Reimagining the Abstract Image: Using GANs to Develop Minimal Sketches of Real Object Images

Levi Lian
levilian@stanford.edu

Amber Yang
yanga@stanford.edu

Benjamin Yeh
bentyeh@stanford.edu

1. Introduction

Machine learning has enabled many recent advances in drawing sketches and creating abstract art. Several recent efforts have focused on creating realistic sketches and imitations of human-drawn sketches [2] or generating photorealistic images from simple sketches. [3, 6, 1] Here, we take the opposite approach and consider the problem proposed by [2] of creating minimal sketches of objects such that the sketch still abstractly resembles said objects. We ultimately hope that this project will help to expand the abstract artist's creativity in how common objects can be represented by providing sketches of how to envision things minimally.

2. Problem Statement

We can break down our objective—creating minimal sketches of objects such that the sketch still abstractly resembles said objects—into two components: (1) creating minimal sketches, and (2) ensuring that the sketch abstractly resembles the original object. We note that there exist two common representations of "sketches": as a series of strokes (vector graphics), or as a set of pixels (raster graphics). [2, 4] We chose the latter since it is more amenable to convolutional neural networks. To quantify the "minimalism" of a raster sketch, we propose the following metric: ratio of black (object) to white (background) pixels, where the dimensions of the image are fixed *a priori* (e.g., 256 x 256).

To evaluate how well a sketch abstractly resembles the original object, we consider two approaches. The first is human perceptual studies, as previously done by [3, 6]. The second is dataset-specific: given a paired photo-sketch dataset, for example, we could compute how similar the generated sketch is to the target sketch. For unpaired datasets, there is no perfect objective; however, we can build a classifier to verify whether the generated sketch falls into the same category (e.g., cat or airplane) as the input image.

3. Datasets

3.1. Paired Dataset

First, we will work with the Sketchy dataset, which consists of 125 categories (e.g., airplane, cat, etc.), each with 100 unique photographs, each paired with roughly five human-drawn sketches for a total of 12,500 unique photographs of objects and 75,471 human sketches. [4] Both the photos and sketches are made available as 256x256x3 (RGB) images; we convert the sketches to binary 256x256 images before use.

Note that a paired dataset like Sketchy can still be used to train an unpaired dataset model like CycleGAN. [6]

3.2. Unpaired Datasets

Next, we will work with Quick Draw, an unpaired sketch database, and CIFAR-10, an unpaired image database. The Quick Draw dataset consists of 325 categories of sketches, each with a varying number of unique sketches per category for a total of 50 million sketches given in jpg format. The CIFAR-10 dataset consists of 60,000 color unique images in 10 classes for 6,000 images per class of object.

Similar to our first approach with the paired image-sketch database, we will create an "image-to-sketch" generator, which we can borrow our implementation from the image to sketch encoder used for the paired dataset approach, implement a sketch discriminator, and then implement a sketch-to-image generator (similar to the decoder used for the paired dataset).

4. Technical Approach

4.1. Baseline: Edge Detection

For a simple baseline model, we used a 2D Sobel filter. [5]

4.2. Paired Dataset Model: pix2pix

Since we had access to a paired dataset, Sketchy, [4] we first experimented with the pix2pix conditional GAN (cGAN) image-to-image translation model. [3] Like unconditional GANs, cGANs consist of a generator G that learns to fool a discriminator D. Unlike unconditional GANs, however, both the cGAN generator and the discriminator observe the input image–in other words, the cGAN generator is "conditioned" on an input image rather than random noise. The pix2pix objective function is

$$G^* = \arg\min_G \min_D \mathcal{L}_{cGAN}(G, D) + \lambda\mathcal{L}_1(G) \quad (1)$$

where $\mathcal{L}_{cGAN}$ is the cGAN loss

$$\mathcal{L}_{cGAN}(G, D) = \mathbb{E}_{x,y} \log D(x, y) + \mathbb{E}_{x,z} \log(1 - D(x, G(x, z))) \quad (2)$$

and $\mathcal{L}_1$ is the L1 loss (which encourages sharper images than L2) [3] between the generated sketch and the target sketch

$$\mathcal{L}_1(G) = \mathbb{E}_{x,y,z} \|y - G(x, z)\|_1 \quad (3)$$

with x sampled from the set of photos, y sampled from the set of sketches, and z a noise component, implemented as dropout noise (at both train and test time).

