# Lecture 12: Visualizing and Understanding

Fei-Fei Li & Justin Johnson & Serena Yeung

Lecture 12 - 1 May 16, 2017

### Administrative

Milestones due tonight on Canvas, 11:59pm

Midterm grades released on Gradescope this week

A3 due next Friday, 5/26

HyperQuest deadline extended to Sunday 5/21, 11:59pm

Poster session is June 6

Fei-Fei Li & Justin Johnson & Serena Yeung

Lecture 11 - 2 May 10, 2017

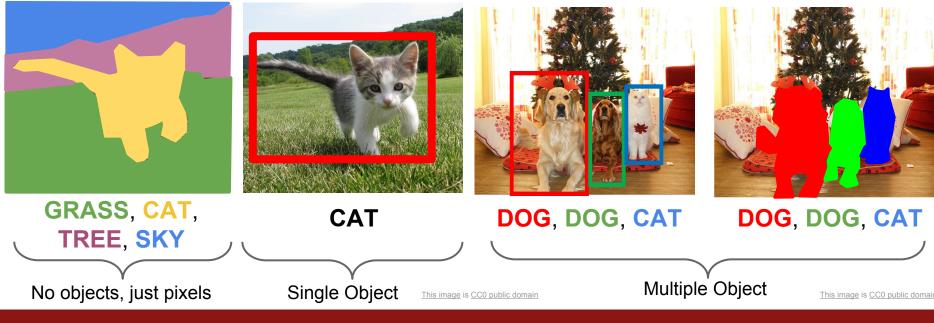
### Last Time: Lots of Computer Vision Tasks

Semantic Segmentation

### Classification + Localization

### Object Detection

#### Instance Segmentation



Fei-Fei Li & Justin Johnson & Serena Yeung

Lecture 11 - 3 May 10, 2017

### What's going on inside ConvNets?

This image is CC0 public domain



Input Image: 3 x 224 x 224

dense 192 128 2048 2048 192 48 128 224 dense densé 1000 192 192 128 Max 2048 2048 pooling Max 128 Max pooling pooling What are the intermediate features looking for?

Class Scores: 1000 numbers

Krizhevsky et al, "ImageNet Classification with Deep Convolutional Neural Networks", NIPS 2012. Figure reproduced with permission.

Fei-Fei Li & Justin Johnson & Serena Yeung

Lecture 11 - 4 May 10, 2017

### First Layer: Visualize Filters



Max Max pooling Ma

Krizhevsky, "One weird trick for parallelizing convolutional neural networks", arXiv 2014 He et al, "Deep Residual Learning for Image Recognition", CVPR 2016 Huang et al, "Densely Connected Convolutional Networks", CVPR 2017

64 x 3 x 11 x 11

Fei-Fei Li & Justin Johnson & Serena Yeung

Lecture 11 - 5 May 10, 2017

Visualize the filters/kernels (raw weights)

We can visualize filters at higher layers, but not that interesting

(these are taken from ConvNetJS CIFAR-10 demo) Weights:

#### 

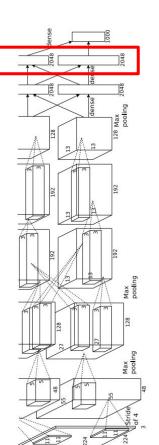
Fei-Fei Li & Justin Johnson & Serena Yeung

Lecture 11 - 6 May 10, 2017

layer 1 weights

### Last Layer

FC7 layer



4096-dimensional feature vector for an image (layer immediately before the classifier)

Run the network on many images, collect the feature vectors

#### Fei-Fei Li & Justin Johnson & Serena Yeung

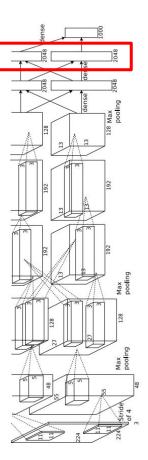
Lecture 11 - 7 May 1<u>0, 2017</u>

# Last Layer: Nearest Neighbors

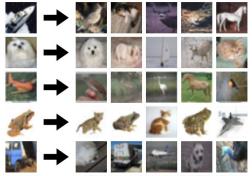
4096-dim vector

Test image L2 Nearest neighbors in feature space





**Recall**: Nearest neighbors in <u>pixel</u> space



Krizhevsky et al, "ImageNet Classification with Deep Convolutional Neural Networks", NIPS 2012. Figures reproduced with permission.

