Visualizing adverse effects of Global Warming on Mountains through CycleGAN

Gunguk Kim
Department of Computer Science
Stanford University
gunguk@stanford.edu

Troy Shen
Department of Computer Science
Stanford University
troyshen@stanford.edu

Abstract

The problem of neural style transfer has been approached in a variety of ways. In our research, we aim to create a visualization of the adverse effects of climate change on the melting of mountains through style transfer. To do so, we build on and modify Zhu et. al's CycleGAN with the purpose of finding a forward mapping function that maps from the domain of snowy mountain images to the domain of non-snowy mountain images. As our motivations dictate that we generally care more about the effectiveness of the forward mapping function, we loosen the constraints on cycle loss consistency in the hopes of generating more photo-realistic forward-mapped images by utilizing weight decay, making changes to the architecture, and using skip layers. We hope our work sheds light on the ways that generative models can be improved upon for future research and in service of solving real-world problems.

1 Motivation and Problem Statement

According to the Intergovernmental Panel on Climate Change, "on average across Western North America, the European Alps and High Mountain Asia, temperatures are warming by 0.54 degrees Fahrenheit per decade."[4] The melting of enormous reservoirs of snow are causing changes in mountain river flows, disrupting plant and wildlife, and increasing the risk of extreme rockslides, avalanches, and mountain floods.

Unless bold policy changes are instituted to counteract climate change, the melting of mountains will only exacerbate ecological issues. Leveraging the fact that people react more strongly to visual cues, our project aims to use neural style transfer to depict how some of the world's most iconic mountain ranges will look if climate change is not properly addressed. Specifically, we will be building on Zhu et. al's CycleGAN in an effort to achieve this style transfer through a generative forward mapping [8].

2 Data Set

We utilize two datasets: 845 photos of Yosemite in the summer and 209 photos of iconic mountain ranges that are rapidly melting due to climate change (Mount Ranier, Mount Everest, and the Kebnekaise Mountain Range).

The Yosemite summer images were pulled from an existing data set curated by Zhu et. al using Flickr API. We manually cleaned the data to get rid of images that did not fit the necessary criteria (e.g. winter photos, people in photos, etc).

The snowy mountain photos were manually scraped using ParseHub's architecture under the criterion that each photo depicted a clear mountain top or mountain range with sufficient amounts of snow. The data set was split using a 90:10 ratio of training to test set.

3 Approach

3.1 Model and Methodology

The goal of the CycleGAN is to learn image-to-image translation from unpaired datasets. It achieves this through the use of two generators and two discriminators by learning two mappings: one seeks to map images from the domain of snowy mountain images to the domain of non-snowy mountain images while the other seeks to map images from the domain of the latter to the domain of the former [9].

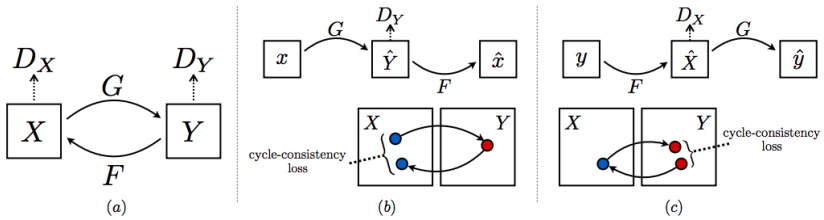


Figure 1: The CycleGAN Model

The overall objective (below) is to minimize the loss function, namely the adversarial loss of both generators and the cycle consistency loss. The cycle consistency loss aims to account for the fact that translating from an snowy mountain image to a non-snowy mountain image and back again should result in the original image.

$$\mathcal{L}(G, F, D_X, D_Y) = \mathcal{L}_{GAN}(G, D_Y, X, Y) + \mathcal{L}_{GAN}(F, D_X, Y, X) + \lambda \mathcal{L}_{cyc}(G, F) \qquad (1)$$

3.2 Initial Errors and Lessons from Forest to Deforest Dataset

We initially wanted to use an unpaired data set of forest images and deforested images to depict the effects of deforestation around the world. However, we quickly discovered a weakness of CycleGAN, namely its ineffectiveness at performing geometric transformations.

