Sketch2Face: Using CycleGAN to Produce Photo-Like Images from Unpaired Sketches

Authors: Jerry Meng, Connor Quinn, and Danny Takeuchi
Dept. of Computer Science
{zmeng90, ctquinn, dtakeuch}@stanford.edu

Abstract

This project tackles the problem of converting pencil sketches to photorealistic images. Previous work has focused on using traditional CNNs with reasonable results, but we used the newer CycleGAN architecture to fully utilize a dataset of unpaired sketches and faces. With the CycleGAN architecture, we experimented with different CNN architectures for the generators and discriminators. With a generator initially implemented in the related "Pix2Pix" Conditional GAN architecture, our system was able to output relatively convincing color schemes for photorealistic images given black-and-white sketches.

Code Repositories We Used:

Cycle GAN: https://github.com/jerryMeng100/CycleGAN-TensorFlow

Data, Neural Style Transfer, Pix2Pix: https://github.com/jerryMeng100/sketch2face

Mask-RCNN-Shiny: https://github.com/huuuuusy/Mask-RCNN-Shiny

1 Introduction

In both police TV dramas and real world criminal cases, law enforcement often only possesses witness description-based sketches to guide their search for potentially harmful individuals. This, among other interesting applications, motivated us to create a deep learning framework for our CS230 project that will reverse engineer photorealistic images from even rudimentary facial sketches. This sort of application has already been tried before in 2016 using basic CNNs, with some success. However, we plan to employ a novel Cycle-consistent Generative Adversarial Network (CycleGAN) technique in the hope that the adversarial nature of the GAN will generate even more realistic photos. By using the CycleGAN architecture, our model will be able to learn sketch-to-photo transformations (and vice versa) without requiring paired sketches and photos.

2 Data

We used two main datasets to train our models:

Image-sketch pairs to run the neural style transfer baseline were sampled from the CUHK Face Sketch Database (CUFS). We also employed this paired dataset to train our initial Pix2Pix explorations and power the sketch data in our CycleGAN.

We used the NVIDIA FFHQ faces dataset, containing 70,000 photorealistic faces (we used a smaller, 1,800 item dataset to avoid the data quantity asymmetry compared to the amount of sketch data), as our photos database for the CycleGAN implementations. (https://github.com/NVlabs/ffhq-dataset).

3 Methods

3.1 Architectures & Project Pipeline:

The bulk of our project came in the form of choosing and tuning the neural network architecture we would employ to tackle our problem. We began with a paired photo-sketch Neural Style Transfer network as a baseline, which indeed struggled to color sketches' faces with consistent skin tones. From there, we moved into the world of GANs by employing a Conditional GAN (a.k.a. Pix2Pix), again with paired photo-sketch data, which was able to very effectively translate paired sketches to images. However, with only a data set of about 100 paired photo-sketches available for this model, all from the same artist, we determined the model overfit the data and had no capability of generalizing to diverse new sketches.

To solve this data scarcity problem, we settled on our final GAN model: CycleGAN. To elaborate, CycleGANs differ from traditional GAN architectures in that they process two unpaired categories of data, X and Y, and for each category, has a generator which tries to map X to Y or vice versa, with a corresponding discriminator for each direction that attempts to determine if the output is "real." CycleGAN's unique point is that it has a cycle-consistency loss (described in Zhu et al) that would allow us to train a model without the need for paired sketches and photos, dramatically expanding and diversifying the frontier of our dataset.

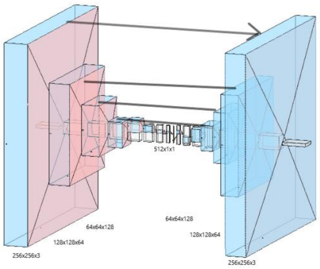


Figure 1: Pix2Pix Generator Architecture

We started with a TensorFlow implementation of a CycleGAN by vanhuyz on Github. The generator architecture is shown in Figure 2 below, and is based on a set of convolutions, a set of residual convolutions, and a set of deconvolutions to map an input image to an output image of the same dimension. However, inspired by how effectively Pix2Pix could perform on paired data, we wondered if replacing this generator structure with the Pix2Pix architecture could improve performance. This generator architecture is shown below in Figure 1 above, and is based on a set of convolutions followed by deconvolutions, but with residual skips across each of the layers of equal size:

After implementing these architectures, we found that both struggled with what we called "coloring fragmentation." Essentially, because the sketches' backgrounds often had leftover graphite/charcoal that resembled the color of the subject's skin tone, both models "learned" to place skin-like color all over the background. We solved this by transforming our project into a pipeline that first employed a Mask-RCNN network to crop out the photos' and sketches' subjects (i.e. remove the background) and surround them with a plain color background. We experimented with both "green screen" and white backgrounds in this regard, finding that the white backgrounds yielded solid performance while green backgrounds caused "image negative errors" regularly (described below).

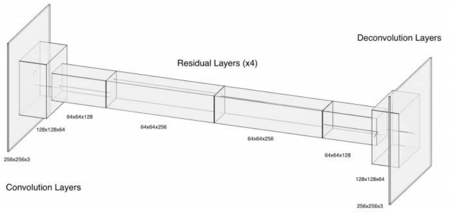


Figure 2: CycleGAN Generator Architecture

Additionally, we tried experimenting with different types of PatchGAN discriminator architectures. The most important thing we experimented with was the receptor field size. We tried out 1x1, 16x16, and 70x70 fields, ultimately settling with the 70x70 PatchGAN as it produced the most defined outputs. The 1x1 provided little spatial information and the 16x16 gave detailed images but ran into the issue of containing some tiling artifacts. This was largely alleviated by the 70x70 PatchGAN.

3.2 Applying CycleGAN

As mentioned above, our CycleGAN is powered two generator nets, one to map sketches to photos and one to map photos to sketches, and two discriminators, one to discern "real" sketches and one to discern "real" photos. At each "step" in the training process, one sketch and one photo (unpaired) are inputted to each generator. The discriminators then determine if their outputs are "real" to determine loss. In addition, the dual generator takes the first generator's input and attempts to reconstruct the original inputs as another determiner of loss. Gradients are then computed and parameters updated. This simplified structure is illustrated in Figure 3.

The CycleGAN's various neural networks' parameters are updated with the Cycle-Consistency Loss function, on which more details can be found in the original CycleGAN paper:

$$J = J^{(D1)} + J^{(G1)} + J^{(D2)} + J^{(G2)} + \lambda J^{cycle}$$

3.3 Hyperparameters and Training the Model

Before finally training our models, we performed a few small test runs for hyperparameter tuning. We primarily tweaked the hyperparameters λ_1 and λ_2, which controlled the weights of the cycle consistency loss. The λ's were coefficients that would be multiplied by the cycle-loss from X->Y->X versus Y->X->Y. We found that when $\lambda_2 = 2 * \lambda_1$, the system performed poorly, but when $\lambda_1 = 2 * \lambda_2$, the system did quite well in learning the color of features on the face.

From there, we ran both networks for as many steps (i.e. input of 1 unpaired sketch and photo into the Cycle GAN) as possible until we noticed performance was no longer improving. This resulted in 70,000 steps for the conventional CycleGAN and 30,000 steps for the Pix2Pix Architecture revision. One curious finding is that, despite the similarly-sized generator architectures, the Pix2Pix generator version trained roughly 50% more quickly, which we suspect owes itself to the greater number of residual skips in that architecture.

