Image-to-Image Translation
(CSE 803 Project Progress)

Asadullah Hill Galib (Single Project)
CSE, Michigan State University
galibasa@msu.edu

Abstract

Image-to-image translation is a popular and growing field in computer vision that deals with many sorts of mapping between an input image and an output image. This field is booming currently thanks to the incorporation of deep learning techniques, particularly generative modeling techniques. It has a variety of applications, such as image synthesis, segmentation, style transfer, restoration, and pose estimation, etc. In this project, edge-to-image - an application of image-to-image translation will be explored. The goal of this project is to generate colored images from edge images using a generative model. Conditional GAN-based architecture will be incorporated to accomplish the goal.

Figure 1. Examples of Image-to-Image Translation (image source: [7])

1. Problem Definition

The goal of image-to-image translation is to transfer images from one domain to another while keeping the content representations intact. Due to the introduction of deep learning techniques, notably generative modeling approaches, this subject is now thriving. It has a wide range of applications, including translation of images from day to night, translation of images from season to different seasons, translation of edge to image, translation of semantic segmentation to image, translation of satellite photographs to Google Maps, translation of black and white photographs to color, translation of sketches to color photographs, etc. Figure 1 refers some examples of image-to-image translation.

Generically, all of the aforementioned applications have tried to generate output images condition on the input images. This conditional theme for the image to image translation is promulgated by Isola et al. [5] in their benchmark study in this field. They employed a model called the conditional Generative Adversarial Network [6], or cGAN for image-to-image translation in general, where the generation of the output image is conditional on the input image. This cGAN is a type of GAN [3] architecture. Followed by this benchmark work, a lot of variations of the GAN architecture are proposed for image-to-image translation, such as CycleGAN [9], StarGAN [2], AttGAN [4], etc.

This project aims at generating images from edges/sketches. Given, edge image representation and the corresponding original image, a cGAN model will be trained as described in [5] for generating and discriminating produced/output image. The discriminator is given a source image and a target image and is asked to decide whether the target is a realistic transformation of the source image. Adversarial loss is used to train the generator, which encourages it to generate believable images in the target domain. L1 loss between the generated image and the intended output image is also used to update the generator. The generator model is encouraged to construct credible translations of the original image as a result of the additional loss.

So, any dataset containing edge image - original image pair can be used here. This kind of datasets exists, such as Danbooru [8], Pokemon [1], etc. Also, automatic edge detected (using edge detector techniques) from sample images can also be used as the dataset. In terms of evaluation, one or more of the following pixel-level error metrics will be used: mean absolute error, mean squared error, cross-entropy. According to [5], the underlying U-net-based (with skip connection) generator has to be implemented first. On

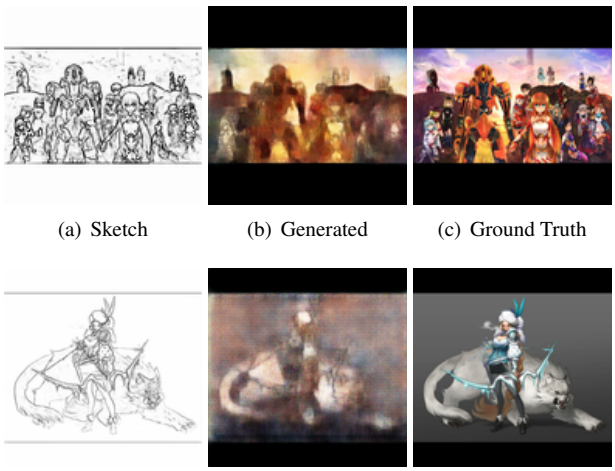
top of that, the GAN architecture has to be implemented. Finally, the U-net-based Generator and the full GAN architecture will be compared.

2. Progress

So far, the author read through and understand the concepts and resources of this project, such as U-net, GAN, cGAN, and the based study [5], etc. Also, dataset selection, data processing, data exploration are carried out. The first building block of the GAN architecture: the U-net-based generator is implemented and evaluated. The details of the progress is described in this section.

According to the project aim of generating colored images from sketches/edges, a relevant large-scale dataset is selected. That is Danbooru Sketch Pair [8]: a large collection of 128x128 anime pictures and sketch pair dataset converted from Danbooru2017. First, the image-sketch pairs are extracted and then processed. Images are converted into RGB and normalized to 0 means. Training, validation, and test sets are created from the full dataset.

The first part of the GAN architecture: Generator is implemented using the underlying U-net convolutional architecture. This U-net-based Generator consists of an encoder-decoder pair. The encoder down-samples the original image sequentially using Conv2D, ReLU, Batch Normalization layers with strides. The decoder takes the down-sampled version of the original image and starts up-sampling it in a reversed way and generates the output. To guard against spatial information losing and to avoid vanishing/exploding gradient problems, skip connections are used from the individual layer of the encoder to the corresponding layer of the decoder.



(a) Sketch (b) Generated (c) Ground Truth

(d) Sketch (e) Generated (f) Ground Truth

Figure 2. Sketch to Image using UNet-based Generator

To train the Generator U-net, L1 loss (mean absolute error) is used in the loss function. L1 loss is calculated pixel-wise. The model optimizes using the loss function to generate images as close to the original images. Sample generated images using U-net Generator are shown in Fig. 2. The learning curve is depicted in Fig. 3.

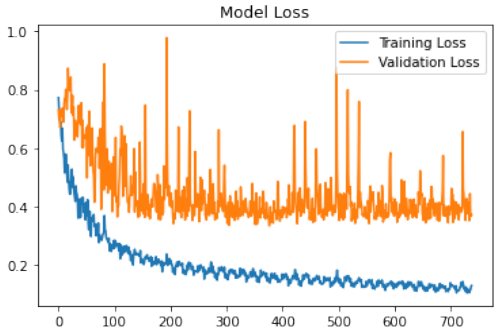


Figure 3. Learning curve of the U-net Generator (using a fraction of the dataset and 50 epochs)

3. Issues/Challenges

The author finds it difficult to manage the large dataset (10.8 GB) and train it without an easily accessible GPU. So far, Google Colab is used for GPU access but managing the large dataset on the cloud is troublesome. So, the experiments are carried out by using a small fraction of the full dataset and a small number of epochs. Hopefully, this issue will be mitigated by using other alternatives, like MSU HPCC.

4. Next Step

The following works have to be carried out next:

- Implementing the Discriminator part and combining the full GAN architecture.
- Evaluating, fine-tuning, and writing report.
- Optional works (if possible in time): Run on other image-to-image translation applications, especially image-to-edge-to-image generation.

References

[1] Doron Adler. Sketch2pokemon, Oct 2019.

[2] Yunjey Choi, Minje Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, and Jaegul Choo. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8789–8797, 2018.

[3] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020.

[4] Zhenliang He, Wangmeng Zuo, Meina Kan, Shiguang Shan, and Xilin Chen. Attgan: Facial attribute editing by only changing what you want. *IEEE transactions on image processing*, 28(11):5464–5478, 2019.

[5] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017.

[6] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.

[7] Yingxue Pang, Jianxin Lin, Tao Qin, and Zhibo Chen. Image-to-image translation: Methods and applications. *arXiv preprint arXiv:2101.08629*, 2021.

[8] Wuhecong. Danbooru sketch pair 128x, Nov 2019.

[9] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232, 2017.