Fast Artistic Style Transfer w/Tensor Flow

Brian Henderson, Sami Ellougani

Introduction

Our goal is simple, transfer the style of one piece of art to another. However, we want to be able to do this fast. Using a combination of Leon A. Gatys *A Neural Algorithm of Artistic Style*, Justin Johnson's *Perceptual Losses for Real-Time Style Transfer and Super-Resolution*, and Dmitry Ulyanov's *Instance Normalization: The Missing Ingredient for Fast Stylization*, we were able to style our own images. Special thanks to Logan Engstrom, for the base that we used to make this possible.

Fast Style Transfer vs. Not Fast

Fast Style Transfer Steps:

Con:

- Training takes a long time and can be tedious to train networks for all styles desired. (6-12hrs per)

Pro:

- Use that network to train any image on demand using that **specific** style. (10 seconds).
- Time efficient if styling multiple images

Regular Style Transfer

Cons:

- Need to train the style while applying to the image.
- Takes lots of time and resources.

Leon A. Gatys *A Neural Algorithm of Artistic Style*

- ❖ The algorithmic basis of this process is unknown
  - ➢ Therefore, other key areas of visual perception has been demonstrated using Deep Neural Networks
- ❖ This artificial system is based on Deep Neural Networks, and creates artistic images of high perceptual quality
- ❖ The system uses neural representations to separate and recombine content and style of arbitrary images
- ❖ This Neural Algorithm provides an understanding of how humans create and perceive artistic imagery

Justin Johnson's ***Perceptual Losses for Real-Time Style Transfer and Super-Resolution***

- Typical feed-forward convolutional neural networks use a per-pixel loss between the output and base image.
- Parallel experiments define and optimize perceptual loss functions based on high-level features extracted from the pretrained networks.

- Proposes the use of perceptual loss functions for training feed-forward networks for image transformation.

Dmitry Ulyanov's ***Instance Normalization: The Missing Ingredient for Fast Stylization***

- Discusses how a minor change in the stylization architecture results in a significant qualitative improvement in the output image.
  - Replace batch normalization with instance normalization layers, and keep it at test time instead of freezing and simplifying as done in batch normalization.
- The normalization process allows to remove instance-specific contrast from the content image, which simplifies image output generation.

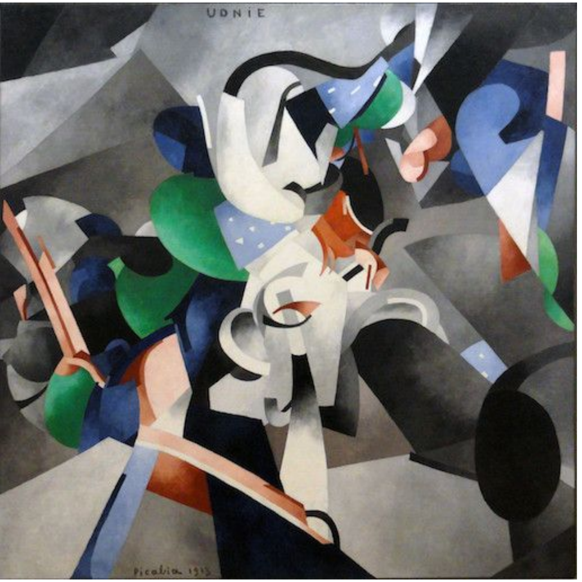
VGG-19

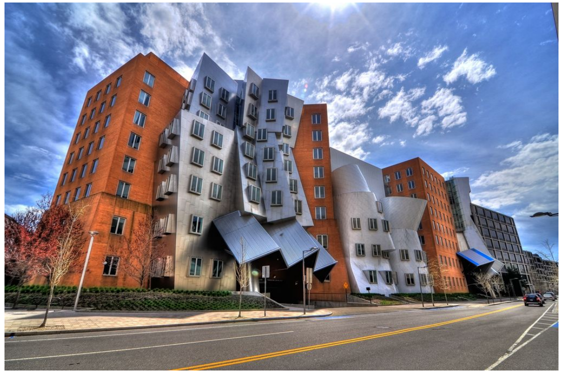
- VGG stands for Visual Geometry Group
- There are two major models
    - VGG-16
    - VGG-19
- Both models differeniate in the amount of layers
- VGG-19 is a 19 layer model that is pretrained against a subset of the ImageNet database
- VGG-19 is trained on more than a million images and can classify images into 1000 object categories.
- As a result, the model has learned rich feature representations for a wide range of images.

