Style Transfer

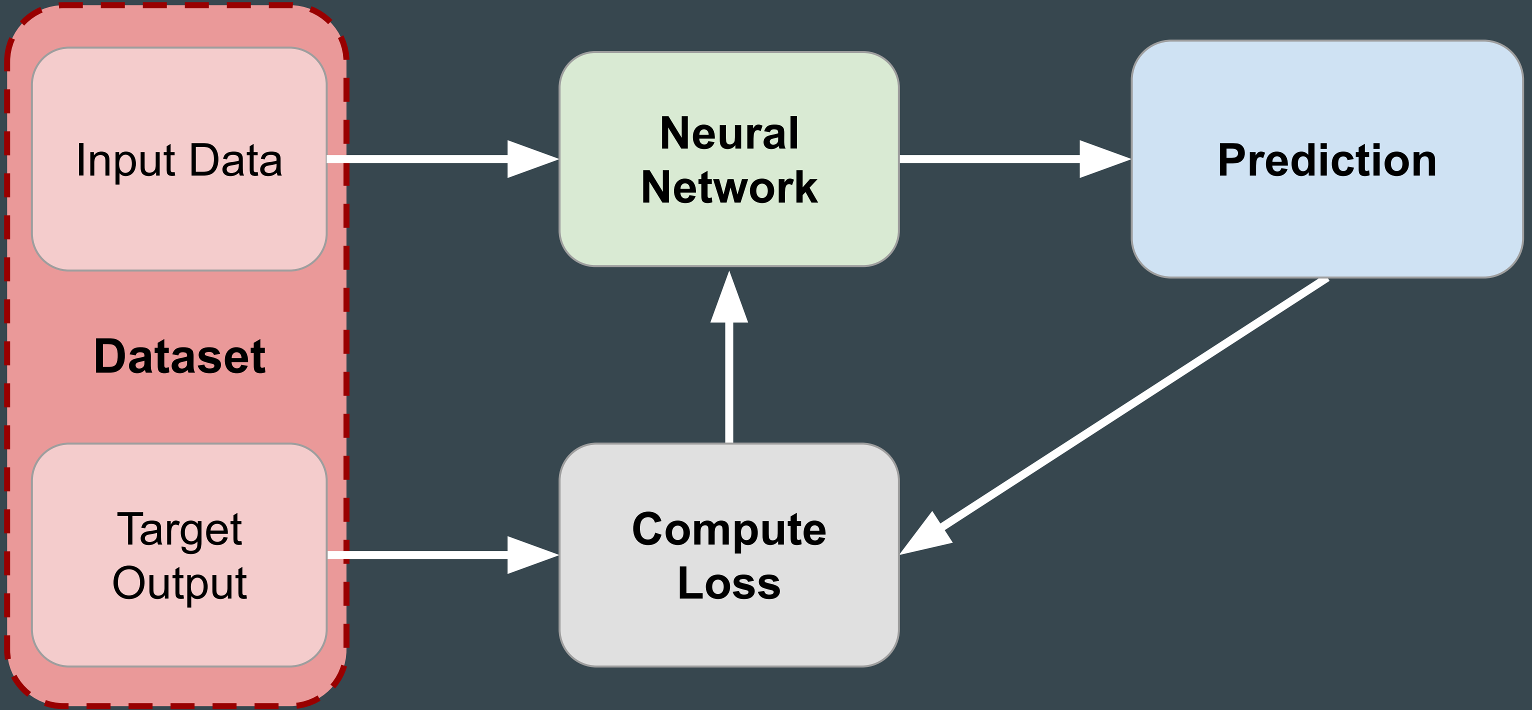
Presented by: Alex Ayala & Piper Wolters

What is style transfer?

Given a *content* image and a *style reference* image...

...generate an image that retains the content of the original image but appears to be painted in the style of the reference image

