Realistic Colorization of Portrait Line Art

Mehul Agarwal
Department of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213
mehula@andrew.cmu.edu

Julia Shuieh
Department of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213
jshuieh@andrew.cmu.edu

Abstract

This paper aims to explore line-art colorization of human faces using Generative Adversarial Networks. Past colorization models have either focused on digital art (taken from or inspired by comic-books and mangas) or black and white photographs, each employing different architectures and models. Adding realism by exploring sketches of human faces brings in its own set of challenges. However, as there are no datasets for this problem, we will compare different lineart generators and use them to construct our training dataset for our pix2pix-based model.

1 Introduction

Lineart colorization techniques (using Generative Adversarial Networks and otherwise) have been explored in depth by the Machine Learning community. But as per our knowledge, they have been restricted to paintings and comic-book/manga pictures. We wish to explore its use case in more realistic scenarios, with human subjects. Owing to the variety of human face datasets, this task becomes truly large-scale.

In terms of real-life applications, this could set a standard for human and machine-drawn line-art, with the ability now to compare the drawing to the actual face. This would be very important for identifying people from police sketches or missing people posters. It can also be used to imagine a person with different hair-color, clothing and styles. Finally it also brings in creative applications with simple line-art being used to generate completely new never-seen-before realistic faces and characters.

2 Related Work and Background

2.1 pix2pix

pix2pix was one of the pioneer works in unsupervised image to image translation. Given two datasets X and Y, pix2pix would use GAN (Generative Adversarial Network) based methods to transform an image from domain X to domain Y. They did this by training a mapping $G : X \rightarrow Y$ adversarially (using the standard Generator-Discriminator co-training in GANs) such that any image $G(x_i)$ is indistinguishable from any $y_i \in Y$.

This technique has had wide success in the field of computer vision.

2.2 Manga and cartoon line art colorization

Following the success of GAN based techniques in CV, many papers have used this technique to train models to colorize drawn art. This technique has especially seen success in manga colorization, going from lineart to colorized mangas.

34th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, Canada.

This is due to the breadth of dense manga datasets. A few notable papers include [1] and [7]. This has also been adapted to online tools such as Petallica Paint.

A few techniques we use to generate our dataset such as using sketchKeras and sketch simplification (discussed below) are inspired from data augmenting techniques used in these papers.

3 Task Setup and Data

Owing to the specificity of our project, there were not any datasets that directly could be used to train our models. To address that, we decided to generate lineart from popular face datasets using pre-trained models to train alongside ground truth full-color faces, including the Tufts, UTK, and Flickr datasets. However, the Tufts dataset only had very similar portraits with gray backgrounds and the UTK dataset were cropped only to faces, often missing context like hair and clothes. Hence, for the baseline, we decided to take portraits from the Flickr dataset [2] as it had more diversity in the surroundings and types of portraits.

Following initial experiments, we quickly realized that the lineart in the training set greatly impacted the quality of generated images. Therefore, we used three different pre-trained models to generate lineart for training.

3.1 ArtLine

We generated the lineart using a model called ArtLine [1]. ArtLine employs self-attention, Progressive Resizing and a Perceptual Loss/Feature Loss based on VGG16. It was trained over APDrawing dataset and the Anime sketch colorization pair dataset. It creates fairly high quality line art for portraits, though we did notice that compared to hand-drawn sketches, it does not have much shading and tends to not include lighter features. This causes our baseline models to falter on shading-heavy hand-drawn sketches.

3.2 sketchKeras

sketchKeras [4] combines the results of a high-pass algorithm via OpenCV and Neural Network-based holistically-nested edge detection (HED) such as PaintsChainer's lnet to generate lineart. It is quite a common model used to train digital art colorization models, giving very succesful results. Though it provides better shading than Artline, since it hasn't been trained on human faces, the lines generated are extremely light.

3.3 sketchKeras and Sketch Simplification

To alleviate the lightness of the sketchKeras model sketches, we run it through the sketch simplification model [3]: a convolutional Networks for rough sketch cleanup. This is a technique employed by a lot of the digital art colorization models. However, in the context of human faces, their effectiveness is hard to determine, since for many sketches, they get rid of a lot of context such as eyes, hair, etc. reducing a face to just a few lines.

Figure 1: Generated lineart from all three models.

4 Baselines

We decided to use pix2pix as our main baseline model as it is one of the standard image-to-image translation models, and it can be easily trained with our dataset. Although we believe it has not been used with this specific problem, it has previously been used with similar problems, such as coloring more cartoon-ish sketches. To set up our training, we generated the lineart of 1000 images and used 800 of those for training for 200 epochs. For each image, the colored and lineart versions had to be side-by-side to one image for training. The lineart was generated using the three methods described earlier.

