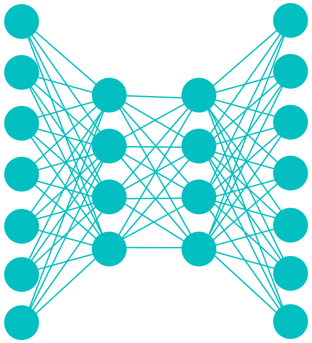
Lecture Notes for

Neural Networks and Machine Learning

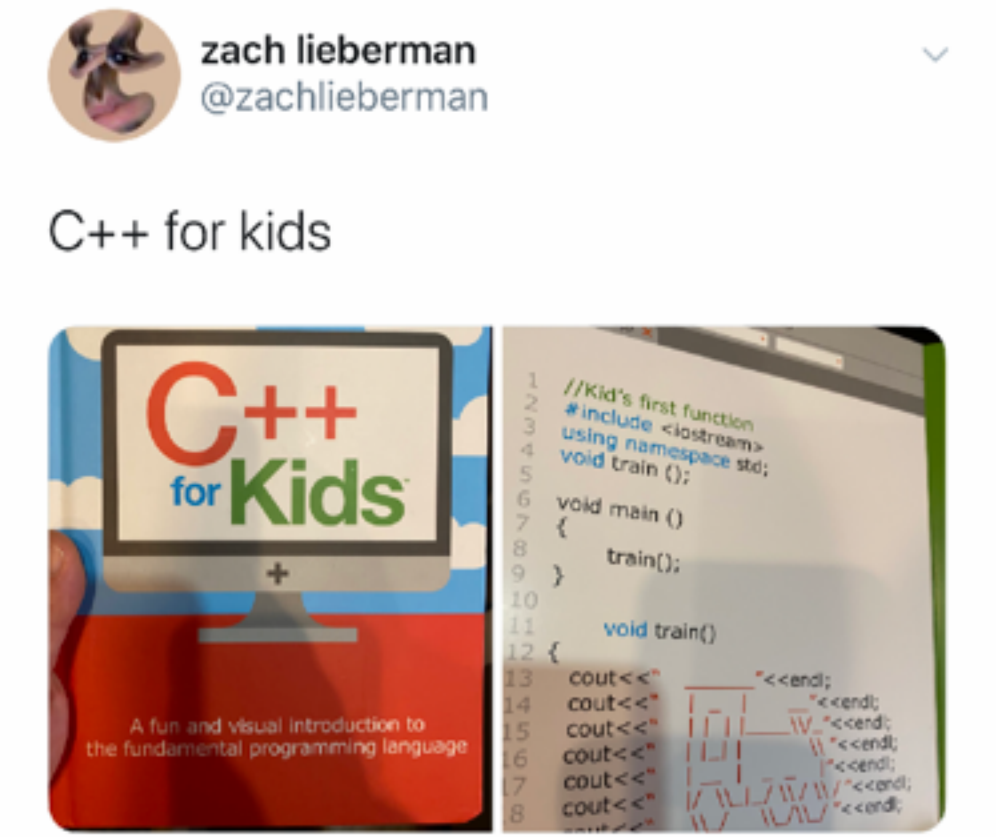
Neural Style Transfer
Model Optimization



Logistics and Agenda

- Logistics
  - Next Assignment: Style Transfer
  - Next Lecture: “Fast Style Transfer” Student Presentation
- Agenda
  - *A History of Style Transfer (last time)*
  - *Image Optimization Algorithms (last time)*
  - Student Paper Presentation
  - Model Optimization Algorithms (today)
  - One Shot Algorithms (today)
  - Evaluating Style Transfer Performance
  - Extensions in Other Domains

Model Optimization Based Style Transfer



zach lieberman
@zachlieberman

C++ for kids

C++ for Kids

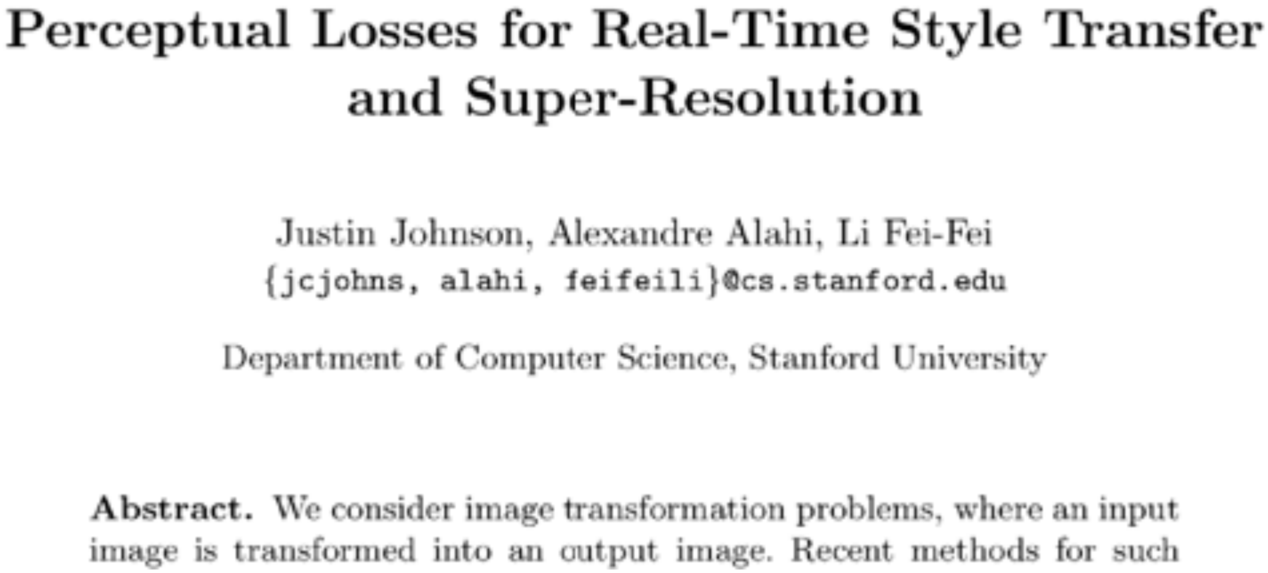
A fun and visual introduction to the fundamental programming language

```
//Kid's first function
#include <iostream>
using namespace std;
void train ();

void main ()
{
        train();
}

        void train()
{
   cout<<"   ________"<<endl;
   cout<<"   | _ |      __"<<endl;
   cout<<"   | | | |_____\V_"<<endl;
   cout<<"   | |_| |        \\ "<<endl;
   cout<<"   |     |          |"<<endl;
   cout<<"   |__   | _  _  \/"<<endl;
   cout<<"   |/ \\_|_/ \\/ \\/"<<endl;
   cout<<"   \\_/    \\_/\\_/
```