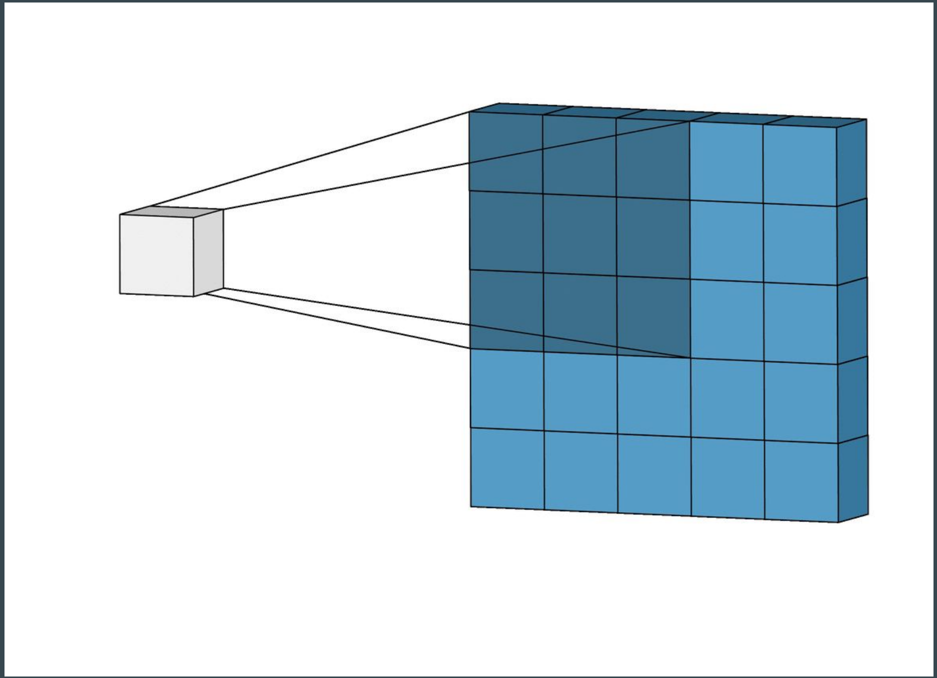
Quick Deep Learning Overview



Convolutional Layers

A specific type of mathematical transformation that utilizes **convolutions** in place of regular matrix multiplication.

A convolution applies filters to your input data. These filters are tuned during training.

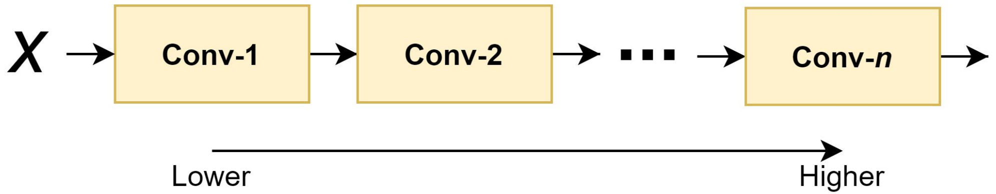


Convolutional Neural Network (CNN)

- Neural Network architecture that consists of multiple convolution layers in series

- Lower levels tend to capture the low-level details (lines and edges)

- Higher levels tend to capture the high-level details (complex shapes and structure)



height

width

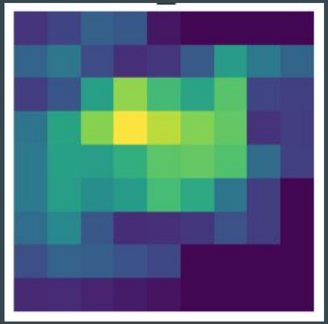
$$\overbrace{\sigma_j(x)}^{\text{Output of conv-}j} \in \mathbb{R}^{C_j \times H_j \times W_j}$$

channels

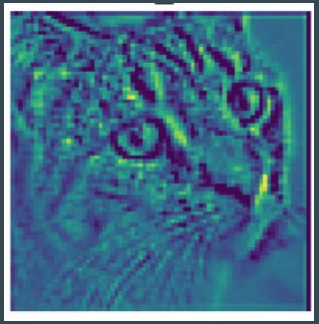
In Class Problem - Sort Intermediate Representations

Based on the idea that ***lower layers*** of a CNN tend to capture ***low-level details*** such as edges and lines, and ***higher layers*** tend to capture ***high-level details*** such as structure…

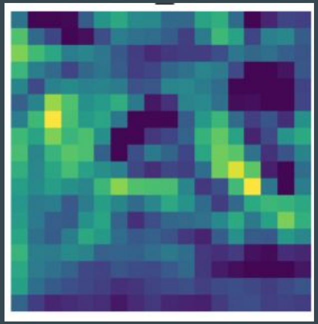
Order the following intermediate representations from lowest layer → highest layer



