Photo Retouching with Generative Adversarial Nets

By Jason Housman

Task Statement

- Not necessarily a function that solves a problem.
- Computationally performing a task originally only possible for humans
- Application for other problems

The Task:

Feed an image into a model and output a retouched version of that image.

Normally performed in photo editing softwares such as photoshop and lightroom.

Applicable to other Problems

Contributes to the literature of computer vision.

Pattern matching in image blending.

Picture to picture translations.

Quantifying the quality of aesthetics.

Performing the Task

- Machine Learning fits very well into the problem of image editing.
  - Pattern matching of what creates beauty in photography.
- Supervised Task vs Unsupervised Task:
  - Supervised: Feeding edited images and targeted images
  - Unsupervised: No real possible use.
  - Semi Supervised: Generate images from the ground up, discriminate with labels

Preparation for the Model

Input:

- Untouched images as inputs, retouched images as target
- Images in the RGB Color space.
- Images transformed for robustness.
- Convolutional feature set.

Untouched Image

Target Example

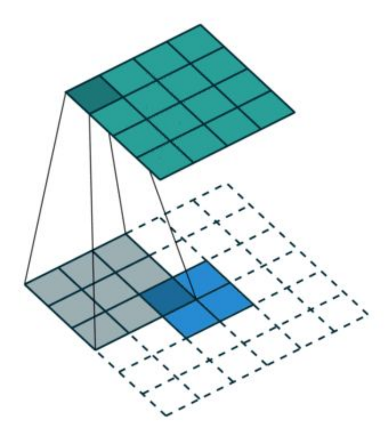
Generative Adversarial Networks

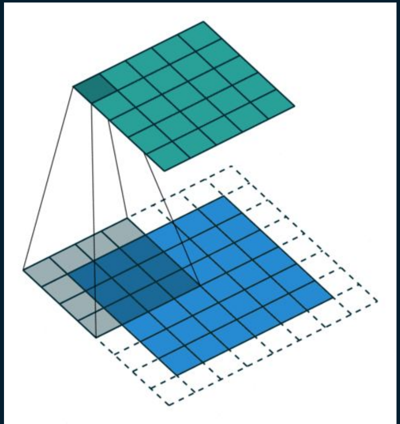
Model Architecture Part 1

The Generator:

Consists of an encoder model and a decoder model.

- Encoder:
  - Input is fed into the encoder, and its dimensionality is shrunk down through convolution.
- Decoder
  - After convolution, image is fed through Decoder.
  - Its dimensionality is increased through deconvolution.





Model Architecture Part 2

The Discriminator:

- The discriminator determines the quality of the output given labels.
  - CGAN Twist
  - Fake Discriminator
    - Examines the output created by the generator and determines legitimacy.
  - Real discriminator
    - Examines an actual target and determines legitimacy

Generator Discriminator Interaction:

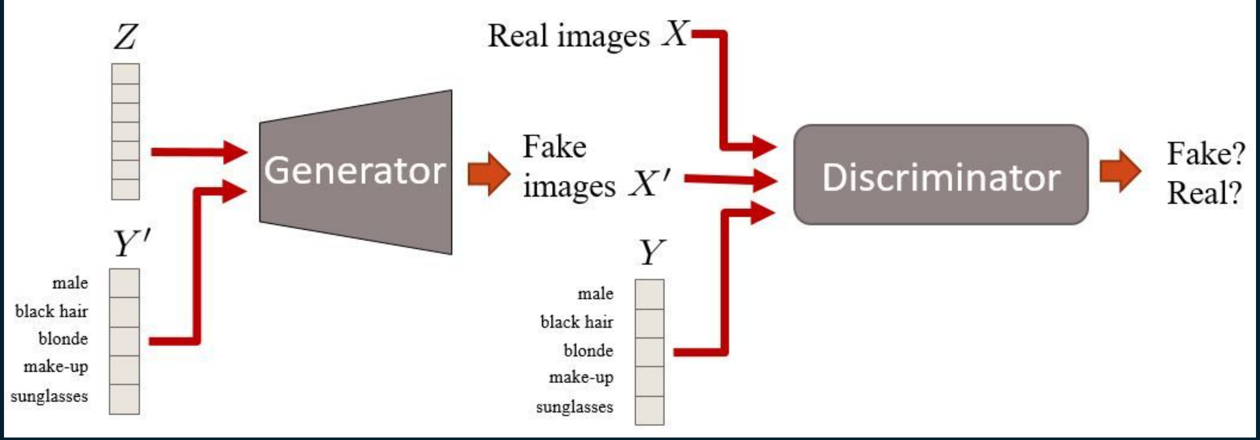
Generator creates images from scratch.

