Ganilla : Generative Adversarial Networks for Image to Illustration Translation

Ganilla Icecream: Zoe Beckman, Minna Kimura-Thollander, Liyaan Maskati, Katherine Sang.
Brown University

Abstract

We implemented the paper GANILLA: Generative Adversarial Networks for Image to Illustration Translation, which performs image translation and stylization using children's book illustrations as the source. Though typical style transfer projects focus on the works of famous fine artists like Van Gogh and Monet, we explore unorthodox illustration domains and train a modified CycleGAN from scratch in an attempt to reproduce different artistic styles. We trained our model on unpaired images to translate landscape photographs to illustrations using (1) stills from Studio Ghibli films and (2) David McKee's children illustration books.

1. Introduction

Image translation and stylization is a common problem within computer vision, but often requires paired images in two domains which can be difficult to obtain. Therefore, much of contemporary computer vision research centers on models that can produce convincing results without training on paired images. GANILLA is one such model that can be multi-purposed for image stylization, image colorization, and general translation (ex. converting a winter landscape turned into a summer landscape). [1]

GANILLA's primary use case is as a tool for artists working in industry. In animation, compositing and developing backgrounds is a time-consuming task that is typically done by Photoshopping photos and painting on top of them. GANILLA can expedite this process by converting landscapes into the target illustration style, saving artists valuable time. Additionally, the model may be used as a novel source of entertainment, given the success of other photo and video editing platforms such as Instagram and Snapchat.

2. Related Work

Previous model architectures that work with style transfer and image translation include Pix2Pix and CycleGAN. The GANILLA architecture is a modification of CycleGAN.

Pix2Pix is a GAN that can be used for a variety of tasks such as image colorization and remote sensing. However, a limitation of Pix2Pix is that it requires paired images from the source and target domain. [2]

CycleGAN was developed as a response to models such as Pix2Pix. CycleGAN does not require paired images, which allows it to be flexible for a variety of tasks. CycleGAN is unique because it combines two generators such that the input of the first generator must match the output of the second. This structure solves the issue of paired images. [3]

CycleGAN leverages two other architectures: ResNet for its generators and PatchGAN for its discriminators. Residual networks are often used for image recognition, thus are useful for downsampling images. For the discriminator, PatchGAN is particularly robust in that it assesses multiple patches of the image to determine if it is real or fake. [4] We will discuss these architectures in more detail in the next section.

3. Method

3.1. Data

We used two illustration datasets: one set of images from Studio Ghibli films and one set of illustrations from David McKee's children's book series *Elmer*. The Studio Ghibli images were scraped from Google Images using a Python script, amounting to 778 images in total. The David McKee images were scraped from OpenLibrary using Selenium, which amounted to 274 images in total. Our landscape dataset was obtained from Kaggle and contains over 3000 images.

Figure 1: Four McKee examples.

Figure 2: Four Studio Ghibli examples.

3.2. Model

The GANILLA architecture includes two generator-discriminator pairs. The first generator translates from landscape photograph to illustration, and its corresponding discriminator assesses whether the generated image is an illustration or not. The second generator translates from illustration to landscape photograph, and its corresponding discriminator determines whether the generated image is a photograph or not. This setup is shown in Figure 3. Under this architecture, we expect that the intermediary generated image will retain the composition of the input image but inherit the style of the illustration.

GANILLA contains a weighted sum of losses to optimize our generator, which include (1) the cyclical loss and (2) the identity loss. The cyclical loss enforces that there is no difference between the input image to the first generator and the output of the second generator, as we expect that the second generator reconstructs the input to the first. On the other hand, the identity loss enforces that the output of the generators should match the input when the input is from the target domain.

