Lego-ization of Images using CycleGAN

Caroline Zhang (czhan157), Trey Wiedmann (tweidman)

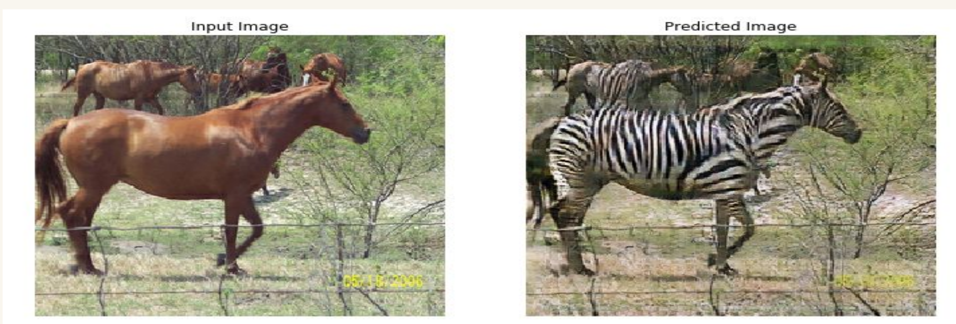
BROWN

CSCI 1430

Motivation

Neural networks in deep learning have become a powerful tool in style and domain transformations, and have been especially successful with media such as images. Using two GAN models in a cyclic fashion has let us move samples from one model to another in an efficient manner to transfer styles, and without the need for paired data.

We wanted to try out image style transfer where the style domains relied upon learning 3D features such as shadows and perspective.



Preprocessing

Data Collection

1. Scraped images of lego sets and objects from Flickr, Wikimedia, Lego, Amazon, and various blogs that talked about Legos.
2. Our object-object model was trained on around 2400 images, and our scene-scene model was trained on around 1300.

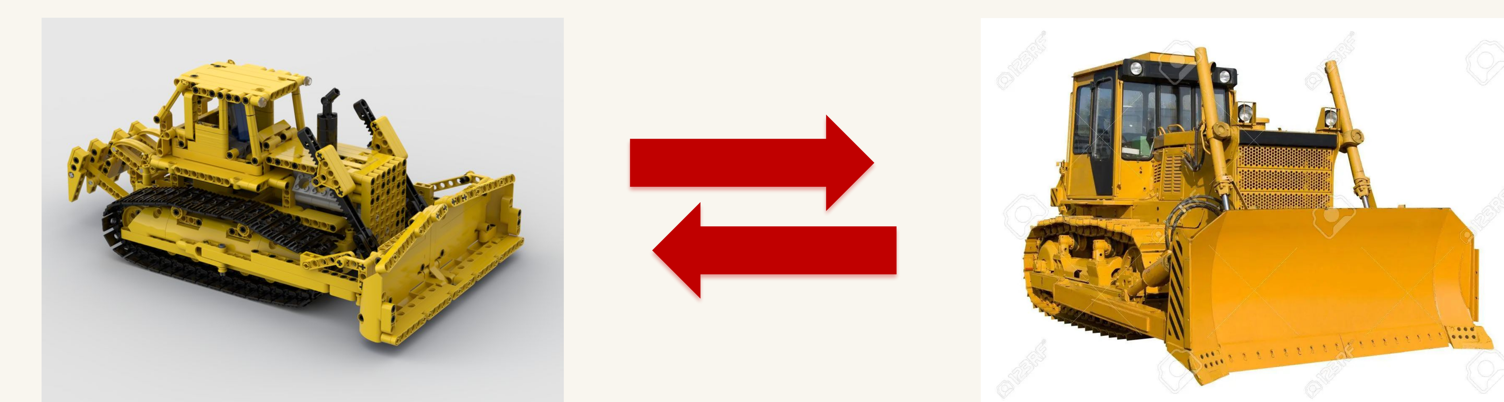
Data Preprocessing

- We preprocessed the images into 3D (256x256x3) arrays.
- We padded the object and Lego set images with white so that they were a square, and then resized
- We did the same for the Lego scene images (though those were mostly square to begin with). For the Flickr scene images, we cropped at the center
- We also did a lot of manual dataset cleaning