Paper Presentation



Perceptual Losses for Real-Time Style Transfer and Super-Resolution

Justin Johnson, Alexandre Alahi, Li Fei-Fei
{jcjohns, alahi, feifeili}@cs.stanford.edu

Department of Computer Science, Stanford University

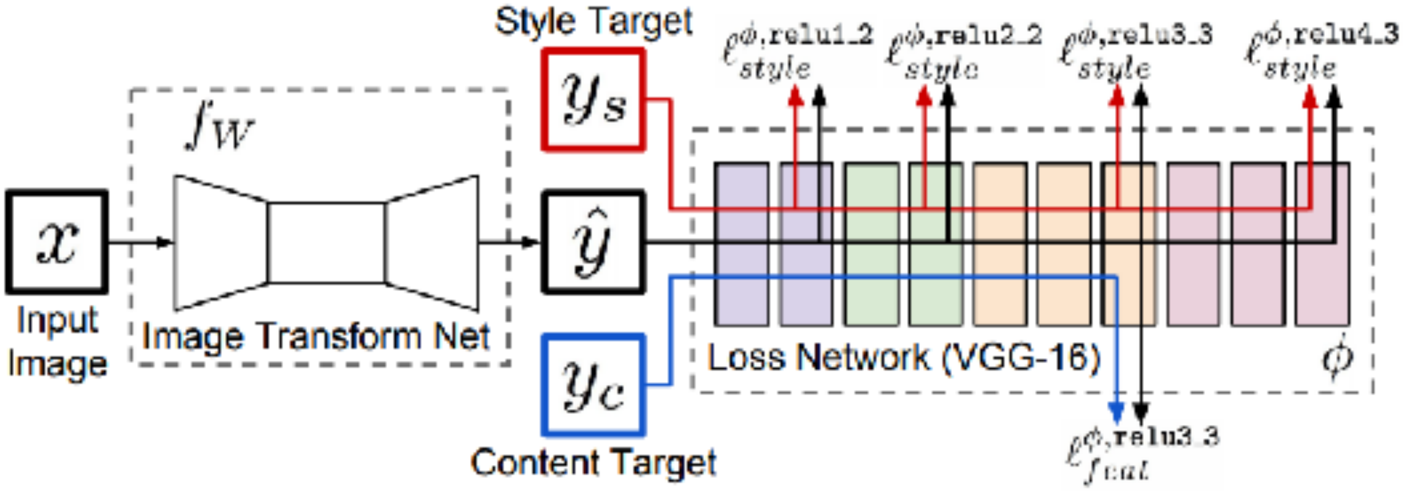
Abstract. We consider image transformation problems, where an input image is transformed into an output image. Recent methods for such

One other thought!

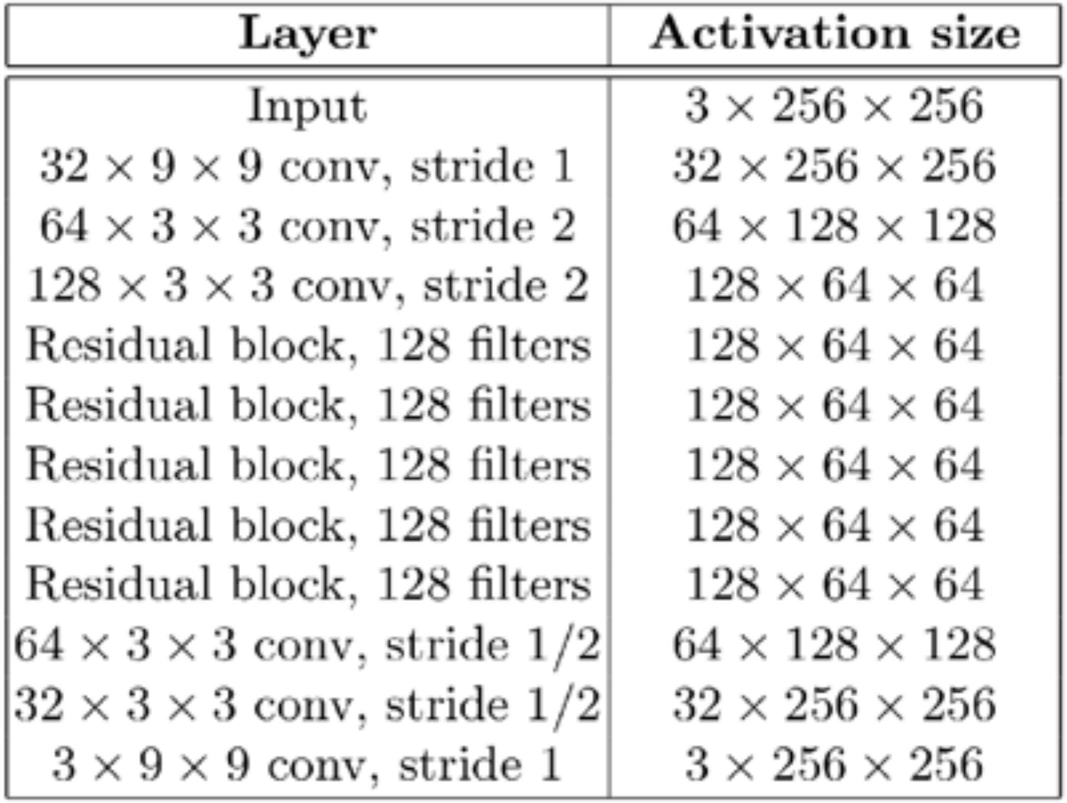
- I hate the term **perceptual loss**!
- It's marketing! Joy!
- …but not good science verbiage
- …like the term global warming
  - it can introduce subjective opinion, misunderstanding through misleading labeling
- There is nothing perceptual here, its a neural network
- A better description of the loss:
  - Convolutional Gram Loss?
  - Information Distillation Covariance?
  - Weighted Grammian Norm (WGN Loss)?
- Just don’t say “perceptual” in this class (so you don’t fail)

