CSC421 Programming Assignment 4

Yubo Wang 1002138377

Part 1: Deep Convolutional GAN

Implement the Discriminator of the DCGAN

1.

We can use the following formula to compute the padding size:

$O = \left\lfloor \frac{I-K+2P}{S} + 1 \right\rfloor$ were I is input width, O is output width, P is padding

$O - 1 = \frac{I-K+2P}{S}$

$(O-1)S = I - K + 2P$

$2P = (O-1)S - I + K$

$P = \frac{(O-1)S-I+K}{2}$

From the question, we know that $S = 2, K = 5$

$I = 2O$ for first three layers

$P = \frac{2(O-1)-2O+5}{2} = \frac{2O-2-2O+5}{2} = 1.5 \approx 2$ to include the floor in the original equation

$I = 4O$ for last layer

$P = \frac{5-2-2}{2} = 0.5 \approx 1$

2.

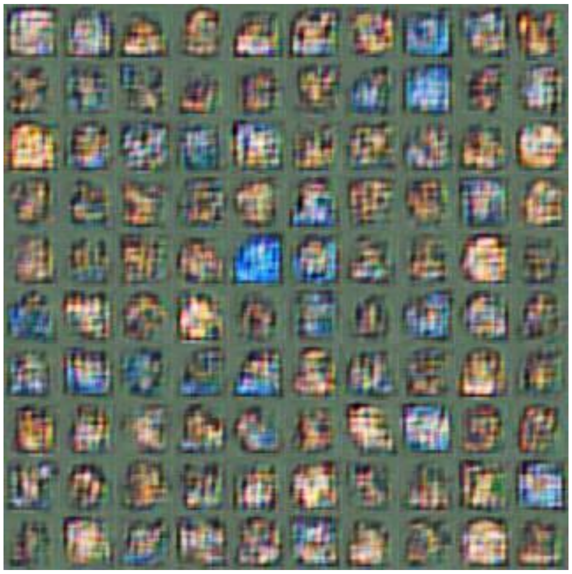
In code file

Generator: In code

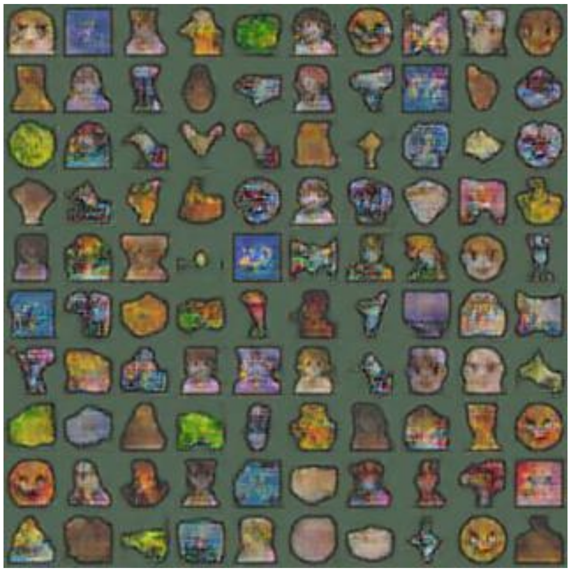
Training Loop: In code

Experiment:

1.



Iteration 200



Iteration 5000

From these two images, we can see that at iteration 200, the generated emoji is like mosaic. Both shape and color are not accurate. At iteration 5000, although the generated emoji still does not have vivid details, the color and shape are much more accurate compared with previous iterations.

Part 2: CycleGAN

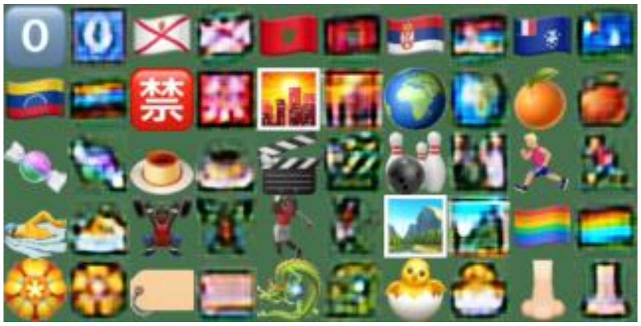
Generator: In code

CycleGAN Training Loop: In code

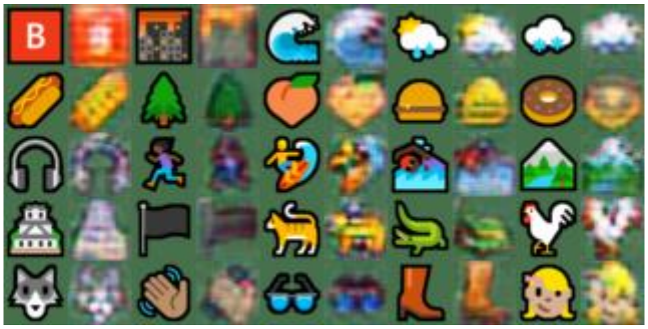
Cycle Consistency: In code

CycleGAN Experiments:

