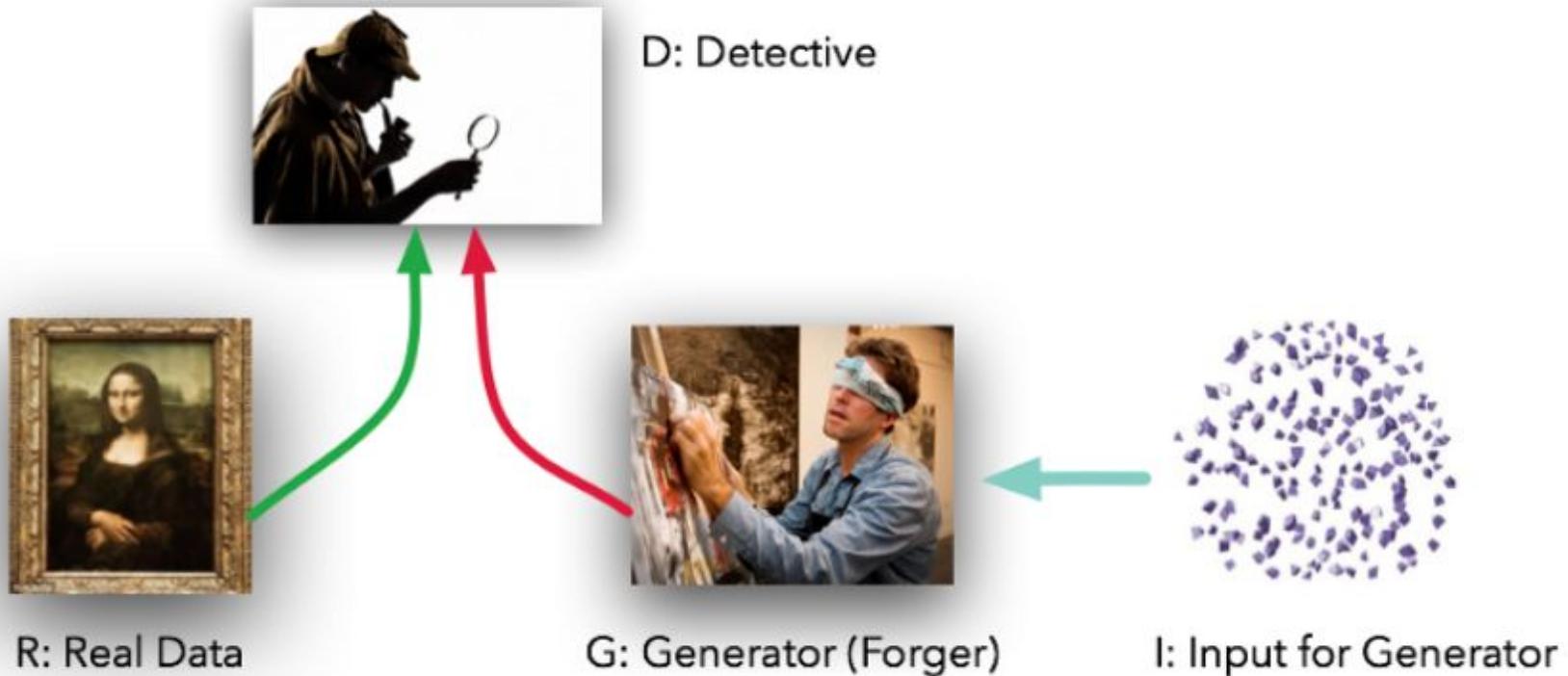


Album covers with GAN

Nikhil Nadig & Óttar Guðmundsson

Generative Adversarial Network



Original project idea



Image 4: A damaged version of Fisher Price – Little people. The damaged clip is the middle one.



Image 2: A damaged version of Fisher Price – Little people. The damaged clip is the middle one.



Image 5: The correct version of Fisher Price – Little people



Image 3: The complete version of Fisher Price – Little people



→



Image 6: A replaced image using the network.