#### Fei-Fei Li & Justin Johnson & Serena Yeung

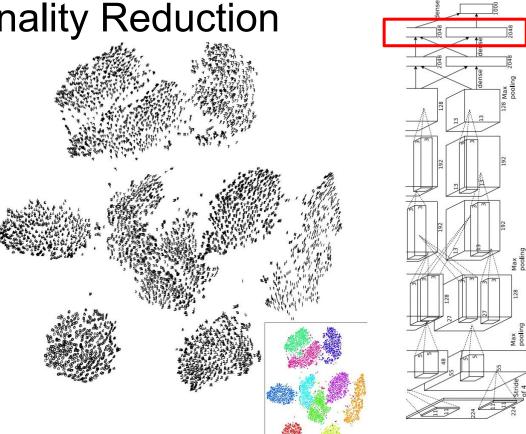
Lecture 11 - 8 May 10, 2017

### Last Layer: Dimensionality Reduction

Visualize the "space" of FC7 feature vectors by reducing dimensionality of vectors from 4096 to 2 dimensions

Simple algorithm: Principle Component Analysis (PCA)

More complex: t-SNE



Van der Maaten and Hinton, "Visualizing Data using t-SNE", JMLR 2008 Figure copyright Laurens van der Maaten and Geoff Hinton, 2008. Reproduced with permission.

Fei-Fei Li & Justin Johnson & Serena Yeung

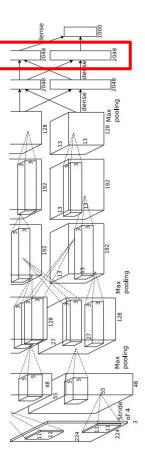
Lecture 11 - 9 May 10, 2017

### Last Layer: Dimensionality Reduction



Van der Maaten and Hinton, "Visualizing Data using t-SNE", JMLR 2008 Krizhevsky et al, "ImageNet Classification with Deep Convolutional Neural Networks", NIPS 2012. Figure reproduced with permission.





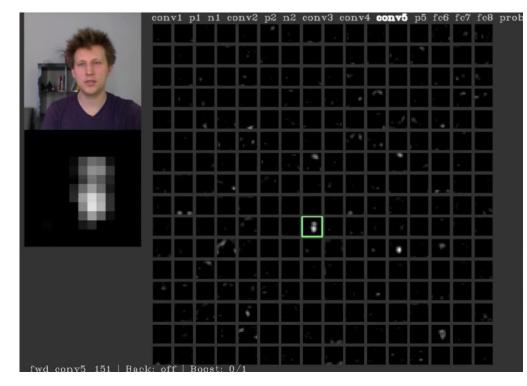
See high-resolution versions at <a href="http://cs.stanford.edu/people/karpathy/cnnembed/">http://cs.stanford.edu/people/karpathy/cnnembed/</a>

#### Fei-Fei Li & Justin Johnson & Serena Yeung

Lecture 11 - 10 May 10, 2017

# **Visualizing Activations**

conv5 feature map is 128x13x13; visualize as 128 13x13 grayscale images



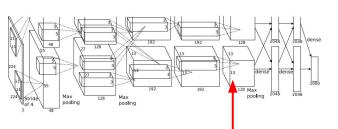
Yosinski et al, "Understanding Neural Networks Through Deep Visualization", ICML DL Workshop 2014. Figure copyright Jason Yosinski, 2014. Reproduced with permission.