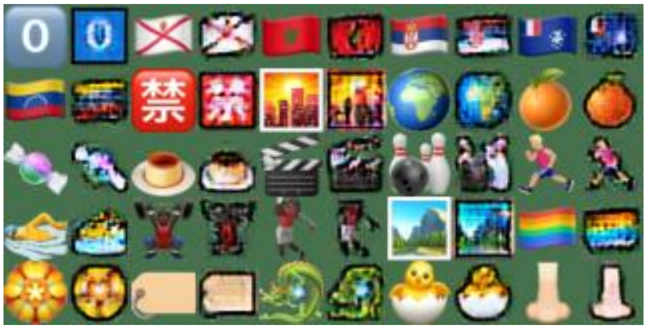
1.



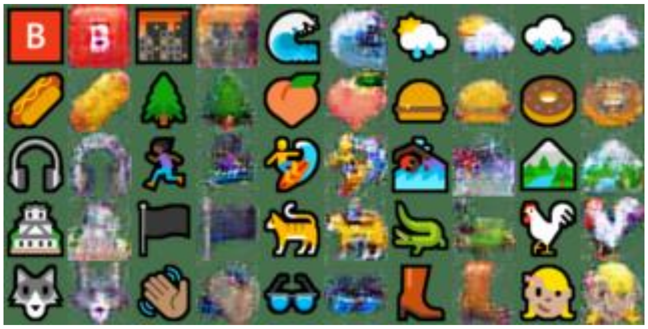
Iteration 200 X to Y



Iteration 200 Y to X



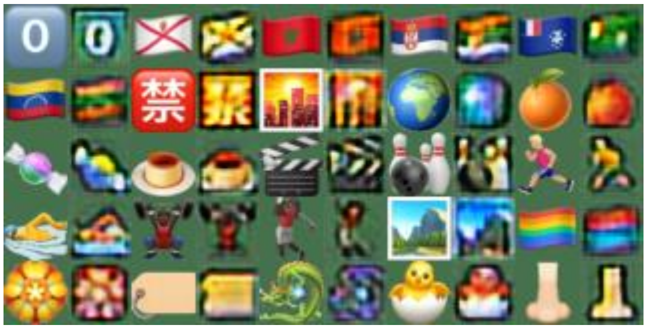
Iteration 5000 X to Y



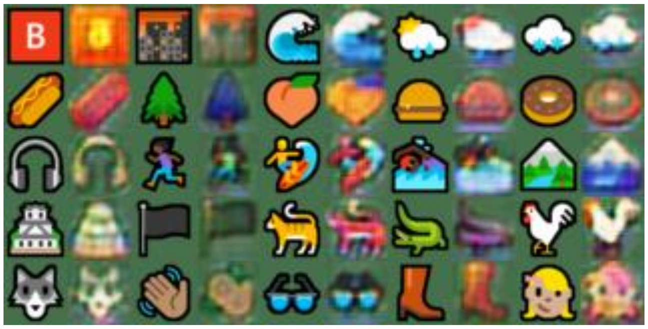
Iteration 5000 Y to X

2.