Johnson Paper Recap (if needed)

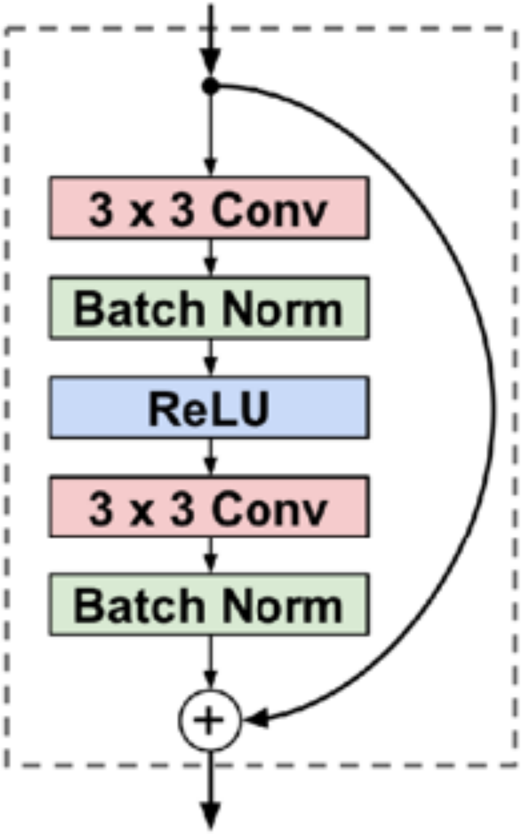
- **Basic Idea**: replace image optimization with single pass, fully convolutional network to perform the transformation
- Loss functions stay identical to Gatys
  - ○ “Content” through activations difference
  - ○ “Style” through Grammian differences
- Instances are normalized along the channel input
- No bias in filters

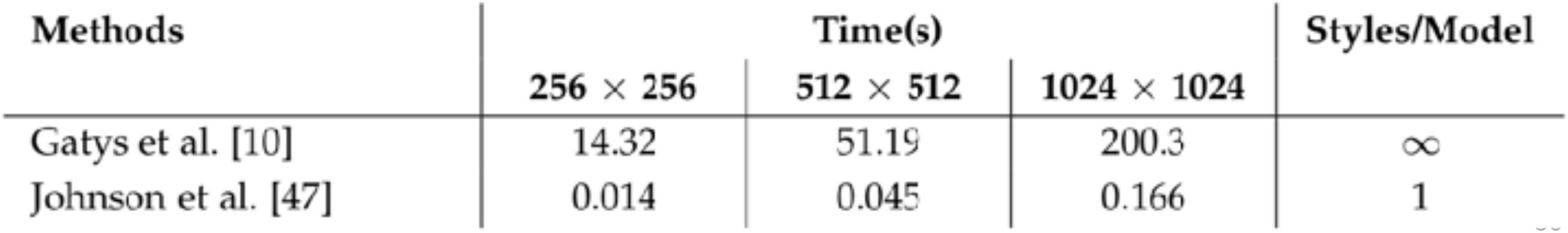

Style Target
y_s
$\ell_{style}^{\phi,relu1\_2}$
$\ell_{style}^{\phi,relu2\_2}$
$\ell_{style}^{\phi,relu3\_3}$
$\ell_{style}^{\phi,relu4\_3}$
f_W
x
Input Image
Image Transform Net
$\hat{y}$
y_c
Content Target
Loss Network (VGG-16)
ϕ
$\ell_{feat}^{\phi,relu3\_3}$

Specifics of the Architecture



Layer	Activation size
Input	$3 \times 256 \times 256$
$32 \times 9 \times 9$ conv, stride 1	$32 \times 256 \times 256$
$64 \times 3 \times 3$ conv, stride 2	$64 \times 128 \times 128$
$128 \times 3 \times 3$ conv, stride 2	$128 \times 64 \times 64$
Residual block, 128 filters	$128 \times 64 \times 64$
Residual block, 128 filters	$128 \times 64 \times 64$
Residual block, 128 filters	$128 \times 64 \times 64$
Residual block, 128 filters	$128 \times 64 \times 64$
Residual block, 128 filters	$128 \times 64 \times 64$
$64 \times 3 \times 3$ conv, stride 1/2	$64 \times 128 \times 128$
$32 \times 3 \times 3$ conv, stride 1/2	$32 \times 256 \times 256$
$3 \times 9 \times 9$ conv, stride 1	$3 \times 256 \times 256$