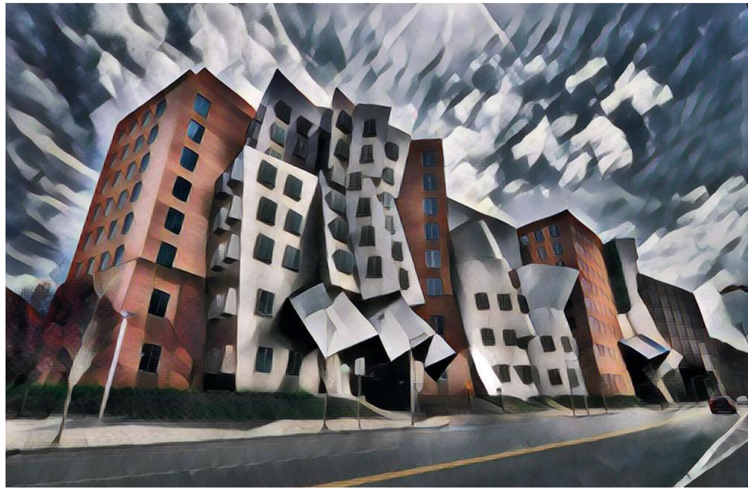
Data Set

❖ The dataset we are utilizing the most is the Microsoft Coco dataset
❖ The Microsoft Coco dataset is mainly utilized for Object segmentation, recognition in context, and captioning
❖ The Coco dataset consists of 330,000 images, 1.5 million object instances, 80 object categories, and 5 captions per image