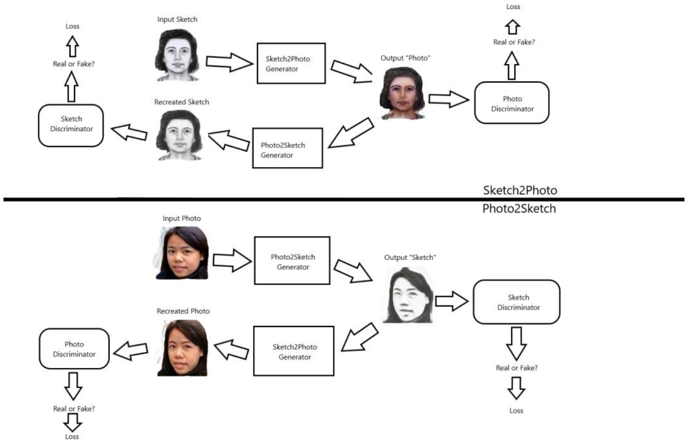


Figure 3: CycleGAN Overall Structure

4 Results

We ran the default generator CycleGAN for 70,000 steps and tracked the outputs of both the "sketch2face" and "face2sketch" directions. We can compare this with with our Pix2Pix generator architecture version, which we ran for about 30,000 steps. Figure 4 contains a sample of the sketch2face direction of these models, with the top from the default generator version and the bottom from the Pix2Pix generator version. Left to right: input sketch, generated photo from input sketch, reconstructed sketch from generated photo.

Both show some cases of background color artifacts, which we suspect resulted from some leftover background that the RCNN neural net did not crop out during our data pre-processing. In both cases, the architectures learns to color faces with reasonable skin shading and even clothes/accessories coloring, though each architecture tended to yield slightly differing facial hues as shown (and across many more examples).

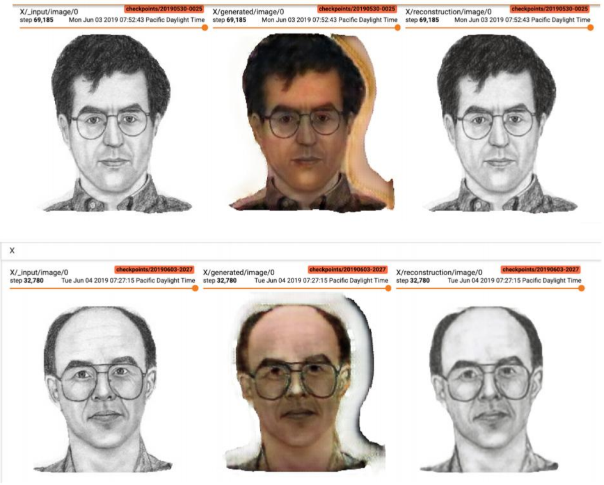

X/_input/image/0
checkpoints/20190530-0025
step 69,185 Mon Jun 03 2019 07:52:43 Pacific Daylight Time
X/generated/image/0
checkpoints/20190530-0025
step 69,185 Mon Jun 03 2019 07:52:43 Pacific Daylight Time
X/reconstruction/image/0
checkpoints/20190530-0025
step 69,185 Mon Jun 03 2019 07:52:43 Pacific Daylight Time
X
X/_input/image/0
checkpoints/20190603-2027
step 32,780 Tue Jun 04 2019 07:27:15 Pacific Daylight Time
X/generated/image/0
checkpoints/20190603-2027
step 32,780 Tue Jun 04 2019 07:27:15 Pacific Daylight Time
X/reconstruction/image/0
checkpoints/20190603-2027
step 32,780 Tue Jun 04 2019 07:27:15 Pacific Daylight Time

Figure 4: CycleGAN Step Examples (Top: Default Generator Architecture, Bottom: Pix2Pix Generator Architecture)

4.1 Error Analysis

One of the main sources of training trouble was a color inversion between the input and the generated, shown in Figure 5:

This typically resulted when the training hyperparameters were ill-set and is analogous to becoming trapped in the Cycle-Consistent loss's local minima.

More precisely, the inversion results in an undesirable transformation G from the input to the generated image, but where the other generator F learns to undo the undesirable transformation. This occurred sporadically, and we largely didn't have these cases with better hyperparameters. Another step to take to avoid this is to alternatively freeze generators as in Harley et al, but we did not prioritize this upgrade over changing architectures or hyperparameters.

