estimates is plotted from minus loss of discriminator. Just like the paper claim, the Wasserstein estimates have a decreasing trend throughout the iterations of generators. Although, in this implementation, it fluctuates and not strictly decreasing like in the original paper. **The quality of the generated faces also seems to correlates to the training loss**, which is a characteristic of WGAN that is not found in GAN**.** The time taken to train WGAN is relatively longer than training GAN so I could only train for 10 epochs which is around 30k generator iterations whereas the paper train for 600k generator iterations. Maybe this is one of the reasons why their training loss graph looks smoother than the training loss I obtained as it is more zoomed out.

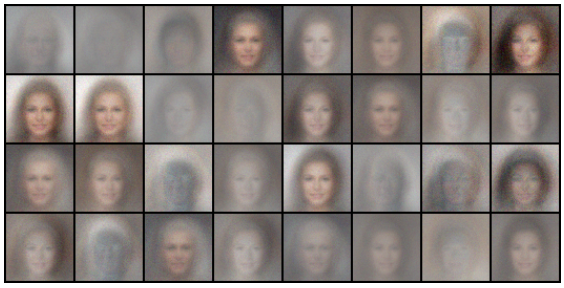
2. WGAN (Generator: MLP_512, Discriminator: MLP_512)





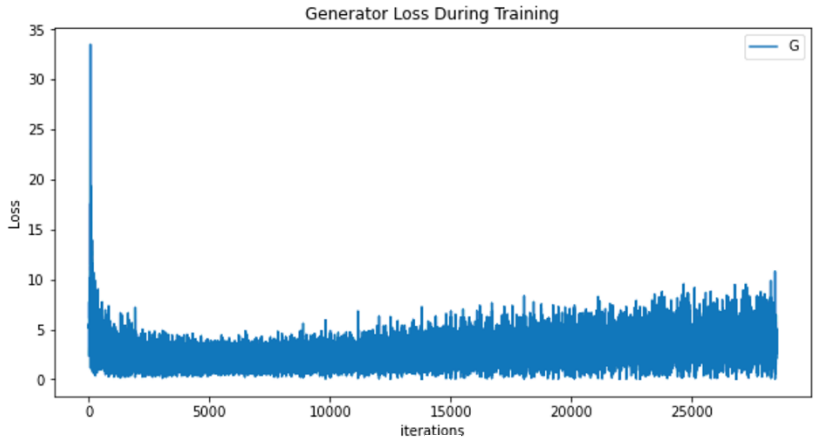
5

6



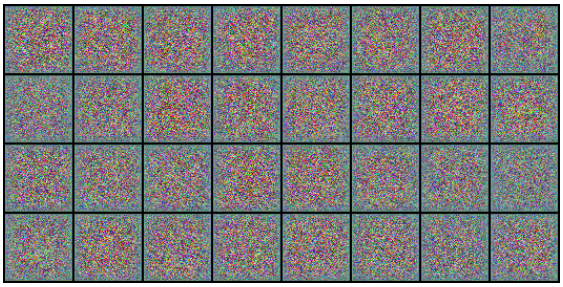
WGAN using MLP with 4 hidden layers with 512 units at each layer. The training loss also shows correlation with the quality of the generated images. (4), (5), (6) look similar as they all have similar loss.

3. GAN (Generator: DCGAN, Discriminator: DCGAN)



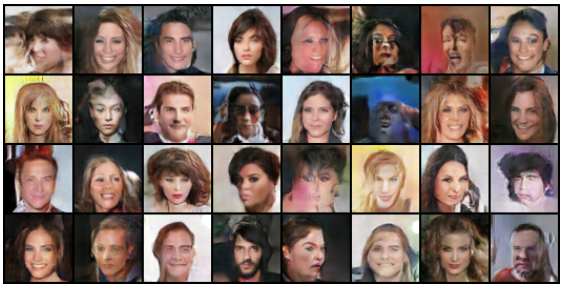
1 2 3 4 5 6

1

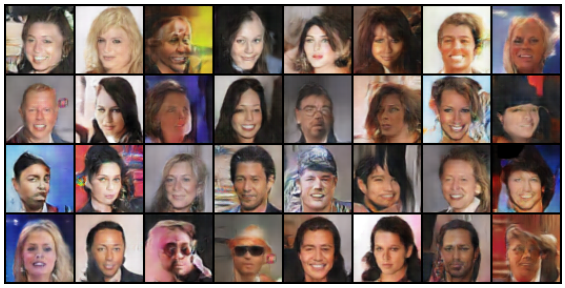


2

3

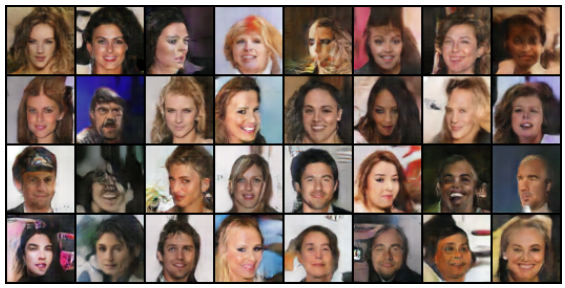


4



5

6



Unlike in WGAN training above, GAN training loss do not give any information about the quality of generated images. Training GAN is faster than training WGAN and the results of generating images based on celebA dataset is comparable.