5 Proposed Model

For our proposed model, we continue to use pix2pix. However, for our training data, we combined the generated images from all three lineart models, as well as the black and white version of the source image. As we have found that the training time scales approximately linearly with the number of training images, and we wanted to use more images for our final model, we used data parallelization.

In our experiments, we increased our number of source training images to 1000 and 2000, which led to 4000 and 8000 total trainin images respectively. We tested the training time for 1 GPU compared to 4 GPUs. In the 1 GPU case, the instance had a Tesla T4 GPU with 16 GB memory, while in the 4 GPU case, the instance had Tesla V100 GPUs with 16 GB memory.

6 Analysis of Results

For all three synthetic datasets, our results for generated lineart of portraits look fairly realistic, especially for the faces. There are more differences in the background, but they are still usually colored realistically. However, there are some areas that have some noise or unsmooth coloring, particularly where there would be shadows or contouring in a photograph. They did not perform well in non-generated lineart, or actual lineart drawn by real people. We believed this is due to the lack of shading or other components in the generated lineart that may be present in the real lineart. All three synthetic datasets give wildly different generated coloring, implying the type of lineart we train the model on significantly impacts results.

Table 1: Performance Metrics for Baseline Models

	G_{GAN}	G_{L1}	D_{real}	D_{fake}
ArtLine	1.125	17.400	0.023	0.661
sketchKeras	1.773	29.065	0.011	0.214
sketch + simple	4.221	20.207	0.011	0.034

6.1 Artline

Figure 2: Generated images from ArtLine model.

This model performed best reconstruction of color for images similar to the dataset (the 200 images left for testing). It was successful in getting subtleties such as strokes of hair color, skin and teeth. From a more subjective point of view, the generated images had a water-color painting look to them.

We also tested this model with images not generated with the ArtLine model, such as hand-drawn lineart. We can see that the results are vastly different, with most of the image being colored with considerable amount of noise. This may be due to the images having more shading compared to the generated lineart, but even the handdrawn police sketch with minimal shading had unrealistic coloring. This suggests that perhaps we need to explore other models for generating our portraits to better represent these kinds of lineart in our training data.

6.2 sketchKeras

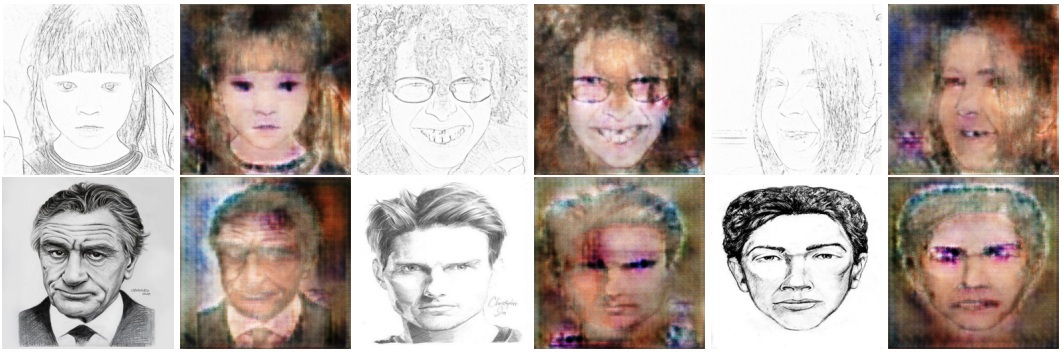

Figure 3: Generated images from sketchKeras + simplified model.

Judging by the performance of this model on the 200 images left for testing, it was only successful in getting general context missing eyes, hair color, etc. From a more subjective point of view, the generated images look as though printed from a dot-matrix printer that has run of ink.

However, this model performed comparatively better with images not generated with the ArtLine model, such as hand-drawn lineart, isolating background, defining boundaries and identifying general facial structure.

6.3 sketchKeras + sketch simplified

Figure 4: Generated images from sketchKeras + simplified model.

Owing to the general context such as eyes, hair and facial features missing from the lineart, the model had an even more vague reconstruction. Unsurprisingly, it missed hair color and style reconstruction, going for blonde/black hair around the general face.

However, surprisingly, this model performed relatively the best with images not generated with the ArtLine model, such as hand-drawn lineart. It successfully isolated the background in most cases and set firmer face and hair boundaries, perhaps owing to the larger number of lines in the hand-made sketches compared to the rarer sporadic number in the training examples.

6.4 Combined dataset

We used data parallelization using 4 GPUs to train in a data-parallel manner. This was useful given the density of the model (2000 source images => 8000 total images) compared to 1000 source images for the baseline models. From Table 3, we can see that this did benefit the training time by a significant amount, and this would be even more significant if we had more epochs and/or training images.

Owing to a larger dataset with all three sketching models and black and white images, the combined dataset performed the best on variety of scenarios, including the hand-drawn lineart. However, interestingly, for the hand-drawn lineart fed to the model, there were interesting pink artefacts left in some images (refer to Figure 5).

