Forest-to-Coast Unpaired Image Translation with Cycle Consistency

Stylianos Rousoglou
Department of Computer Science
Stanford University
steliosr@stanford.edu

Lydia Maria Tsiverioti
Department of Mechanical Engineering
Stanford University
ltsiver@stanford.edu

Abstract

We develop models for unpaired translation from images of forests to images of coasts, and vice versa, using conditional generative adversarial networks (cGAN). A potential road block in image to image translation is the absence of paired examples when performing a novel task. cGANs are capable of robustly learning a loss function and using it to train mappings between image spaces, without having to be trained on image pairs. We study the implications and performance of different generator and discriminator architectures, and evaluate the results in terms of aesthetic value and faith to the original, which we measure through cycle consistency loss.

1 Introduction

Image to image translation, such as translating aerial pictures to maps and translating scenes captured in daylight to scenes captured at night, often arises in fields such as Computer Graphics. Current literature focuses on cGANs in developing mappings between image spaces. Without cGANs, such mappings and their loss functions have to be developed on a case by case basis. The power and robustness of cGANs lies in their ability to simultaneously train a discriminator that deciphers "fake" images and a generator that alters an input image to "trick" the discriminator. Additionally, cGANs condition their discriminator and generator on auxiliary information, which makes it easier to guide the data generation process.

2 Related work

Before cGANS, less robust methods addressed image to image translation. Pathak et al.[1] attempt filling in missing image parts using L2 losses without an adversarial loss component. They output blurry image parts since L2 minimizes mean pixel wise error. Hertzmann et al.[2] train a network to learn a filter and then apply it to a source image. Unlike Isola et al [3] they do not use adversarial losses and therefore have to provide their network with pairs of unfiltered and filtered images as a training set. Metz et al.[4] use GANs in learning ojbect vs scene representations for generative modeling and exhibit promising results. Zhao et al.[5] supplement GAN by building an "energy based" discriminator that distributes "higher energies" to generated images. Isola et al. supplement classical GAN by conditioning their discriminator and generator on auxiliary information. In the case of image to image translation, the "noise" is the input image. We use Isola et al. pix2pix network as a starting point and explore different generator/discriminator frameworks.

3 Dataset and Features

We use the "Forests" and "Coasts" collections from MIT's LabelMe dataset [8] and employ a 90/10 training/test split. There is no need for preprocessing, since our images are 256 by 256 in size. We opted for a limited volume of training data, around 500 images for each class, due to the computational cost of training a cGAN, the short time frame of the project, and the limited computational resources we have at our disposal, specifically one GPU.

4 Methods

We use a conditional GAN for image generation and training. Conditional GAN learn a mapping from input image x and noise vector z to y, G : {x,z}→ y. The generator is tasked to produce images the discriminator will categorize as "real." Conversely, the discriminator aims at detecting the generator's "fakes." Both discriminator and generator are trained simultaneously.

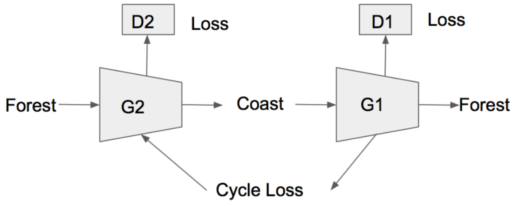
CS230: Deep Learning, Winter 2018, Stanford University, CA. (LateX template borrowed from NIPS 2017.)

4.1 Loss function

For the function $G : X \rightarrow Y$, and the discriminator D_Y, the conditional GAN objective entails an adversarial loss

$$L_{GAN}(G, D_Y) = E_y[logD_Y(y)] + E_x[log(1 - D_Y(G(x)))]$$

We introduce a cycle consistency loss to enforce $F(G(X)) = X$ [10] as seen in Figure 1. This extra objective helps our generated image maintain certain aspects of the original image. We train all 4 networks simultaneously, and both "ways" (only one direction is shown in Figure 1) Half of the time our input is coast images and the other half is forest images.