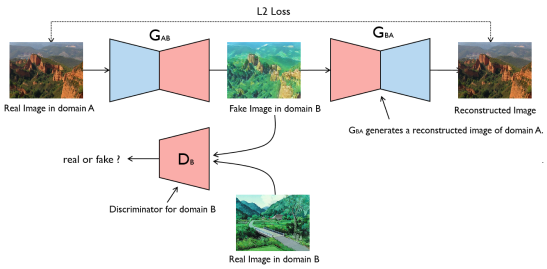

Figure 3: The GANILLA architecture.

The generator follows an encoder-decoder model to downsample/compress the source image and upsample it into a stylized version. The downsampling phase utilizes four layers, each of which contain two Resnet-18 blocks, max pooling, and instance normalization. We modified the original architecture in the paper by removing two of the four layers due to concerns about runtime and out-of-memory errors. The outputs from the downsampling blocks are concatenated with our upsampling layers, which allows us to pass the lower level features from the downsampling blocks to the upsampling blocks. As a result, the original image composition is preserved.

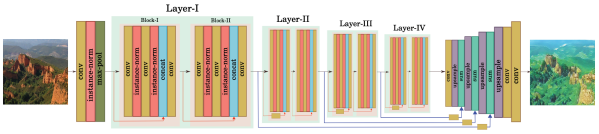

Figure 4: The generator architecture.

The discriminator is a PatchGAN, a CNN that takes in input images of size (256 x 256). For every (70 x 70) image patch, it determines whether the image is real or fake, and then averages the predictions of each patch to determine if the overall image is real or fake.

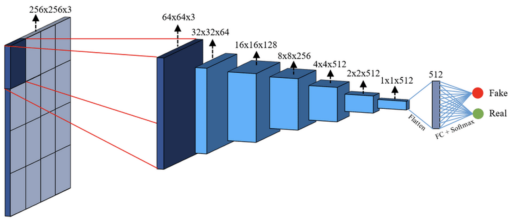

Figure 5: The discriminator architecture.

With regards to train time, we trained the model for 200 epochs, at a rate of 6 to 8 minutes per epoch on a V100 GPU with 30 GB of RAM. We had over 100+ hours in training.

4. Results

We implemented and trained two GANILLA models for the two sets of data, David McKee and Studio Ghibli. In this section, we present both qualitative and quantitative metrics to evaluate our results.

4.1. Qualitative Metrics

The model trained on McKee's illustrations preserved the general outline of the landscape images; in particular, the light and dark values of the images are consistent, even if the colors are more reminiscent of McKee's color palette. Additionally, all of the generated images seem to follow a more flattened style that resemble McKee's actual illustrations. However, we note that none of the images are particularly convincing as landscapes, which can likely be attributed to the fact that our dataset for McKee is very small.

Figure 6: Results for David McKee for epochs 50-100.

In comparison, the model trained on Studio Ghibli images was much more successful. The generated images preserve the composition of the landscape while transferring the signature Ghibli style (note the saturated viridian greens & cerulean blues and the painterly texture). While Studio Ghibli landscapes tend to have realistic composition and color, they do have an illustrative style that make them distinct from photographs, which our model appears to emulate.

Something interesting to note about these results is that our outputs tend to have a green tinge to them. Many Studio Ghibli films take place in fields and forests, thus most of our training dataset consists of green images. After making this observation, we were curious to see how the model would stylize a black and white image. As shown in the bottom left corner of our results table, when the model was given a greyscale forest, it colored the image green.

Figure 7: Results for Studio Ghibli for epochs 100-150.

More generated images can be found in our Github repository under the results directory.

4.2. Quantitative Metrics

For quantitative metrics, we evaluated the efficacy of image reconstruction on both of the models we trained. The loss that we compute minimizes the difference between the input image to the first generator and the output of the second generator, as they should be identical. Thus, our quantitative metrics assess the difference between these two images.

