



2-Stage Conditional GAN for Sketch Auto-Coloring

Yu-shun Cheng

ysc270@stanford.edu

Introduction

Sketch auto-coloring has been a wild west, where lots of different approaches have been taken.

This project aims to achieve equal or better than state-of-the-art results, by adopting a novel idea coined by a new paper published in Nov 2018.

Approach

I chain 2 GANS back-to-back to get a finer image. The 1st GAN outputs the draft, which is then fed into the 2nd GAN for refinement. Both networks are conditioned on the same color hint for generator and discriminator. The loss functions are picked as:
 $G Loss : 0.01 * BCE(fake being real) + L1 (pixels)$
 $D Loss : BCE (fake being fake, real being real)$
Noise is sometimes required to bring G up to speed against D. Here I use random label flipping as noise for D, then I gradually reduce it as G gets stronger.

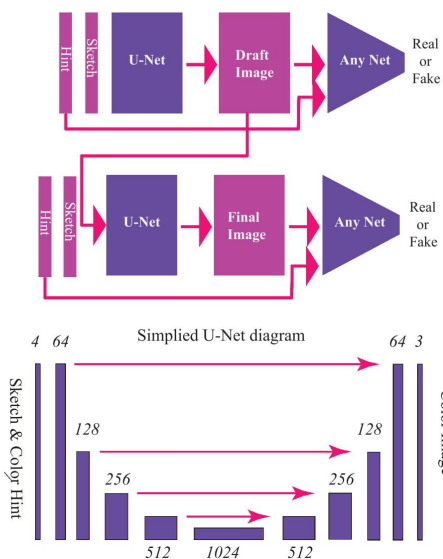
Dataset

- Danbooru2017 512px is used as the ground truth and source for generating the input images.
 - Naive edge detection to extract the "sketch"
 - A large gaussian kernel to blur the image as color hint
- Output images from first stage are then saved to disk in volume as another dataset to train the second GAN in parallel.

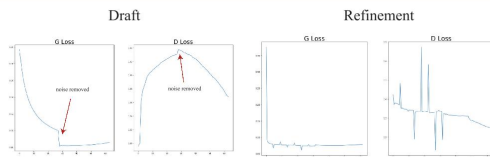
References

- Zhang et al. **Two-stage sketch colorization** Nov 2018
- kvfrans **Deepcolor** Mar 2017
- Ronneberger et al. **U-Net: Convolutional Networks for Biomedical Image Segmentation** May 2015
- Nazeri et al. **Image Colorization with GAN** Mar 2018

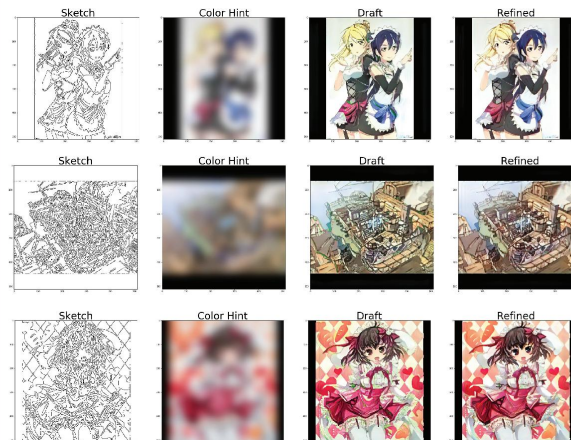
Architecture



Training



Result



Discussion

- It's not hard to observe that a few artifacts are corrected by the refinement stage:
 1. Color drift
 2. Color leak
 3. Thick lines
 4. Horizontal / vertical stripes
- We can use 2 smaller, easier to train networks to get higher quality images even with a very simple duplication scheme.
- Refinement generator trained with draft images collected from old epochs can be used to refine draft images collected from newer epochs.

Future Work

- Instead of a random white-out scheme for hint images to reduce model reliance on full coloring, which still is flawed, come up with a more robust way of giving hints.
- Naive edge detection seems to work okay to a certain extent, but it limits the amount of details an image can have, it's worthwhile to investigate how to best turn a colored image to sketch, by knowing the artist's workflow.
- Modify the loss to encourage color contrast.
- Build better tools