Neural Style Transfer

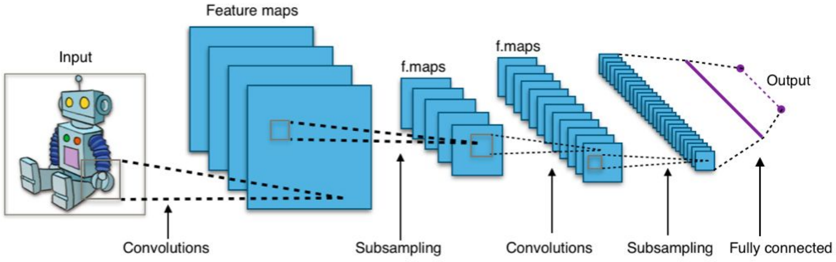
BOSTON UNIVERSITY
MACHINE INTELLIGENCE
COMMUNITY

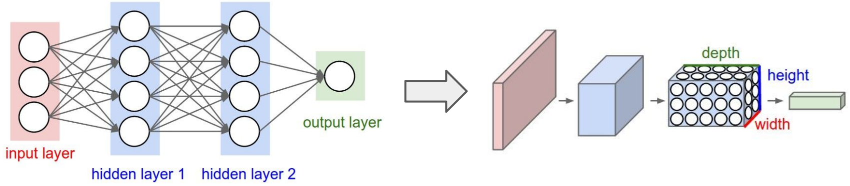
Tammy Qiu
Rex Wang
Dec. 11, 2017

Agenda

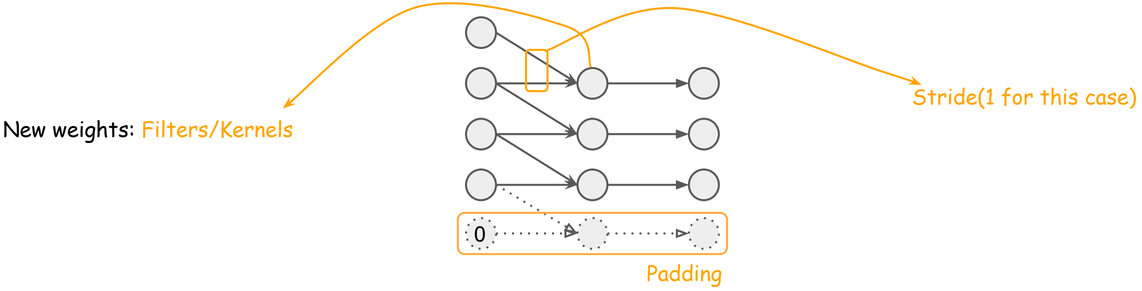
- Convolutional Neural Networks
- What is Neural Style Transfer (NST)?
- How Neural Style Transfer works?
- Some appealing results of NST
- What is Fast Neural Style Transfer?
- Art parts!!
- Real-Time Demo
- Wrapping up for “Fundamentals of Deep Learning” - MIC Workshop series

Recap: Convolutional Neural Networks



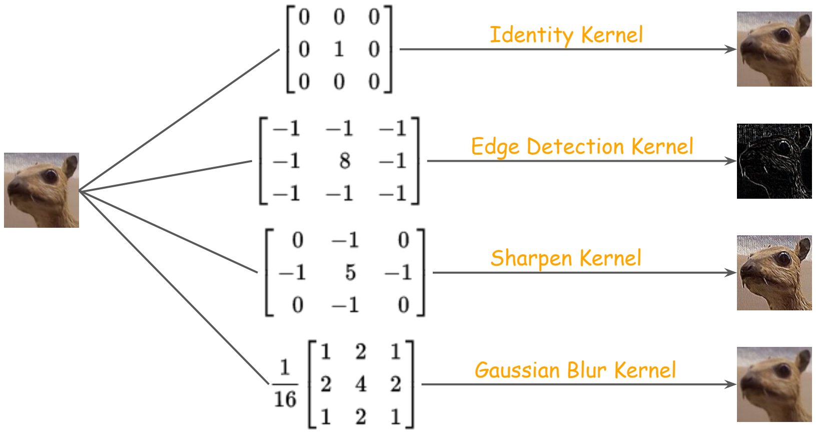


- Neural Network with convolutional layers: layers use convolutions instead of general matrix multiplications.
- ConvNet architectures make the explicit assumption that the inputs are images(this is true for most of the time)
    - Process visual information hierarchically
    - Feed-forward