The only images that the model properly failed on were comic book lineart (with characters that had more cartoony features rather than realistic).

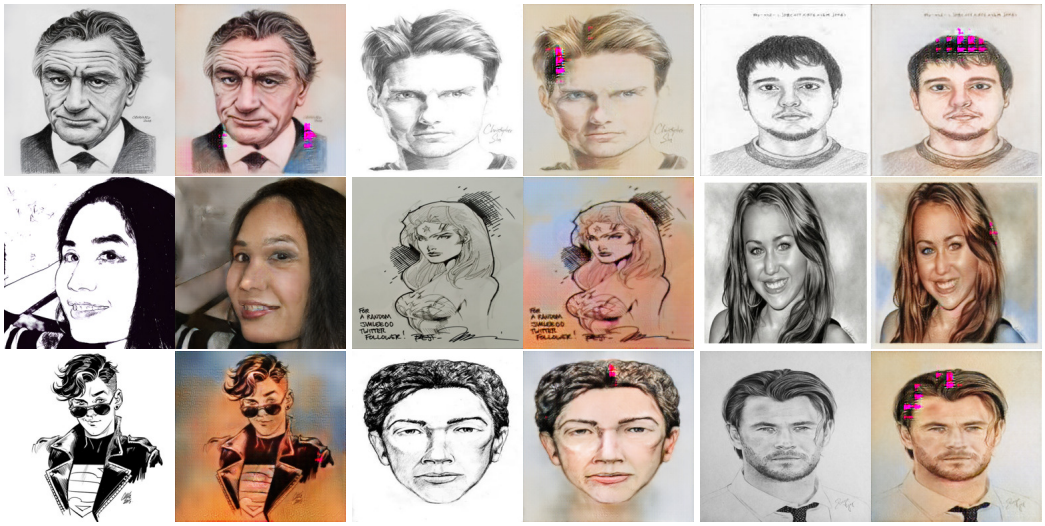

Figure 5: Generated images from 2000 training set for final model.

Table 2: Performance Metrics for Final Model

Number of Source Images	G_{GAN}	G_{L1}	D_{real}	D_{fake}
1000	2.224	24.073	0.000	0.245
2000	2.267	27.352	0.000	0.303

Table 3: Epoch Time for One GPU vs Multi-GPU

Number of Source Images	1 GPU	4 GPUs
1000	287	156
2000	563	241

7 Future Work and Limitations

Due to the limited time to work on this project, we did not explore possible training setups as much as we have wanted. To improve the quality of the colorization, more source images or hyperparameter

tuning may return better results. We can also explore different lineart generators that may represent actual lineart better. Due to some noise in the resulting images, we can also explore post-processing techniques that would clean up noise.

If possible, we can create a new dataset to train this kind of model with. For example, a police sketch dataset could be used, where we can colorize the police sketch to the actual person. However, the usability of this dataset may depend on the quality of the police sketch, as some sketches may not be as realistic.

Finally, images often include parts that may be colored in many different ways, such as the color of a person's hair or clothes. It would be nice to describe or sloppily color (give hints) parts of the images, and have the model follow the hints and color in the image in a realistic manner. Similar models have been trained for comic and cartoon characters, so this suggests that this is achievable with realistic images.

References

[1] Ci, Y., Ma, X., Wang, Z., Li, H., & Luo, Z. (2018). User-guided deep anime line art colorization with conditional adversarial networks. *Proceedings of the 26th ACM International Conference on Multimedia*. https://doi.org/10.1145/3240508.3240661

[2] Isola, P., Zhu, J.-Y., Zhou, T., & Efros, A. A. (2017). Image-to-image translation with conditional adversarial networks. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. https://doi.org/10.1109/cvpr.2017.632

[3] Madhavan, V. (2022). *ArtLine* [Source code]. https://github.com/vijishmadhavan/ArtLine

[4] Nvidia. (2019). *Flickr Faces HQ Dataset* [Data set]. https://github.com/NVlabs/ffhq-dataset

[5] Simo-Serra, E. (2019). *Sketch Simplification* [Source code]. https://github.com/bobbens/sketch_simplification

[6] Zhang, L. (2017). *sketchKeras* [Source code]. https://github.com/lllyasviel/sketchKeras

[7] Zhang, L., Li, C., Simo-Serra, E., Ji, Y., Wong, T.-T., & Liu, C. (2021). User-guided line art flat filling with split filling mechanism. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. https://doi.org/10.1109/cvpr46437.2021.00976

[8] Zhu, Jun-Yan. (2022). *CycleGAN and pix2pix in PyTorch* [Source code]. https://github.com/junyanz/pytorch-CycleGAN-and-pix2pix