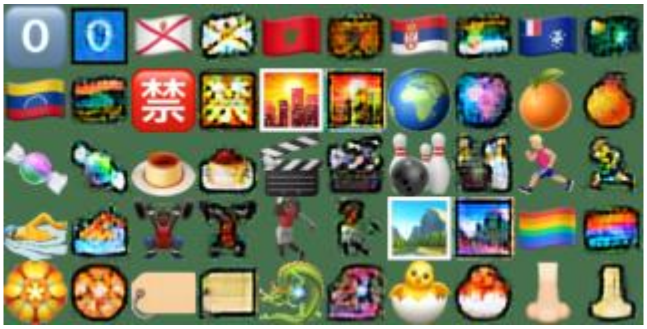
Change random seed from 4 to 30



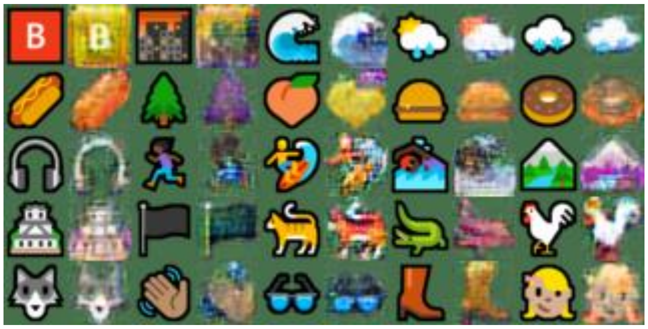
Iteration 200 X to Y



Iteration 200 Y to X



Iteration 5000 X to Y

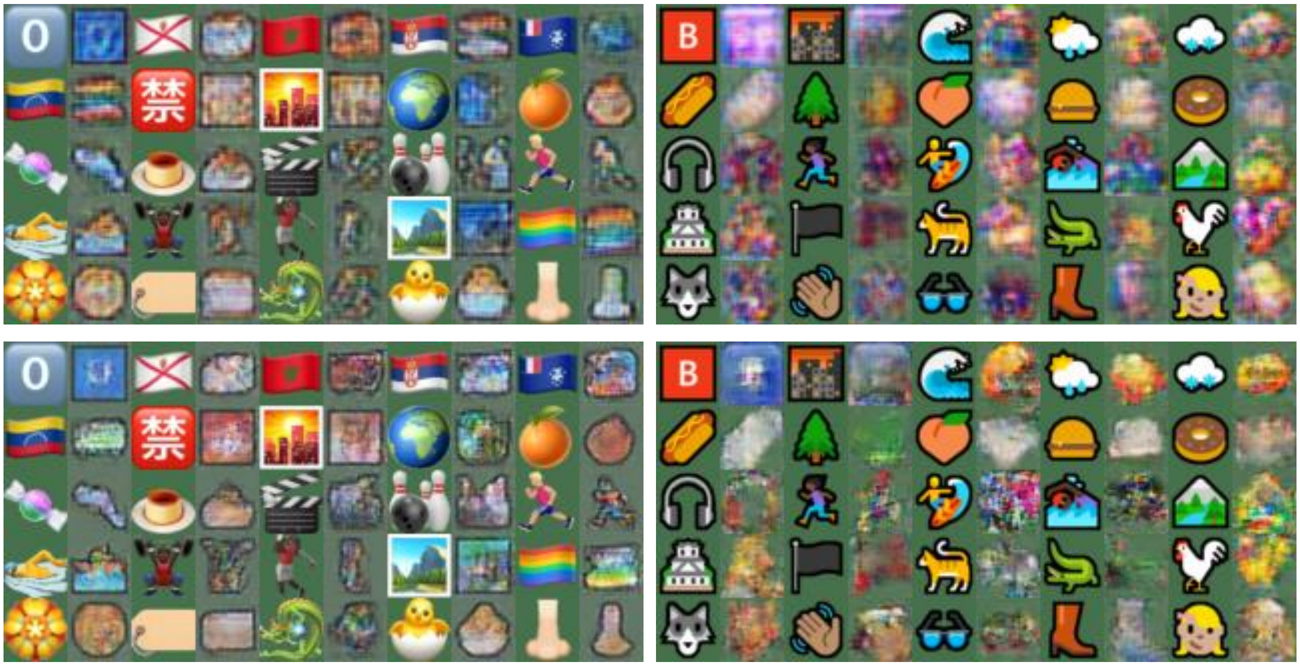


Iteration 5000 Y to X

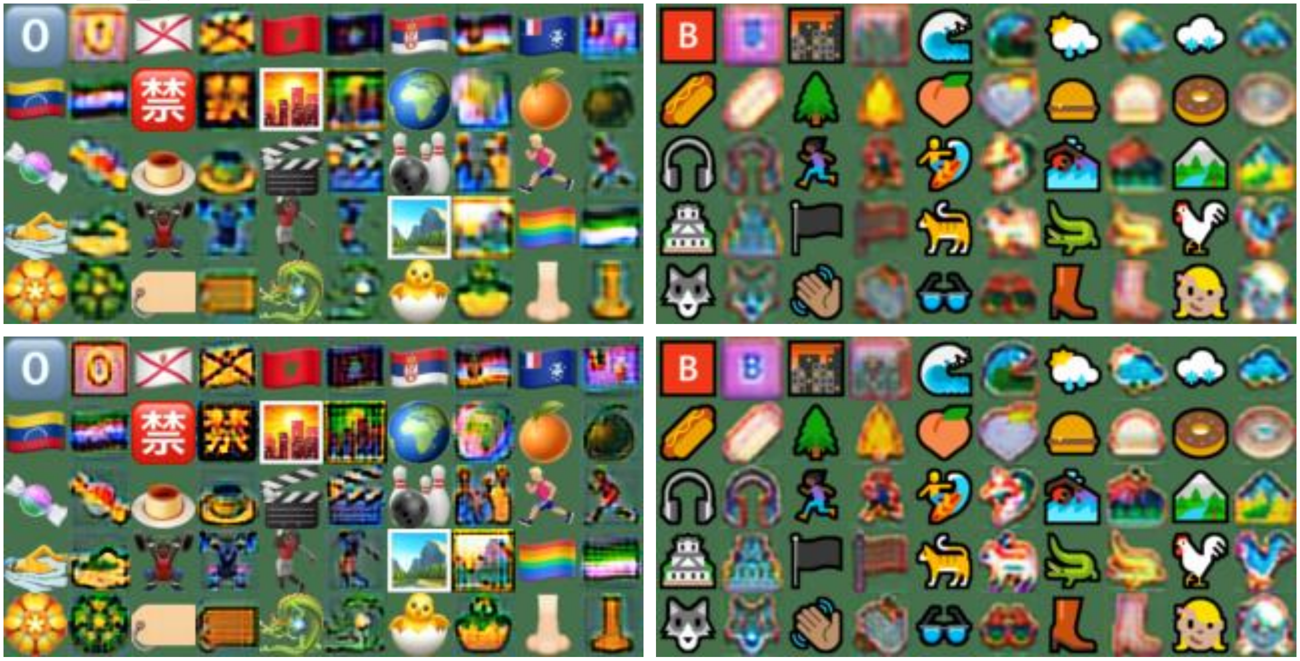
The color of generated emoji is not same as color of previous configuration since random seed just randomly initialized the initial weight. Therefore, we need to pick an appropriate seed to have the best result.

