Ganimate : CycleGAN for Human to Cartoon Translation

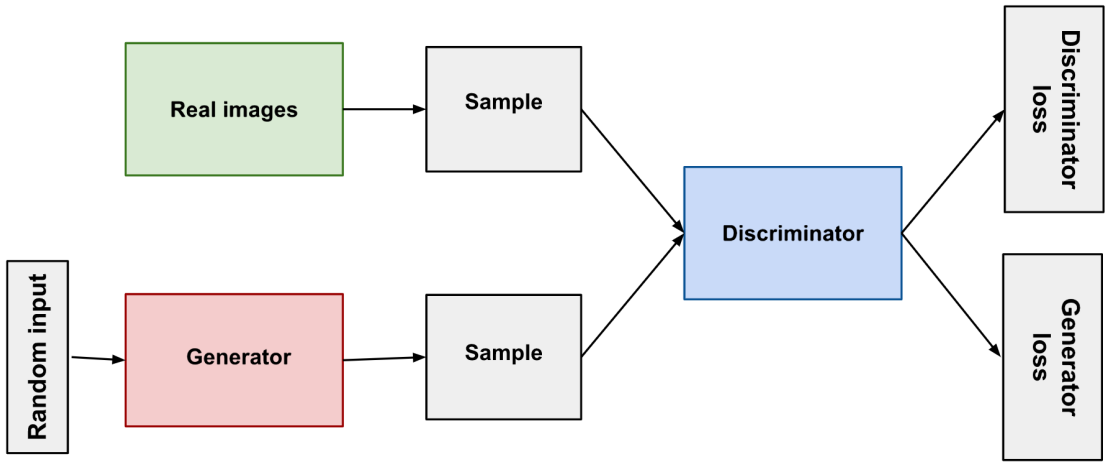
Jack Smith, Abdulhamed Alothaimen

Univeristy of Michigan

Winter 2021

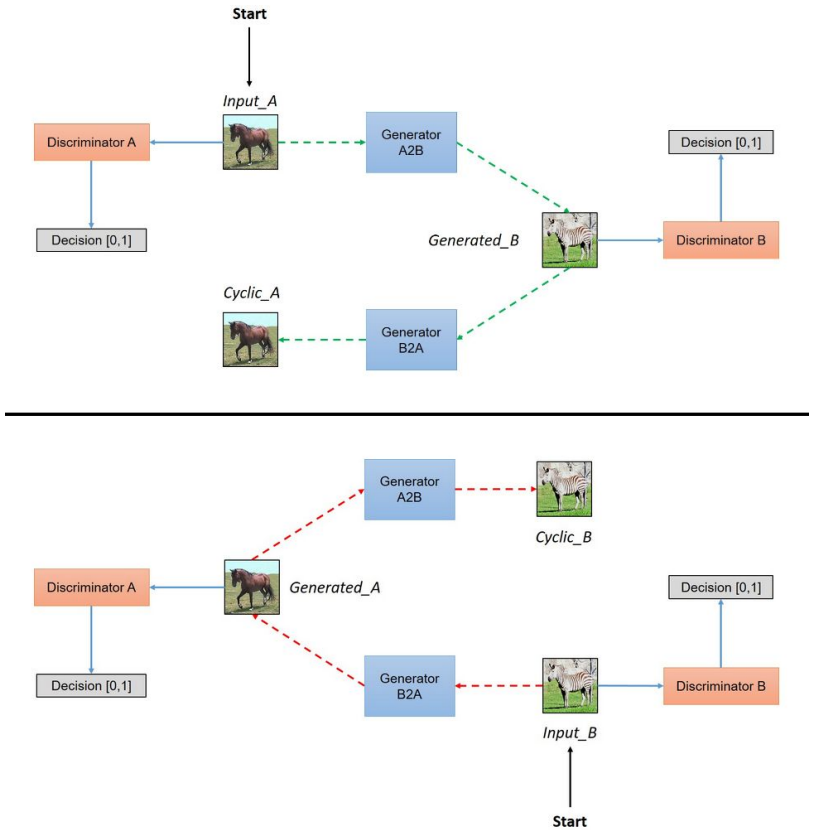
Introduction to GANs

- Traditional GANs consist of a generator and discriminator.
  - Generators take noise as input and are trained to produce an image of something you train it on (ie. human faces)
  - The discriminator takes an image as input and determines whether it is produced by the generator or is a real image
- **Error of GAN:** $E(G, D) = \frac{1}{2}\,\mathbb{E}[1 - D(x)] + \frac{1}{2}\,\mathbb{E}[D(G(x))]$
- **Objective:** $\max_G \min_D [E(G, D)]$