Methods	Time(s)			Styles/Model
	256×256	512×512	1024×1024	
Gatys et al. [10]	14.32	51.19	200.3	∞
Johnson et al. [47]	0.014	0.045	0.166	1

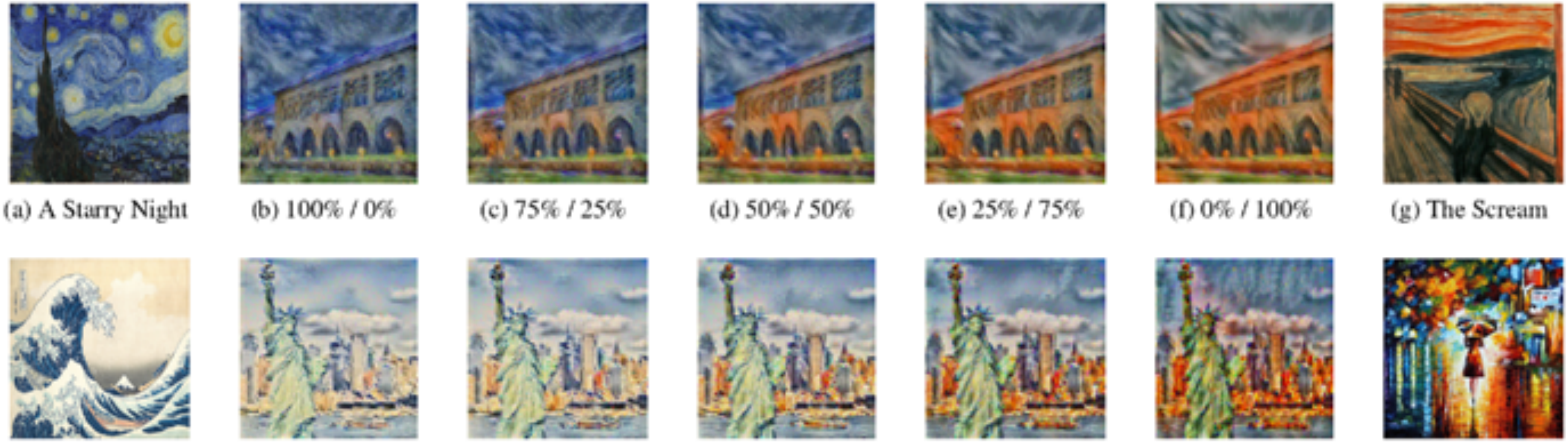
Johnson Paper Extensions

- Multi-style transfer:

Set of Style Images

For each Style Image

$$\mathcal{L}_s(I_{s0}, \ldots, I_{sN}, I_{new}) = \sum_{i=0}^{N} \beta_i \sum_{l \in L_s} \|G_{si}^{(l)} - G_{new}^{(l)}\|^2$$



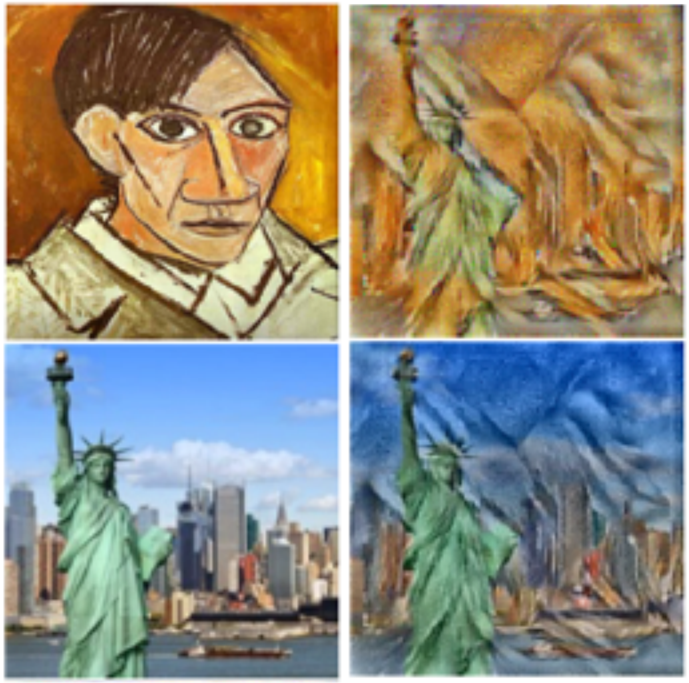
(a) A Starry Night (b) 100% / 0% (c) 75% / 25% (d) 50% / 50% (e) 25% / 75% (f) 0% / 100% (g) The Scream