To encourage the generation of simpler sketches, we additionally add a penalty term (effectively counting the number of black pixels)

$$\mathcal{L}_{npix}(G) = \mathbb{E}_{x,z} \|1 - G(x, z)\|_1 \quad (4)$$

on top of the pix2pix objective (Equation 1) for a final objective function

$$G^* = \arg\min_G \min_D \mathcal{L}_{cGAN}(G, D) + \lambda\mathcal{L}_1(G) + \gamma\mathcal{L}_{npix}(G). \quad (5)$$

4.3. Unpaired Dataset Model: CycleGAN

The second method we used is CycleGAN. [6] Instead of relying on paired datasets as in pix2pix, CycleGAN learns mapping functions between two domains X and Y given two sets of datasets each belonging to one domain. CycleGAN uses the same adversarial loss function defined for pix2pix, but includes an additional cycle consistency loss function in our final objective function. This cycle consistency loss is what enables the training to learn the correct mapping between two domains, without a one-to-one mapping. Intuitively, cycle consistency means when we apply a backward generator onto the result, we should get an image similar to the original input. That is, for the mapping function $G : X \rightarrow Y$, $F : Y \rightarrow X$ and the original image $x \in X$, we have $F(G(x)) \approx x$. We use L1 norm to measure the similarity between $F(G(x))$ and x. Similarly, we can measure the similarity between $G(F(y))$ and y for $y \in Y$. Finally, since we want both mappings G and F to be accurate, the cycle consistency loss is simply the sum of both terms.

5. Preliminary Results

Our preliminary results used the original pix2pix and CycleGAN models without the added loss terms penalizing the number of black pixels in the generated image. This will be addressed in our final work. Additionally, we only trained and evaluated on images in the airplane category of photos in the Sketchy dataset. Eventually, we plan on training over a subset of the categories (say, 100) and evaluating our models on the remaining (say, 25) categories to evaluate our models' generalization performance.

The evaluation metric we used to compare the two aforementioned models with the baseline is the number of black pixels in the generated images of the entire test dataset. Instead of using a single statistic to represent the number of black pixels in the predicted image set, we show a distribution of the number of black pixels below.

As we can see (Figure 3), the CycleGAN model for the unpaired dataset seemed to outperform the pix2pix model for the paired dataset. In our preliminary tests, we noticed that after training for several epochs, the predicted sketch for the pix2pix model did not change between different sample inputs we tested with. We will work on investigating this matter past this milestone.

Below we show the predicted sketch image of the same input on the rightmost side - $n02691156\_9966.jpg$ in the Sketchy dataset - generated from the two aforementioned models.

References

[1] W. Chen and J. Hays. SketchyGAN: Towards Diverse and Realistic Sketch to Image Synthesis. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9416–9425, Salt Lake City, UT, USA, jun 2018.

[2] D. Ha and D. Eck. A Neural Representation of Sketch Drawings. In *6th International Conference on Learning Representations, ICLR 2018 - Conference Track Proceedings*, Vancouver, BC, Canada, feb 2018. International Conference on Learning Representations, ICLR.

[3] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-Image Translation with Conditional Adversarial Networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5967–5976, Honolulu, HI, USA, jul 2017.

[4] P. Sangkloy, N. Burnell, C. Ham, and J. Hays. The sketchy database: learning to retrieve badly drawn bunnies. In *ACM Transactions on Graphics*, volume 35, pages 1–12. Association for Computing Machinery, jul 2016.

