GANime: Generating Anime and Manga Character Drawings from Sketches

Tai Vu (taivu@stanford.edu), Robert Yang (bobyang9@cs.stanford.edu)

Problem and Motivation

Motivation: Producing fully colorized drawings from sketches is a large, costly bottleneck in the manga and anime industry.

Problem: The project aims to automate line art colorization with deep learning. The inputs are sketch drawings of anime characters.. The outputs are high-quality colorized images.

Solution: Neural Style Transfer, C-GAN and CycleGAN.

Data and Input Pipeline

Dataset:

- Anime Sketch Colorization Pair (Kaggle).
- 17769 pairs of sketch-color anime character images, 14224 examples for training and 3545 instances for testing. Each of them is an RGB image of size 512 x 1024.

Input pipeline:

- The training example and the ground truth in the same image are separated.
- All images are rescaled the image to 256 x 256 resolution and normalized to the range [-1, 1]
- We chose a batch size of 32 and shuffled the training instances in every epoch.
- We used random cropping and random mirroring.

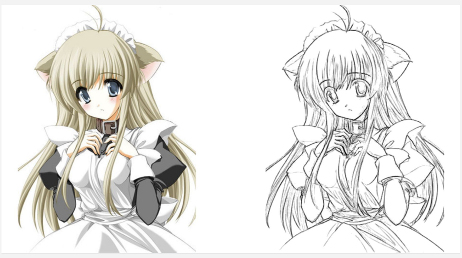


Figure 1. Color and sketch drawings in the same image.

Models

Neural Style Transfer - Baseline

Modeling: The training example is inputted as the "content" image, and the ground truth is inputted as the "style" image. We observe if the style (colors) from the ground truth can be transferred onto the generated image.

Two implementations:

- Fast Style Transfer - Arbitrary Image Stylization, provided by Tensorflow Hub.
- The original style-transfer algorithm with a pretrained VGG19 network. We trained the model for 1000 epochs.

Models, cont.

C-GAN (Pix2Pix)

Modeling: A generator G generates a colorized image from an input sketch image, while a discriminator D takes as inputs a sketch image and a color image and determines whether the color image is real or fake.

Architecture: The generator is a U-Net. The discriminator is a PatchGAN.

We also modified Pix2Pix with an added total variation loss.

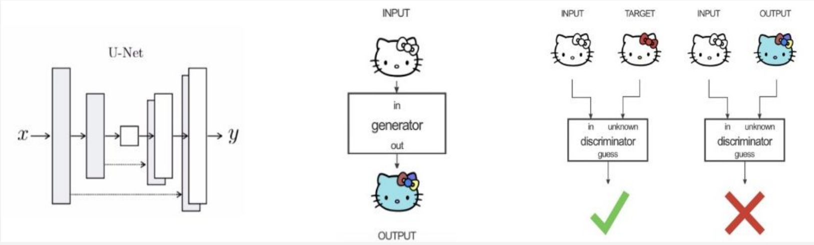


Figure 2. U-Net architecture (left), Generator (middle), Discriminator (right)

CycleGAN

Modeling: A generator G generates a colorized image from a sketch image, while a generator F generates a sketch image from a colorized image. Two discriminators D_X and D_Y distinguish between real and fake images.

Architecture: The generators are U-Nets. The discriminators are PatchGANs.