Fei-Fei Li & Justin Johnson & Serena Yeung

Lecture 11 - 11 May 10, 2017

# **Maximally Activating Patches**

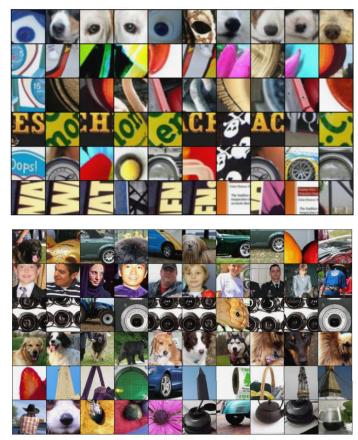




Pick a layer and a channel; e.g. conv5 is 128 x 13 x 13, pick channel 17/128

Run many images through the network, record values of chosen channel

Visualize image patches that correspond to maximal activations



Springenberg et al, "Striving for Simplicity: The All Convolutional Net", ICLR Workshop 2015 Figure copyright Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, Martin Riedmiller, 2015; reproduced with permission.

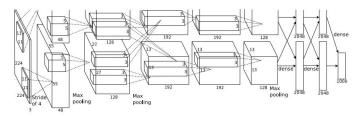
#### Fei-Fei Li & Justin Johnson & Serena Yeung

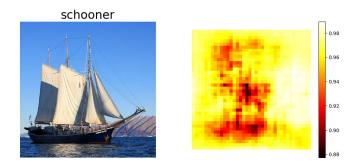
#### Lecture 11 - 12 May 10, 2017

# **Occlusion Experiments**

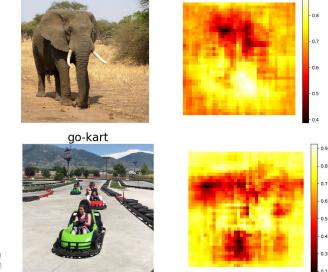
Mask part of the image before feeding to CNN, draw heatmap of probability at each mask location







African elephant, Loxodonta africana



Zeiler and Fergus, "Visualizing and Understanding Convolutional Networks", ECCV 2014

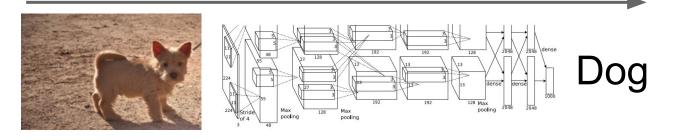
Boat image is CC0 public domain Elephant image is CC0 public domain Go-Karts image is CC0 public domain

#### Fei-Fei Li & Justin Johnson & Serena Yeung

#### Lecture 11 - 13 May 10, 2017

### Saliency Maps

#### How to tell which pixels matter for classification?



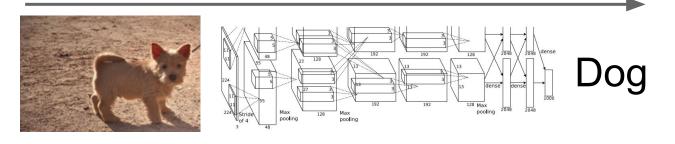
Simonyan, Vedaldi, and Zisserman, "Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps", ICLR Workshop 2014. Figures copyright Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman, 2014; reproduced with permission.

#### Fei-Fei Li & Justin Johnson & Serena Yeung

#### Lecture 11 - 14 May 10, 2017

# Saliency Maps

#### How to tell which pixels matter for classification?

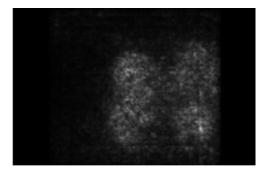


Compute gradient of (unnormalized) class score with respect to image pixels, take absolute value and max over RGB channels

Simonyan, Vedaldi, and Zisserman, "Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps", ICLR Workshop 2014.