[5] S. van der Walt, J. L. Schönberger, J. Nunez-Iglesias, F. Boulogne, J. D. Warner, N. Yager, E. Gouillart, T. Yu, and the scikit-image contributors. scikit-image: image processing in Python. *PeerJ*, 2:e453, 6 2014.

[6] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2242–2251, Venice, Italy, oct 2017.

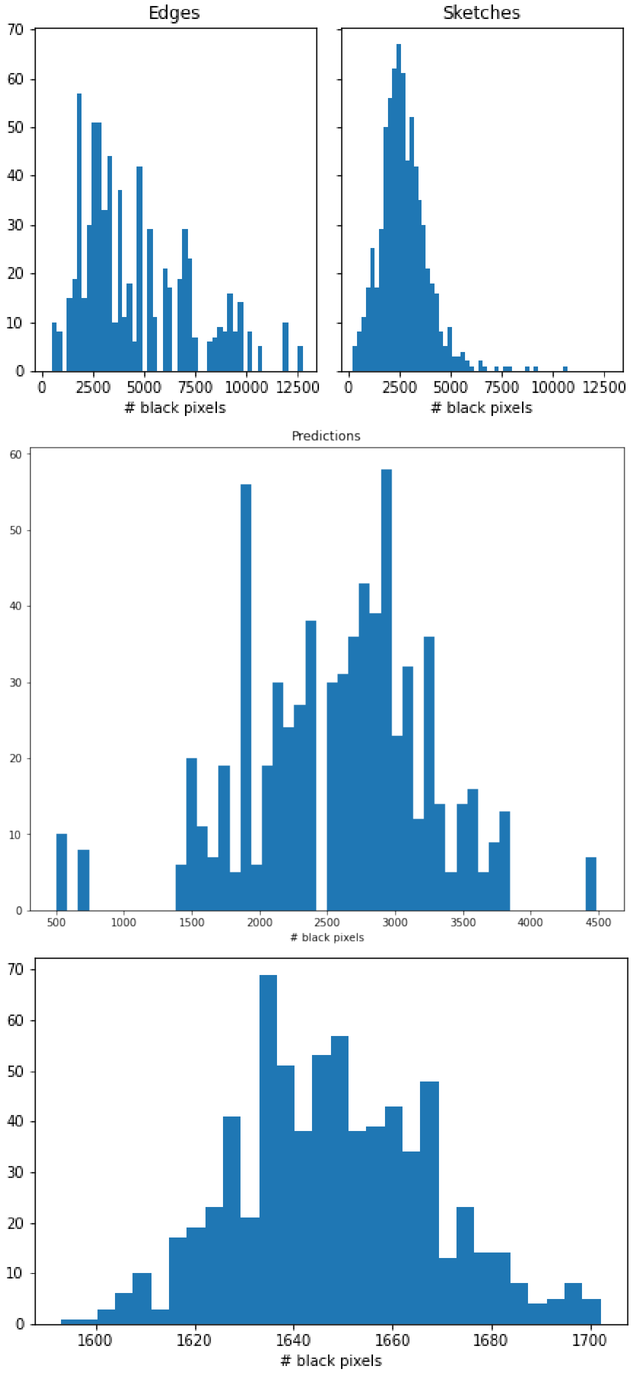

Figure 1. Distributions of the number of black pixels generated by edge detection, the original sketches, CycleGAN, and pix2pix (left-to-right, top-to-bottom). The total number of pixels in a 256 x 256 image is 65536.