Test Experiment from Author







Experiments We Ran

UDNIE
Picabia 1913

*Experiments were run on Google’s gCloud Service using a NVIDIA Tesla K80 GPU

00

WEIGHTS

EPOCHS: 3

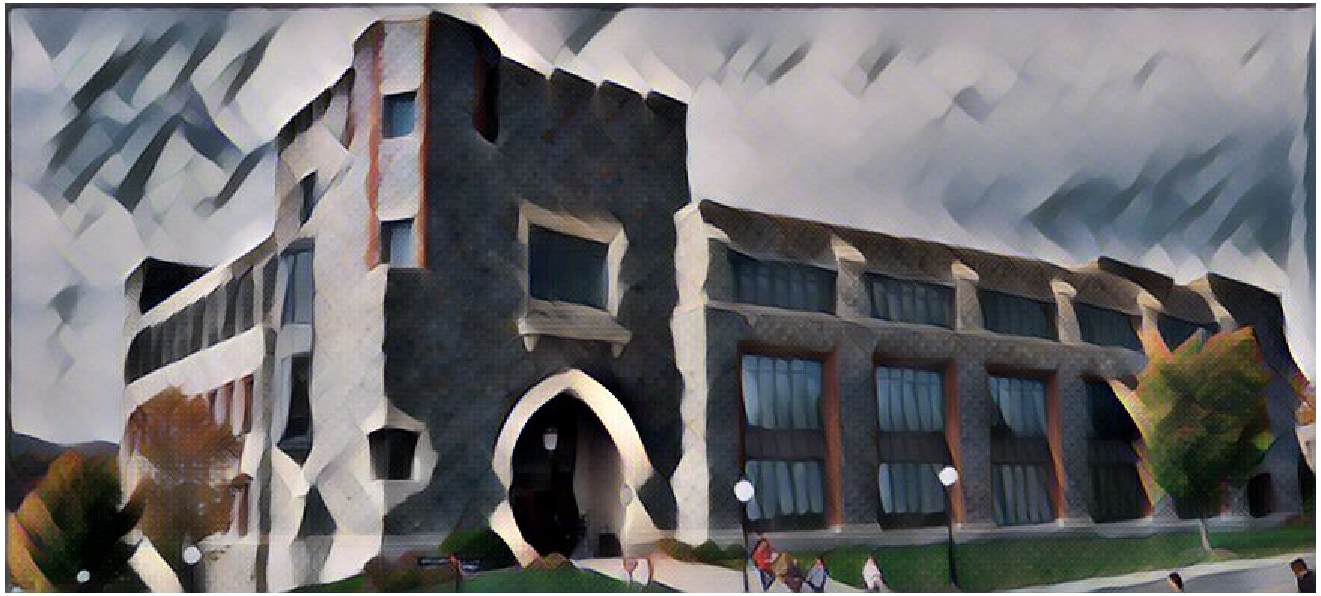
BATCH SIZE: 4

CONTENT WEIGHT: 7.5e0

LEARNING RATE : 1e-3

STYLE WEIGHT: 1e2

TIME TO COMPLETE: ~24hrs



01

WEIGHTS

EPOCHS: 2