3.

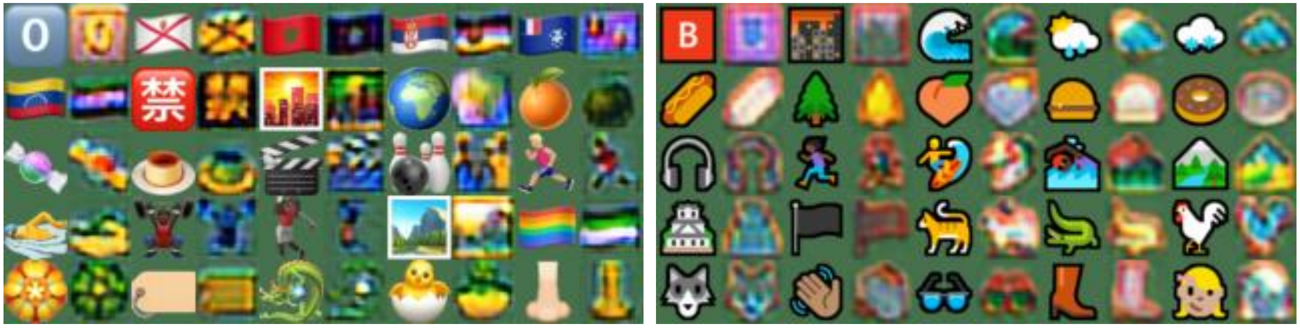
For lambda_cycle = 0

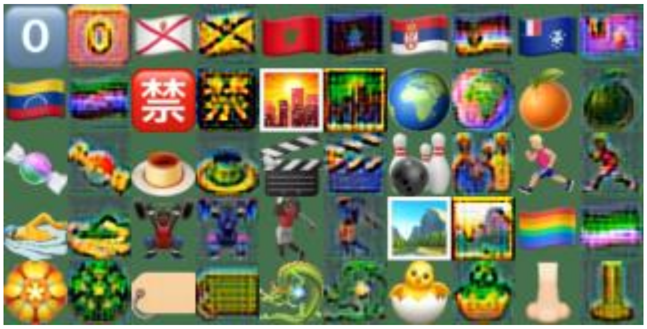
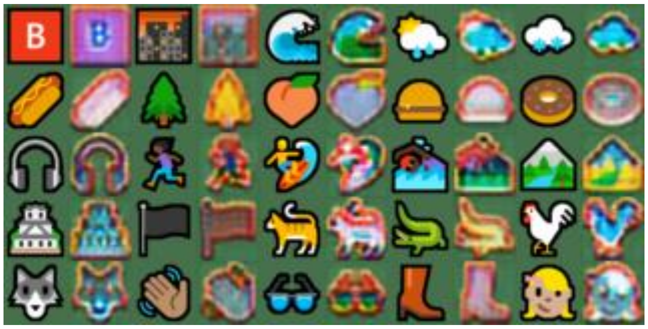


For lambda_cycle = 0.1



For lambda_cycle = 0.5



By comparing these 3 new cases and the original one, we can conclude that outer shape, detail and color are obviously better with cycle consistency loss, but when the lambda goes too high, the color of the generated emoji looks more saturated, which leads to worth result.

The reason of this change is that the cycle consistency loss can keep the detail feature of the original emoji. With appropriate lambda, we can get a more similar and accurate generated emoji compared with model without cycle consistency loss or model with high cycle consistency loss.