DrawBot: Turning Bad Sketches into Beautiful Sketches

Jimmie Harris
jdharris@stanford.edu

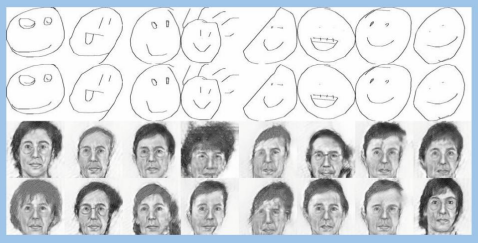
Tayo Falase
tfalase@stanford.edu

Usman Khaliq
usmank@stanford.edu

Problem Statement

How might we design and develop a deep learning application that takes as its input a rough (or badly) made sketch and transforms it into a drop-dead masterpiece? As designers, two of us have often struggled to bring our ideas to life in an aesthetically pleasing manner, while all three of us would dearly love to have a tool that makes it easy for us to churn out beautiful sketches. The input to our algorithm is a bad sketch of a face. We then use a MUNIT [1] network to output a much cleaner and aesthetically accurate image of a face. We found that while the model was able to learn large parameters (face shape and size), it was unable to learn finer details (nose shape, smiling/frowning). We believe that with better data, this model could be used to create more accurate sketches.

Results

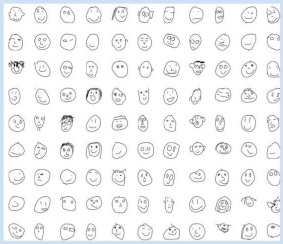


Training results after 80,000 iterations of MUNIT. The model was trained on a set of 300 unpaired bad sketches and good sketches and tested on a set of 100 bad sketches from Quick, Draw!. This demonstrates the ability to create realistic sketches within the confines of the shapes given, but highlights trouble with specific features.

Data Collection

Source Images

"Bad sketches" collected from Google's open-source Quick, Draw! Dataset [2]. We made a Python script to collect the data as images rather than vectorized representations.



Target Images

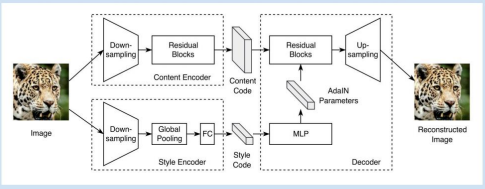
"Good" portrait sketches collected from the CUHK Face sketch database [3]. These images were cropped to 128x128 pixel images and then fed into our model.

Discussion + Future Work

Our model seemed to have learned some parameters from the input images (face size and shape) fairly well. However, it is unable to adjust finer details such as nose shape and smiling/frowning. We believe that the main inhibitor was the quality of the input drawings. If we had been able to use drawings with more detailed features (many of them lack noses, ears, hair, etc) we believe that this model could have learned those parameters as well.

Model

We use **Multimodal Unsupervised Image-to-Image Translation** (**MUNIT**), which takes as input a source domain and target domain. The model creates a shared content space with features common between the two domains, as well as a style space for each domain with features unique to that domain. To translate an image from the source domain to the target domain, MUNIT combines the content code of the source with a randomly-selected style code of the target.



MUNIT's auto-encoder architecture [1].

$$\min_{E_1,E_2,G_1,G_2} \max_{D_1,D_2} \mathcal{L}(E_1,E_2,G_1,G_2,D_1,D_2) = \mathcal{L}_{\text{GAN}}^{x_1} + \mathcal{L}_{\text{GAN}}^{x_2} + \lambda_x(\mathcal{L}_{\text{recon}}^{x_1} + \mathcal{L}_{\text{recon}}^{x_2}) + \lambda_c(\mathcal{L}_{\text{recon}}^{c_1} + \mathcal{L}_{\text{recon}}^{c_2}) + \lambda_s(\mathcal{L}_{\text{recon}}^{s_1} + \mathcal{L}_{\text{recon}}^{s_2})$$

MUNIT's Total Loss Function [1]

References

[1] X. Huang, M.-Y. Liu, S. Belongie, and J. Kautz, "Multimodal Unsupervised Image-to-Image Translation," *arXiv:1804.04732 [cs, stat]*, Apr. 2018.
[2] J. Jongejan, H. Rowley, T. Kawashima, J. Kim, and N. Fox-Gieg. The Quick, Draw! - A.I. Experiment. https://quickdraw.withgoogle.com/, 2016
[3] X. Wang and X. Tang, "Face Photo-Sketch Synthesis and Recognition," IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), Vol. 31, 2009.