E

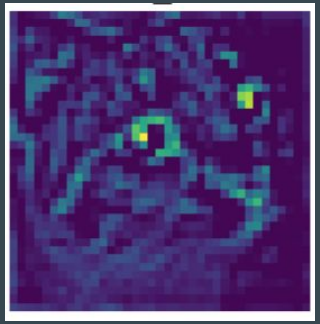


T



L

S



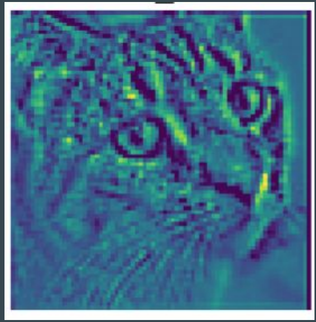
Y

In Class Problem - Sort Intermediate Representations

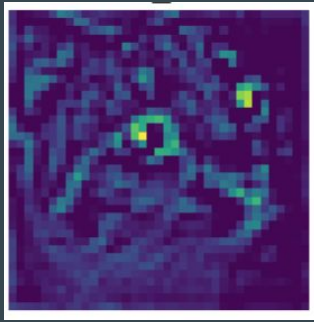
Based on the idea that ***lower layers*** of a CNN tend to capture ***low-level details*** such as edges and lines, and ***higher layers*** tend to capture ***high-level details*** such as structure

Order the following intermediate representations from lowest layer → highest layer

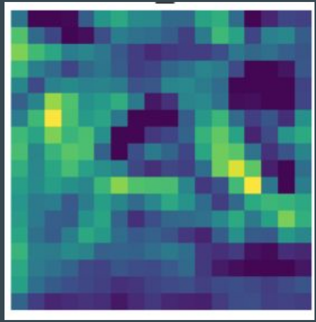
S



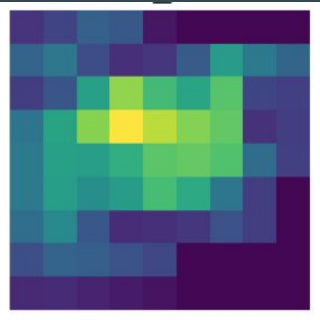
T



Y



L



E

Style Transfer

- Images:
  - Generated Image
  - Content Image
  - Style Image

- The generated image should have similar *content* to the content image and similar *style* to the style image.

- Style Transfer optimizes these two objectives

Perceptual Loss aka “Overall” Loss

- In Deep Learning, we’re always optimizing a loss term

- Perceptual Loss:
  - Content Loss
  - Style Loss

Coefficient or weight for the content loss

Coefficient or weight for the style loss

$$L_P = \lambda_c L_c + \lambda_s L_s$$

Content Loss

Style Loss

Content Loss

- The generated output should have the same *content* as our content image
- Higher layers of a CNN captures this abstract content
- We require that the generated output and the content image are similar within this space

Content Loss

Convolution output of our image

$$L_c = \frac{1}{C_j H_j W_j} ||\sigma_j(x) - \sigma_j(y_c)||_2^2$$

Dimension of convolution output

Convolution output of our *content* image

Gram Matrices

- Intuitively, the dot product is large if two vectors are similar
- Gram Matrices plays off this idea, it considers how similar two filters are
- Performs the "dot product" of two matrices for each combination of channels

$$A \cdot B = |A||B|\cos(\theta)$$

$$\overbrace{\sigma_j(x)}^{\text{Output of conv-}j} \in \mathbb{R}^{C_j \times H_j \times W_j}$$

$$G(x)_{j,c,c'} = \frac{1}{C_j H_j W_j} \overbrace{\sum_{h=1}^{H_j} \sum_{w=1}^{W_j} \sigma_j(x)_{c,h,w} \sigma_j(x)_{c',h,w}}^{\text{"dot product" between 2 matrices}}$$

Style Loss

- The generated output should have the same *style* as our style image
- Lower layers of a CNN captures these artistic designs and colors
- We require that the generated output and the style image have similar Gram Matrices, effectively capturing the style

Gram Matrix for our image

Gram Matrix for our *style* image

$$L_s = ||G_j(x) - G_j(y_s)||_F^2$$

Perceptual Loss aka “Overall” Loss

- In Deep Learning, we’re always optimizing a loss term
- Perceptual Loss:
  - Content Loss
  - Style Loss

Coefficient or weight for the content loss

Coefficient or weight for the style loss

$$L_P = \lambda_c L_c + \lambda_s L_s$$

Content Loss

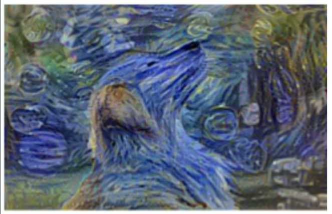
Style Loss

(a) Content / Style (b) $\gamma = 0.1$ (c) $\gamma = 0.2$ (d) $\gamma = 1.0$ (e) $\gamma = 5.0$ (f) $\gamma = 10.0$

Adjusting the Style Coefficient

In Class Problem - Match the Generated Image to the Losses

Given this content image and style reference image, match the two generated images to the loss descriptions!



