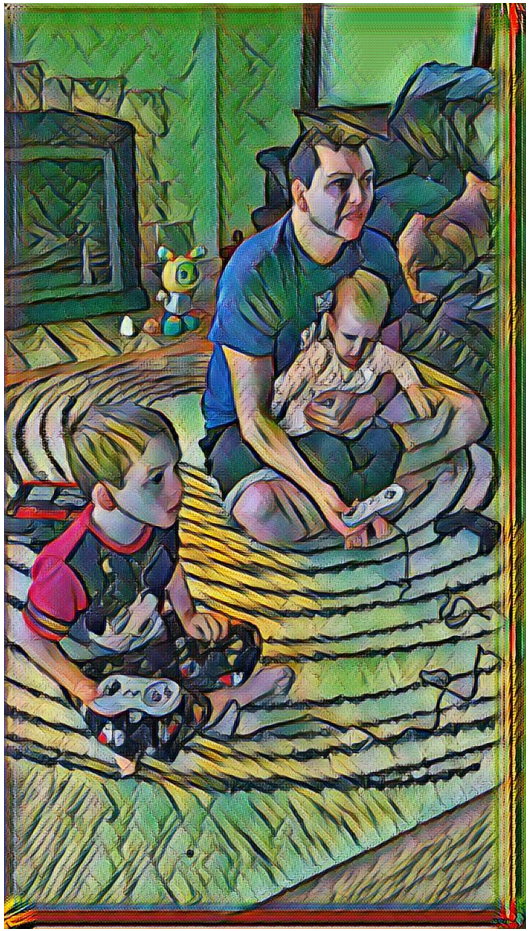
Style Transfer

A cutting-edge Convolutional Neural Network (CNN) application

by Brian McMahon

Objective

- *Introduce Style Transfer*
- *Introduce CNN model*
- *Create a style transfer image*



What is Style Transfer?

- Key image recognition application of CNN
- Cutting edge research topic
- Remake any image using the style of another image
- Model to better understand how a computer, and the human brain, interprets an image
- Machine-generated art

What is a CNN?

- A neural network whose layers are subsampled to focus on individual features, such as edges
- Subsequent layers focus on details of increasing complexity
- Neurons together then connected to cover the entire image
- Key applications include image and video recognition and natural language processing

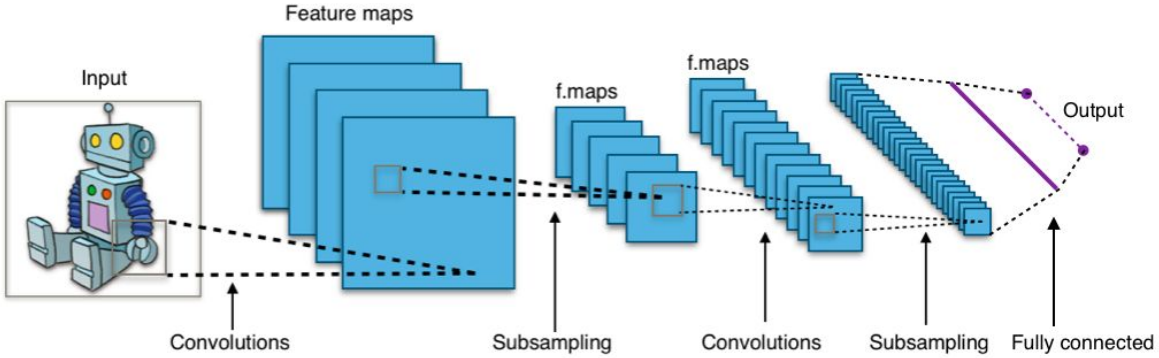


Image source: Wikipedia.