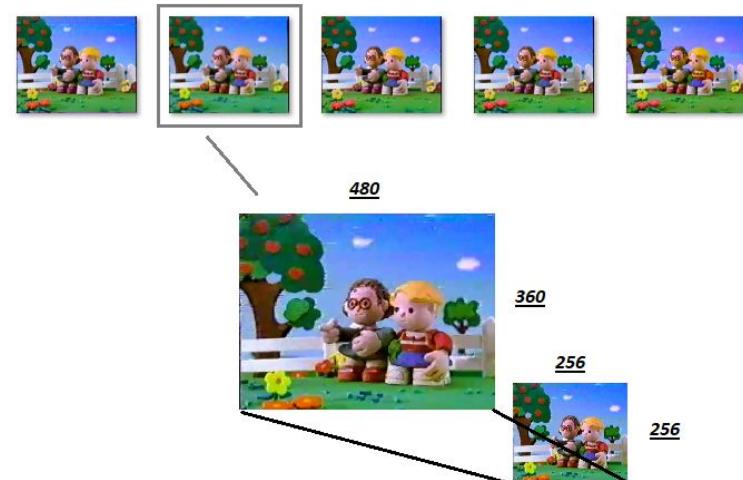
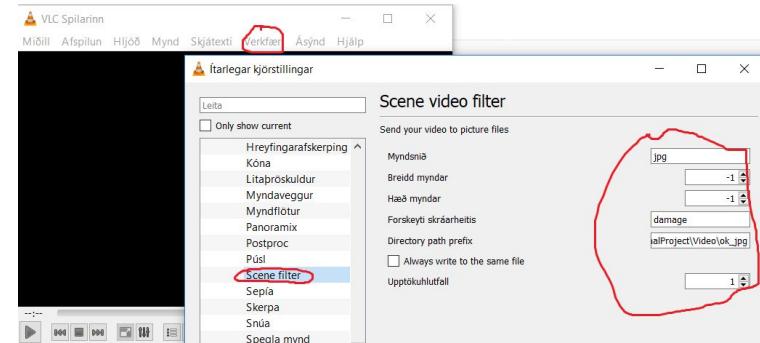
Data gathered and prepared

