draw2pix: Generative Adversarial Networks for Art School Rejects

Cedrick Argueta[1], Kevin Wang[2]

[1]Department of Computer Science, Stanford University, Stanford, CA 94305
[2]Department of Physics, Stanford University, Stanford, CA 94305

Motivation

Image-to-image translation has produced many cool results in recent years, especially regarding unsupervised translation. Even without labeled data, **CycleGAN** has been able to generate stunning images. We wanted to explore this model by translating sketches to photorealistic images.

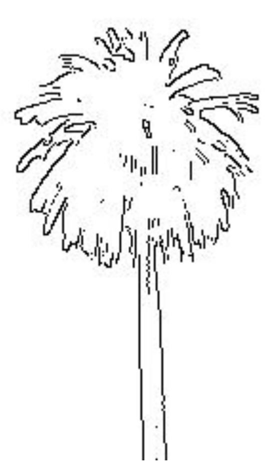
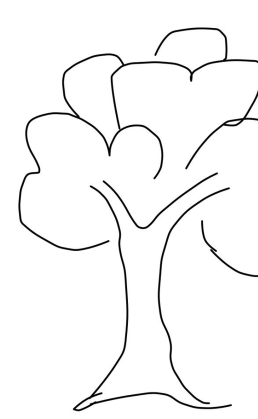
Overview

We implement a version of CycleGAN that includes various **regularization** and stabilization techniques applied in other types of GAN models, and use it to perform **sketch-to-photo** image translation.
Additionally, we experiment with a more advanced loss function for generators that enforces **feature-level cycle consistency**.
We discuss the trade offs that come from using **automatically generated edge maps** vs. hand-drawn sketches, and demonstrate various techniques that alleviate a majority of the generalization issues.

Data

Training is primarily done with an ImageNet synset of **palm trees**. We perform training on photos of trees and automatically generated edge maps of trees. The edge maps are created with the **Canny edge detection** algorithm. We test our model by transforming hand-drawn sketches of trees to photos of trees.

Flipping and normalization was applied to all inputs, while ColorJitter was applied to photo inputs.

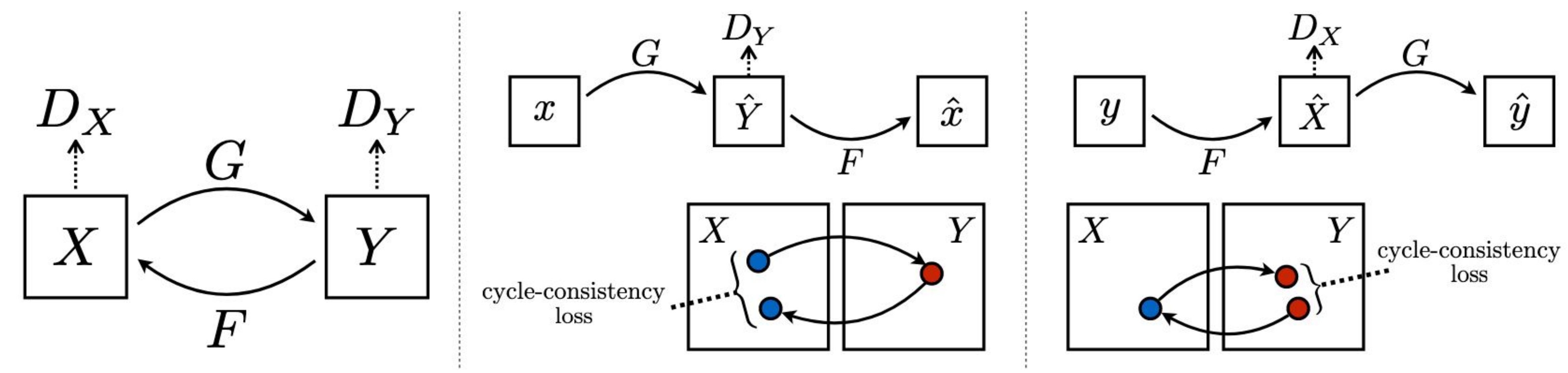




Methods

CycleGAN has two generators and two discriminators: one for each domain. Each generator learns a mapping function from its domain to the other, and are trained **adversarially** with the discriminators. Additionally, the generators enforce **cycle consistency**: that an image can be run through both generators and remains unchanged.