$$L_{cycle} = -\frac{1}{m_{gen}} \sum_{i}^{m_{gen}} ||G_2(G_1(x_{c2}^{(i)})) - x_{c2}^{(i)}|| - \frac{1}{m_{gen}} \sum_{i}^{m_{gen}} ||G_1(G_2(x_{c1}^{(i)})) - x_{c1}^{(i)}||$$

$$L_{overall} = L_{GAN}(G_1, D_1) + L_{GAN}(G_2, D_2) + \lambda L_{cycle}$$

Figure 1: CycleGAN architecture and associated loss function

4.2 Generator

4.2.1 U-Net

The U-Net generator [6] consists of a contracting path (encoder) and a expansive path (decoder), and each contain 8 repeated applications of convolution/deconvolution, ReLU and max pooling. Additionally, it concatenates the ith layer's feature map to that of the $(n - i)$ th layer where $1 < i < 8, n = 16$, to avoid loss of border pixels. Figure 2 gives a high level overview of U-Net's architecture.

4.2.2 ResNet

The second generator is based off of ResNet 6 and ResNet9. In both networks He et al. [7] use skip connection between their encoder and decoder that bypass convolutional layers to form residual blocks. Each residual block is a sequence of: convolution, BatchNorm, ReLU, convolution, BatchNorm. Residual blocks supplement a simple feedforward neural network by dealing with problems such as vanishing gradient. Additionally each encoding/decoding block consists of 2 repeated applications of convolution/deconvolution, InstanceNorm, ReLU.

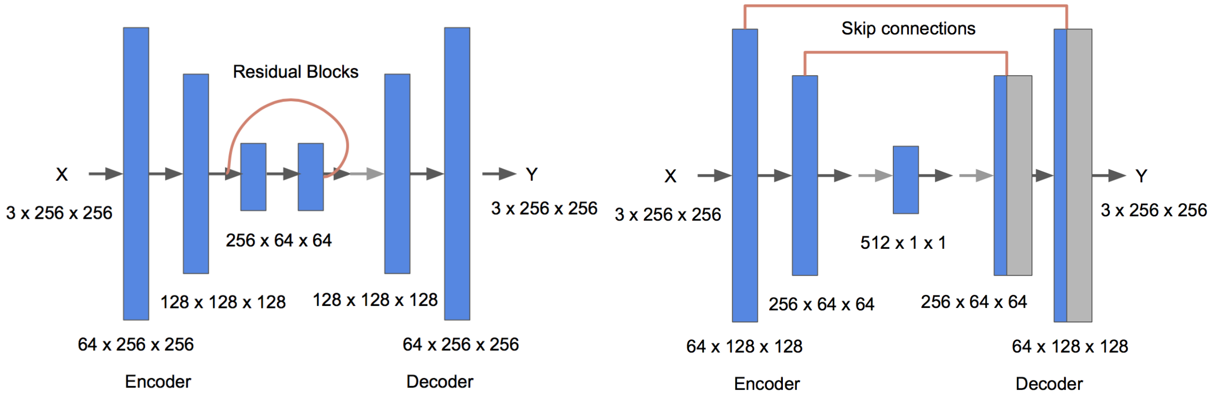


Figure 2: Left to right: ResNet architecture, U-Net architecture

4.3 Discriminator

4.3.1 PatchGan

PatchGan sequentially samples 70 by 70, in size, patches from our original image and classifies them real or fake [3]. It passes each one through n layers of convolution, BatchNorm, LeakyReLU and averages all responses to decide whether our image as a whole is real or fake. In our study we experiment with $n = 3$ and $n = 6$.

5 Training

During training, we initially kept our discriminator constant and explored different generator architectures. We applied ResNet-9 and U-Net 256 and then turned to shallower networks, namely, ResNet-6 and U-Net 156. During these tests we used PatchGAN with 3 layers.

After these 4 tests we kept Generator selection constant and varied our Discriminator. Specifically, we applied ResNet-9 and U-Net 256 again, but used a 6 layer PatchGAN instead of a 3 layer on.

Our training data varies greatly so we used Batchnorm to allow more flexibility in learning weights. BatchNorm normalizes each input channel across minibatches and deals with internal covariate shift of layer inputs [9]. Batch Normalization allowed us to use much higher learning rates and be less careful about initialization. For initialization we use Xavier initialization.

also it reduces sensitivity to the initialization; and speeds up training of the CNN

6 Results/Discussion

An instance of epoch transitions is shown in Figure 3 using ResNet-9. Our generator maintains the overall composition of the original image throughout transitions. It translates the bright green clusters of leaves in the center to waves and the darker clusters of branches to water. However, it does not solely rely on the original composition. For instance, there is not a clear horizon at epoch 50, but at epoch 200, we observe a clear border between the sky and the sea. Additionally, at epoch 200 the sky develops a color gradient; it is close to white near the sea and it becomes darker blue further away. Overall, during epoch transition we notice shapes, colors and borders becoming sharper and growing increasingly closer to resembling a coast while maintaining some of the original image's structure.