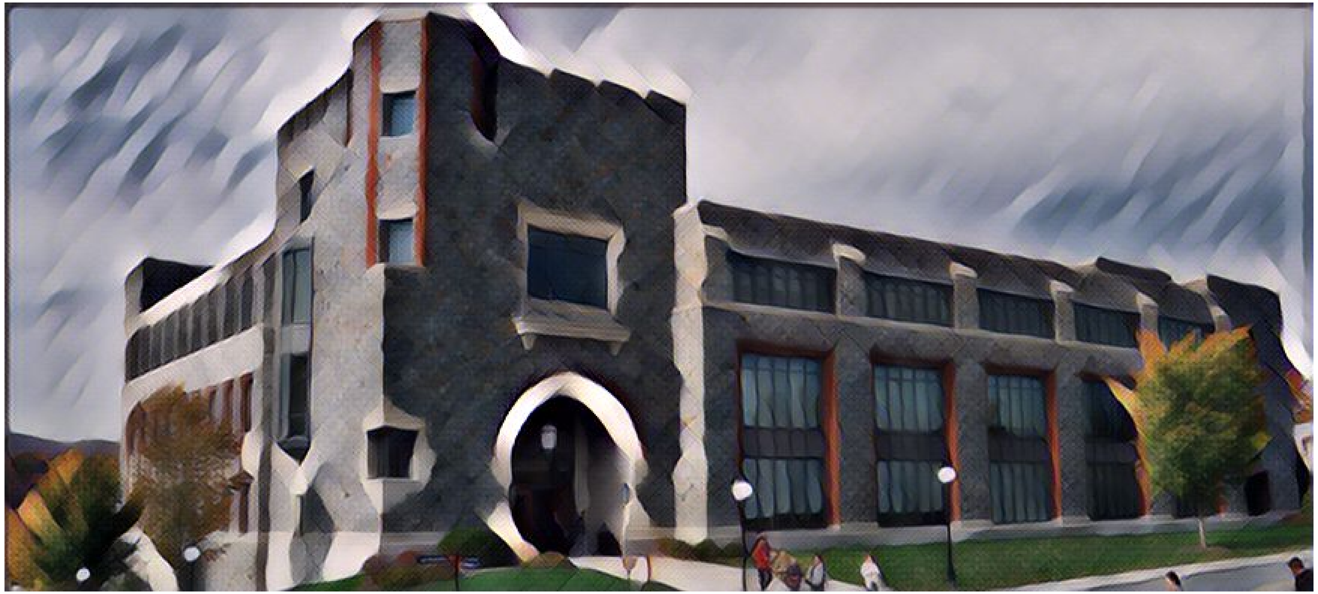
BATCH SIZE: 20

CONTENT WEIGHT: 1.5e1

LEARNING RATE : 1e-2

STYLE WEIGHT : 1e2

TIME TO COMPLETE: ~12hrs



02

WEIGHTS

EPOCHS: 2

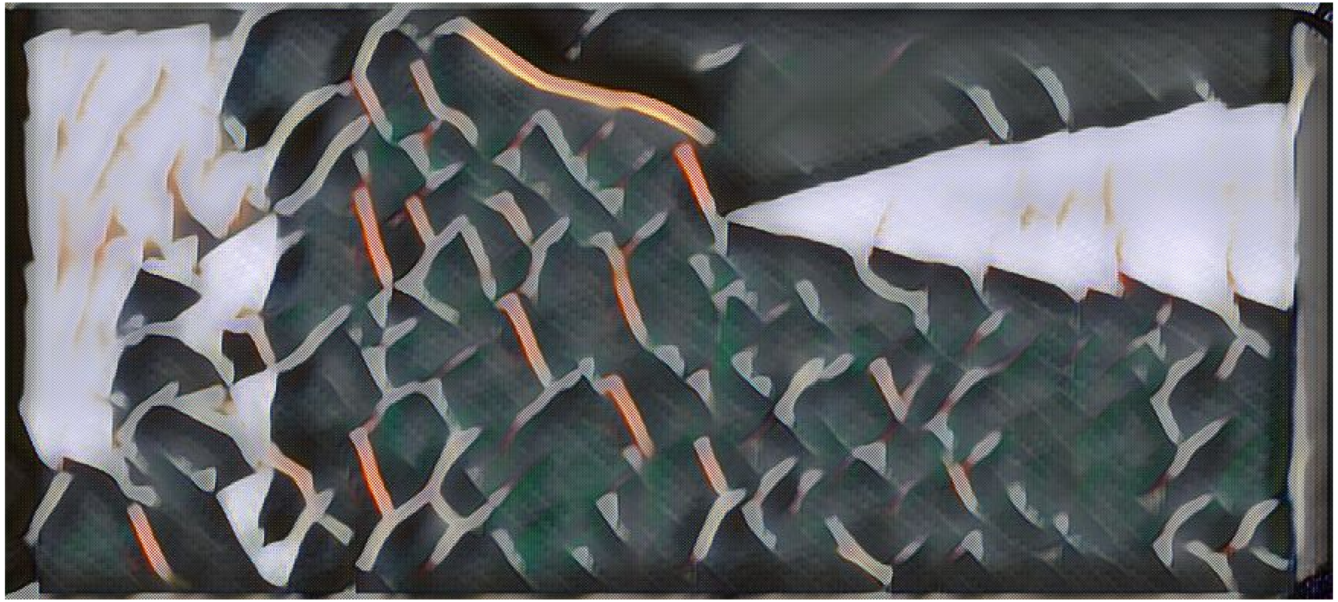
BATCH SIZE: 20

CONTENT WEIGHT: 1.5e1

LEARNING RATE : 1e-2

STYLE WEIGHT : 1e4

TIME TO COMPLETE: ~12hrs



03

WEIGHTS

EPOCHS: 2

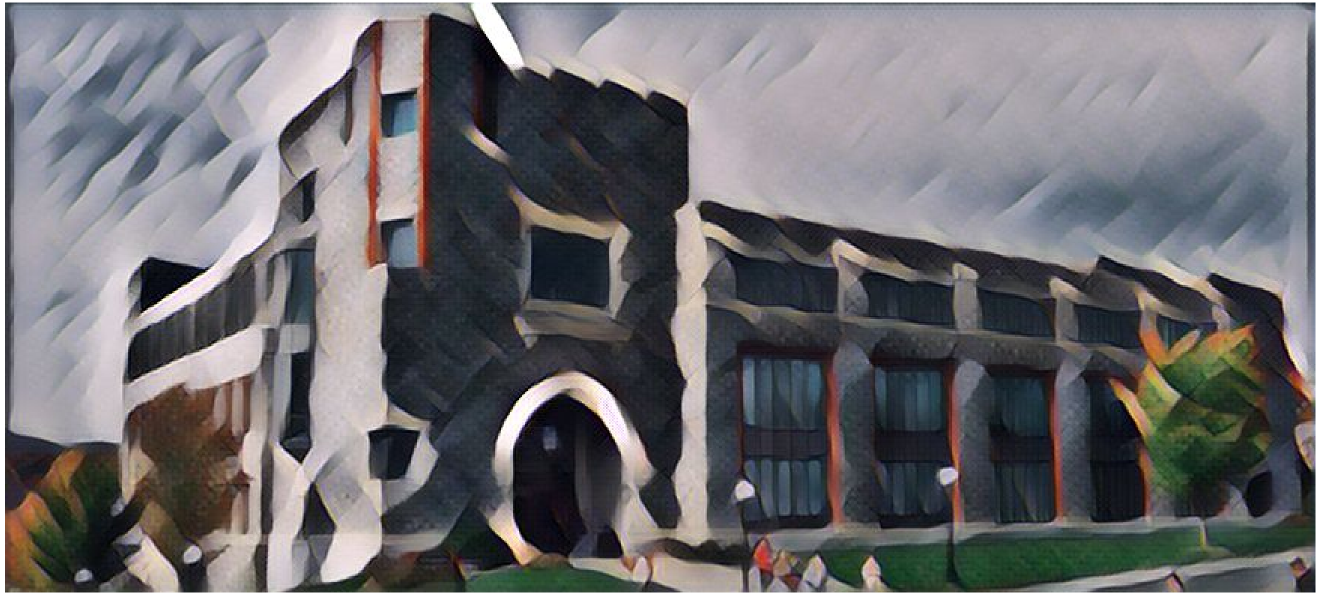