Introduction to CycleGAN

- Aims to solve the problem of unpaired image-to-image translation [1].
- Datasets simply consist of images from both domains.
- Architecture:



Introduction to CycleGAN - Loss Function

- Combines the adversarial loss with a cyclic loss.

$$Loss_{adv}(G, D_B, A) = \frac{1}{m}\sum_{i=1}^{m}(1 - D_B(G_{A2B}(a_i)))^2$$

$$Loss_{adv}(F, D_A, B) = \frac{1}{m}\sum_{i=1}^{m}(1 - D_A(G_{B2A}(b_i)))^2$$

$$Loss_{cyc}(G_{A2B}, G_{B2A}, A, B) = \frac{1}{m}\sum_{i=1}^{m}[G_{B2A}(G_{A2B}(a_i)) - a_i] + [G_{A2B}(G_{B2A}(b_i)) - b_i]$$

$$Loss_{full} = Loss_{adv} + \lambda Loss_{cyc}$$

Our Project Aim

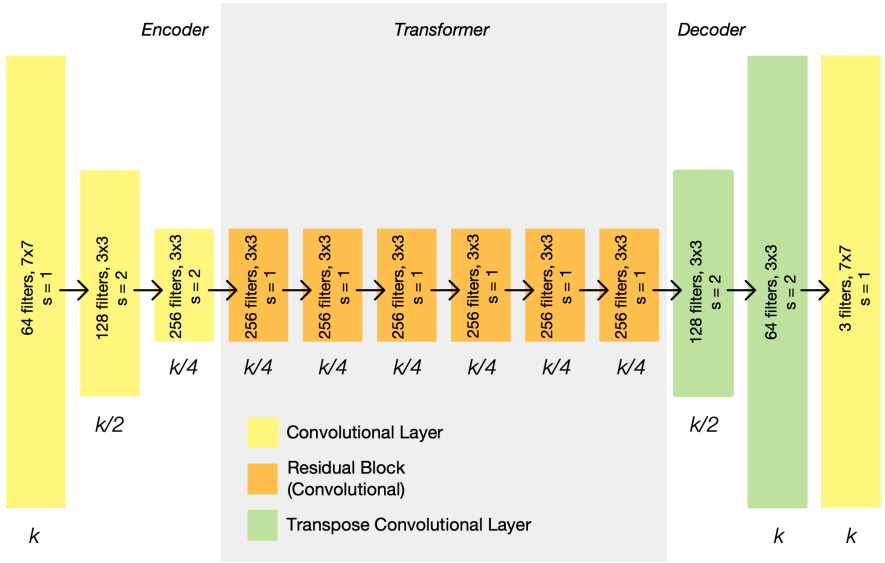
- Train CycleGAN to map between human faces and cartoon faces.
- Compare results to other deep learning methods used to achieve similar goals, such as XGAN.

Dataset

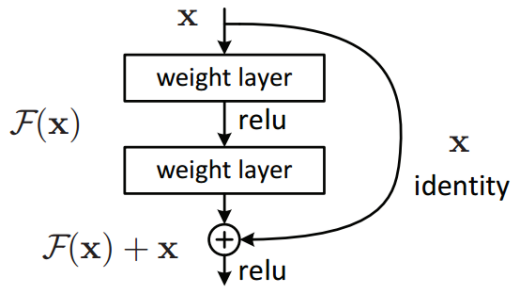
- Data from the domain of human faces consisted of 2000 images from the CelebA dataset.
- Cartoon data was 2000 images from the Bitmoji dataset on Kaggle.
- Test and Train splits:
  - 1900 images in train set
  - 100 images in test set
- We transformed images to be 128x128 prior to training in order to scale to our computing power.