Large content loss & large style loss

Small content loss & large style loss

Small content loss & small style loss

Large content loss & small style loss

In Class Problem - Match the Generated Image to the Losses

Given this content image and style reference image, match the two generated images to the loss descriptions!

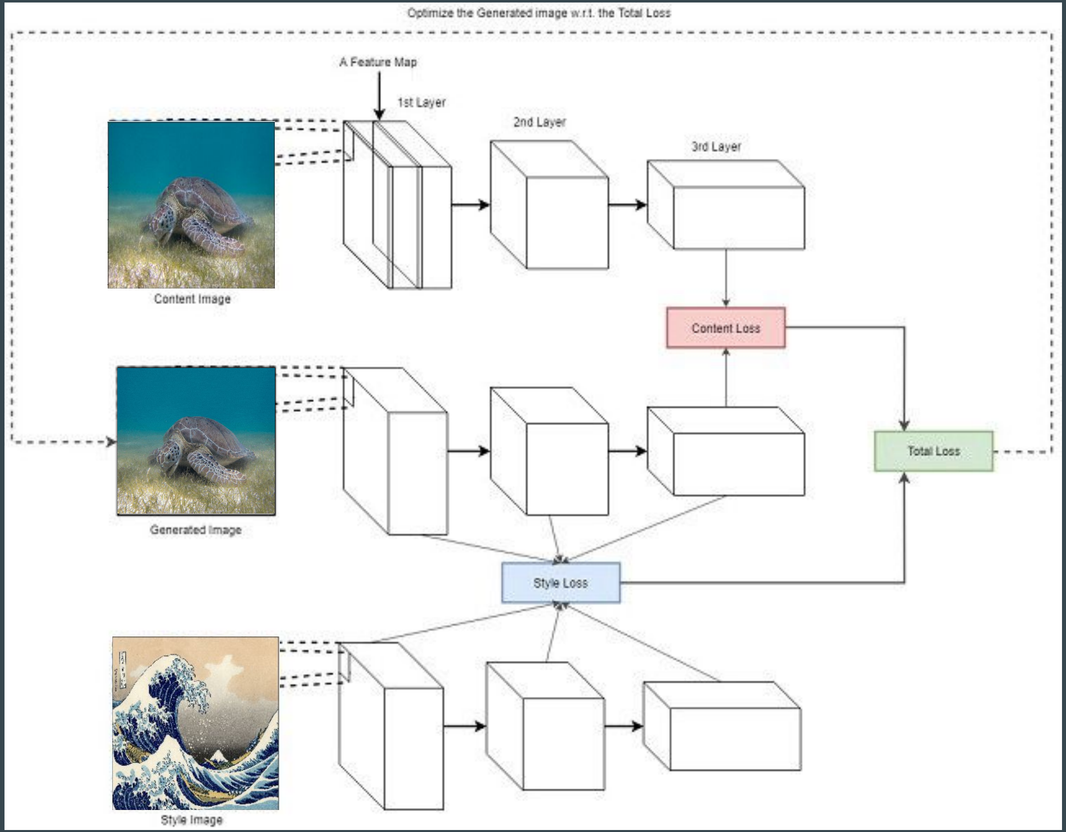
Large content loss & large style loss

Small content loss & large style loss