(a) Forest Input (b) Deforest Input (c) Forest Output (d) Deforest Output

Figure 2: Example Output on Epoch 16

As seen with this example on epoch 16, the algorithm fails to learn the absence of trees in the deforest domain and thus cannot perform the geometric transformation of deleting trees in the forward mapping function. Despite our unsuccessful findings with the original dataset, we observed that CycleGAN was very good at learning textual and color changes. This knowledge helped us think of and curate the current datasets which we believed would perform better with the CycleGAN algorithm.

3.3 Modifications to Original CycleGAN Model

Notably, because our project motivation was to create a visualization of the adverse affects of the melting of mountains, we wanted to prioritize the effectiveness of the forward mapping from the domain of snowy mountain images to the domain of non-snowy mountain images. The effectiveness of the reverse mapping was less important to us. With this in mind, we made the following changes to CycleGAN.

3.3.1 Weight Decay

The loss function:

$$\mathcal{L}(G, F, D_X, D_Y) = \mathcal{L}_{GAN}(G, D_Y, X, Y) + \mathcal{L}_{GAN}(F, D_X, Y, X) + \lambda \mathcal{L}_{cyc}(G, F) \quad (2)$$

contains a λ that controls the relative importance of the two objectives: generation and cycle-consistency. The original architecture maintains a constant lambda of 10. As a result, it makes the assumption that generating an image to the other domain is just as important as ensuring the minimum cycle-consistency loss.

Low cycle-consistency loss is important in the beginning to ensure the generators avoid unnecessary changes and thus generate images that share the same structural similarity with the inputs. However, further in training, maintaining a constant λ produces unfavorable results by generating images that look unrealistic to the human eye since cycle consistency is enforced at the pixel level [6]. As a result, the textures of the original input image persists in many output images. As depicted in Figure 6(b), without weight decay, remnants of snow still remain since the constant λ enforces a small cycle consistency loss.

One solution to this problem is to use weight decay to gradually reduce λ. This effectively states that as training furthers, it is less important to mimic the original domain, and more important to generate an image in the other domain. Implementing weight decay improves many generated images to look more like the other domain.

The loss function with weight decay therefore becomes

$$\mathcal{L}(G, F, D_X, D_Y, t) = \mathcal{L}_{GAN}(G, D_Y, X, Y) + \mathcal{L}_{GAN}(F, D_X, Y, X) + \lambda_t \mathcal{L}_{cyc}(G, F) \quad (3)$$

where t represents the current epoch.

We chose to decay λ linearly with t to approach but not reach 0 as it increases toward the last epoch [5].

```
# Decay step
decay = self._lambda_a.eval() / (self._max_step + 1e-1)

# Update
sess.run(tf.assign_sub(self._lambda_a, decay))
```

3.3.2 Architecture Modifications

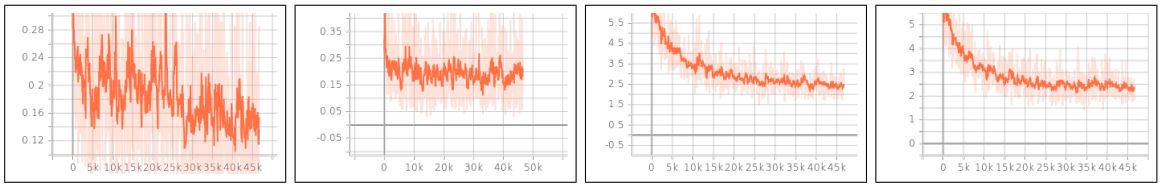
In addition to the weight decay, we also wanted to create a more complex network to better results. We added three resnet blocks with the following characteristics:

Padding: Symmetric padding

2D Convolutional layers: Strides of 3, Padding of 1, Standard Deviation of the Gaussian being sampled from is 0.02