Generator Implementation

- Following the paper, the generator architecture was implemented:

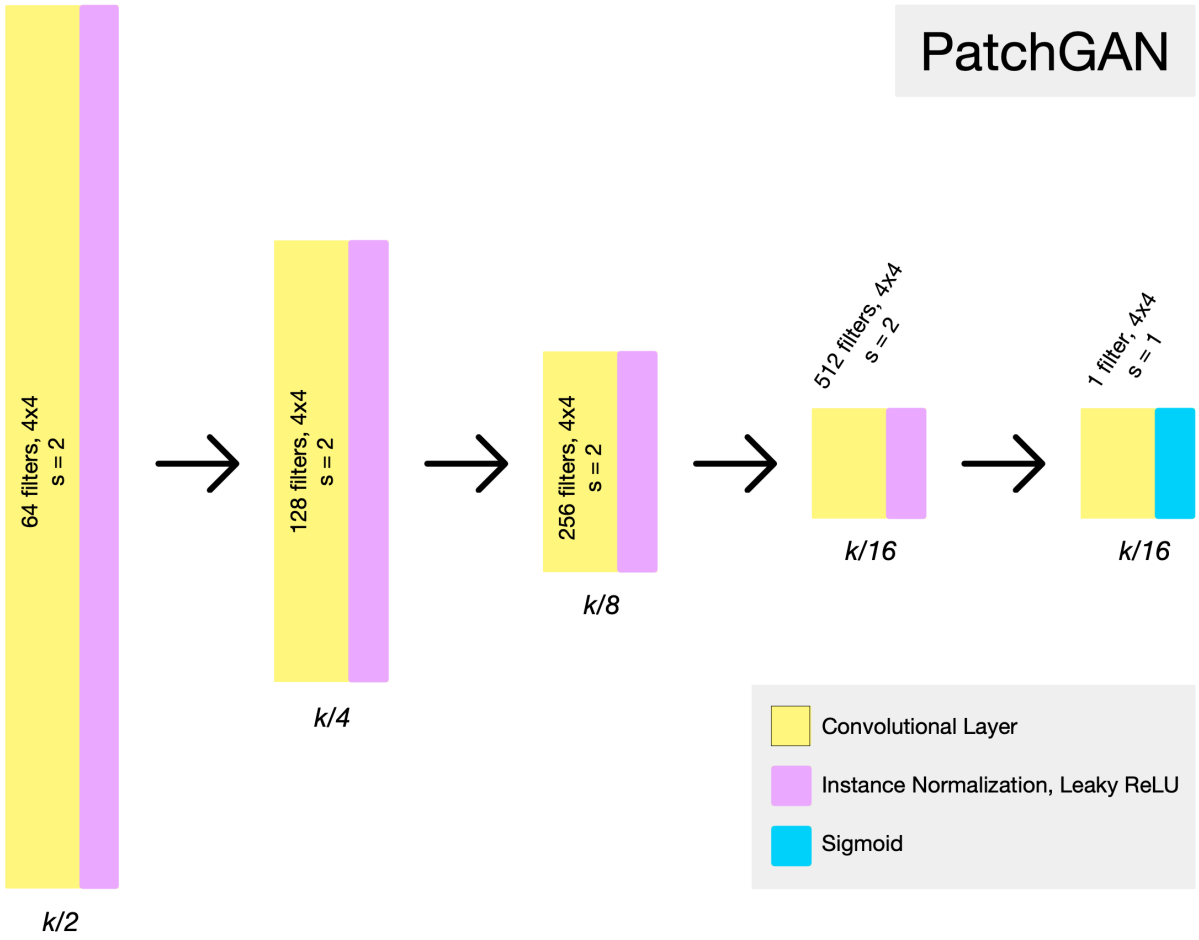


- Where each of the residual blocks is:



Discriminator Implementation

- CycleGAN uses PatchGAN architecture for the discriminators.
- Our implementation ended up being similar to:

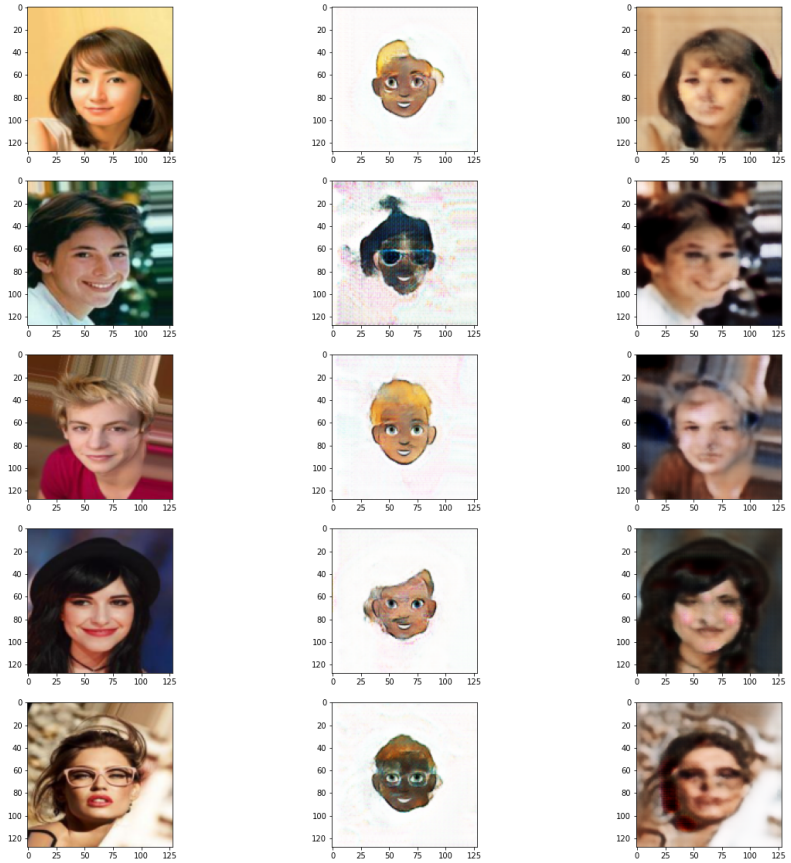


Training Details

- Hyperparameters (Letting A be the human domain and B the cartoon domain):
  - Batch Size $= 1$
  - $num\_epochs = \begin{cases} 31 & D_A, D_B, G_{B2A} \\ 15 & G_{A2B} \end{cases}$
  - Learning Rate $= 0.0002$
- Discriminators were trained on a buffer of past images.
  - The buffer contained 50 past-generated images.
  - Buffer is updated each iteration through the training data.

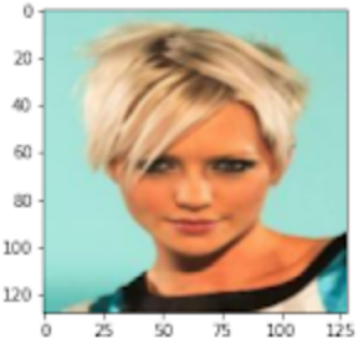
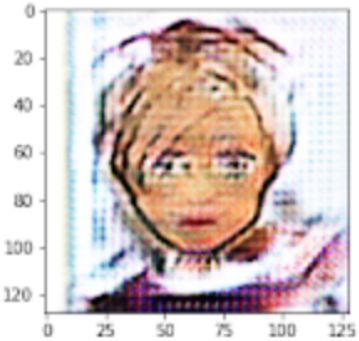
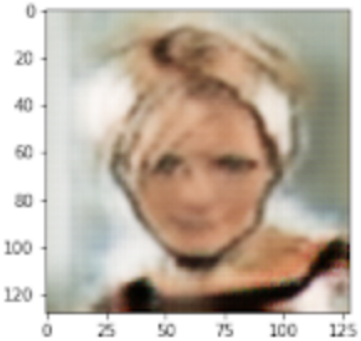
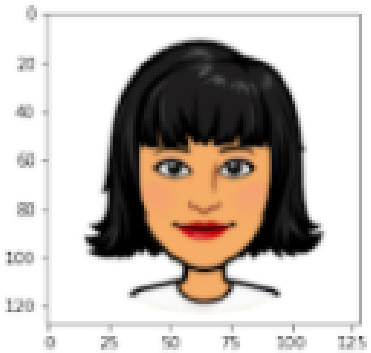
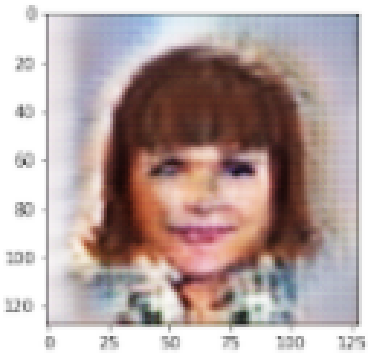
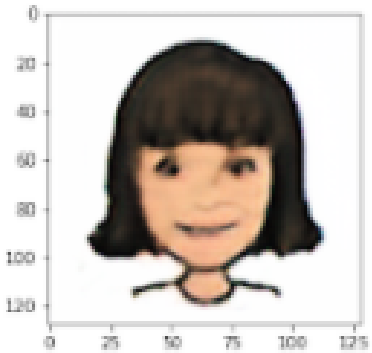
The Generator Gambit