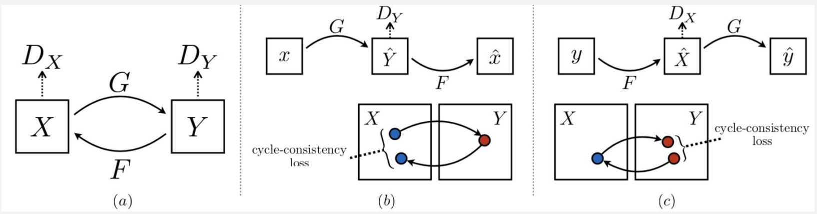


Figure 3. CycleGAN generators and discriminators

Experiments and Results

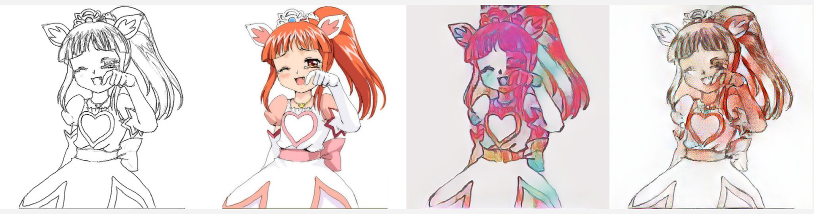


Figure 4. Results of Neural Style Transfer. Training example (first), ground truth (second), generated image - 1st model (third), generated image - 2nd model (fourth).

Neural Style Transfer: The same colors from the style image have been transferred onto the generated image, but the location of those colors are different than expected. Also, the first implementation produced an output image with many distinct colors, while the second one focused on transferring the main color of the style image to the generated outcome.

Experiments and Results, cont.

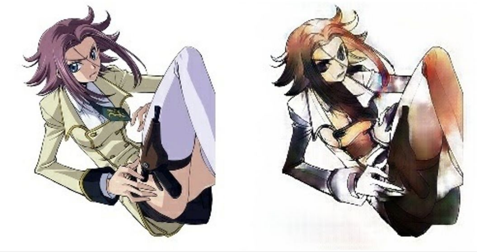


Figure 5. Results of CycleGAN. Ground truth (left), generated image (right).

CycleGAN: Visually, this model performs slightly better than the baseline, but far from perfect. The model focuses on two colors (black and brown), but doesn't learn to produce different colors.

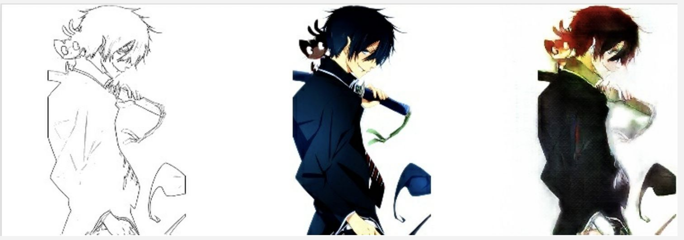
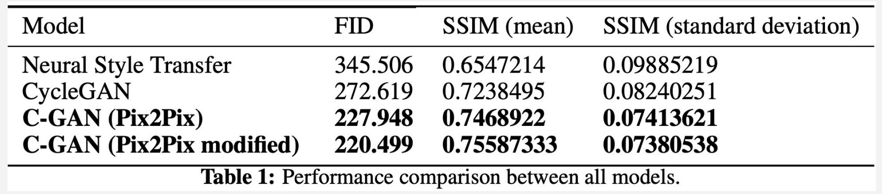


Figure 6. Results of C-GAN, epoch 146. Sketch (left), ground truth (middle), generated image (right).

C-GAN: Pix2Pix excels at coloring details of hair, skin and eyes. However, this is still imperfect with sophisticated images with many details, and sometimes still smudges dark colors. C-GAN has high performance overall. The added total variation loss further improves the outcomes by removing high frequency artifacts.



Model	FID	SSIM (mean)	SSIM (standard deviation)
Neural Style Transfer	345.506	0.6547214	0.09885219
CycleGAN	272.619	0.7238495	0.08240251
C-GAN (Pix2Pix)	**227.948**	**0.7468922**	**0.07413621**
C-GAN (Pix2Pix modified)	**220.499**	**0.75587333**	**0.07380538**

Table 1: Performance comparison between all models.