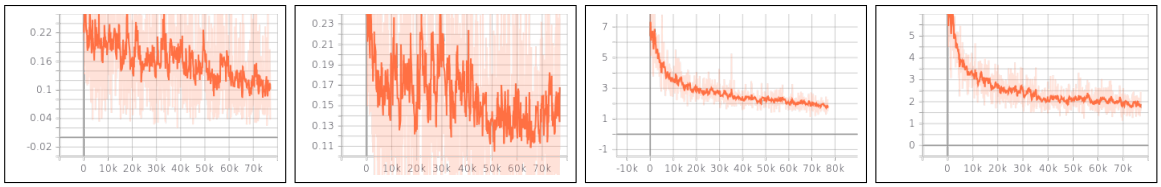
We also used a skip connection in the hopes that it would provide information of features in the image that could be passed on to further layers in the network.

4 Results



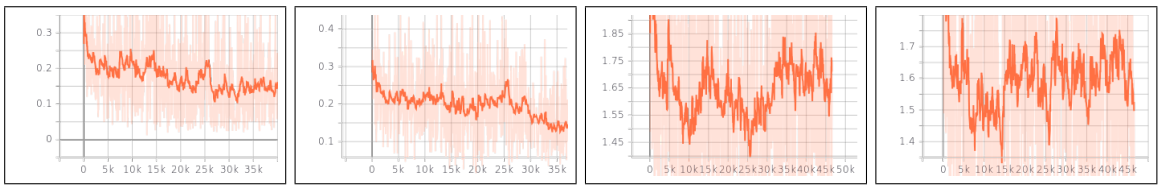
(a) Discriminator A Loss (b) Discriminator B Loss (c) Generator A Loss (d) Generator B Loss

Figure 3: Initial Loss Functions



(a) Discriminator A Loss (b) Discriminator B Loss (c) Generator A Loss (d) Generator B Loss

Figure 4: Loss Functions with Weight Decay



(a) Discriminator A Loss (b) Discriminator B Loss (c) Generator A Loss (d) Generator B Loss

Figure 5: Loss Functions with Architectural Changes and Weight Decay

5 Analysis

5.1 Initial Run

The results from the initial run coincide with our intuition of CycleGAN's architecture. During the early stages of training, the generator losses are quite high due to generating images that do not look like the other domain: the model is still learning the features of the mapping. However, as training progresses, the generator loss plateaus and converges since each new epoch does not improve in image quality.

Furthermore, we notice that the discriminator loss is quite low throughout the training. However, it is and should be extremely high in the first few runs as the generated images do not resemble the original images due to its focus on learning the basic features such as color and edges.

5.2 Weight Decay

As mentioned, weight decay loosens the cycle consistency requirement. As a result the generators are able to create images that are look more like the other domain (see Figure 6(c)), effectively causing the generator loss to decrease as depicted in Figure 4. This can be clearly seen by comparing the loss of both Generator A and Generator B at similar steps.

5.3 Architectural Changes

The architectural changes' results were surprising. Overall, the quality of the generated images for the deeper network with skip were better as shown in Figure 7. The deeper network, as it over fits the training set, also maintained the structural edges of the original image as show in Figure 7(c). However, on average, the deeper network worsened the goal of removing the snow. In many photos, remnants of snowed remained.

6 Future Work

There are many possibilities to improve upon this model. The following are just a few such modifications worth pursuing:

Utilize HarmonicGAN: In a paper published by Zhang et al. titled "Harmonic Unpaired Image-to-image Translation", the images explore an improved CycleGAN architecture called HarmonicGAN [8]. Its improvements can be seen by more realistic images that leave less of a trace of the the original domain. For example, in a horse to zebra setting, CycleGAN sometimes has a brown undertone to

(a) Input Image (b) Generated w/o Weight Decay (c) Generated w/ Weight Decay

Figure 6: Comparison between images generated with and without weight decay.