- The map from human to cartoon began to severely overfit, so it was trained less than every other network.



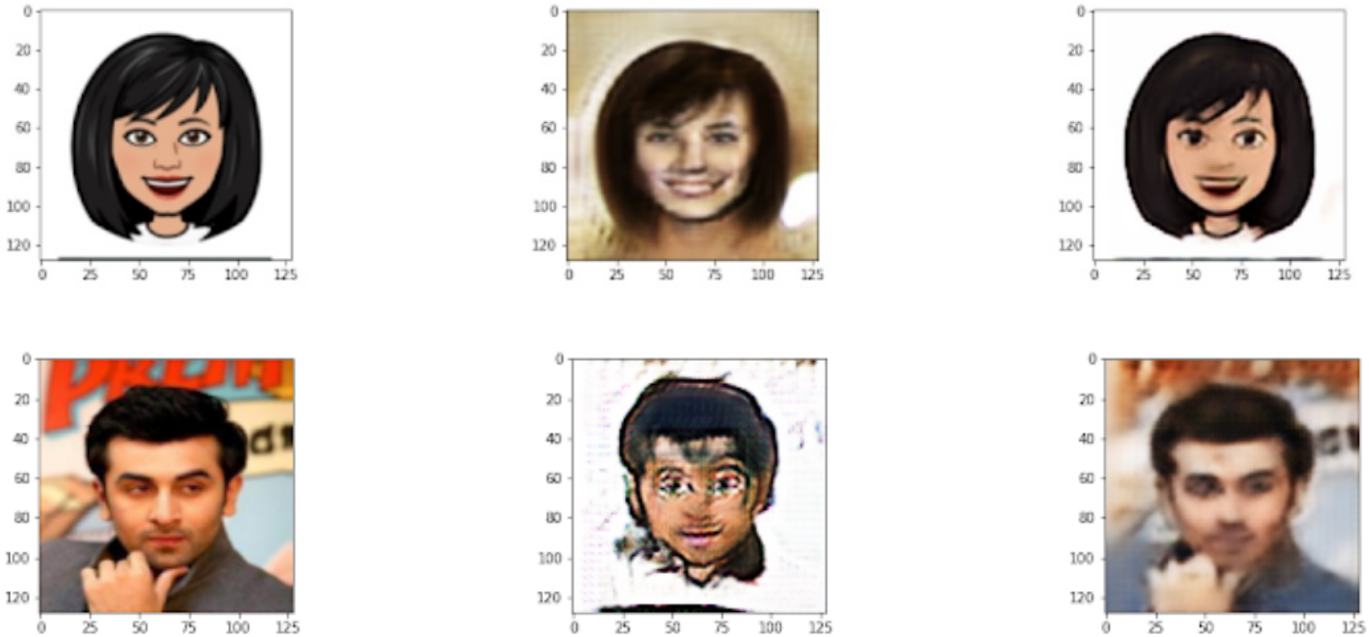
Initial Results

- Looked okay, but there was some checkerboarding happening with the generators.