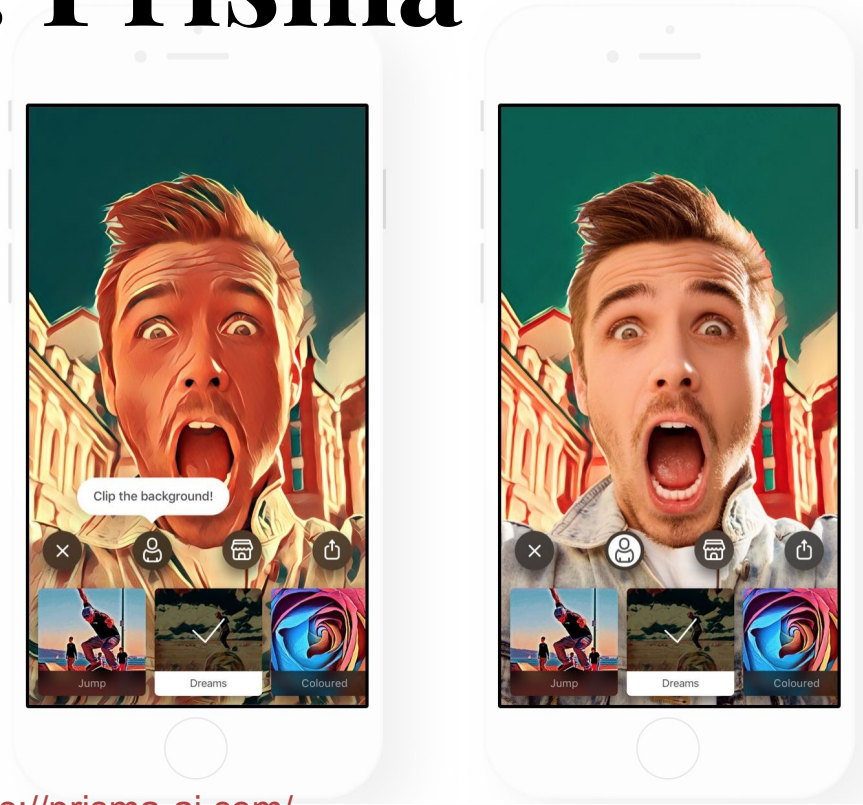
Notable Apps: Deep Dream

- Initially created by Google to study image recognition
- Now a popular form of abstract art

