CycleGAN for sim2real Domain Adaptation
Anchit Gupta, Jonathan Booher
Stanford University

Summary
- Robot policies trained in massively distributed sim
- Need to bridge perceptual gap to use these in the real world
- Train a GAN to convert from one domain to the other!
- Use GAN for one shot sim2real transfer

Problem Setup
- Need to separate dynamics mismatch from perceptual mismatch which we aim to bridge
- Hence set up two version of the SawyerLift task in RoboSuite[2] with differing visual properties
- Collect data using random exploration in both environments, also augment with images collected from trained policy rollouts

Policy Training
- Used Distributed Proximal Policy Optimization implemented in [2].
  - Figure: Policy network combines both the pixel input with the low dimensional robot joint data and feeds it into a LSTM module.

CycleGAN
- CycleGAN augments GAN loss to encourage two generators to be inverses.
  - \( E_{\mu, i,j}(x) = ||F(y_i) - y_j||^2 \)
- Pure Cycle GAN works well for individual images but can vary greatly based on small variations in the input image (e.g. small robot movement).
- Use shift loss \([4]\) for each generator to address this problem
  - \( E_{\mu, i,j}(x) = ||F(y_i) - F(y_j)|| \)

Results (so far)
- We collected a dataset of 4800 images on the 'simple' domain, 3000 from random exploration and 1800 from rollouts of our trained policy as well as 500 images from the 'complex' domain.
- The trained policy achieves a score of 200 on its source domain ('simple') and only 90 when naively transferred to the 'complex' domain.
- Train Cycle GAN on a balanced dataset
  - Tendency to shift and rotate objects in space
  - Common generator errors include:
    - artefacts
    - disappearing gripper
- Transfer using Cycle GAN from imbalanced dataset yielded mean reward of 112 but with high variance.

References