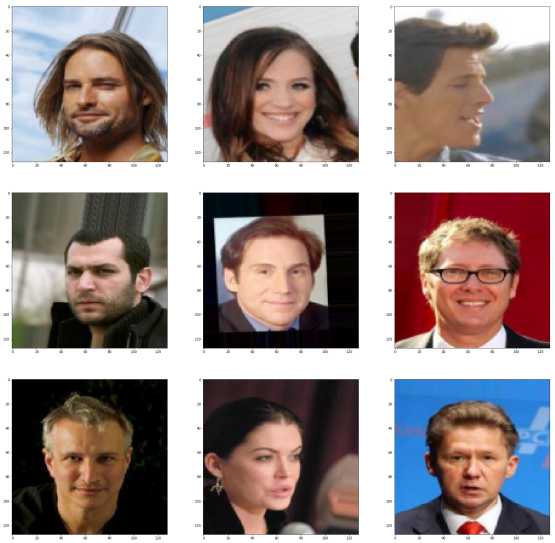
Checkerboarding Fix

- Modified the upsampling portion of the generators to not use transposed convolution.
- Instead used nearest neighbor interpolation along with regular convolution [2]
- Improved results:

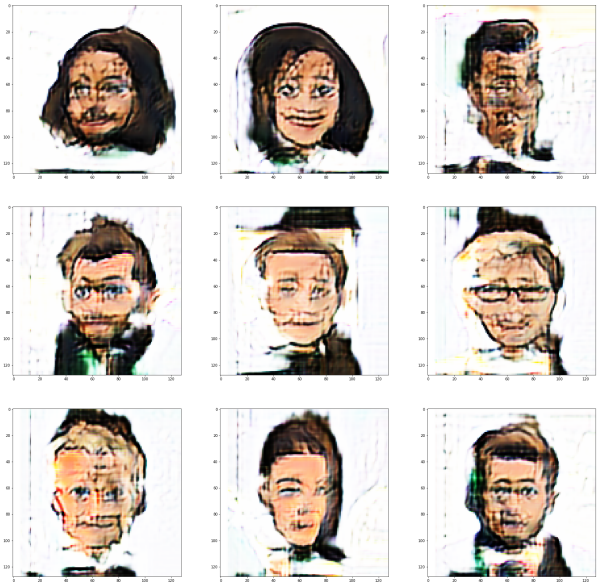


Test Results

- Some amount of modal collapse was experienced, but test results generally looked convincing.



Real Images

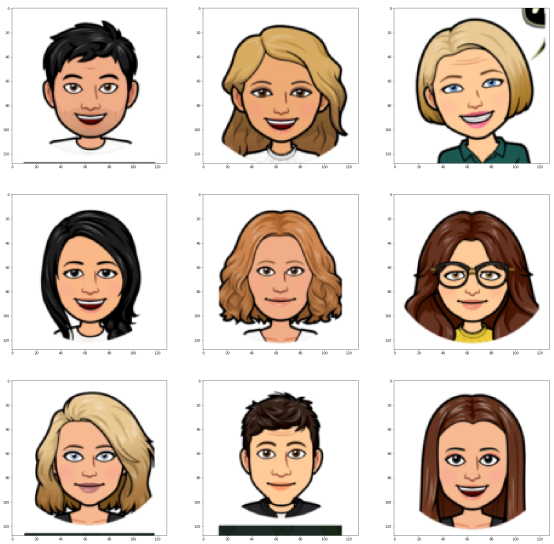


Generated Images

Cycled Images

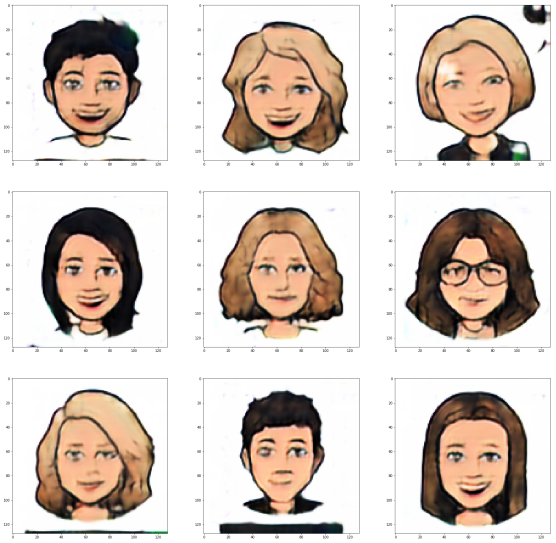
Test Results

- Some amount of modal collapse was experienced, but test results generally looked convincing.