(a) Input Image (b) Without Architectural Change (c) With Architectural Change

Figure 7: Comparison between images generated with and without architectural change

the generated zebra pattern. The paper introduces "a smoothness term over the sample graph to attain harmonic functions to enforce consistent mappings during the translation."

$$\mathcal{L}_{HarmonicGAN}(G, F) = \mathcal{L}_{CycleGAN}(G, F) + \lambda_{Smooth} \times \mathcal{L}_{Smooth}(G, F) \tag{4}$$

The paper reports that in the context of medical imaging, radiologists preferred the HarmonicGAN generated images over CycleGAN images 95% of the time.

Latent Variable Generators Another potential option is the use of latent variable generators. This would work by encoding images from one domain into a latent space and then decoding the latent space into images in the other domain. The use of latent variable generators makes it possible to use other notions of consistency in the overall objective, namely latent variable consistency.

Flow Normalization CycleGAN only imposes what is referred to as a "soft-cycle consistency penalty", meaning there is no guarantee that the forward and backward mappings are true inverses of each other. In AlignFlow, Grover et. al propose the notion of using pairs of invertible flow models to ensure exact cycle consistency, meaning cycle consistency loss effectively becomes zero[3].

7 Conclusion

Ultimately, we were largely successful at using CycleGAN to create visualizations of the adverse effects of climate change on melting mountains. We were able to learn three major insights regarding the effectiveness of CycleGAN on unpaired image pairings: (1) CycleGAN in its current state is unable to perform geometric transformations successfully; instead it is used most effectively for texture swaps, (2) hyper-tuning the λ function greatly affects the output and can be manipulated to generate outputs favorable to one's requirements or desires (in this case, emphasizing the effectiveness of the forward mapping function), and (3) creating deeper architectures generally improves the photo-realism of images but is not as successful at style transfer.

Contributions and Acknowledgements

Although we collaborated on all tasks together, we specialized and spear-headed the following:

Gunguk Kim: Dataset Extraction and Curating, Weight Decay, Tensorboard Setup

Troy Shen: Paper Research, AWS Setup, Architecture Modifications

We would like to thank our project mentor Jo Chuang for his support and guidance throughout the course of the project, in addition to the entire teaching staff and Professor Ng.

Code

https://github.com/troyshen17/CS230-Final-Project

References

[1] Berwyn, B., Dixit, K., & Berwyn, B. (2019, October 10). In the Mountains, Climate Change Is Disrupting Everything, from How Water Flows to When Plants Flower. Retrieved from https://insideclimatenews.org/news/07102019/mountain-climate-change-disruption-glaciers-water-ecosystems-agriculture-plants-food.

[2] Carson, R. (n.d.). Losing Paradise: Climate change is changing Mount Rainier. Retrieved from https://media.thenewstribune.com/static/pages/rainier/.

[3] Grover, A., Chute, C., Shu, R., Cao, Z., & Ermon, S. (2019, May 30). AlignFlow: Cycle Consistent Learning from Multiple Domains via Normalizing Flows. Retrieved from https://arxiv.org/abs/1905.12892.

[4] News, B. B. I. C. (2019, October 7). Fire, Rock Slides, and Floods: UN Report Details Growing Warming Threat in the Mountains. Retrieved from https://www.kqed.org/science/1948659/fire-rock-slides-and-floods-un-report-details-growing-warming-threat-in-the-mountains.

[5] Vasani, D. (2019, November 18). This thing called Weight Decay. Retrieved from https://towardsdatascience.com/this-thing-called-weight-decay-a7cd4bcfccab.

[6] Wang, T., & Lin, Y. (n.d.). CycleGAN with Better Cycles ssnl.github.io. Retrieved from https://ssnl.github.io/better_cycles/report.pdf.

[7] Yang, H. (2018, December 14). leehomyc/cyclegan1. Retrieved from https://github.com/leehomyc/cyclegan1.

[8] Zhang, Rui & Pfister, Tomas & Li, Jia. (2019). Harmonic Unpaired Image-to-image Translation.

[9] Zhu, J.-Y., Park, T., Isola, P., & Efros, A. A. (2017). Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks. 2017 IEEE International Conference on Computer Vision (ICCV). doi: 10.1109/iccv.2017.244