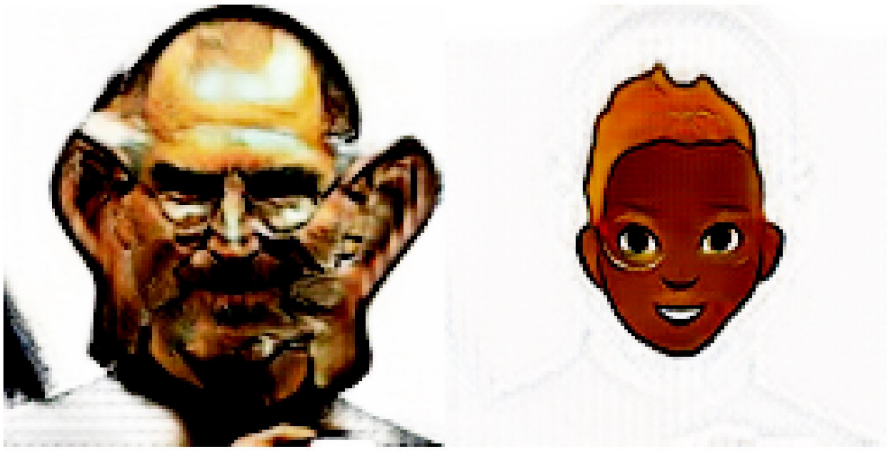
ECE 188: Real Time Unpaired Photo-to-Cartoon Translation

Lucas Tindall Samuel Sunarjo

June 15, 2019



1 DESCRIPTION

In this project we build a system for real time translation of human faces to cartoon stylized faces. We trained CycleGANs on different cartoon styles to perform the transformations. The end product is a real time application which displays a camera feed where real human faces are replaced by the selected cartoon style.

1.1 Concept

We really like cartoons so we wanted to see our lives as cartoons. With this as our inspiration we decided to build a system that could translate a real time video feed into a cartoon stylized version. While there exist methods for video translation we decided to approach this by independently translating individual frames. To accomplish this translation we train cyclic generative adversarial neural networks to transform back and forth between real and cartoon domains. This transformation is challenging to learn as we did not use any paired dataset containing conversions between real human faces and corresponding cartoon faces.

1.2 Technique

Our technical architecture is heavily based on the paper "Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks" by Jun-Yan Zhu and et al [1]. The generative network consists of strided convolutional layers, residual blocks, and fractionally strided convolutional layers; the discriminator network uses a PatchGAN architecture [2]. This approach allows the use of an unpaired dataset which fits our desired dataset and application. This is made possible from using an objective function that consists of the adversarial loss and the cycle consistency loss over the two domains of image X and Y. The first domain of images X is real human faces which we detect and extract from the camera feed, while the second domain Y depends on whichever face dataset we choose to train the model on.

We have also attempted an alternative architecture for this task, CartoonGAN [3]. The generator network is similar to the generator of CycleGAN, and the discriminator is a simple patch-level discriminator network. Their loss consists of adversarial loss and content loss.

1.3 Process

We trained the CycleGAN model from scratch on multiple different datasets of faces, such as the hand-drawn caricature CaVI dataset [4], anime face dataset [5], Simpsons face dataset [6], and 2D cartoon avatar dataset [7]. Each model is set to train up to 200 epochs, which could take up to 3 days, or manual termination based on the visual results that were generated and stored during training. The training was done on a single computer with a single Nvidia GEFORCE GTX 1080 GPU. Most of the models were terminated manually after training for 20 - 100 epochs.

We have also tried training the CartoonGAN model on these datasets but the results are not as good as the CycleGAN model. This architecture had a much longer training time compared to the previous and therefore we were not able to obtain visually significant results.

We wrote a simple script in Python which captures images from the camera feed, detects and extracts human faces in the image using a face detector from OpenCV, translates the image using one of the trained CycleGAN model, and overlays it on the camera image and displays it locally on the device such as computer with webcam. For one instance of human face in the captured image, the system is able to achieves roughly 2 frames per second performance in real time.

1.4 Result

Our results will mainly consist of comparisons between the original image and the translated version of the original image in the different domains that we trained on. For CycleGAN, Figure 1 and Figure 5 show the generated images in cycle between domain X (human face) and Y (Simpsons face) and Figure 2 shows the translated version of the actor Keanu Reeves in different image domain. For CartoonGAN, Figure 3 and Figure 4 showed the generated outputs in the anime face domain and hand-drawn caricature respectively. The results don't to be successfully translated, which is why we chose CycleGAN as our main architecture instead.

Figure 1: CycleGAN Results; first column contains original image ($Real_A$, $Real_B$), second column contains fake transformation ($Fake_B = Generator_A(Real_A)$, $Fake_A = Generator_B(Real_B)$), third column contains the reconstruction ($Reconstruction_A = Generator_B(fake_B)$, $Reconstruction_B = Generator_A(fake_A)$), and the fourth column contains identity ($Identity_B = Generator_B(real_A)$, $Identity_A = Generator_A(real_B)$)

Figure 2: CycleGAN results; left to right: original image, The Simpsons domain, Anime domain, hand drawn caricatures domain, cartoon domain

Figure 3: CartoonGAN trained on anime faces

1.5 Reflection

Our generated results are somewhat understandable with a bit of interpolation and imagination in our brain. Certainly longer training and perhaps larger dataset will help the model generate better results, which we were unfortunately unable to do due to time constraint and limited resources.

We believe the part of overlaying the translated face onto the original image definitely can be improved further with different approaches. Currently we simply replace the pixels on the original

Figure 4: CartoonGAN trained on hand drawn caricatures

image based on the bounding box from the OpenCV detection. Other interpolation technique may help in smoothing the image better for human visuals. Another aspect would be the region to overlay. Currently the translated faces cover most of the background pixels around the faces. Instead of overlaying onto the pixels in the bounding boxes, it might be potentially better to only overlay onto the face region through some sort of semantic segmentation to further preserve the background.

Incorporating depth information and aspect ratio may also improve the translation results. One good example of this issue is illustrated in the paper cover photo of Steve Job and his translated version of the hand drawn domain in the bottom left image; the translated ears are obviously too big because of the distance from the face to the "camera" is different for the original image and the hand-drawn domain.

2 Code

https://github.com/ucsd-ml-arts/ml-art-final2-ltss

3 Result

Link to example web cam video: Download Video

Figure 5: More CycleGAN results

References

[1] CycleGAN and pix2pix in PyTorch,
`https://github.com/junyanz/pytorch-CycleGAN-and-pix2pix`

[2] Image-to-Image Translation with Conditional Adversarial Networks,
`https://arxiv.org/pdf/1611.07004.pdf`

[3] CartoonGAN,
`https://github.com/znxlwm/pytorch-CartoonGAN`

[4] CaVI Dataset,
`https://github.com/lsaiml/CaVINet`

[5] Anime Face Dataset,
`https://github.com/Mckinsey666/Anime-Face-Dataset`

[6] Simpsons Faces Dataset,
`https://www.kaggle.com/kostastokis/simpsons-faces`

[7] Cartoon Set Dataset,
`https://google.github.io/cartoonset/`