Download two videos, damaged and undamaged

VLC to extract images from video

Select a sequence to try it on

Resize images to 256x256



The code

Load data to tensors

Add noise to images

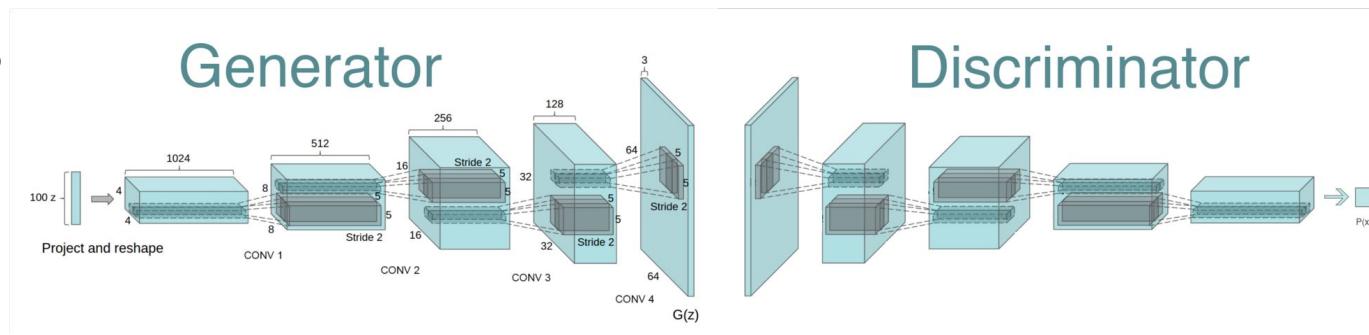
Setup Generator

Setup Discriminator

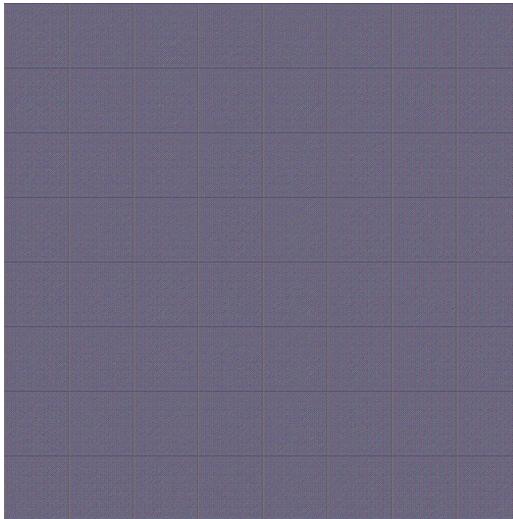
Create a batch job of 5x5 images

Train for N many rounds with values from paper

Save images and checkpoints at epoch 100 and 500



First training 15 CPU hours (2017.11.28)



Epoch 0



Epoch 649

Something is happening but it's pretty slow

Floydhub to the rescue!

50 GPU hours for \$33



```
(C:\ProgramData\Anaconda3) C:\Users\ottarg\Documents\SWED_NEURAL\FinalProject>floyd login
```

```
New version of CLI (0.10.26) is now available. To upgrade run:  
pip install -U floyd-cli
```

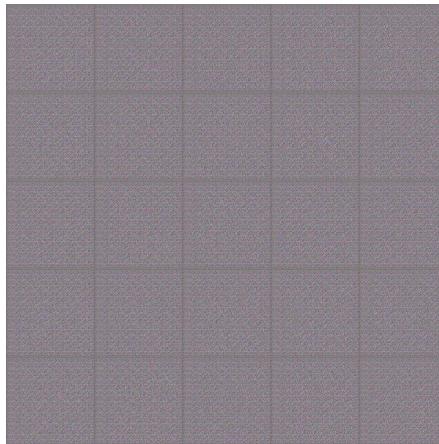
```
Authentication token page will now open in your browser. Continue? [Y/n]: y  
Please copy and paste the authentication token.  
This is an invisible field. Paste token and press ENTER:  
Login Successful
```

```
(C:\ProgramData\Anaconda3) C:\Users\ottarg\Documents\SWED_NEURAL\FinalProject>floyd init gan_album
```

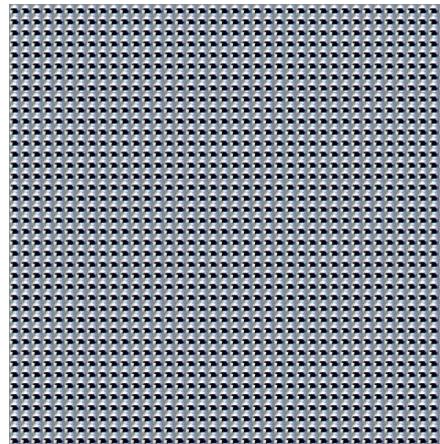
```
New version of CLI (0.10.26) is now available. To upgrade run:  
pip install -U floyd-cli
```

```
Project "gan_album" initialized in current directory
```

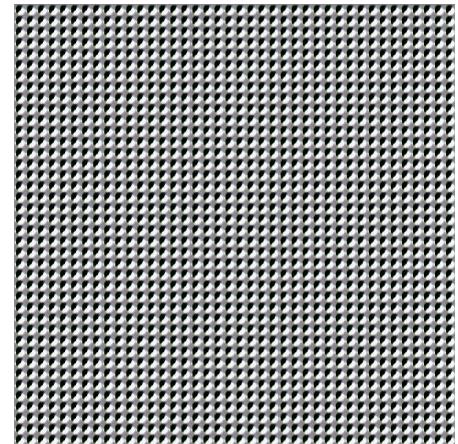
Second training 8 GPU hours (2017.12.1)