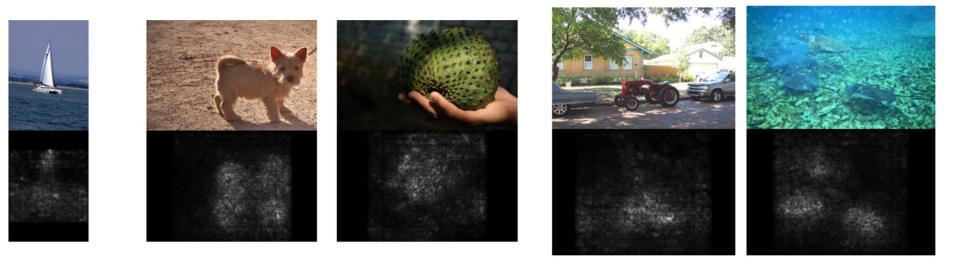
Figures copyright Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman, 2014; reproduced with permission.

#### Fei-Fei Li & Justin Johnson & Serena Yeung



#### Lecture 11 - 15 May 10, 2017

### Saliency Maps



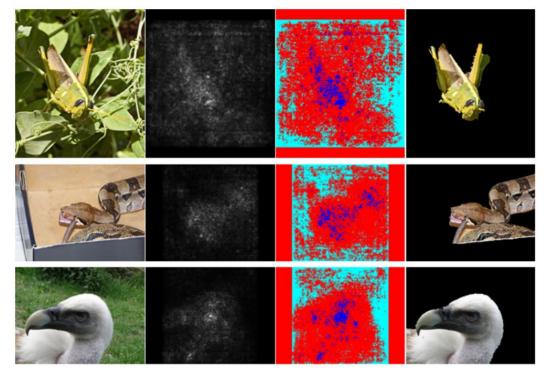
Simonyan, Vedaldi, and Zisserman, "Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps", ICLR Workshop 2014.

Figures copyright Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman, 2014; reproduced with permission.

#### Fei-Fei Li & Justin Johnson & Serena Yeung

#### Lecture 11 - 16 May 10, 2017

### Saliency Maps: Segmentation without supervision



Use GrabCut on saliency map

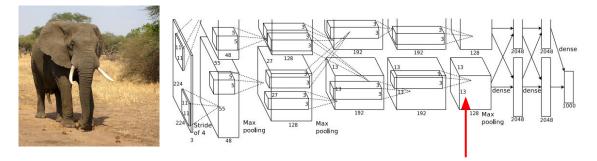
Simonyan, Vedaldi, and Zisserman, "Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps", ICLR Workshop 2014.

Figures copyright Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman, 2014; reproduced with permission. Rother et al, "Grabcut: Interactive foreground extraction using iterated graph cuts", ACM TOG 2004

Fei-Fei Li & Justin Johnson & Serena Yeung

Lecture 11 - 17 May 10, 2017

### Intermediate Features via (guided) backprop



Pick a single intermediate neuron, e.g. one value in 128 x 13 x 13 conv5 feature map

Compute gradient of neuron value with respect to image pixels

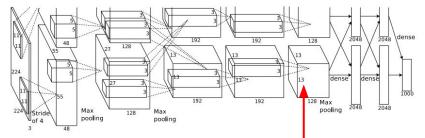
Zeiler and Fergus, "Visualizing and Understanding Convolutional Networks", ECCV 2014 Springenberg et al, "Striving for Simplicity: The All Convolutional Net", ICLR Workshop 2015

Fei-Fei Li & Justin Johnson & Serena Yeung

Lecture 11 - 18 May 10, 2017

### Intermediate features via (guided) backprop





Pick a single intermediate neuron, e.g. one value in 128 x 13 x 13 conv5 feature map

Compute gradient of neuron value with respect to image pixels

 $\begin{array}{c|cccc} ReLU \\ \hline 1 & -1 & 5 \\ \hline 2 & -5 & -7 \\ \hline -3 & 2 & 4 \end{array} \begin{array}{c} 1 & 0 \\ \hline 2 & 0 \\ \hline 0 & 2 \end{array}$ 