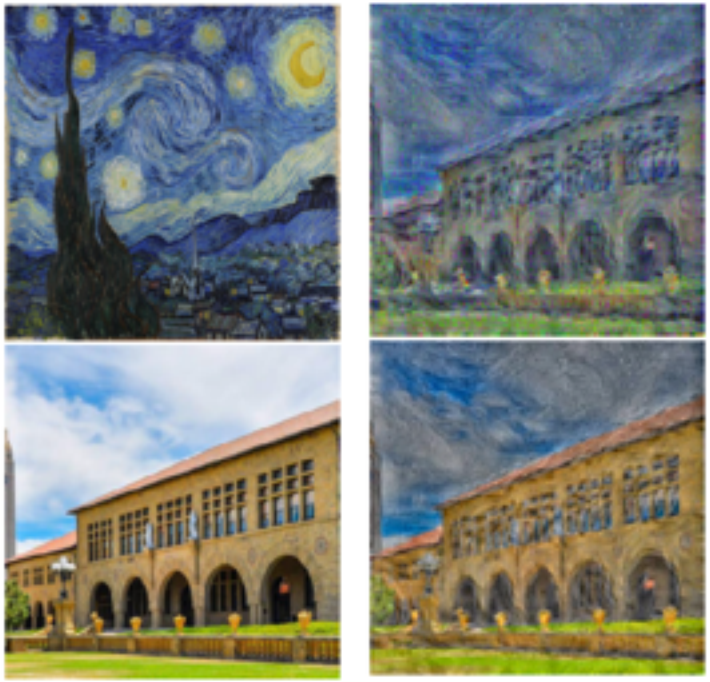
(h) The Great Wave (i) 100% / 0% (j) 75% / 25% (k) 50% / 50% (l) 25% / 75% (m) 0% / 100% (n) Rain Princess

Final projects from: http://cs231n.stanford.edu/reports/2017/pdfs/428.pdf

Fast Style Paper Extensions

- Color Preservation:
  - Apply style to luminance only
  - Reapply color channels from original image (blurred)
  - $HS\{V_{\text{transform}}\}$



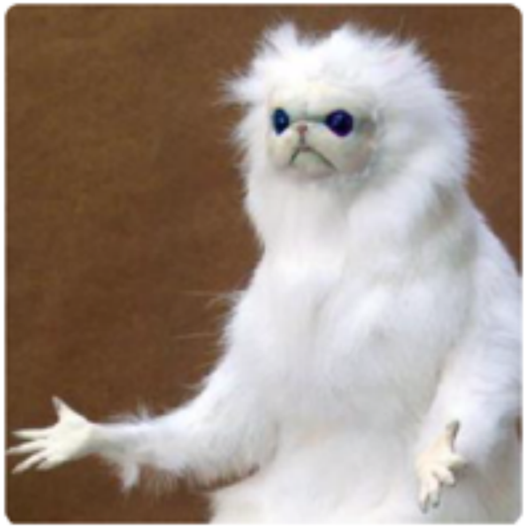


Final projects from: http://cs231n.stanford.edu/reports/2017/pdfs/428.pdf

Model Based Optimization Style Transfer

Johnson, *et. al* Fast Style Transfer

Jake Carlson

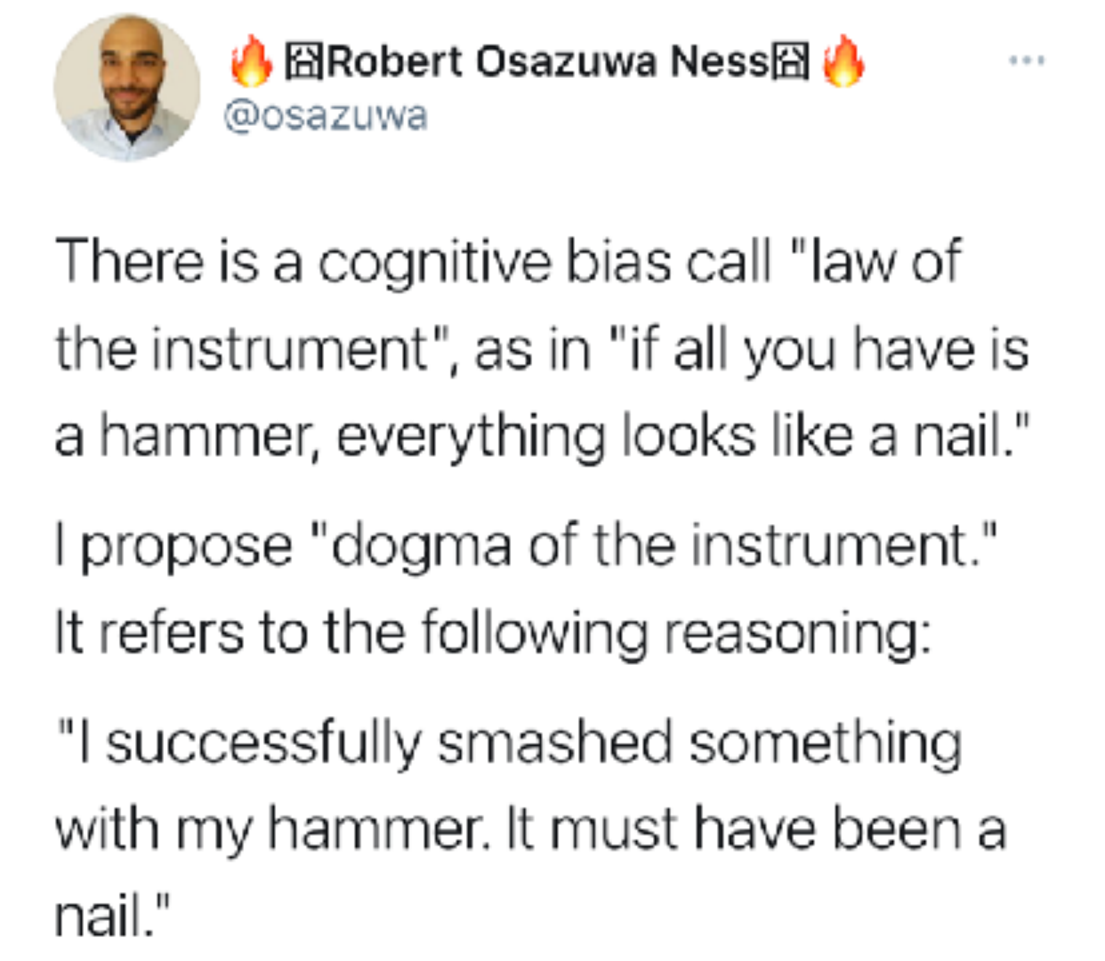


Justin Ledford

Luke Wood

Follow Along: LectureNotesMaster/03 LectureStyleTransfer.ipynb
Training: https://github.com/8000net/StyleTransfer/blob/master/Style_Transfer_Training.ipynb