https://deepdreamgenerator.com/
https://github.com/google/deepdream

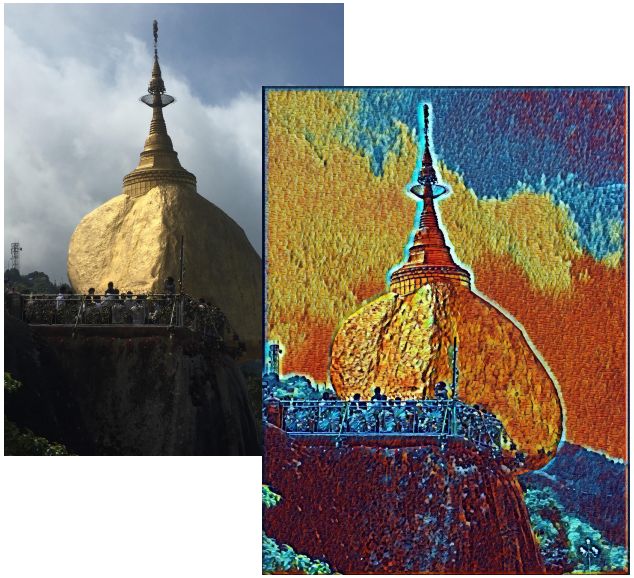
Notable Apps: Prisma

PRISMA



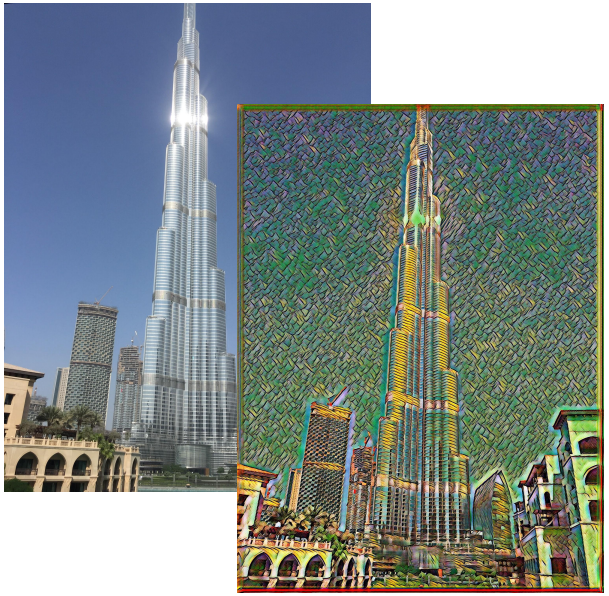
https://prisma-ai.com/

Example Outputs



Rain Princess,
Leonid Afremov

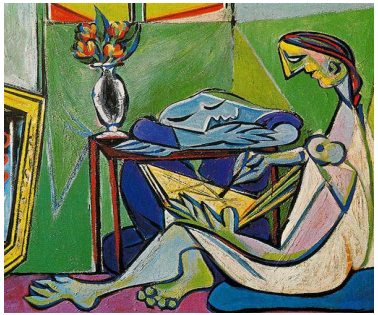
Udnie,
Francis Picabia



La Muse,
Pablo Picasso

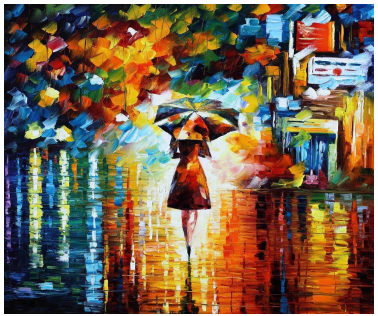
Trained Input Styles

Udnie,
Francis Picabia



La Muse,
Pablo Picasso

The Scream,
Edvard Munch



Rain Princess,
Leonid Afremov

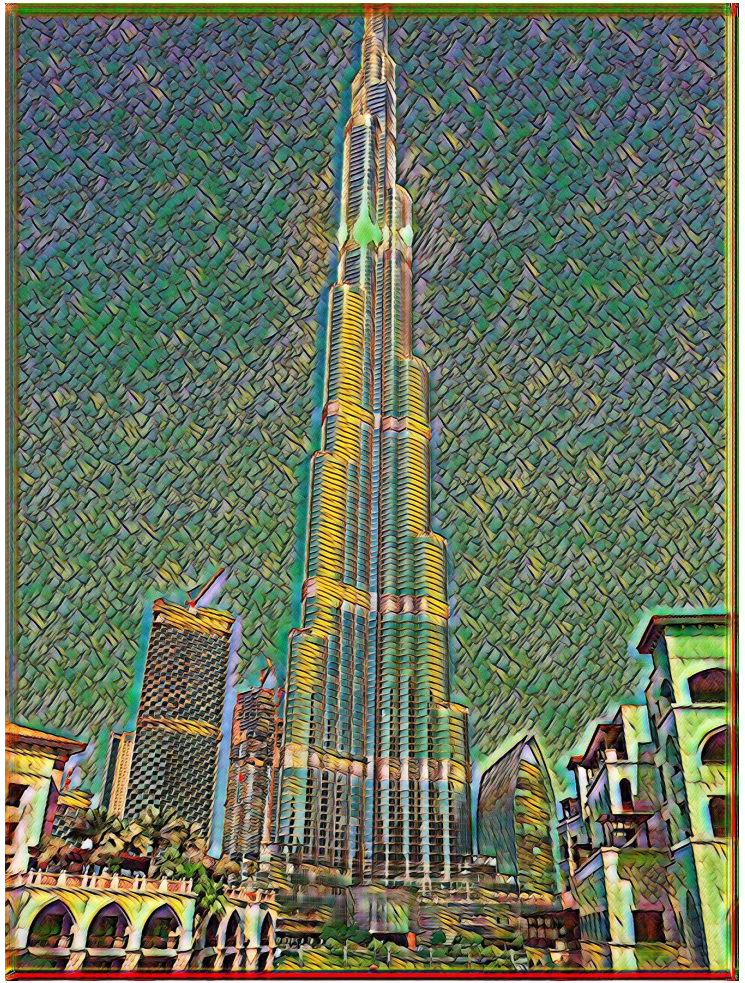
The Great Wave at Kanagawa,
Katsushika Hokusai