De clussend access	-2	0	-1		
Backward pass: backpropagation	6	0	0	-	
a a confer e fra Garriero					

0 -1 3

Backward pass:<br/>"deconvnet"030601203Backward pass:<br/>guided<br/>backpropagation0000003

Forward pass

	2	-1	3				
	-2	3	-1				
->	6	-3	1				
	2	-1	3				
	-2	3	-1				
←	6	-3	1				
	2	-1	3				

0

Images come out nicer if you only backprop positive gradients through each ReLU (guided backprop)

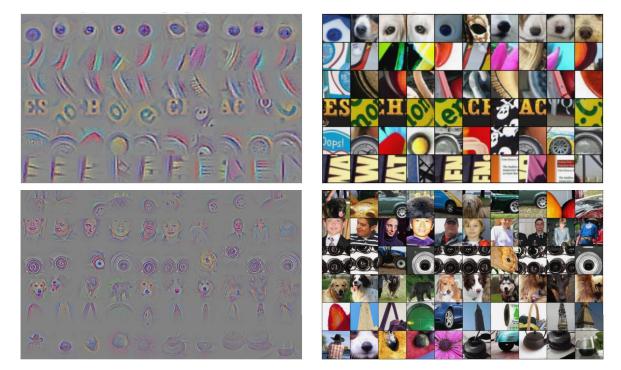
Figure copyright Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, Martin Riedmiller, 2015; reproduced with permission.

Zeiler and Fergus, "Visualizing and Understanding Convolutional Networks", ECCV 2014 Springenberg et al, "Striving for Simplicity: The All Convolutional Net", ICLR Workshop 2015

Fei-Fei Li & Justin Johnson & Serena Yeung

#### Lecture 11 - 19 May 10, 2017

### Intermediate features via (guided) backprop



Zeiler and Fergus, "Visualizing and Understanding Convolutional Networks", ECCV 2014 Springenberg et al, "Striving for Simplicity: The All Convolutional Net", ICLR Workshop 2015 Figure copyright Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, Martin Riedmiller, 2015; reproduced with permission.

Fei-Fei Li & Justin Johnson & Serena Yeung

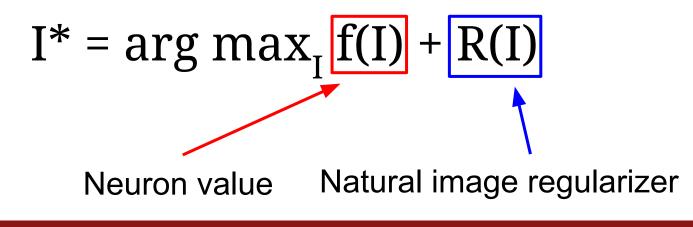
Lecture 11 - 20 May 10, 2017

#### (Guided) backprop:

Find the part of an image that a neuron responds to

#### Gradient ascent:

Generate a synthetic image that maximally activates a neuron



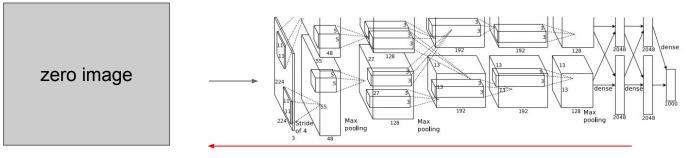
Fei-Fei Li & Justin Johnson & Serena Yeung

Lecture 11 - 21 May 10, 2017

1. Initialize image to zeros

$$\arg\max_{I} S_c(I) - \lambda \|I\|_2^2$$

score for class c (before Softmax)



Repeat:

- 2. Forward image to compute current scores
- 3. Backprop to get gradient of neuron value with respect to image pixels
- 4. Make a small update to the image

Fei-Fei Li & Justin Johnson & Serena Yeung

Lecture 11 - 22 May 10, 2017

$$\arg\max_{I} S_c(I) - \lambda \|I\|_2^2$$

Simple regularizer: Penalize L2 norm of generated image

Simonyan, Vedaldi, and Zisserman, "Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps", ICLR Workshop 2014. Figures copyright Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman, 2014; reproduced with permission.