One Shot Transfer



🔥囧Robert Osazuwa Ness囧🔥
@osazuwa

There is a cognitive bias call "law of the instrument", as in "if all you have is a hammer, everything looks like a nail."

I propose "dogma of the instrument." It refers to the following reasoning:

"I successfully smashed something with my hammer. It must have been a nail."

The Problem of Training

- One network is only capable of one style transformation
  - Great for trained filters in an app
  - Does not work for generic transfer of one style to another
- How to define a transformation on a content and style image that transfers style into the content?
  - …without any “per style” training
  - …but still using the “grammian style losses”
  - …and allows for infinite customization?
- We need to first cover **whitening** and **coloring** transformations

Aside: Whitening and Coloring

- Signal processing terms applied typically to spectra of signals
- **Whitening** is the process of removing correlation in a signal
  - For a matrix, the whitening transform:
    - makes covariance nearly diagonal (identity)
    - using only linear operations
- **Coloring** is the process of adding the observed correlation back into a signal
  - For a matrix, the coloring transform
    - makes the covariance of signal A as close as possible to signal B (the target)
    - with only linear operations to A

Aside: Whitening with SVD

- Singular Value Decomposition (SVD) decomposes a matrix into three elements
  - $A = U\Sigma V^T$
  - U is eig-vec of AA^T and V is eig-vec of A^TA
  - where U and V are orthogonal matrices such that

    $UU^T=I$ and $VV^T=I$
  - Σ is a diagonal matrix of the singular values
- the matrix $UV^T=A_w$ is a whitened version of the signal A such that

  $A_w A_w^T=I$
- since the Gram of a layer activation is $A A^T=G$, the whitened signal has $A_w A_w^T = I = G$
  - which would have “no style” according to Gatys

Coloring with SVD

- Suppose we have two activations
  - $A_c = U_c \Sigma_c V_c^T$ and $G_c = A_c A_c^T$
  - $A_s = U_s \Sigma_s V_s^T$ and $G_s = A_s A_s^T$
- We can transfer the Gram matrix of A_s to A_c using coloring
  - To color the matrix:

$$A_{new} = U_s \Sigma_s V_c^T$$

$$A_{new}\, A_{new}^T = G_s$$

  - But A_{new} is more similar to A_c than A_s
- That is exactly what we want for style transfer!
- *Can also achieve coloring and whitening through the Eigen decomposition, but is less numerically stable*
  - *(but can be faster)*

Whitening and Coloring

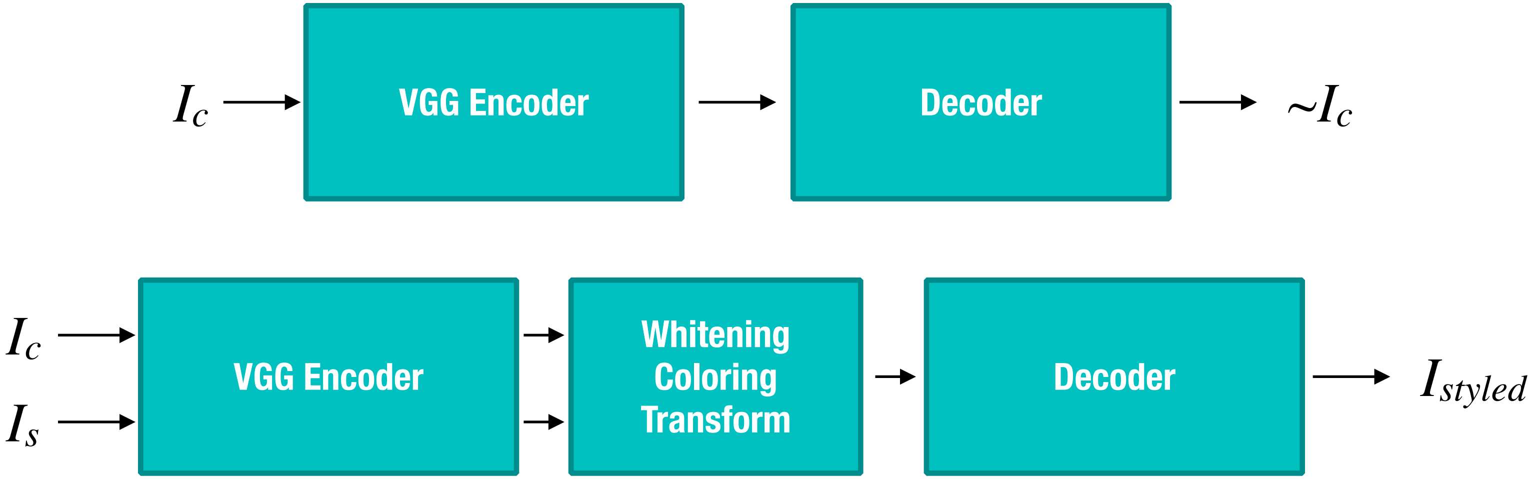
What does this all mean?