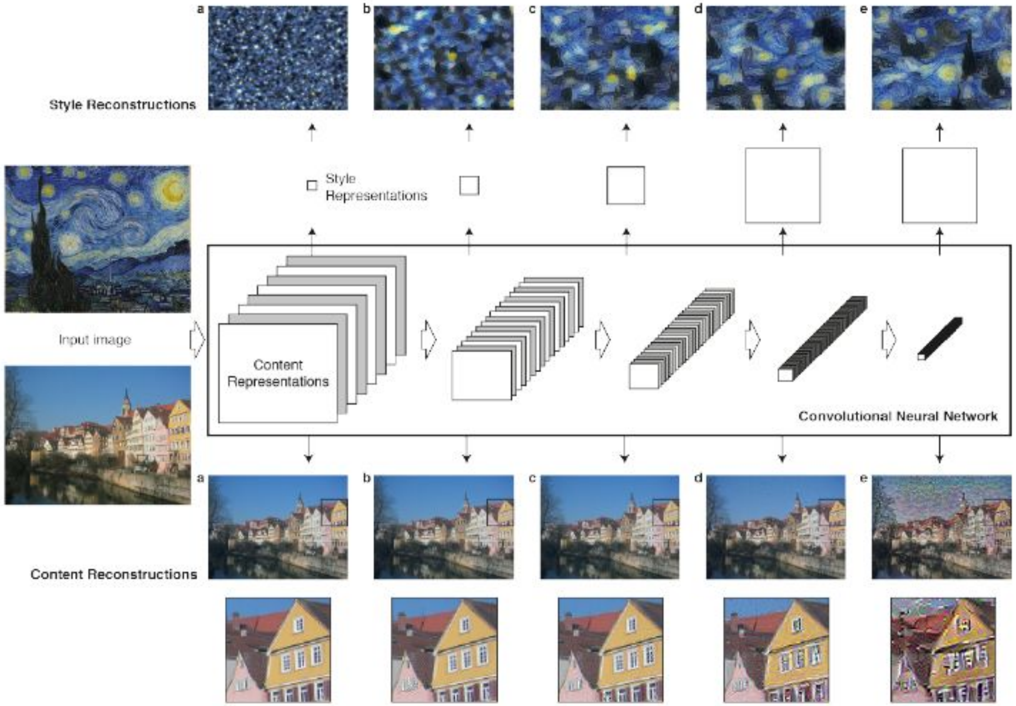
Convolutional Neural Networks - Kernels

- Kernels can learn/extract specific representations/features of the hierarchical data (images), which can be used to process images(by doing a convolution between a kernel with an image).



- Each output layer of neurons/kernels can be regarded as a collection of image filters,
  - So-called **feature maps**: differently filtered versions of the input image (this is important to Neural Style Transfer)
- Popular architectures CNNs: LeNet, AlexNet, ZF Net, GoogleNet, VGGNet, ResNet, etc.
- Recommended Materials: Stanford CS231n Convolutional Neural Networks for Visual Recognition

So what is Neural Style Transfer (NST)?



M I C

How does Neural Style Transfer work?

- Transfer style by updating pixels in image iteratively through backpropagation
- Three images involved:

Content image:
Picture we want to transfer style onto

Style image:
Artwork whose style we want to transfer

Pastiche:
Final stylized image
Initialized to random/white noise

Defining the Loss Function

- **Objective**: minimize loss so pastiche matches content of content image and style of style image
- **Content** Loss
  - $\vec{p}$: original content image
  - $\vec{x}$: pastiche
  - l : layer

Feature Representation of $\vec{p}$

$$L_{content}(\vec{p}, \vec{x}, l) = \frac{1}{2} \sum_{ij} (F^l_{ij} - P^l_{ij})^2$$

Feature Representation of $\vec{x}$

M I C

Defining the Loss Function

- **Style** Loss
  - **Gram Matrix**: inner product between the vectorised map *i* and *j*
  - Computes correlations between different features

$$G^l_{ij} = \sum_k F^l_{ik} F^l_{jk}$$

  - Minimize mean-squared distance between entries of Gram matrix from style image and Gram matrix of pastiche

$$E_l = \frac{1}{4N_l^2 M_l^2} \sum_{i,j} (G^l_{ij} - A^l_{ij})^2$$

M I C

Defining the Loss Function

- **Total Style Loss** across multiple layers

$$L_{style}(\overrightarrow{a}, \overrightarrow{x}) = \sum_{l=0}^{L} w_l E_l$$

M I C

Defining the Loss Function

- Total Loss: Jointly minimize distance of a white noise image from
    - Content representation of the photograph in one layer of the network
    - Style representation of the painting in a number of layers of the CNN

$$L_{total}(\overrightarrow{p}, \overrightarrow{a}, \overrightarrow{x}) = \alpha L_{content}(\overrightarrow{p}, \overrightarrow{x}) + \beta L_{style}(\overrightarrow{a}, \overrightarrow{x})$$

- α, β : weighting factors for content and style reconstruction respectively

M I C

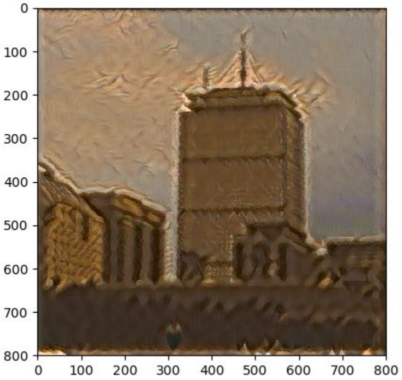
Art Time!!

