

No Game No Driving

--Transfer driving task via cycleGAN

Zhipeng Fan N16246016

Ben Ahlbrand N18797462

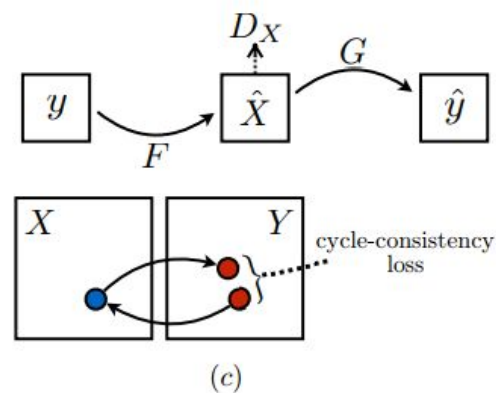
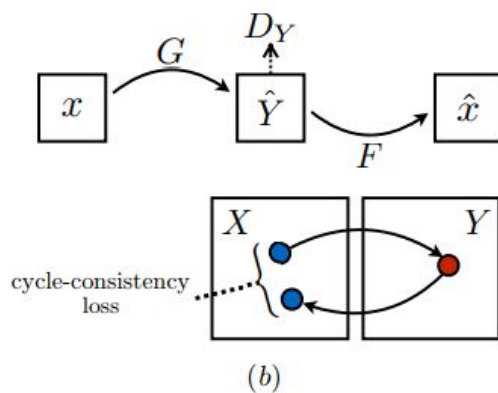
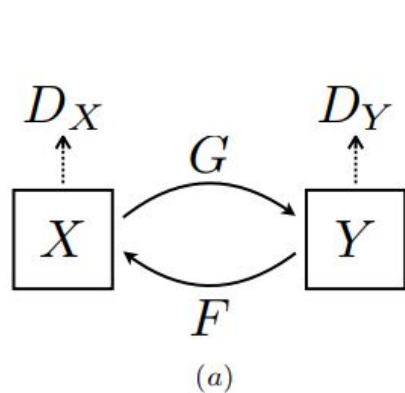
Hui Wei N17048100

Motivations

- Real world scenes contain less sticky situations, which leads to underfitting models in self driving algorithms for tricky cases.
- The evolution of computer graphics made computer games the perfect setting for training self-driving cars (less need for large amount of human annotations).
- How to transfer autonomous driving AI trained on Games to real-world settings slow down the progress of migrations.
- We present to conduct the image domain transfer (Computer Game \Leftrightarrow Real World) via cycleGAN
- Who doesn't love Games!!!

Intuitions of CycleGAN

1. **Machine Translation** => Introduces the Cycle Consistency (“back-translation”).
2. **Adversarial loss** => matching from source domain to target domain
3. **Cycle consistency loss** => Prevent mapping from contradicting each other
4. Enables domain transfer over **unpaired training dataset** rather than paired one.



CycleGAN architecture

- Adversarial loss

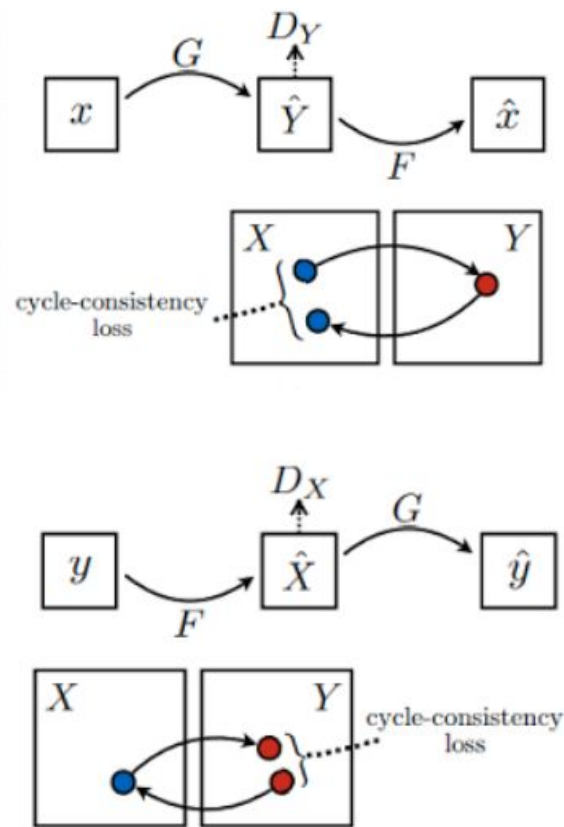
$$\mathcal{L}_{\text{GAN}}(G, D_Y, X, Y) = \mathbb{E}_{y \sim p_{\text{data}}(y)} [\log D_Y(y)] \\ + \mathbb{E}_{x \sim p_{\text{data}}(x)} [\log(1 - D_Y(G(x)))]$$