Developing an intuition…

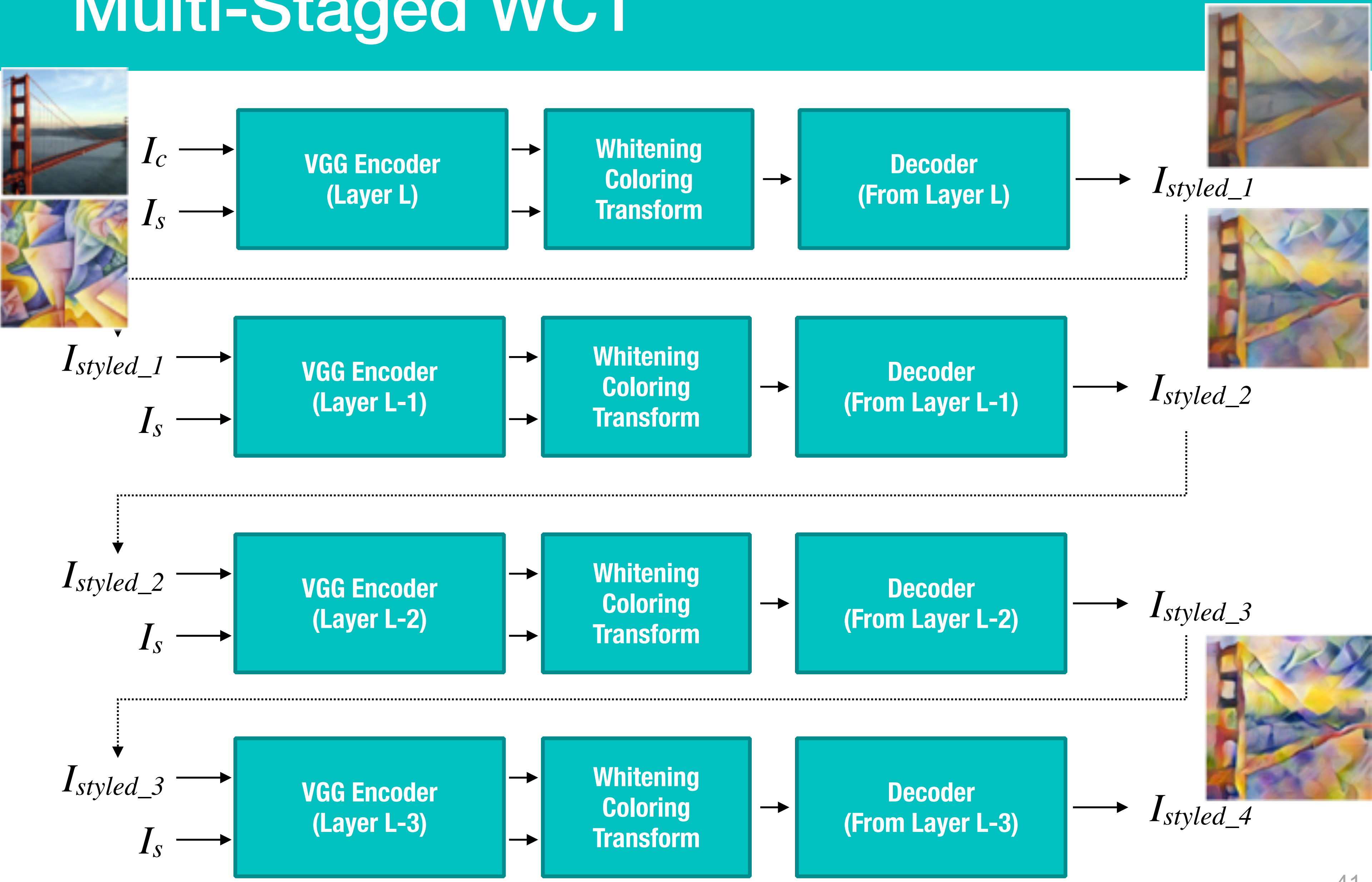
```
Follow Along:
03b WhiteningColoringExample.ipynb
```

How to use this for style transfer?

- If we can learn to **reconstruct an image from VGG**…
  - …we can **whiten content activations**
  - …then **color with desired style Grammian**
- The resulting reconstruction should have largely the same content, but style from the colored activations
- …One transformation network for any style!



Multi-Staged WCT



I_c → I_s → VGG Encoder (Layer L) → Whitening Coloring Transform → Decoder (From Layer L) → I_{styled_1}

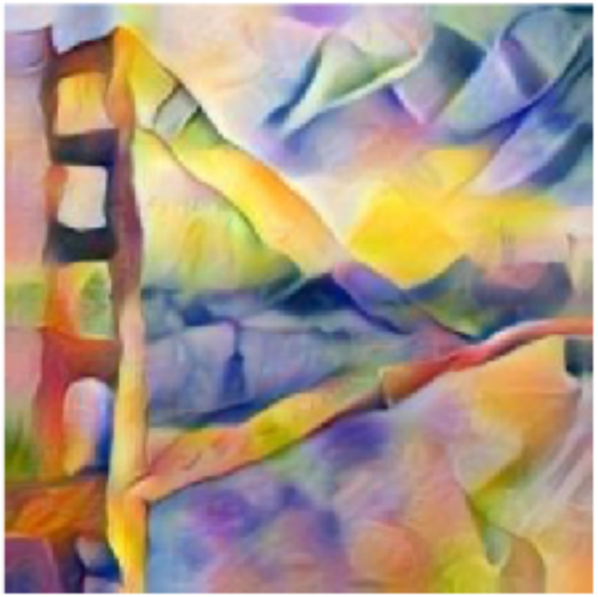
I_{styled_1} → I_s → VGG Encoder (Layer L-1) → Whitening Coloring Transform → Decoder (From Layer L-1) → I_{styled_2}