Epoch 0

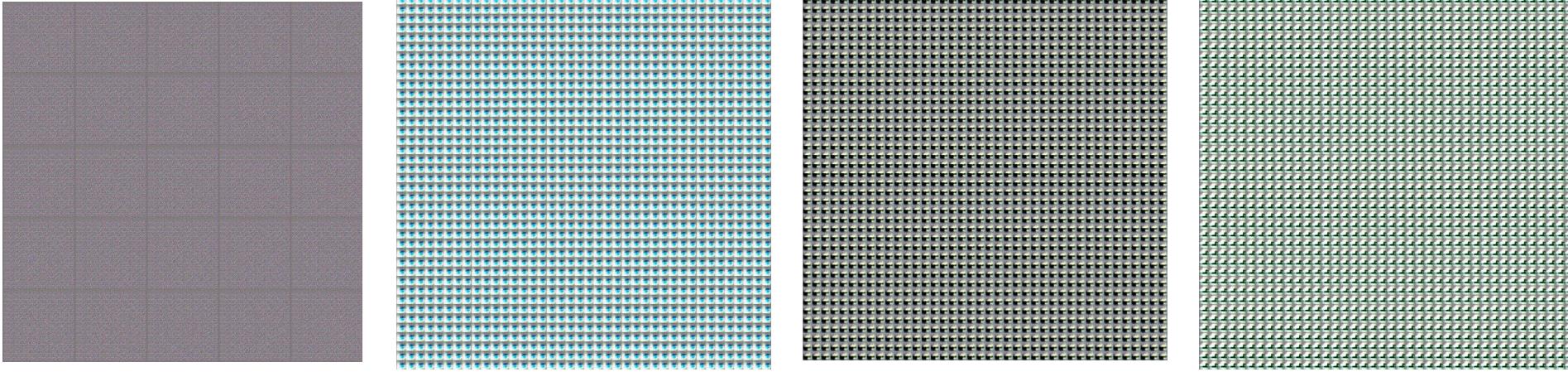


Epoch 2100



Epoch 4900

This is not what we hoped for

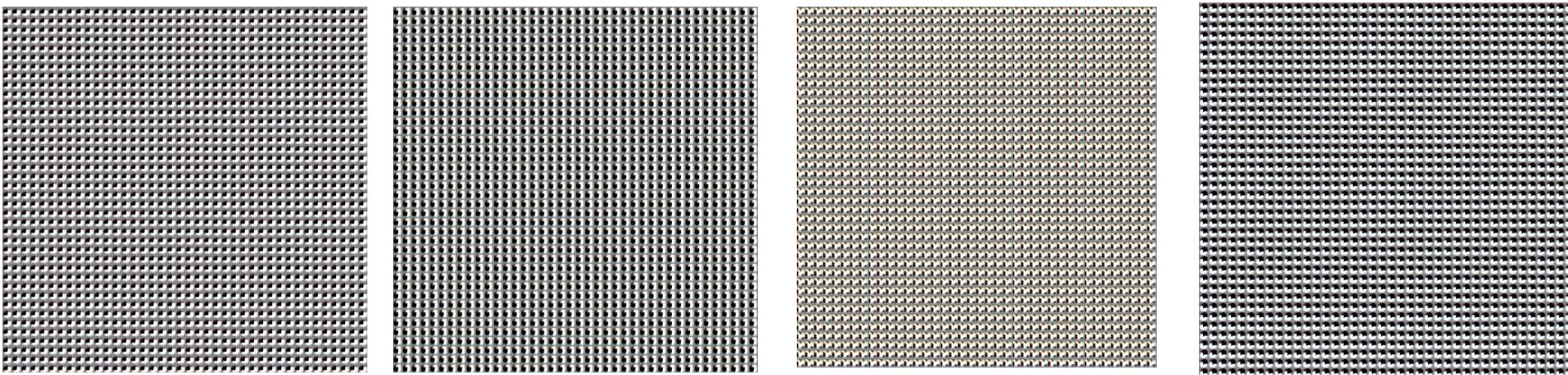


Epoch 0

Epoch 500

Epoch 1000

Epoch 1500



Epoch 2000

Epoch 2500

Epoch 3000

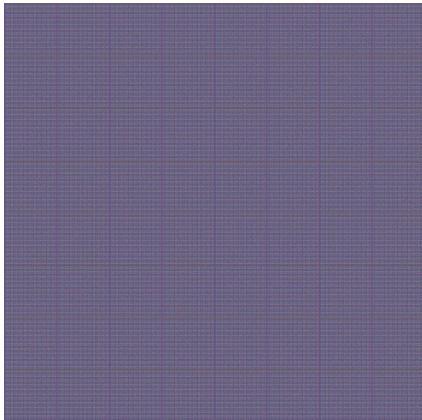
Epoch 3500

Tweak

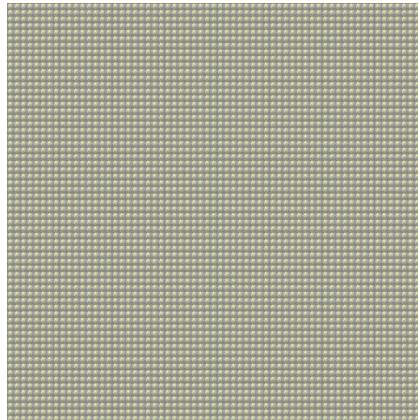
- Contrast range 0.95 to 1.05
- Noise input range -0.05 to 0.05
- Learning rate 0.004
- Learning rounds 2x
- Clip values range -0.005 to 0.005



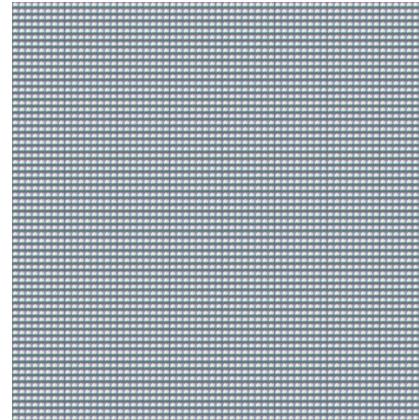
Third training 7 GPU hours (2017.12.4)



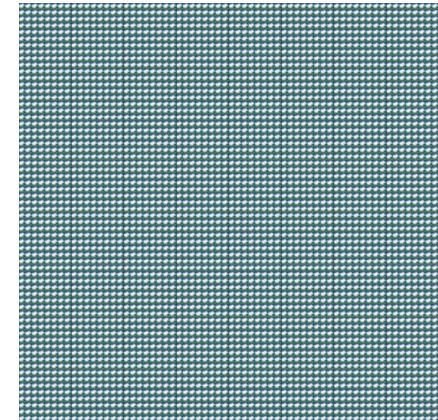
Epoch 0



Epoch 200



Epoch 400



Epoch 600...

This is not getting any better

What is happening?

Training epoch: [0],Discriminator Loss:-138.006332,Generator Loss:43.682674

Training epoch: [200],Discriminator Loss:-344.098785,Generator Loss:151.174179

Training epoch: [400],Discriminator Loss:-344.139771,Generator Loss:151.173889

Training epoch: [600],Discriminator Loss:-344.132874,Generator Loss:151.173401

Training epoch: [800],Discriminator Loss:-343.962830,Generator Loss:151.134995

Training epoch: [1000],Discriminator Loss:-344.140411,Generator Loss:151.179016

Training epoch: [1200],Discriminator Loss:-344.054413,Generator Loss:151.081757

...

Problem

All of the images look the same so the discriminator gets really good at knowing what is fake and what isn't.