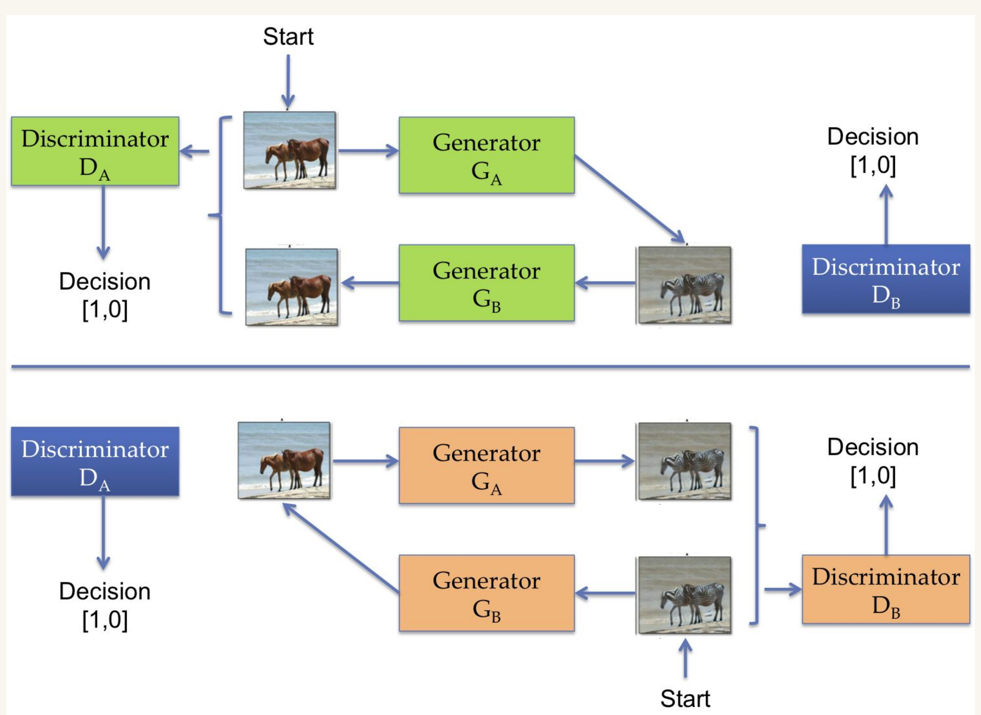
Goal

State Goals

1. Successfully convert a picture of an object or scene to an image of the equivalent lego object or scene



Model Architecture



Model Architecture

1. We trained 2 models: one for real-world-object to lego-object transfer, the other for real-world scene to lego-scene transfer
2. CycleGAN architecture: Using 2 discriminators and 2 generators
3. Our generators and discriminators consisted of convolutional, instance normalization, and ReLU layers, as well as residual blocks
4. We implemented Image Pooling to help the generator stay ahead of the discriminator

Discussion

Assessment:
We found that when we inputted an image of a real-world object into the object-object model, the output bore little resemblance to a lego. We think we need to experiment more with different losses, as well as the amount of identity loss to use.
With the scene-scene model, we found that when we inputted an image of a real-world scene into the model, the output image had a more lego-like quality in that it had developed a pattern similar to the top view of lego pieces.
We plan to continue training both models and experiment with different architecture complexities, such as the number of layers, size of the kernels, etc.

Limitations:
One of the limitations was data collection, as it was difficult to find large quantities of high quality images of lego sets and lego scenes. We also struggled with standardization of the lego images, as the "style" apparent in a lego image depends on the perspective – for instance, the top view of a lego brick would feature lots of small circles, while the side view would be a solid block of color.

Future Directions:
We would like to try training on model on a more standardized set of lego images (from the same angle) as well as implementing fine-tuning with paired data.