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

$$\mathcal{L}_{cyc} = \mathbb{E}_{x\sim p_X}[\|F(G(x)) - x\|_1] + \mathbb{E}_{y\sim p_Y}[\|G(F(y)) - y\|_1]$$



Neural Network Architecture

- Generator is an encoder, transformer, and decoder
  - **Encoder** is convolutional layers down to latent space
  - **Transformer** is ResNet blocks that transform to next domain
  - **Decoder** is resize-convolutional layers that generate output
- Discriminator is **PatchGAN**, outputs a matrix of values indicating whether receptive field of value is real or fake

Label Regularization

- Label **smoothing**
  - Rather than have targets of 0 or 1, have targets uniformly distributed between (0, 0.3) and (0.7, 1.2)
  - Introduces noise in discriminator target, weakens discriminator
- Label **flipping**
  - Randomly flip target labels
  - Adds more noise to discriminator, even more regularization

Evaluation Metrics

- **FID Score** - measures diversity and quality of generated images
- **Reconstruction loss** - measures quality of inverse mapping

Training Details

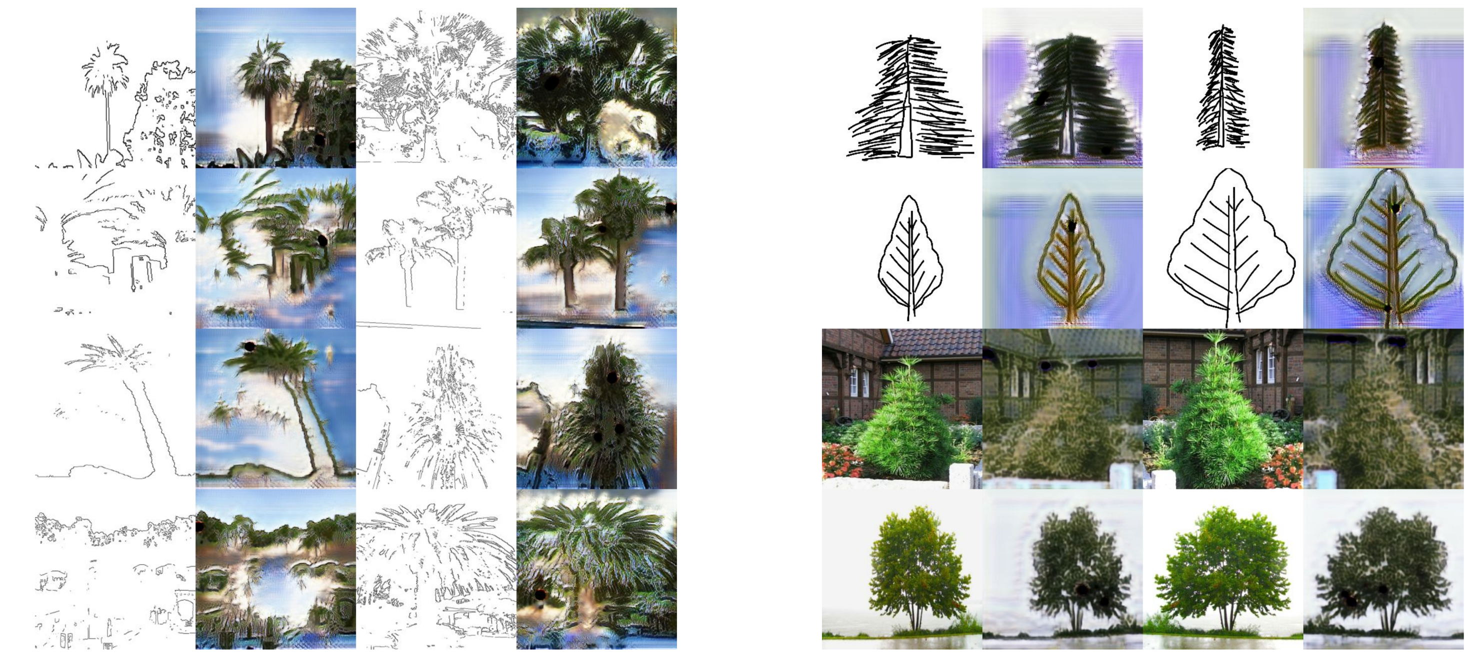
- **Replay buffer** maintained, periodically shows old generated images to discriminator so it doesn't 'forget' old artifacts
- Discriminator loss halved to reduce gradient update
- Weights initialized with normal distribution about 0 with std 0.02
- All losses are implemented in **least-squares** fashion
- Same hyperparameters as original CycleGAN