We took 100 test images to generate the following assessments. The three metrics that we used were:

- *MSE*: measures the squared difference between each pixel. Lower MSE is better.
- *PSNR*: assesses the amount of noise in the compressed image. Higher PSNR is better.
- *SSIM*: quantifies image quality degradation between two images. Unlike MSE and PSNR, SSIM accounts for the difference in compositional structure between the two images. A value closer to $+1$ is better.

Our quantitative metrics for both McKee and Studio Ghibli are presented in the tables below:

Epoch	MSE	PSNR	SSIM	
50	688.722	2030.513	0.621	
100	620.396	2089.216	0.600	BEST
150	1101.864	1816.460	0.572	
200	913.559	1901.338	0.585	

Figure 8: Quantitative metrics for David McKee.

Epoch	MSE	PSNR	SSIM	
50	471.162	2238.550	0.738	
100	454.322	2236.677	0.725	BEST
150	581.006	2120.833	0.649	
200	621.682	2099.616	0.643	

Figure 9: Quantitative metrics for Studio Ghibli.

As seen in Figures 8 and 9, both models achieved the best performance around epoch 100. This was surprising to us, as the original GANILLA paper states that they trained their models for 200 epochs. Since training GANs is relatively unstable, it is likely that our GAN model experienced some mode collapse as the epoch count increased.

Something that is interesting to note is that the SSIM metric is the highest in the earlier epochs. This may be because in the earlier epochs, our illustration generator performs less radical modifications to the given image, and thus may be easier to convert back into the input image. In general, the MSE and PSNR metrics are worse at epoch 150 and 200; something to investigate in the future would be to load the model at epoch 100 and retrain it to see if the poor results are due to unstable training.

4.3. Technical Discussion

The largest architectural modification we made was reduce the number of layers in our generator. We were initially concerned that this reduction would be too extreme and produce poor results. However, we observed that our generated images for Studio Ghibli retained the composition of the original image while introducing the color palette and style of the illustrations, meaning we were successful.

One of the failings of our generator for Studio Ghibli is that its outputs tend to have a tinge of green. This can be seen most clearly when testing black and white input. However, we attribute this tendency towards green to the

skew of our training data rather than model architecture itself. Studio Ghibli films often take place in nature, so many of our training images happen to be green. A similar effect can be seen with the David McKee outputs, in which the original color of the landscape is often distorted in favor of light greens, yellows, and reds which resemble McKee's original illustrations.

Another technical issue with our architecture is that it only allows training on a batch size of one. Since the architecture does not allow batching, our model trains quite slowly. After doing some research online, we found that that similar architectures such as CycleGAN have significantly worse performance when images are batched. Thus, it may be worthwhile to investigate how the framework could be modified to support batching while still retaining good results.

5. Conclusion

We observed overall success in reimplementing GANILLA. While being unable to obtain enough data resulted in issues for our David McKee model, our Studio Ghibli model was particularly successful in retaining the composition of the input images and applying color and texture reminiscent of Ghibli films.

With regards to improvement, there are plenty of ways our model can be refined. For instance, finding illustrators who have a more comprehensive body of work to use as a training dataset should radically improve the results of our model. Additionally, modifying the architecture to support batching should help decrease the training time.

In terms of possible extensions, we noticed that our resulting model was able to semi-successfully colorize and stylize greyscale images with the color and texture profile of the training dataset (as shown in Figure 7). A line of further inquiry worth investigating would be to train a more robust network that is capable of transferring illustrative color and style to both RGB and greyscale images.

References

[1] E. Akbas P. Duygulu S. Hicsonmez, N. Samet. Ganilla: Generative adversarial networks for image to illustration translation, 2015. 1

[2] T. Zhou A. A. Efros P. Isola, J. Zhu. Image-to-image translation with conditional adversarial networks, 2018. 1

[3] P. Isola A. A. Efros J. Zhu, T. Park. Unpaired image-to-image translation using cycle-consistent adversarial networks, 2014. 1

[4] S. Ren J. Sun K. He, X. Zhang. Deep residual learning for image recognition, 2015. 1