- Cycle Consistency loss

$$\mathcal{L}_{\text{cyc}}(G, F) = \mathbb{E}_{x \sim p_{\text{data}}(x)} [\|F(G(x)) - x\|_1] \\ + \mathbb{E}_{y \sim p_{\text{data}}(y)} [\|G(F(y)) - y\|_1].$$

- Full objective

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



Implementation Details

- **To stabilize the training and generate higher quality results**
 - Using least square loss instead of negative log likelihood [1]
 - G: $E_{x \sim P_{data}} [(D(G(x)) - 1)^2]$
 - D: $E_{y \sim P_{data}} [(D(y) - 1)^2] + E_{x \sim P_{data}} [D(G(x))^2]$
- **Network architecture:**
 - Generator: encoder-decoder structure
 - c7s1-32 => d64 => d128 => r128 * 6 => u64 => u32 => c7s1-3
 - Discriminator: classification network in fCNN fashion
 - c64 => c128 => c256 => c512
 - c7s1-32: 7x7 conv-InstanceNorm-ReLU with 32 filters and stride of 1
 - d64: 3x3 conv-InstanceNorm-ReLU with 64 filters
 - r128: residual block contains 2 3x3 conv layers
 - u64: 3x3 fractional-strided-conv-InstanceNorm-ReLU with 64 filters

Implementation Details

- **Dataset:**

- Real world data comes from the cityscapes datasets, developed for segmentation[2]
- Game data comes from ECCV 2016 paper that is originally developed for segmentations[3]



- [2] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The Cityscapes Dataset for Semantic Urban Scene Understanding," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.
- [3] Richter, S. R., Vineet, V., Roth, S., & Koltun, V. (2016, October). Playing for data: Ground truth from computer games. In *European Conference on Computer Vision* (pp. 102-118). Springer International Publishing.

Result

(~1.5k training images, 375 and 425 test images, 200 epochs)

Real Scene



Game Scene
(After transferred)



Recovered Scene
(from the game scene)



Result

(~1.5k training images, 375 and 425 test images, 200 epochs)

Game Scene



Real Scene
(After transferred)



Recovered Scene
(from the real scene)



Intermediate Results

Epoch 2



Epoch 17



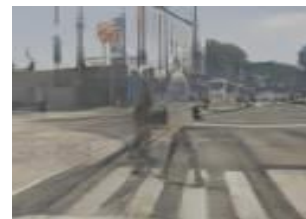
Epoch 23



Epoch 51



Epoch 132



Epoch 154

Result

(~2.5k training images, 375 and 425 test images, 200 epochs)

Real Scene



Game Scene
(After transferred)



Recovered Scene
(from the game scene)



Result

(~2.5k training images, 375 and 425 test images, 200 epochs)

Game Scene



Real Scene
(After transferred)



Recovered Scene
(from the real scene)



Result

(High Resolution & larger Net ~1.5k training images, 375 and 425 test images, 200 epochs)

Real Scene



Game Scene
(After transferred)



Recovered Scene
(from the game scene)



Result (High Resolution & larger Net ~1.5k training images, 375 and 425 test images, 200 epochs)

Game Scene



Real Scene
(After transferred)



Recovered Scene
(from the real scene)



Results in Video

- Real vs Fake (Transferring from Game to Real world image)



Analysis

Strengths:

1. It turns out that we can get good results transferring styles between two unpaired datasets.
2. Using the cycle loss function, we can recover the original scene to the maximum degree.
3. Using higher resolution images with larger networks produces more clear and vivid images, but significantly longer to train

Analysis

Limitations:

1. For complex scenes, transfer images might be distorted and blurry, mainly on the border due to size of training images
2. Generating vivid real scene images from simulated images in Game is more difficult compared to producing game images from real scene
3. No regularizations over consecutive frames, leading to jittering in consecutive frames
4. Increasing # of training samples doesn't improve the results much
5. inconsistent results with slight variations in illumination in scene

Results in Video

- Real vs Fake (Transferring from Real World to Game image)