Fake Discriminator is fed the input images and the generated images.

Real Discriminator is fed the input images and target images.

- When fake discriminator is correct, generator knows to improve further.
- When real discriminator is incorrect, discriminator knows to improve.
- Proceeds until convergence

Generative Adversarial Networks



Generative Adversarial Nets were conceived by researchers at University of Montreal. It has become fairly popular in the deep learning world.
Introduced in 2014 in a paper, primarily written by Ian J. Goodfellow.

"Generative Adversarial Training is the coolest idea in machine learning in the last 10 or 20 years"

- Yann LeCun, Director of AI Research at Facebook

Dataset

- MIT Adobe FiveK Dataset
- 5000 Original Images retouched by 5 professionals.
- Total Size: 30,000.
- Preprocessing: Combine original and retouched image into singular image.
- Resized to 256x256.
- My own images for testing.
- Training Set: 4850 Image Pairs
- Test Set: 150 Image Pairs

Execution Process

Training:

- Images fed into pipeline
- Image converted to tensor
- Image pair split
- Images randomly flipped or transformed for robustness
- Images fed into Model
  - Input Images fed into encoder
  - Encoded images fed into decoder
- Generated images outputted
- Generated Image and Input Image fed into FAKE Discriminator
  - Discriminator Score Calculated
- Target images fed into REAL Discriminator
  - Discriminator Score Calculated
- Differences between Discriminator scores
  - Searches for Minimum
- Generator success based on score of Fake Discriminator
- Generator score maximized

Implementation

Notable Requirements:

Tensorflow(1.0.1)

Tensorboard

Scipy(0.19.0)

Scikit-image(0.13)

Numpy 1.12

Project Structure:

Argument File - Containing output, input

And execution parameters.

Data processing - Loads data, transforms data,

stores data

Model Utilities - Convolution, Deconvolution,

Batch Normalization, RELU, Leaky Relu

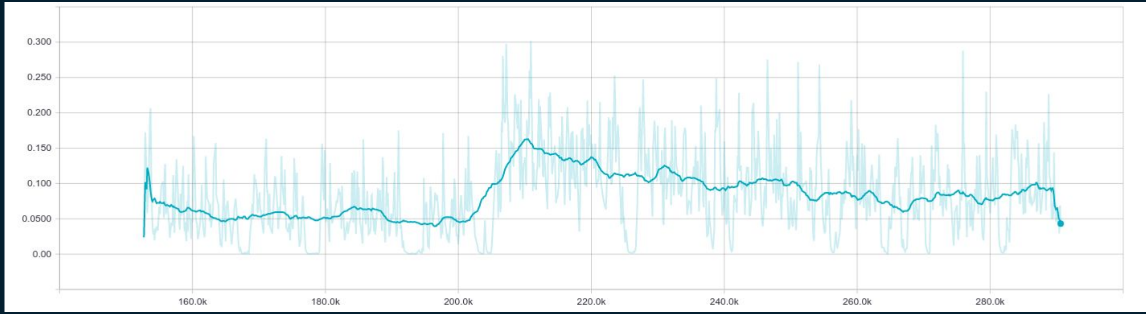
Model Runner - Contains execution path, train, test and