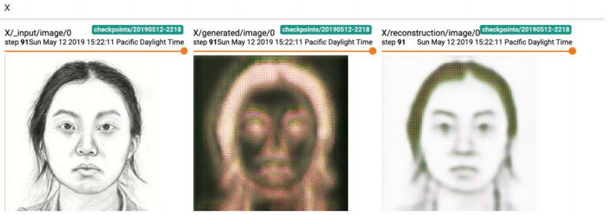

X
X/_input/image/0
checkpoints/20190512-2218
step 91Sun May 12 2019 15:22:11 Pacific Daylight Time
X/generated/image/0
checkpoints/20190512-2218
step 91Sun May 12 2019 15:22:11 Pacific Daylight Time
X/reconstruction/image/0
checkpoints/20190512-2218
step 91 Sun May 12 2019 15:22:11 Pacific Daylight Time

Figure 5: Color Inversion

In addition to the color inversion, we encountered background colors that seeped into the face color many times, which persisted somewhat even in the final result. We suspect this stemmed from both residual backgrounds in the input photos, as well as the GAN struggling to differentiate when facial makeup/accessories should be translated to the generated photo from a base sketch. This is shown in Figure 6:

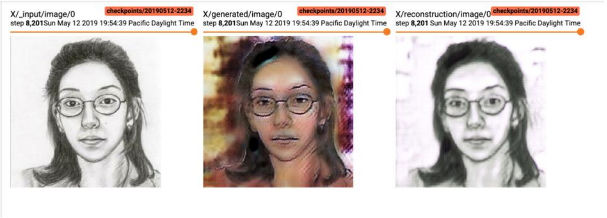

X/_input/image/0
checkpoints/20190512-2234
step 8,201Sun May 12 2019 19:54:39 Pacific Daylight Time
X/generated/image/0
checkpoints/20190512-2234
step 8,201Sun May 12 2019 19:54:39 Pacific Daylight Time
X/reconstruction/image/0
checkpoints/20190512-2234
step 8,201 Sun May 12 2019 19:54:39 Pacific Daylight Time

Figure 6: Background Fragmentation

We got around this by using a CNN which did segmentation to separate the face from the background, then fill all background with white. The main results we have are after this process which largely solved this problem.

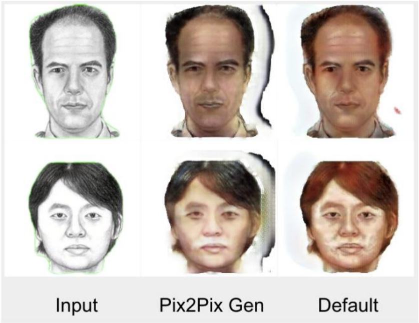
5 Discussion

Our final CycleGAN enabled us to transfer the facial features from low resolution sketches while maintaining the input image constraints. The generated sketches and images are robust across a wide variety of face structures, hairstyles, and skin tones, and we believe satisfy our project goal of achieving these transformations. In general, the Pix2Pix Generator CycleGAN yielded "duller" (but arguably more realistic) skin and hair tones, but often produced quite a bit of background fragmentation. In contrast, the CycleGAN model trained more slowly, but produced lusher hair and skin tones. However, this model avoided background fragmentation more consistently than Pix2pix as can be seen in the figure above. We suspect this core color difference owes itself to the residual skips across the entirety of the Pix2Pix generator version's network, allowing it to learn the identity of desired colors early and not over-train to yield less realistic coloration.

While we primarily focused on qualitative image quality, we also calculated quantitative metrics for both of our models. One of which is the "inception score," which gives a metric to measure a combination of a GAN's generated image quality and image diversity, given by the following equation:

$$IS(G) = e^{E_{x \sim p_g}[KL(p(y|x)||p(y))]}$$

We found no significant difference between the two models' sketch2face generators' inception scores despite the still discernible trends with the generated faces. In addition, the Pix2Pix generator model trained more quickly on a step-by-step basis, and only required half the steps of the default CycleGAN to achieve comparable performance to that model. Once again, we suspect the generator's residual skips yielded both of these differences for the Pix2Pix generator version, as the convolutions/deconvolutions are of relatively similar sizes and natures in both generator architectures.