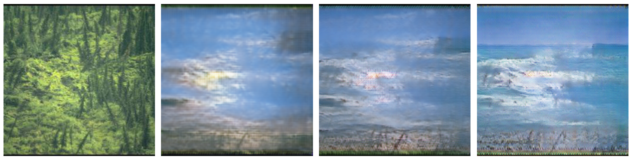


Figure 3: Left to right: original image, generated image at epoch: 50, 100, 200

6.1 Generator evaluation

Figure 4 contains generated images of all of the generators we use (ResNet9, U-net256, ResNet6, U-net128). We focus on the first row for visual analysis. ResNet-9 develops sharper colors and contrasts than ResNet-6. The brown color of the sand is more crisp and the sky does not have yellow undertones. The waves and the rocks are slightly more well defined. The extra residual layers of ResNet9 visibly improve image generation. It is interesting how differently U-net256 and U-net128 interpret the original image. U-net256 develops the horizon noticeably lower than U-net128, as well as all the other networks. This perspective shift is mirrored in the plethora of waves U-net256 fits in its smaller "sea region." U-net256 predictably generates a slightly more realistic image because it is deeper. U-net256, seems to generate the most realistic composition, whereas ResNet9 develops the best color rendering. Similarly for the second row, U-net256 contains the most persuasive composition, while ResNet9 has the most vibrant colors.

We see our visual analysis of generators somewhat mirrored in our quantitative results, as shown in Figure 5. Identity loss pushes the source-target to maintain the previous identity source distribution. For instance, maintaining the color of the original image in the generated image will result in lower identity loss. In our identity loss plots, U-Net 128/256 decrease to values slightly lower than those of ResNet-6/9. In both examples in Figure 4 we observe that ResNet-6/9 images exhibit "blue-er" colors that are more common in images of coasts than in those of forests. Conversely, U-Net 128/256 images have yellow/green undertones, similar to the original forest images.

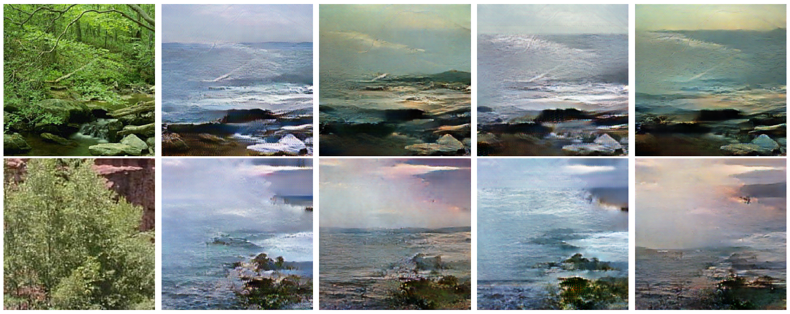


Figure 4: Left to right: original image, generated image using: ResNet9, U-net256, ResNet6, U-net128