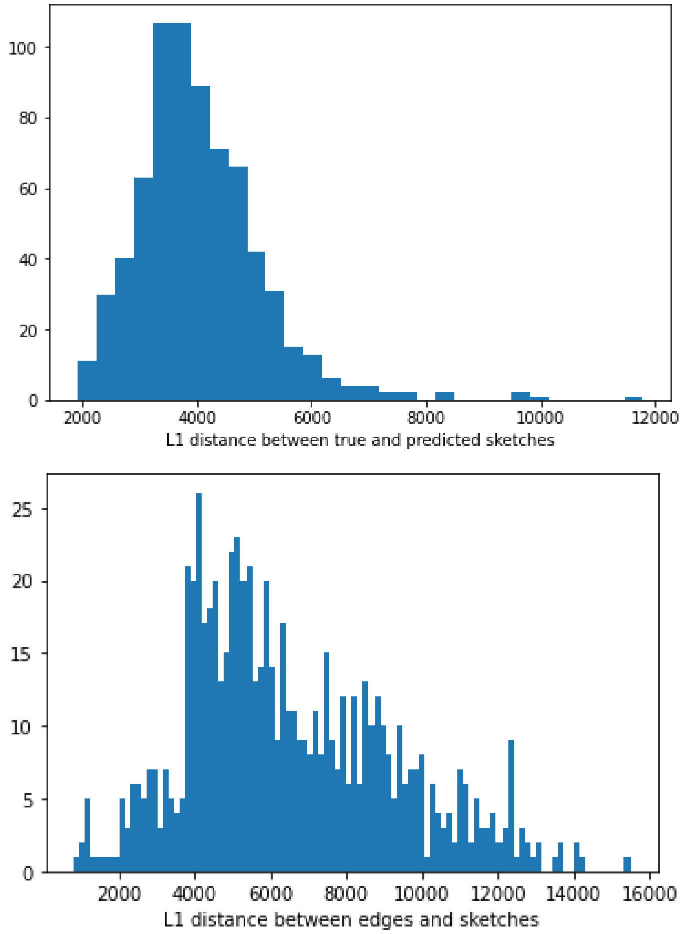

Figure 2. Distributions of L1 distance between target sketch and predicted sketch by pix2pix and edge detection (top-to-bottom).

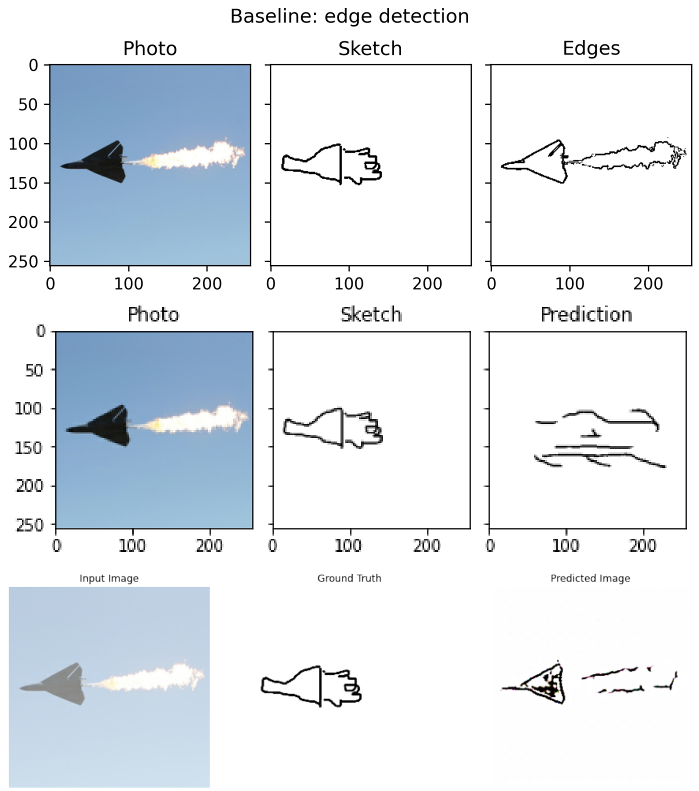

Figure 3. Sketches generated by edge detection, pix2pix (trained 20 epochs), and CycleGAN (trained 40 epochs) (top to bottom).