Model	Inception Score Mean	Std. Dev.
Pix2pix Gen.	1.62	.205
CycleGAN Gen.	1.60	.223

Figure 7: Model Inference Results

5.1 Future Steps

Our current methods can achieve reasonable results on most cases. In sketch to image translation tasks, our model very often succeeded in transfer in the skin tones and facial textures. A future area we would like to pursue is tasks that involve geometric changes to the input, such as generating the maternal or paternal face of the image image. This would require the ability to map to opposite gender counterparts and the ability to distinguish would features should transfer over. We would also like to look into super-resolutions GANs in order to sharpen our generated images.

Although we trained both CycleGAN models for many tens of thousands of steps, other CycleGAN papers have trained networks for over 600,000 steps; as such, letting the algorithm run for longer could yield some more realistic results or prune out undesirable artifacts.

In addition, we might explore adding more robustness to the input sketches. Currently, the sketches are generally rather detailed without verging into complete artistic photorealism, and so we are interested in mixing in both more basic and advanced sketches from various artists to generate a stronger model.

One more interesting project add-on we thought of is in combining super-resolution GANs to sharpen the generated images. Used in either an intermediate step or post-processing, we believe this addition to our project pipeline could greatly enhance the photorealism of our image outputs.

To sharpen our current project, we pondered removing photos that contain makeup or facial accessories from our training set, or appending sketches with these features to our sketch dataset. Both architectures tended to learn colors such as blue and pink around the eyes. Pictures with accessories performed slightly better because they come in a variety of shapes while makeup misled the model. Furthermore, our model should produce images of faces without makeup since the input sketches do not include makeup.

6 Team Contributions and What Code We Wrote:

Jerry: I proposed using CycleGANs as a new method to investigate the problem of sketch to photo, after seeing a paper using CNNs for the same purpose. I researched to find the photos dataset and built scripts to make data uniform and split the data. I experimented with CycleGAN implementations and hyperparameters early on, and worked to use a CNN to resegment our data so that the background in the photos did not interfere with the learning. For the report, I focused on the results and error analysis section.

Connor: I primarily focused on investigating the Pix2Pix architecture initially, and so most of my personal coding entailed implementing the Pix2Pix generator architecture into the CycleGAN system. I also ran many of the different model runs that ultimately lead to our final model, such as testing green backgrounds vs. white backgrounds on the CycleGAN and Pix2Pix generator architectures. For the final report and poster, I primarily focused on the "Methods" sections and created many of the handmade visuals found in this report and on the poster, but also made large contributions across most sections.

Danny: I developed the Neural Style Transfer baseline initially. I then transitioned to researching and implementing various PatchGAN discriminator architectures for both Pix2Pix and CycleGAN. Discriminators of several varying receptor field sizes were tested. I also wrote up the parts pertaining to these, the discussion section, and much of the poster.

7 References

Gatys, Leon A., Alexander S. Ecker, and Matthias Bethge. "A neural algorithm of artistic style." arXiv preprint arXiv:1508.06576 (2015).

Harley, A.W., Wei, S., Saragih, J.M., Fragkiadaki, K. (2019). Image Disentanglement and Uncooperative Re-Entanglement for High-Fidelity Image-to-Image Translation. CoRR, abs/1901.03628.

He, K., Gkioxari, G., Dollar, P., Girshick, R. (2017). Mask R-CNN. 2017 IEEE International Conference on Computer Vision (ICCV). doi:10.1109/iccv.2017.322

Isola, Phillip, et al. *Image-to-Image Translation with Conditional Adversarial Networks*. Berkeley AI Research (BAIR) Laboratory, 2017, github.com/junyanz/pytorch-CycleGAN-and-pix2pix. Accessed 6 May 2019.

Jun-Yan Zhu*, Taesung Park*, Phillip Isola, and Alexei A. Efros. *"Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks"*, in IEEE International Conference on Computer Vision (ICCV), 2017.

Karras, T., Laine, S., Aila, T. (2018). [Flickr-Faces-HQ Dataset (FFHQ)].

W. Zhang, X. Wang and X. Tang. Coupled Information-Theoretic Encoding for Face Photo-Sketch Recognition. *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011.

X. Wang and X. Tang, "Face Photo-Sketch Synthesis and Recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, Vol. 31, 2009.