Feature-level Cycle Consistency

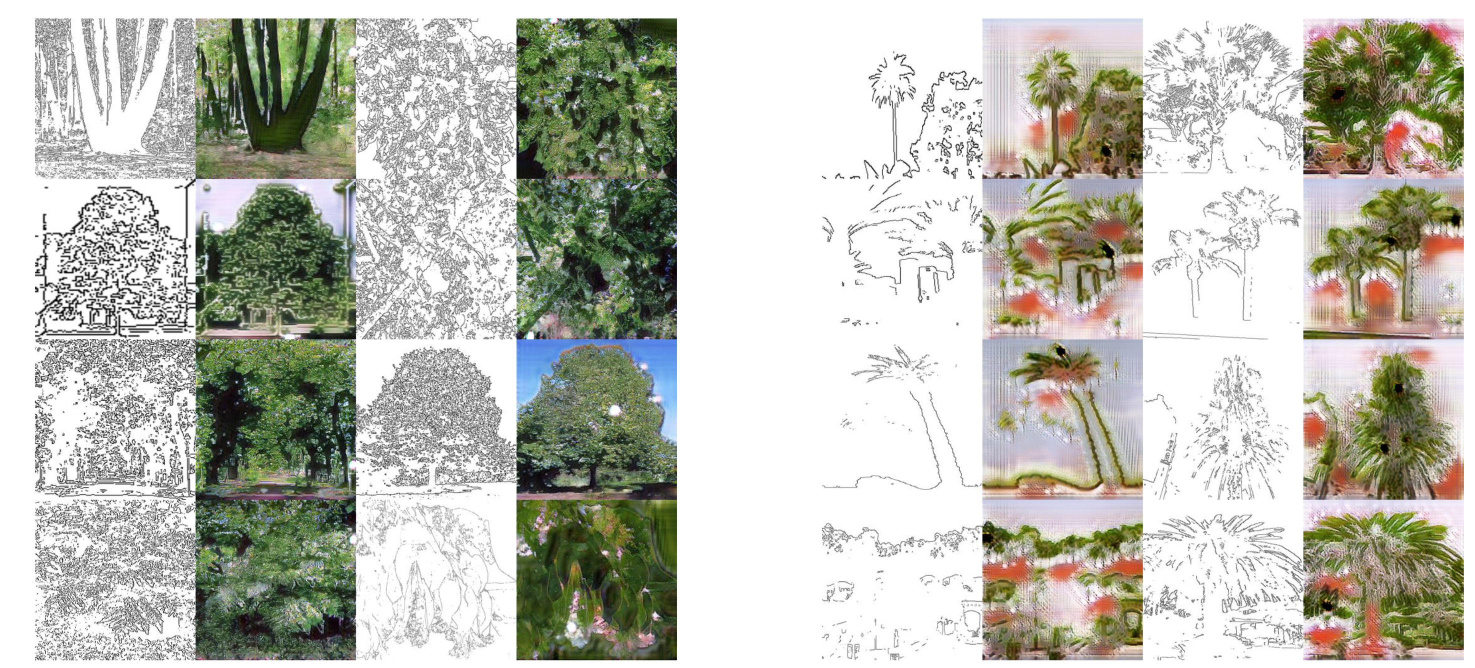
- CycleGAN has pixel-level cycle consistency (L1 loss)
- Instead enforce feature-level cycle consistency, maintain features across translation
- Takes **feature extractor** from discriminator and weights cycle consistency based on output of discriminator

$$\begin{aligned}\mathcal{L}_{cyc} = \mathbb{E}_{x\sim p_X}[D_X(x)(\gamma\|f_{D_X}(F(G(x))) - f_{D_X}(x)\|_1 \\ + (1-\gamma)\|F(G(x)) - x\|_1)] \\ + \mathbb{E}_{y\sim p_Y}[D_Y(y)(\gamma\|f_{D_Y}(G(F(y))) - f_{D_Y}(y)\|_1 \\ + (1-\gamma)\|G(F(y)) - y\|_1)]\end{aligned}$$

Results



Semi-photorealistic images can be created, however the network fails to learn details where no edges are located. The model fails to generalize to sketch inputs, though the inverse mappings learned are excellent.



The model with label regularization can produce photorealistic results when presented with finer detail in sketch/edge map input.
Quantitatively, label regularization improves generalization for sketch-to-photo synthesis. Photos-to-edges is easy to learn because Canny edge detection is just convolutions.

Translation	CoGAN	CycleGAN	CycleGAN w/ LR	CycleGAN w/ BC	CycleGAN w/ all
edges-to-photos	N/A	**344.02**	363.03	449.05	503.70
photos-to-edges	526.59	306.62	260.57	313.71	**255.29**
sketch-to-photo	538.97	491.06	**434.18**	479.18	526.06
photo reconstruction	N/A	**177.50**	221.17	324.89	324.73
edge reconstruction	N/A	**99.34**	173.22	196.56	196.03

Table 1. FID scores for generated photos of palm trees compared to real photos of palm trees, and generated edge maps compared to real edge maps. Reconstructed images are also compared to their original domains. LR = label regularization, BC = better cycle consistency loss

Reconstruction	CycleGAN	CycleGAN w/ LR	CycleGAN w/ BC	CycleGAN w/ all
edges	**33.23**	36.18	37.21	39.97
photos	101.30	**91.50**	107.37	101.21

Table 2. Mean L1 distance between source images and reconstructed images.

Conclusion

- Our implementation with **label regularization outperforms traditional CycleGAN** in sketch-to-photo translation
- Traditional CycleGAN outperforms all other models with edge-to-photo translation
- We show **ability to create realistic images** with other domains
- Generalization is poor - edge maps are not similar enough to sketches
- Future work could involve use of Holistically-nested Edge Detection

References

- P. Isola, J. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. CoRR,abs/1611.07004, 2016.
- J. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. CoRR, abs/1703.10593, 2017.
- T. Wang and Y. Lin. Cyclegan with better cycles.