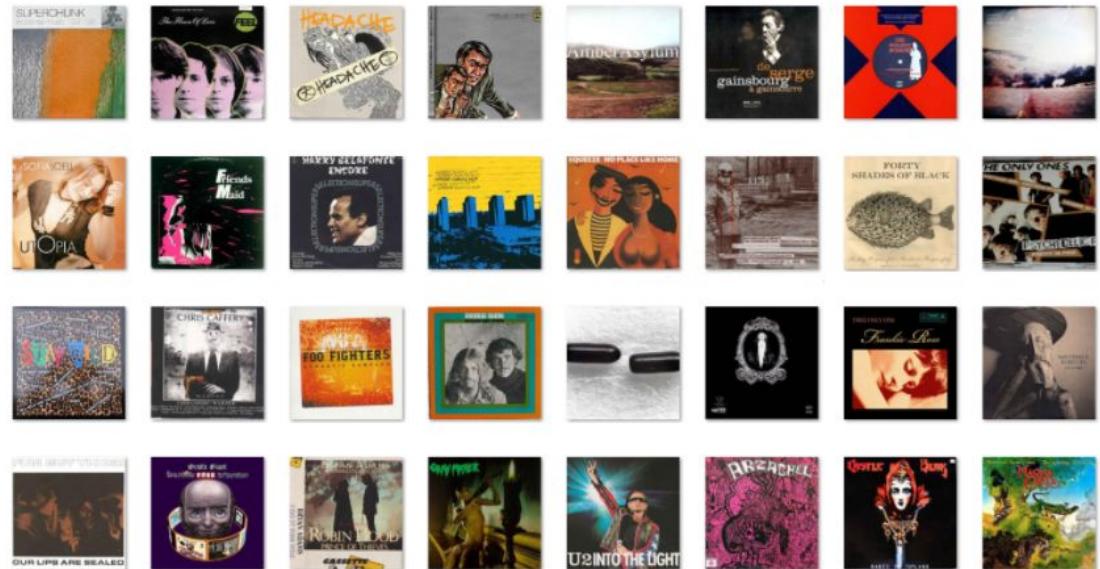
Thus the generator never gets better.

Deliver something with the network - change the idea

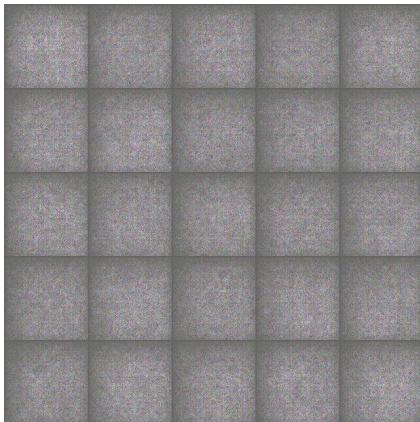
Create album covers

A lot of variety

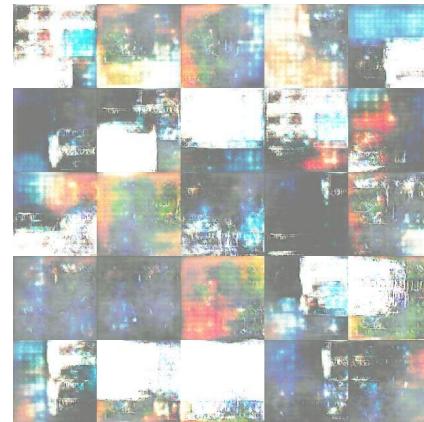
Different colors
and shapes



Fourth training 24 GPU hours (2017.12.5)



Epoch 0



Epoch 200



Epoch 400



Epoch 600

This **is** getting any better

What is happening?

Training epoch: [0],Discriminator Loss:-247.667419,Generator Loss:135.535370

Training epoch: [200],Discriminator Loss:-362.098877,Generator Loss:186.704910

Training epoch: [400],Discriminator Loss:-227.098892,Generator Loss:184.3513499

Training epoch: [600],Discriminator Loss:-169.517609,Generator Loss:-26.418308

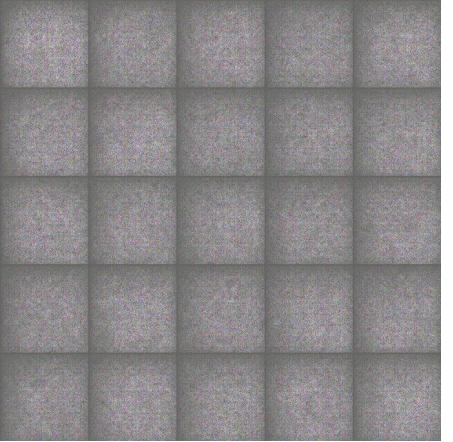
Training epoch: [800],Discriminator Loss:-195.191345,Generator Loss:-44.647148

Training epoch: [1000],Discriminator Loss:-273.622559,Generator Loss:179.810165

Training epoch: [1200],Discriminator Loss:-326.898071,Generator Loss:177.027695

...