#### Fei-Fei Li & Justin Johnson & Serena Yeung

Lecture 11 - 23 May 10, 2017

$$\arg\max_{I} S_c(I) - \lambda \|I\|_2^2$$

Simple regularizer: Penalize L2 norm of generated image



dumbbell

cup



bell pepper

lemon

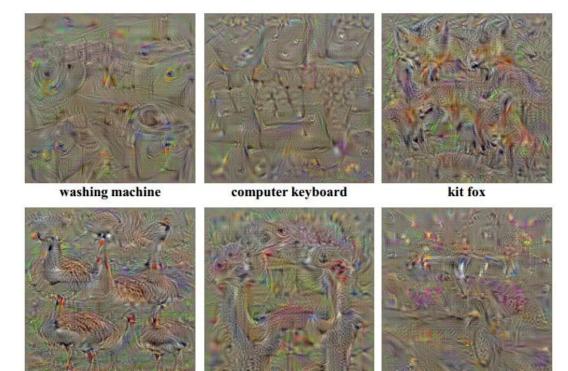
Simonyan, Vedaldi, and Zisserman, "Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps", ICLR Workshop 2014. Figures copyright Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman, 2014; reproduced with permission.

#### Fei-Fei Li & Justin Johnson & Serena Yeung

#### Lecture 11 - 24 May 10, 2017

$$\arg\max_{I} S_c(I) - \lambda \|I\|_2^2$$

Simple regularizer: Penalize L2 norm of generated image



ostrich

goose

Yosinski et al, "Understanding Neural Networks Through Deep Visualization", ICML DL Workshop 2014. Figure copyright Jason Yosinski, Jeff Clune, Anh Nguyen, Thomas Fuchs, and Hod Lipson, 2014. Reproduced with permission.

Fei-Fei Li & Justin Johnson & Serena Yeung

#### Lecture 11 - 25 May 10, 2017

limousine

$$\arg\max_{I} S_c(I) - \lambda \|I\|_2^2$$

Better regularizer: Penalize L2 norm of image; also during optimization periodically

- (1) Gaussian blur image
- (2) Clip pixels with small values to 0
- (3) Clip pixels with small gradients to 0

Yosinski et al, "Understanding Neural Networks Through Deep Visualization", ICML DL Workshop 2014.

#### Fei-Fei Li & Justin Johnson & Serena Yeung

Lecture 11 - 26 May 10, 2017

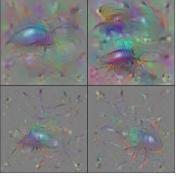
$$\arg\max_{I} S_c(I) - \lambda \|I\|_2^2$$

Better regularizer: Penalize L2 norm of image; also during optimization periodically

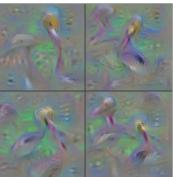
- (1) Gaussian blur image
- (2) Clip pixels with small values to 0
- (3) Clip pixels with small gradients to 0



Flamingo







Pelican



Yosinski et al, "Understanding Neural Networks Through Deep Visualization", ICML DL Workshop 2014. Figure copyright Jason Yosinski, Jeff Clune, Anh Nguyen, Thomas Fuchs, and Hod Lipson, 2014. Reproduced with permission. Indian Cobra

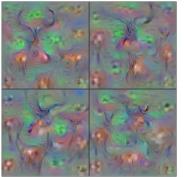
Fei-Fei Li & Justin Johnson & Serena Yeung

Lecture 11 - 27 May 10, 2017

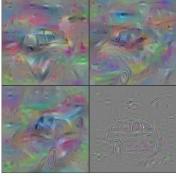
$$\arg\max_{I} S_c(I) - \lambda \|I\|_2^2$$

Better regularizer: Penalize L2 norm of image; also during optimization periodically

