Deep Sketch Drawer

Yipeng He, Di Bai, Wantong Jiang

{yipenghe, dibai, wantongj}@stanford.edu

Overview

Many efforts have been done to produce sketch-to-image generation and image retrieval from sketches. Few work addresses the opposite direction: generate human sketches from images.

We explored how to use deep learning approaches to generate sketch-style images.

Motivation:
- Existing work on generating sketches are more about generating sketches given human sketch input
- Explore ways how AI can draw sketches based on realistic photos

Dataset & Data Preprocessing

- Sketchy Dataset
  - 125 categories
  - 100 photo per category
  - average 4-5 sketches after data cleaning
- Data preprocessing
  - Removed poor quality sketches according to annotation files
  - Manually selected over 1000 photos for experiments
  - Wrote program to select the most similar sketch based on spatial distance with cosine similarity
  - Resized images to different sizes
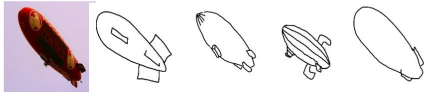
References

1. Sangkloy, P., Burnell, N., Ham, C., & Hays, J. (2016). The sketchy database: learning to retrieve badly drawn bunnies. ACM Transactions on Graphics (TOG), 35(4), 119.
2. Gatys, L. A., Ecker, A. S., & Bethge, M. (2015). A neural algorithm of artistic style. arXiv preprint arXiv:1508.06576.Hed detector
3. Xie, S., & Tu, Z. (2015). Holistically-nested edge detection. In Proceedings of the IEEE international conference on computer vision (pp. 1395-1403).
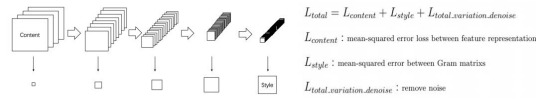4. Anish Athalye, Neural Style, (2015), GitHub repository, https://github.com/anishathalye/neural-style

Approaches & Architectures

Architecture search:
- We first tried simple convolutional layers with mean squared error loss, but the result was not promising
- We then framed the problem as to get a "bad" edge detector, and thus we turned to holistic-nested edge detector network and explored methods to generate sketches from there

Neural Style Transfer:
- We tried to generate human-style sketches by using CNN-based neural algorithm for art style (https://github.com/anishathalye/neural-style).Run through the HED network and then apply neural style transfer to get the result
  - Content image: edge images from real photos.
  - Style images: sketches from the database.
- The model syntheses images mixing content image with style from human sketches. We use our sketches as style sources and trained for 1000 iterations for style reconstruction.
- The results (shown on the right side) shows out that the 'sketchy style' learned are pretty similar to a thinning algorithm, as well as making the image a bit scribbled.

$$L_{total} = L_{content} + L_{style} + L_{total\_variation\_denoise}$$

$L_{content}$: mean-squared error loss between feature representations

L_{style} : mean-squared error between Gram matrixs

$L_{total\_variation\_denoise}$: remove noise

Conditional GAN:
- We developed a Conditional GAN based on a simple GAN network used on quickdraw dataset
- We resized the image to 28*28 and made inverted the black-white background and strokes
- HED edge map images were passed in to serve as conditional images

Generator: 28*28 condition image ->3136 dense->7*7*64 reshape->up sample with convolutional transpose layer-> sigmoid map

Discriminator: 28*28 conditioned image -> convolution layers with 64, 128, 256, 512, 1 filters->binary cross entropy loss

Example of Generated Images

Neural Style Transfer results:
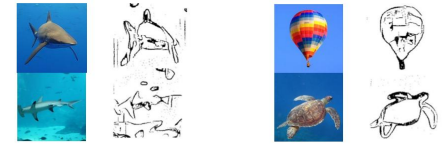
Conditional GAN results (blimp results, below is edge map):

10000 iterations

500 iterations
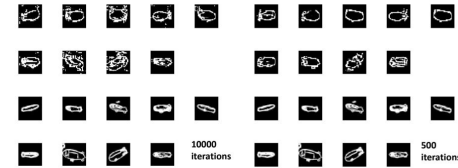
Conclusion & Discussion

- Sketch images are hard to generate with pure convolutional neural networks
  - Information is sparse
  - For the same picture, different humans can draw differently, so the model is essentially asked to learn a mapping that can represent many humans' sketching style
- GAN models are promising for generating sketch images
- Neural Style Transfer can be used to generate realistic sketch images

Future Work

- Develop better discriminator and generator models using pre-trained models
- Use RNN and LSTM with SVG format sketch data as model and output
- Try performing cross-domain embedding and cycle GANs