Training epoch: [3000],Discriminator Loss:-368.880188,Generator Loss:145.526917



Epoch 0



Epoch 500



Epoch 1000



Epoch 1500



Epoch 2000



Epoch 2500



Epoch 3000



Epoch 3500

Pretty pictures from epoch 500 - 1500





imagemode12000.jpg



imagemode13000.jpg



imagemode14000.jpg

```
saver = tf.train.Saver(variables_to_restore)
ckpt = tf.train.latest_checkpoint(generatedPathVersion)
print("latest checkpoint")
print(ckpt)
saver.restore(sess, ckpt)

coord = tf.train.Coordinator()
threads = tf.train.start_queue_runners(sess=sess, coord=coord)

# Create a random noise
randomNoise = np.random.uniform(-0.5, 0.5, size=[BATCH_SIZE, insertDimention]).astype(np.float32)
generatedImage = sess.run(fake_image, feed_dict={random_input: randomNoise, is_train: False})
save_images(generatedImage, sampleMatrix ,generatedPath + '/image' + ckpt.split('/')[-1] + '.jpg')
```

OUTCOME

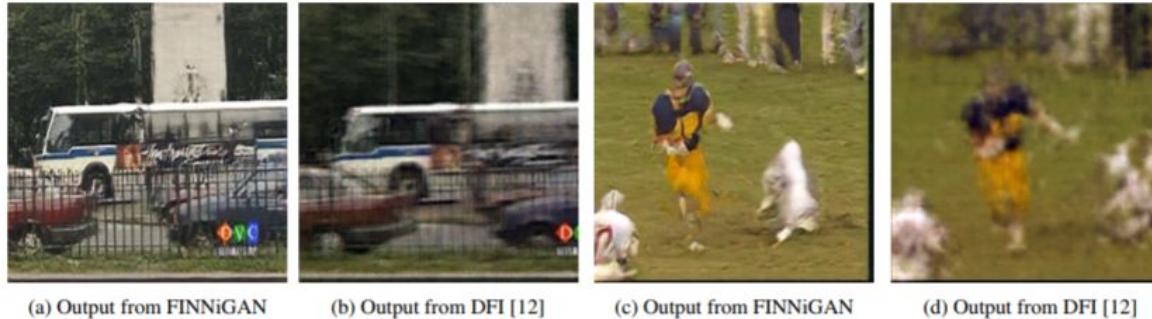
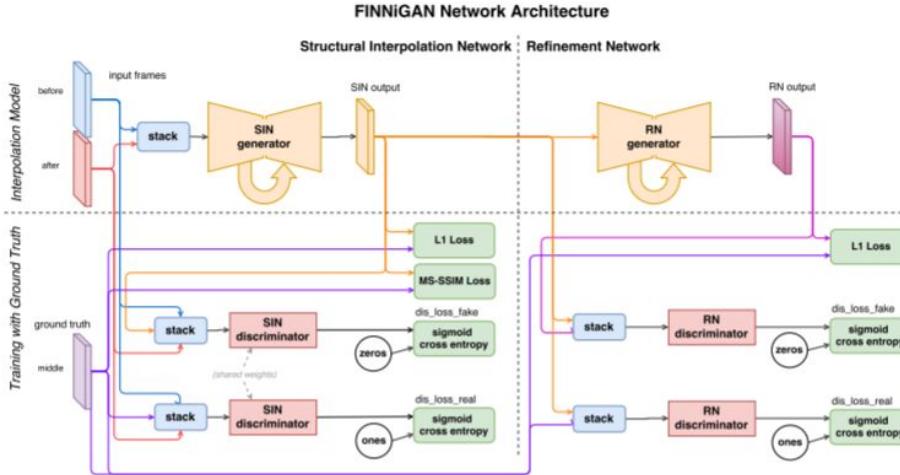


Learned some Tensorflow and Numpy

Knowledge of external computation from Floydhub

Learned about the limitations GAN

Incredible album covers



Note: solution to the problem solved at Stanford

Bonus slide - First try using style transfer methods



→



→