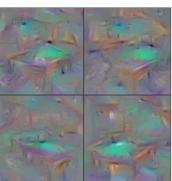
- (1) Gaussian blur image
- (2) Clip pixels with small values to 0
- (3) Clip pixels with small gradients to 0



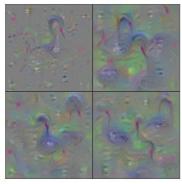
Hartebeest



Station Wagon



**Billiard Table** 



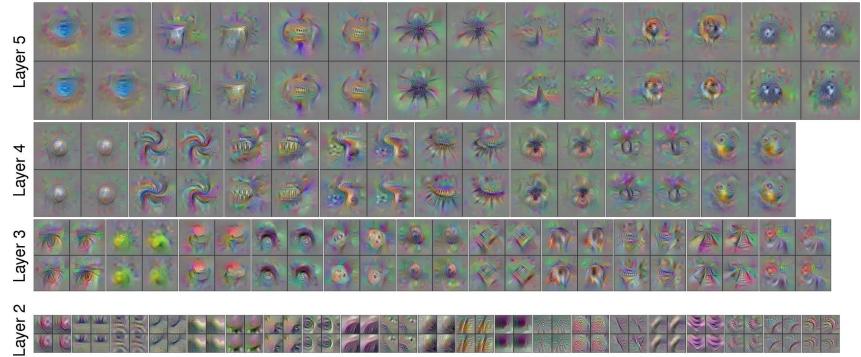
Black Swan

Yosinski et al, "Understanding Neural Networks Through Deep Visualization", ICML DL Workshop 2014. Figure copyright Jason Yosinski, Jeff Clune, Anh Nguyen, Thomas Fuchs, and Hod Lipson, 2014. Reproduced with permission.

Fei-Fei Li & Justin Johnson & Serena Yeung

Lecture 11 - 28 May 10, 2017

Use the same approach to visualize intermediate features

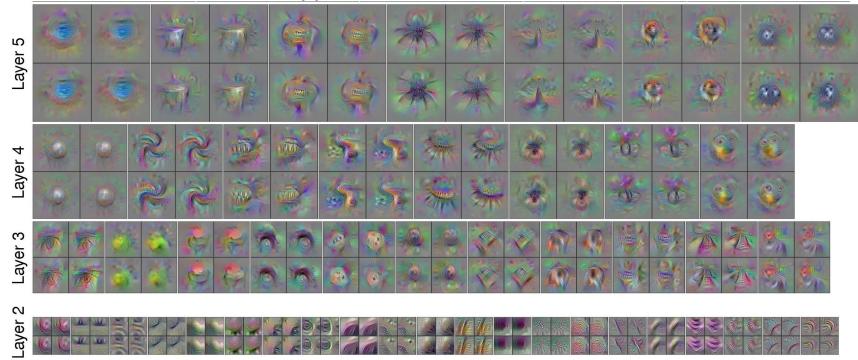


Yosinski et al, "Understanding Neural Networks Through Deep Visualization", ICML DL Workshop 2014. Figure copyright Jason Yosinski, Jeff Clune, Anh Nguyen, Thomas Fuchs, and Hod Lipson, 2014. Reproduced with permission.

Fei-Fei Li & Justin Johnson & Serena Yeung

Lecture 11 - 29 May 10, 2017

Use the same approach to visualize intermediate features



Yosinski et al, "Understanding Neural Networks Through Deep Visualization", ICML DL Workshop 2014. Figure copyright Jason Yosinski, Jeff Clune, Anh Nguyen, Thomas Fuchs, and Hod Lipson, 2014. Reproduced with permission.