Content Images

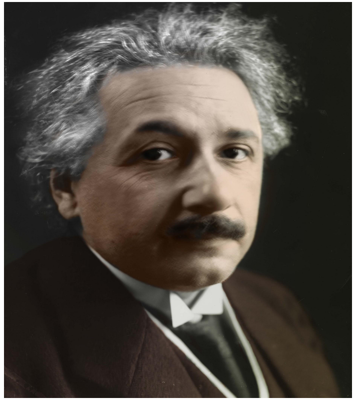
Neural Style Transfer with PYTORCH

+

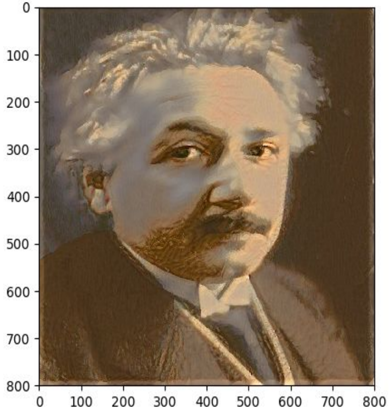
1000 iterations



+



1000 iterations

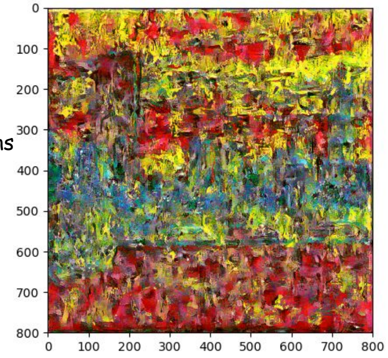


M I C

Neural Style Transfer with PYTORCH

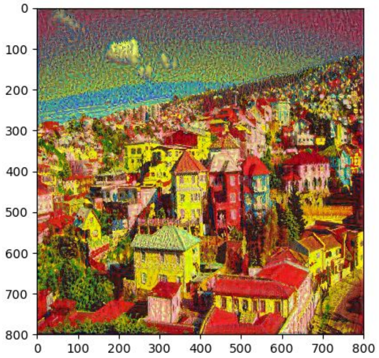
+

1000 iterations



+

1000 iterations

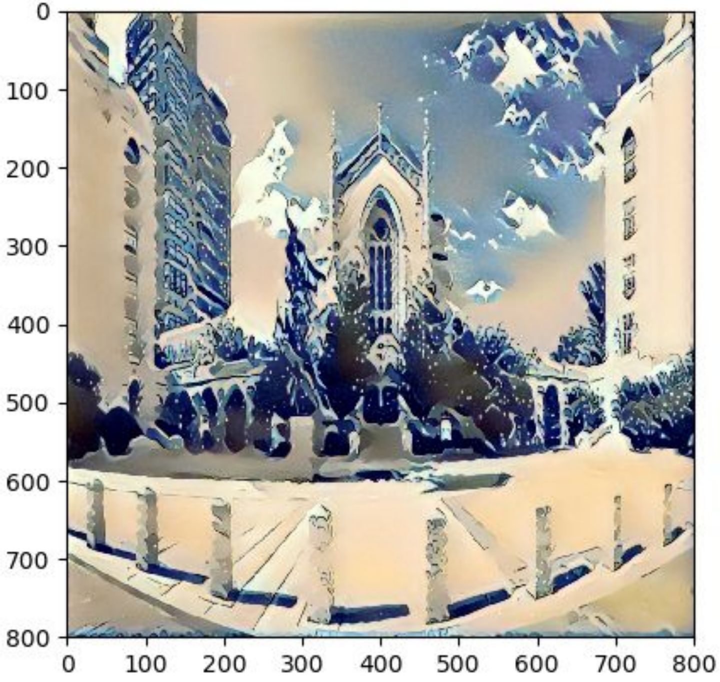


M I C

Neural Style Transfer with PYTORCH

+

1000 iterations

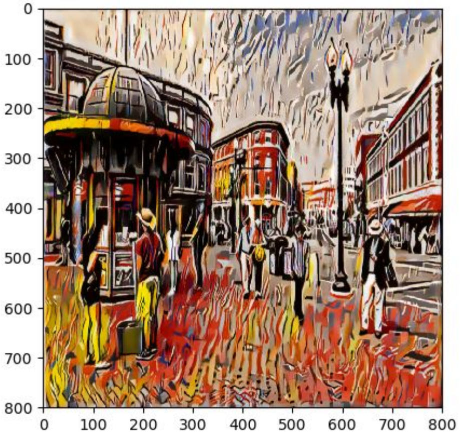


M I C

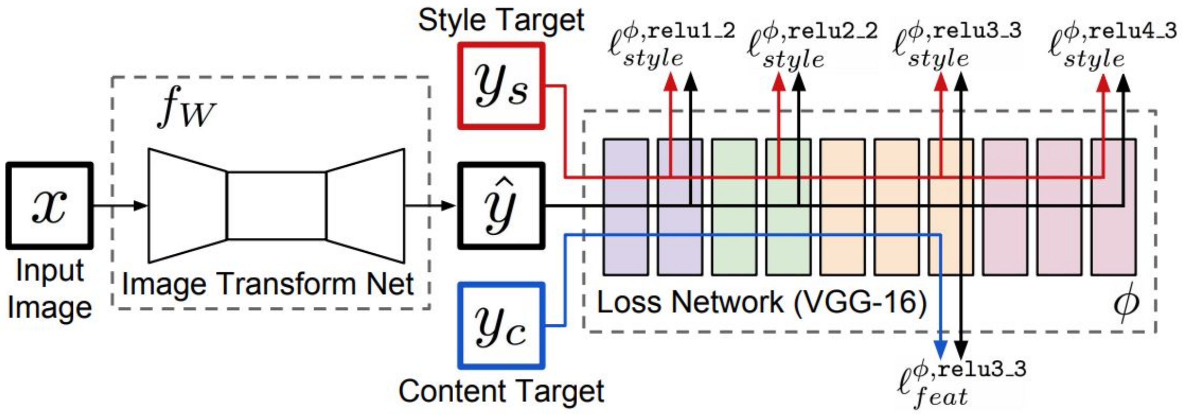
Neural Style Transfer with PYTORCH

+

1000
iterations



One More Thing: Fast Neural Style Transfer!



- Take an untrained Image Transformation Network
- Train network to learn the style of the artwork
- Feed-forward once
- “Perceptual Losses for Real-Time Style Transfer and Super-Resolution”

What does this all mean?

- "Offers path forward to algorithmic understanding of how humans create and perceive artistic imagery"
- What does this mean for art?
- People are already beginning to explore this exciting intersection:
  - http://www.creativeai.net/
  - https://www.artnome.com/
  - https://medium.com/artists-and-machine-intelligence
    - https://ami.withgoogle.com

M I C

References & Further Reading

1. Gatys, Leon A., Alexander S. Ecker, and Matthias Bethge. "**A neural algorithm of artistic style.**" arXiv preprint arXiv:1508.06576 (2015).