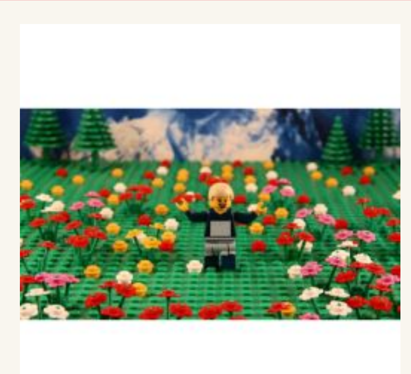
Results

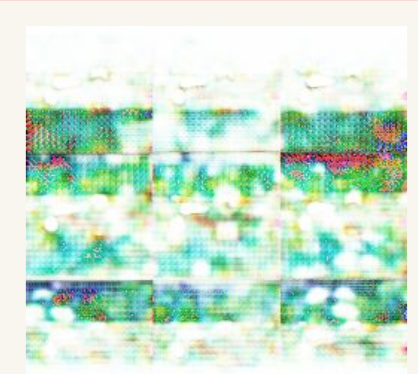
Scene-Scene

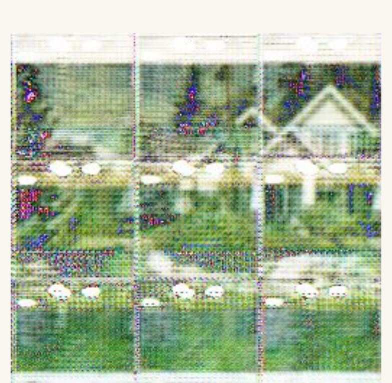

Input Scene


After 1 Epoch

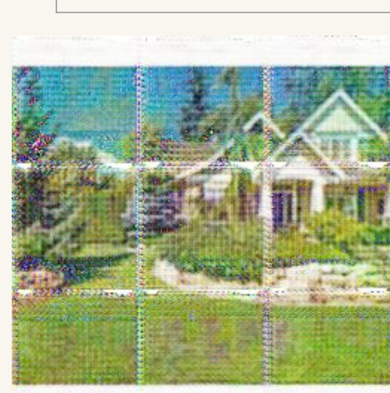
Epoch	Cycle Loss
1	1116.67
5	1115.39
15	1115.02
30	1114.9


Input Scene

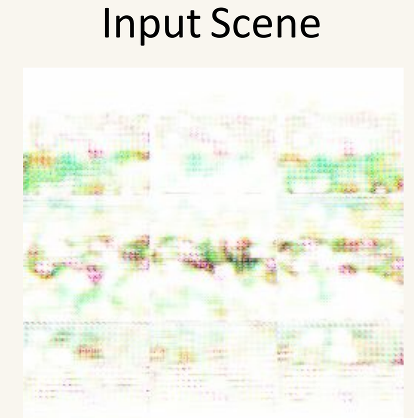

After 1 Epoch

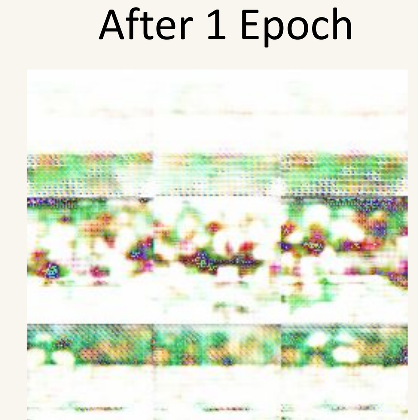

After 6 Epochs


After 20 Epochs


After 30 Epochs


After 30 Epochs

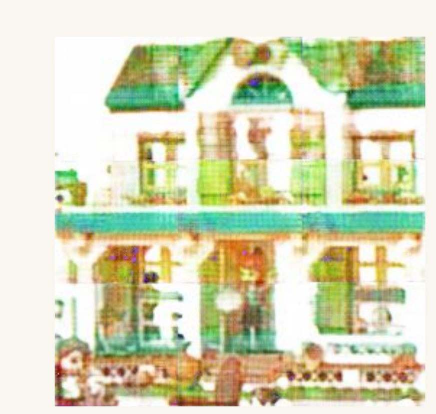

After 20 Epochs


After 6 Epochs

Object-Object

From top left to bottom: Input image, output after 22 epochs, output after 72 epochs

Epoch	Cycle Loss
1	0.6871
5	0.6579
10	0.6313
15	0.6305
25	0.5761
35	0.5723
50	0.5605



From top right to bottom: Input image, output after 22 epochs, output after 72 epochs

References

[1] Zhu, Jun-Yan, et al. "Unpaired image-to-image translation using cycle-consistent adversarial networks." Proceedings of the IEEE international conference on computer vision. 2017.

Acknowledgements

We would like to thank our mentor TA Anh for his help and insight, Professor Tompkins for his teaching, and all the CV TAs for their help throughout this semester :)