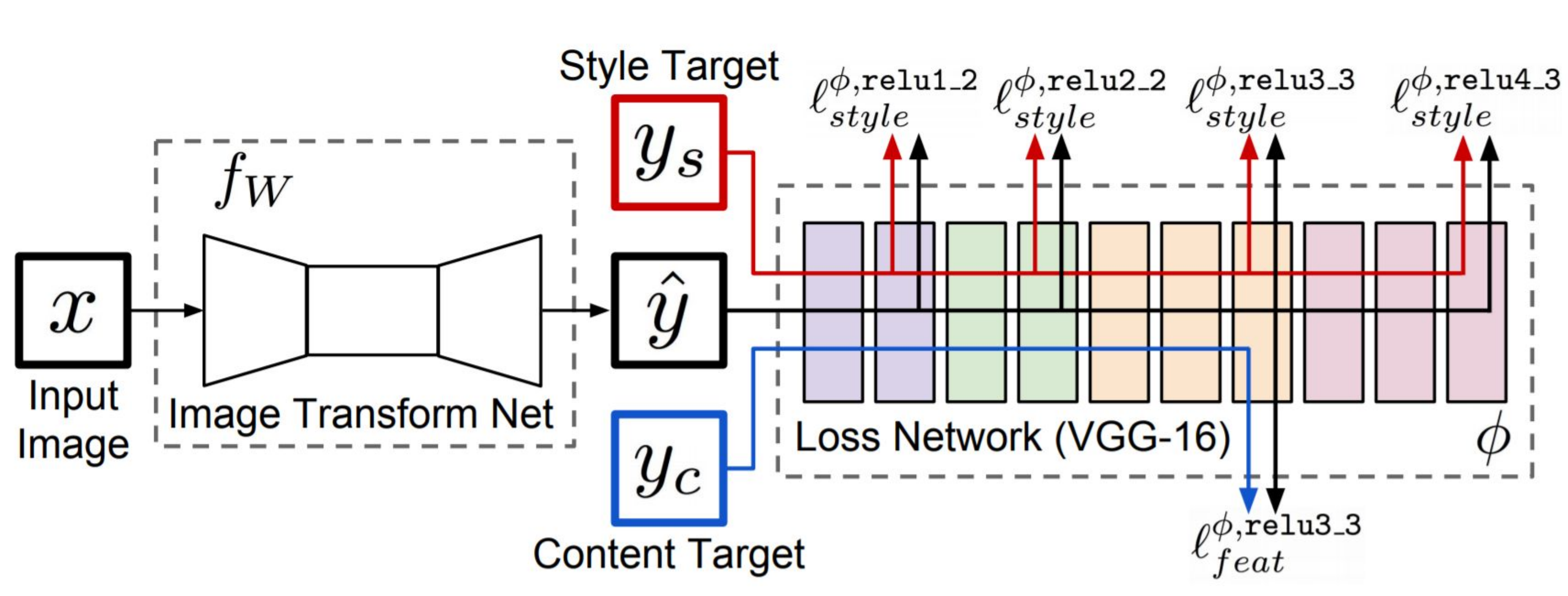
Small content loss & small style loss

Large content loss & small style loss

Neural Style Transfer Training

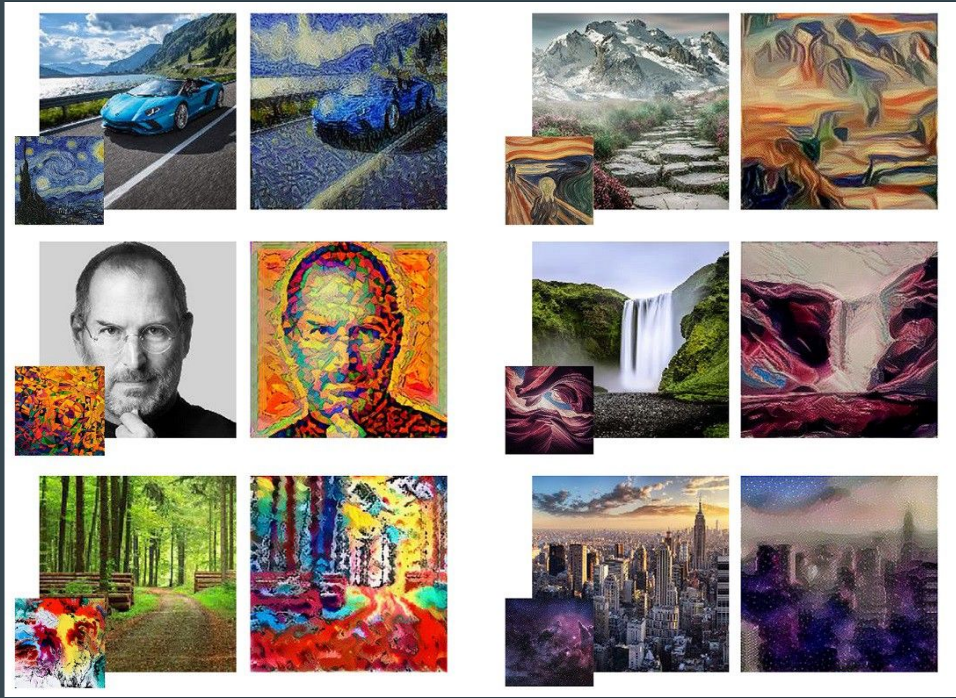


Feed-Forward Style Transfer Learning



Where do we go from here?

- Style Transfer began with images, but expanded towards videos and 3D space
- Consumer tools utilizing Style Transfer are becoming popular



Applications

Applications

Questions??