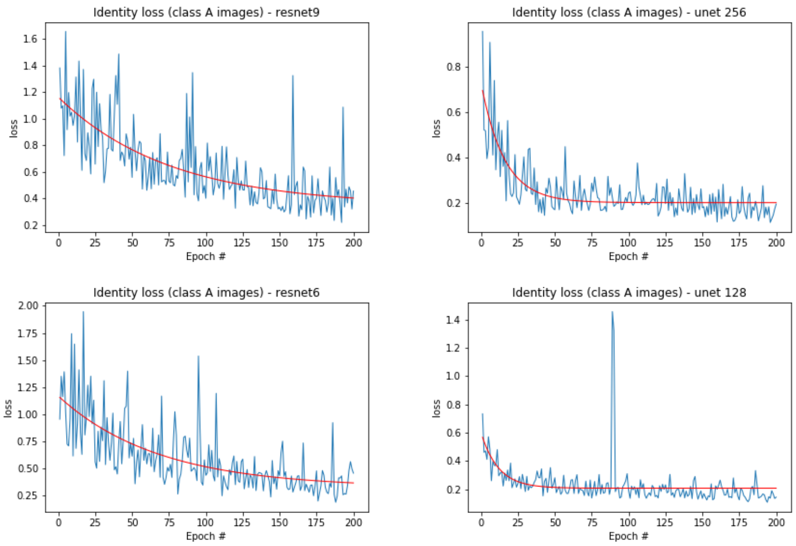


Figure 5: Plots of Identity loss vs epoch for each generator

6.2 Cycle consistency evaluation

In cGAN we introduce a cycle consistency loss to enforce F(G(X)) = X. The recovered cGAN images are shown in Figure 6. All four generated pictures are pretty close to the ground truth. U-net128 seems to be the least blurry and in general has similar vibrant colors to the original image. Converseley, ResNet-9 is the blurriest with the most "washed out" colors.

Our Cycle Loss plots, as shown in Figure 7 validate our visual analysis. Specifically, the U-net128 generated image, our most "crisp" copy of the original image, decays to a lower value than all three of the other generators. Conversely, the ResNet9 generated image, the blurriest recovered image, decays to the highest value compared to all thee other generators.

In general, even though ResNet6/9 produce more "realistic" images of coasts, they have lower Cycle consistency and Identity scores. Alternatively, U-net128/256 have better scores, but they produce less "realistic" images. There is no free lunch!

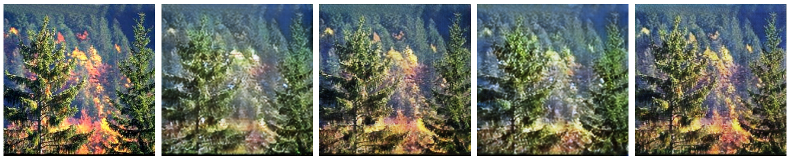


Figure 6: Left to right: original image, recovered image using: ResNet9, U-net256, ResNet6, U-net128

6.3 Discriminator evaluation

We examine discriminator complexity by using the same generator and observing how different number of discriminator layers alter our results. In the first row of Figure 8 we use ResNet-9 with a 3 layer PatchGAN and get a satisfactory representation of a coast. Comparatively, our 6 layer PatchGAN fails to produce a coast image. The generated image is a slightly blurry and noisy version of our original image and by no means looks like a coast. We rule out our choice of generator as an explanation, since we observe the same issue for U-net256. The standard to trick the discriminator is much higher with increasing number of layers and therefore, the generated image has to look much more like forest in order succeed.

Cyclic loss is predictably considerably lower for a 6 layer discriminator as seen in Figure 9. However, one of our objectives is an aesthetically pleasing generated image that resembles a coast. Therefore, adding complexity to our discriminator does not improve our result.

7 Conclusion/Future Work

Overall, cGAN prove to be successful in developing visual mappings between different types of landscape images. Even though the Identity and Cycle objective are crucial in our study we observe that putting too much weight on them produces less aesthetically pleasing images of