I_{styled_2} → I_s → VGG Encoder (Layer L-2) → Whitening Coloring Transform → Decoder (From Layer L-2) → I_{styled_3}

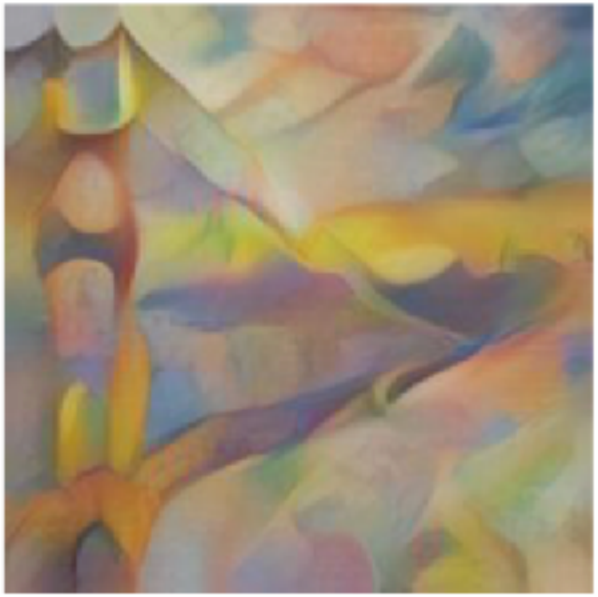
I_{styled_3} → I_s → VGG Encoder (Layer L-3) → Whitening Coloring Transform → Decoder (From Layer L-3) → I_{styled_4}

Why not go the other way?

- Start at earlier layers and apply WCT as we progress through the network
- Paper does not have good explanation, but results are subjectively poorer:

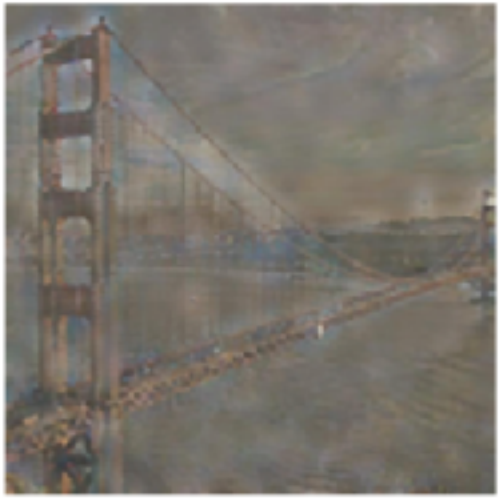


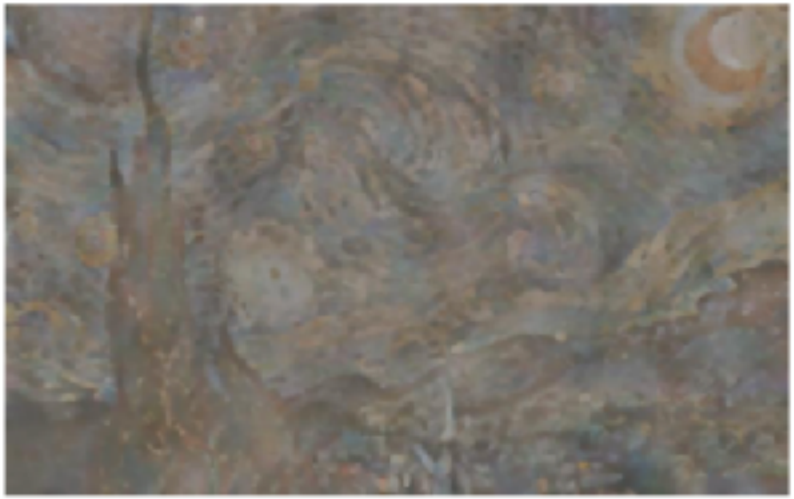
L > L-1 > L-2 > L-3



L-3 > L-2 > L-1 > L

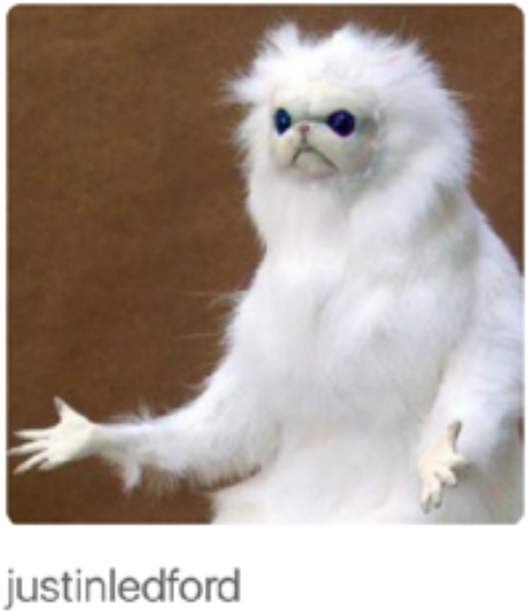
Removing Style? Only Whitening





One Shot Style Transfer

Li, *et. al* Universal Style Transfer



Follow Along: https://github.com/8000net/universal-style-transfer-keras

Lecture Notes for

Neural Networks and Machine Learning

Style Transfer: Model Opt.

Next Time:
Photo Realistic WCT

Reading: None