BATCH SIZE: 20

CONTENT WEIGHT: 6e0

LEARNING RATE : 1e-2

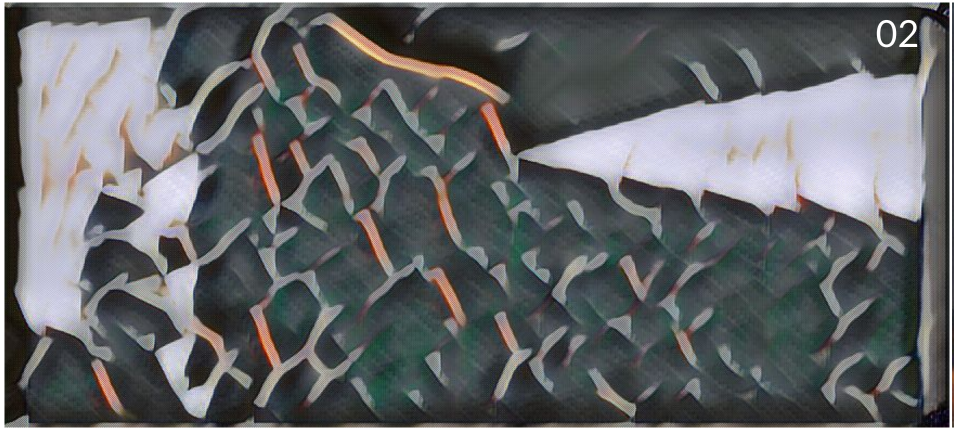
STYLE WEIGHT : 1e2

TIME TO COMPLETE: ~12hrs



00

01


02

03

Resources

https://github.com/lengstrom/fast-style-transfer

https://arxiv.org/abs/1508.06576

http://cs.stanford.edu/people/jcjohns/eccv16/

https://arxiv.org/abs/1607.08022