2. Johnson, Justin, Alexandre Alahi, and Li Fei-Fei. "**Perceptual losses for real-time style transfer and super-resolution.**" European Conference on Computer Vision. Springer International Publishing, 2016.
3. Gatys, Leon A., Alexander S. Ecker, and Matthias Bethge. "**Image style transfer using convolutional neural networks.**" Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016.
4. Ulyanov, Dmitry, et al. "**Texture Networks: Feed-forward Synthesis of Textures and Stylized Images.**" ICML. 2016.
5. Shen, Tianxiao, et al. "**Style Transfer from Non-Parallel Text by Cross-Alignment.**" arXiv preprint arXiv:1705.09655 (2017).
6. Yang, Shuai, et al. "**Awesome Typography: Statistics-Based Text Effects Transfer.**" arXiv preprint arXiv:1611.09026 (2016).
7. Kernel (image processing): https://en.wikipedia.org/wiki/Kernel_(image_processing)

Workshops Recap: Fundamentals of Deep Learning

Sep. 12	Machine Intelligence
Sep. 19	Gradient-Based Learning
Oct. 3	Neural Networks
Oct. 10	Regularization
Oct. 18	Compositional Data
Oct. 31	Transfer Learning
Nov. 7	Sequential Data
Nov. 14	Deep Reinforcement Learning
Nov. 28	Unsupervised Learning
Dec. 12	Neural Style Transfer

2017

M I C

What is NEXT?

- Reading Groups!
- Reinforcement Learning Series?!
- Research Groups!!!

- What do you guys want?

Shoutout to our Sponsors

Boston University Computer Science

Boston University SPARK!

Boston University Software Application and Innovation Lab