Generate

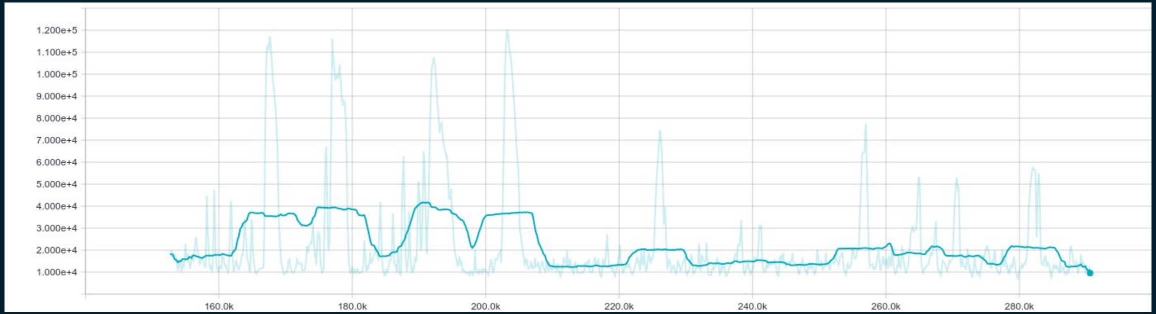
Minimize: Discriminator difference, L2_loss between target and generated image, minimize total generator loss as function of discriminator loss and L2 loss.

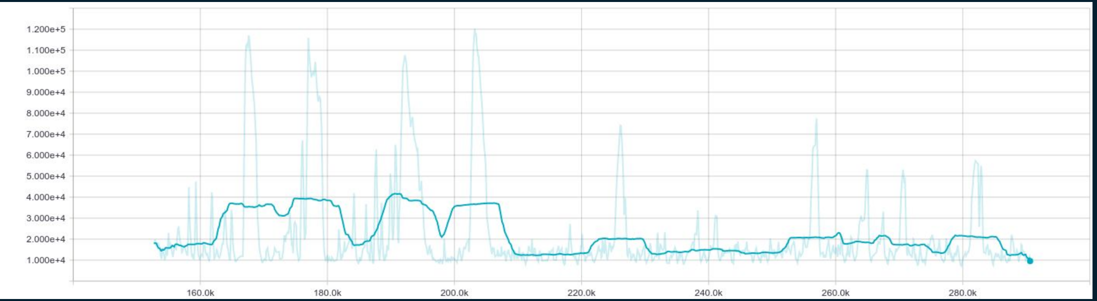
Model Progress Graphics

Discriminator Improvement



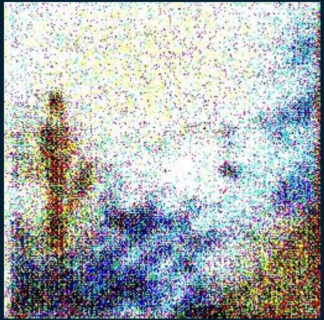
Output Image Improvement





Generator Improvement

The Learning Curve






Training Image Results

The Results with Targets

Test Image Results

Results with No Targets

Other Literature

Plenty of work in computer vision, however specific artistic applications are starting to become more popular.

GP-GAN: Towards Realistic High Resolution Image Blending

- Image concatenation with convincing blending

Google's Magenta - Artistic Applications in Music and Pictures

- Style transfers
- Image filling

Paints-Chainer - Artistically fills in line drawings

- Machine Learning Coloring Book

Concluding Remarks

- Bright future for Generative Adversarial Networks
  - Many variations, DCGAN, CGAN, GAN-T21, InfoGan, many more.
  - Many applications for it.
- Growing desire to combine art and machine learning.
- Model had some notable shortcomings.

References

- Wu, H., Zheng, S., Zhang, J., & Huang, K. (n.d.). GP-GAN: Towards Realistic High-Resolution Image Blending
- Vladimir Bychkovsky, Sylvain Paris, Eric Chan Frédo Durand (2011). Learning Photographic Global Tonal Adjustments with a Database of Input/Output Image Pairs. *The Twenty Fourth IEEE Conference on Computer Vision and Pattern Recognition*
- http://www.cnbc.com/2017/04/17/meet-the-man-who-makes-facebooks-machines-think.html?se=toc&so=cu
- https://github.com/vdumoulin/conv_arithmetic
- https://github.com/JHousmanEdi/CMPS-4720-6720