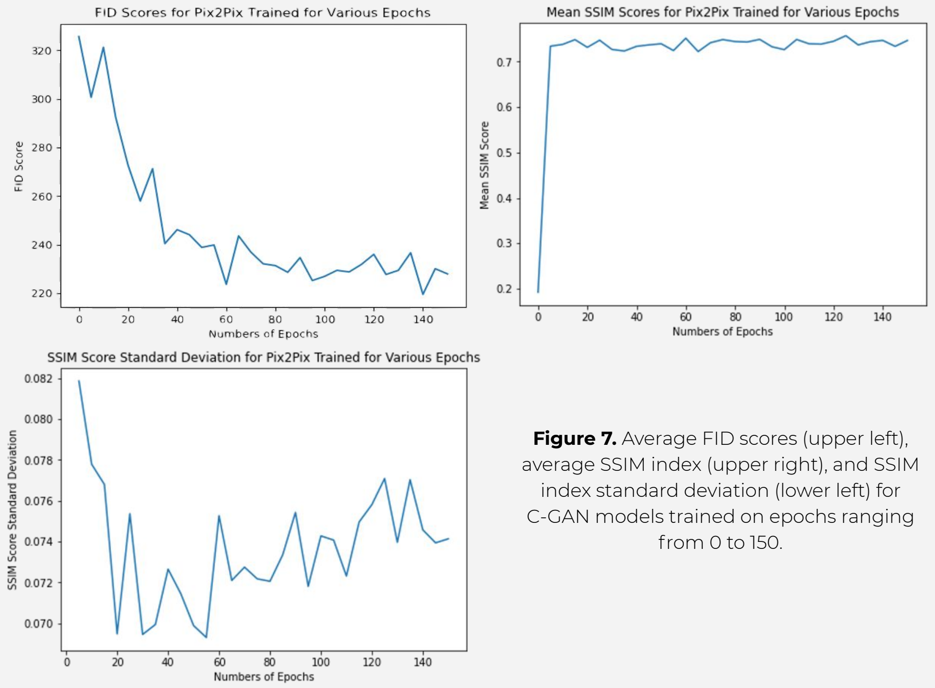


Figure 7. Average FID scores (upper left), average SSIM index (upper right), and SSIM index standard deviation (lower left) for C-GAN models trained on epochs ranging from 0 to 150.

Analysis of Results

C-GAN with a total variation loss has the best performance according to both the FID and SSIM metrics:

- The lowest FID (220.499)): The distribution of C-GAN-generated images is closest to the ground truth distribution.
- The highest average SSIM (0.75587333): C-GAN has the highest overall image quality.
- The lowest SSIM standard deviation (0.07380538): C-GAN has superior robustness to input image complexity.

C-GAN performs better since it quickly overcomes texture and color problems to focus on the harder part of learning detail:

- SSIM improves quickly until epoch 10 and then gradually levels out. The SSIM, which focuses on the texture of the image, is only good at distinguishing bad texture (i.e. grainy or grid-shaped coloring). Thus, the C-GAN focuses on outputting the right texture from the start to epoch 10.
- FID improves quickly until epoch 35 and then gradually levels out. The FID is able to detect both "texture" and "color" problems, so from epoch 10 to epoch 35, the C-GAN focuses on improving its skills at coloring one area with one single color (as opposed to two colors smudged together).
- The FID and SSIM scores improve slightly from epoch 35 onwards, even though qualitatively, the images show improvement until epoch 100. This is because the GAN focuses on learning small details after epoch 35 that the two metrics do not weight heavily.

Conclusion and Future Work

Summary: Our models produce decent outcomes on this anime colorization task, with the modified C-GAN yielding the best performance as it improved past its texture and color problems.

Next Steps:

- Fine-tuning hyperparameters.
- Designing a real-fake experiment to test models' performance based on human perception;
- Using 512 x 512 resolution to generate high-quality outputs.
- Experimenting with alternative models (GANs and conditional VAEs) and modifying network architectures (ResNet, ImageGAN).
- Doing transfer learning with pre-trained weights for Pix2Pix.
- Utilizing different color spaces.
- Conditioning on certain colors to give the user more control.

References

- Gatys, Leon A., et al. "A Neural Algorithm of Artistic Style." ArXiv:1508.06576 [Cs, q-Bio], Sept. 2015. arXiv.org.
- Isola, Phillip, et al. "Image-to-Image Translation with Conditional Adversarial Networks." ArXiv:1611.07004 [Cs], Nov. 2018. arXiv.org.
- Zhu, Jun-Yan, et al. "Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks." ArXiv:1703.10593 [Cs], Nov. 2018. arXiv.org