Shipwreck of the Minotaur,
J.M.W. Turner

Model Overview

- CNN architecture
- Model trains on a painting, learning its unique style (4-6 hours on Nvidia Titan X)
- Model then applies (“evaluates”) the style to another picture (~10 minutes with GPU acceleration)

Based on Logan Engstrom’s GitHub repo located at
github.com/lengstrom/fast-style-transfer



How to “evaluate”

1. Log into computer with compatible GPU or cloud training capability (ie FloydHub, AWS, Google Cloud)
2. Clone the repo at https://github.com/cipher813/fast-style-transfer.
3. (Ideally) create environment with python 3.5, tensorflow 1.0+, scipy and pillow
    a. GPU acceleration is strongly recommended. To set up GPU acceleration, be sure to pip install tensorflow-gpu
    b. Make sure all GPU drivers installed. If using an Nvidia graphics card, install key drivers including CUDA and cuDNN
4. Evaluating a pre-trained model:
    a. Choose one of the pre-trained paintings, saved as .ckpt files in ckpt/
    b. Choose any content image to stylize

continued...

How to “evaluate” (continued)

6. **Navigate to fast-style-transfer and pass the following code via terminal:**

   python evaluate.py --checkpoint ckpt/<.ckpt file> --in-path <path to picture file>> --out-path output/<output picture name>

   Checkpoint is the style to be transferred, in-path is the content image to be modified and outpath is the output filename, placed in output/

   For example:

   python evaluate.py --checkpoint ckpt/udnie.ckpt --in-path examples/content/GoldenRock.jpg --out-path output/GoldenRockUdnie.jpg

If no GPU...

Run StyleTransfer.ipynb, also located in the repo.

Load any content and style images that you wish.

Note that quality of output will not be comparable to the GPU-accelerated method

+

=

GPU Model

Non-GPU Jupyter Notebook

Credit / Further Reading

Research Papers:

Leon A. Gatys, Alexander S. Ecker, Matthias Bethge. ‘A Neural Algorithm of Artistic Style.’ 2 Sept. 2015. https://arxiv.org/abs/1508.06576.

Justin Johnson, Alexandre Alahi, Fei-Fei Li. ‘Perceptual Losses for Real-Time Style Transfer and Super-Resolution.’ 2016. https://cs.stanford.edu/people/jcjohns/eccv16/.

Dmitry Ulyanov, Andrea Vedaldi, Victor Lempitsky. ‘Instance Normalization: The Missing Ingredient for Fast Stylization.’ 6 Nov 2017. https://arxiv.org/abs/1607.08022.

Karen Simonyan, Andrew Zisserman. ‘Very Deep Convolutional Networks for Large-Scale Image Recognition.’ 10 Apr 2015. https://arxiv.org/abs/1409.1556. (CNN; VGG-19)

Blogs:

Harish Narayanan. ‘Convolutional neural networks for artistic style transfer.’ 31 Mar 2017. https://harishnarayanan.org/writing/artistic-style-transfer/.

Credit / Reading (continued)

GitHub Repositories:

Lengstrom. “Fast Style Transfer.” https://github.com/lengstrom/fast-style-transfer.

IISourcell. “How to Do Style Transfer in Tensorflow.” https://github.com/IISourcell/How_to_do_style_transfer_in_tensorflow.

IISourcell. “How to Generate Art.” https://github.com/IISourcell/How-to-Generate-Art-Demo.

Hnarayanan. “Stylist.” https://github.com/hnarayanan/stylist.

Applications:

Oleg Poyaganov. ‘DIY Prisma app with CoreML.’ https://blog.prismalabs.ai/diy-prisma-app-with-coreml-6b4994cc99e1. (Prisma mobile app)

Questions?

Agenda

- *Who is involved? (A person? A company?)*
- *What is the problem that's being addressed?*
- *What data is available, being used, or relevant for a technique?*
- *What techniques are being developed/applied?*
- *How does the application relate to other topics of the class?*
- *How does the application relate to the world? What are the implications? (Opportunities? Risks?)*

How it Works

- Utilizes VGG-Network, a deep Convolutional Neural Network (CNN) architecture for object recognition.
- 16 convolutional and 5 pooling layers of the 19 layer VGGNetwork (VGG-16); avg. vs. max pooling; no fully connected layers