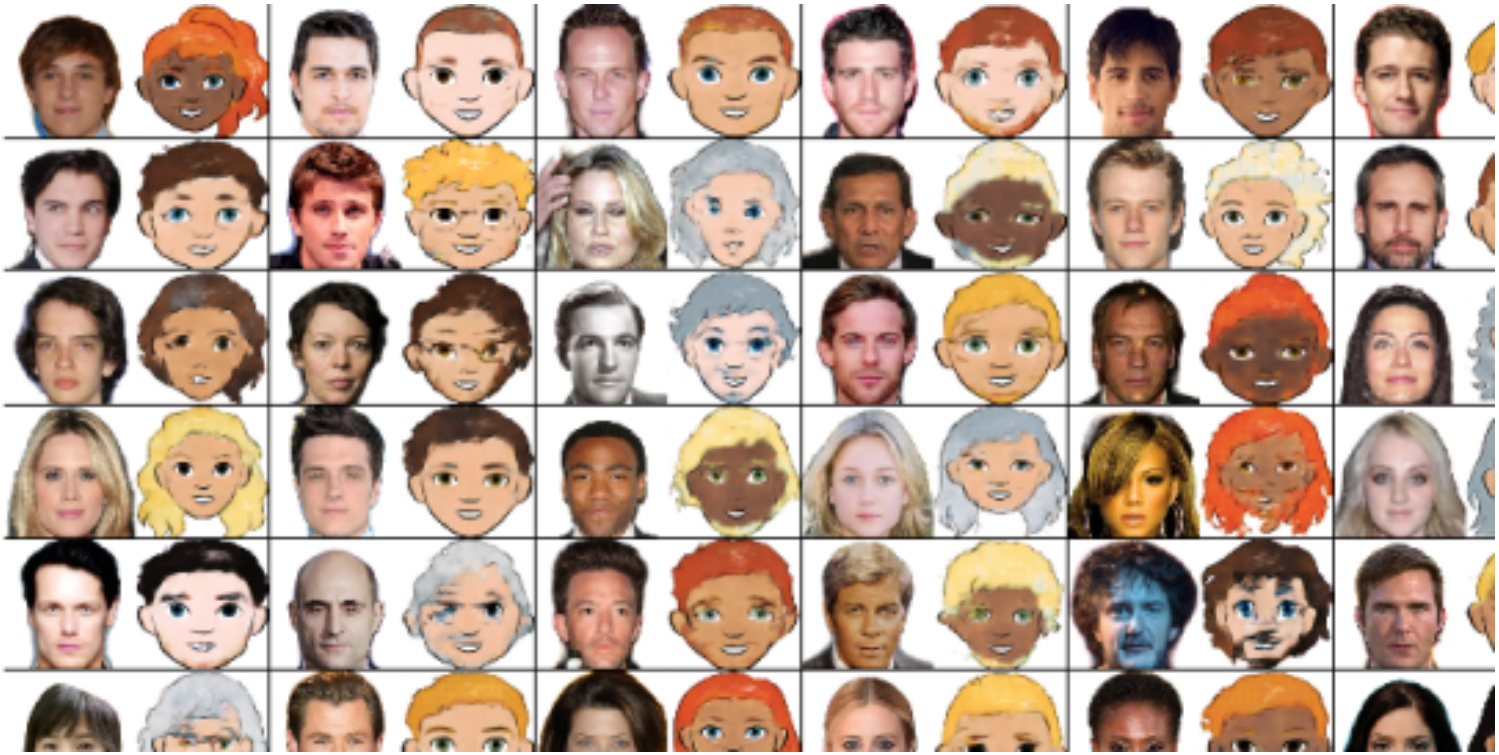
Real Images

Generated Images



Cycled Images

Comparison with XGAN [3]



XGAN generates more believable cartoons but often times they don't resemble the real person.

Areas of Improvement

- Find better datasets
  - The Bitmoji dataset had all-white backgrounds which made it difficult for the generator to produce longer hair.
  - Possibly combat this by finding a human face dataset with all-white backgrounds or cartoon set with varying backgrounds.
- Take more efforts to reduce modal collapse in the human-generated faces.
  - Try things like spectral normalization or fine-tuning the multipliers on our loss functions.

References

1. JY Zhu, et al 2017 https://arxiv.org/pdf/1703.10593.pdf
2. Odena, et al 2016 https://distill.pub/2016/deconv-checkerboard/
3. Royer, et al 2017 https://arxiv.org/pdf/1711.05139.pdf