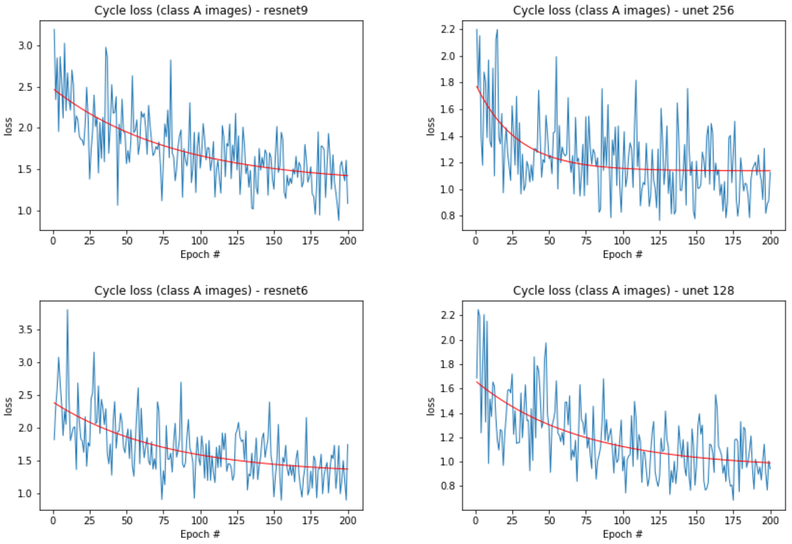


Figure 7: Left to right: Plots of cycle loss vs epoch number

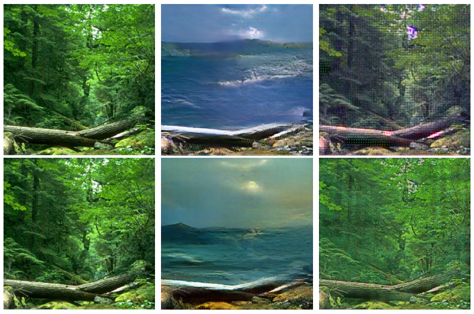


Figure 8: Top row: original image, ResNet-9 + PatchGAN with 3 layers, ResNet-9 + PatchGAN with 6 layers, Botton row: original image, U-net256 + PatchGAN with 3 layers, U-net256 + PatchGAN with 6 layers

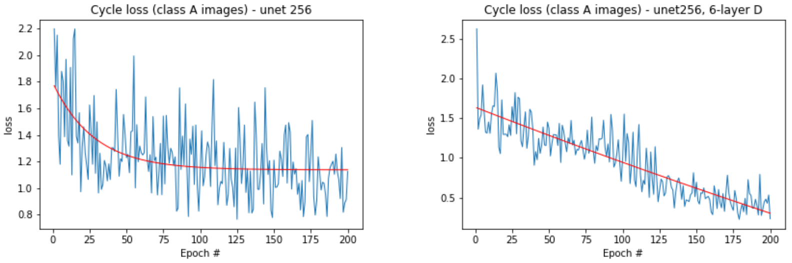


Figure 9: Cycle loss plots for U-Net256 + 3 layer PatchGAN (left), U-Net256 + 6 layer PatchGAN (right)

coasts. We witnessed this when we added more complexity to our discriminator and or used a U-Net generator. Both succeeded in lowering these objectives but outputed images inferior to the coast images associated with ResNet generators and shallower discriminators.

In future iterations, with less of a time constraint, we would train our selected model with more data. Additionally, we would want to explore more generators, such as ResNet 50, and more discriminators, such as ImageGAN.

8 Contributions

Each member contributed equally and in different ways. Stylianos focused more on training models and data post processing while Lydia took the lead in interpreting results and writing the report.

9 Acknowledgements

We would like to thank Vineet Kosaraju for his support during project meetings. Additionally, we are grateful for the AWS computing resources we received through CS230.

10 References

[1] Pathak, D., Krahenbuhl, P., Donahue, J., Darrell, T., & Efros, A. A. (2016). Context encoders: Feature learning by inpainting. *In Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2536-2544).

[2] Hertzmann, A., Jacobs, C. E., Oliver, N., Curless, B., & Salesin, D. H. (2001, August). Image analogies.*In Proceedings of the 28th annual conference on Computer graphics and interactive techniques* (pp. 327-340). ACM.

[3] Zhu, J. Y., Park, T., Isola, P., & Efros, A. A. (2017). Unpaired image-to-image translation using cycle-consistent adversarial networks. *In Proceedings of the IEEE international conference on computer vision* (pp. 2223-2232).

[4] Radford, A., Metz, L., & Chintala, S. (2015). Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv preprint arXiv:1511.06434.

[5] Zhao, J., Mathieu, M., & LeCun, Y. (2016). Energy-based generative adversarial network. arXiv preprint arXiv:1609.03126.

[6] Ronneberger, O., Fischer, P., & Brox, T. (2015, October). U-net: Convolutional networks for biomedical image segmentation. *In International Conference on Medical image computing and computer-assisted intervention* (pp. 234-241). Springer, Cham.

[7] Targ, S., Almeida, D., & Lyman, K. (2016). Resnet in resnet: Generalizing residual architectures. arXiv preprint arXiv:1603.08029. Chicago

[8] http://labelme.csail.mit.edu/Release3.0/

[9] Ioffe, S., Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv preprint arXiv:1502.03167.

[10] Zhu, J. Y., Park, T., Isola, P., Efros, A. A. (2017). Unpaired image-to-image translation using cycle-consistent adversarial networks. Proceedings of the IEEE international conference on computer vision (pp. 2223-2232).