Fei-Fei Li & Justin Johnson & Serena Yeung

Lecture 11 - 30 May 10, 2017

Adding "multi-faceted" visualization gives even nicer results: (Plus more careful regularization, center-bias)

Reconstructions of multiple feature types (facets) recognized by the same "grocery store" neuron



Corresponding example training set images recognized by the same neuron as in the "grocery store" class



Nguyen et al, "Multifaceted Feature Visualization: Uncovering the Different Types of Features Learned By Each Neuron in Deep Neural Networks", ICML Visualization for Deep Learning Workshop 2016. Figures copyright Anh Nguyen, Jason Yosinski, and Jeff Clune, 2016; reproduced with permission.

#### Fei-Fei Li & Justin Johnson & Serena Yeung

Lecture 11 - 31 May 10, 2017



Nguyen et al, "Multifaceted Feature Visualization: Uncovering the Different Types of Features Learned By Each Neuron in Deep Neural Networks", ICML Visualization for Deep Learning Workshop 2016. Figures copyright Anh Nguyen, Jason Yosinski, and Jeff Clune, 2016; reproduced with permission.

#### Fei-Fei Li & Justin Johnson & Serena Yeung

#### Lecture 11 - 32 May 10, 2017

Optimize in FC6 latent space instead of pixel space:



Nguyen et al, "Synthesizing the preferred inputs for neurons in neural networks via deep generator networks," NIPS 2016 Figure copyright Nguyen et al, 2016; reproduced with permission.

Fei-Fei Li & Justin Johnson & Serena Yeung

#### Lecture 11 - 33 May 10, 2017

### Fooling Images / Adversarial Examples

- (1) Start from an arbitrary image
- (2) Pick an arbitrary class
- (3) Modify the image to maximize the class
- (4) Repeat until network is fooled

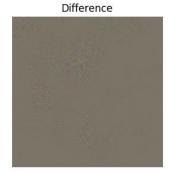
### Fooling Images / Adversarial Examples

#### African elephant

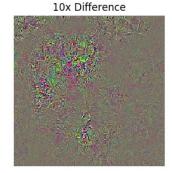




iPod



Difference

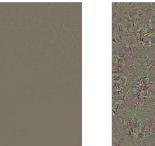


schooner

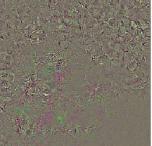








10x Difference



Boat image is CC0 public domain Elephant image is CC0 public domain

#### Fei-Fei Li & Justin Johnson & Serena Yeung

#### Lecture 11 - 35 May 10, 2017

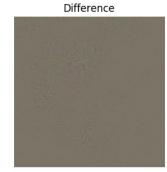
### Fooling Images / Adversarial Examples

#### African elephant

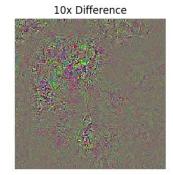




iPod



Difference



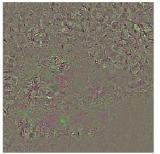
schooner







10x Difference



Boat image is CC0 public domain Elephant image is CC0 public domain

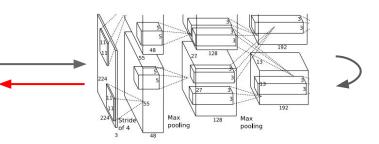
#### What is going on? Ian Goodfellow will explain

Fei-Fei Li & Justin Johnson & Serena Yeung

Lecture 11 - 36 May 10, 2017

Rather than synthesizing an image to maximize a specific neuron, instead try to **amplify** the neuron activations at some layer in the network





Choose an image and a layer in a CNN; repeat:

- 1. Forward: compute activations at chosen layer
- 2. Set gradient of chosen layer equal to its activation
- 3. Backward: Compute gradient on image
- 4. Update image

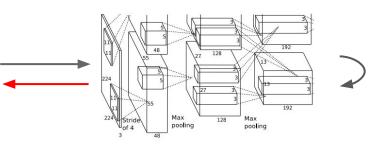
Mordvintsev, Olah, and Tyka, "Inceptionism: Going Deeper into Neural Networks", <u>Google Research Blog</u>. Images are licensed under <u>CC-BY</u>

Lecture 11 - 37 May 10, 2017

### Fei-Fei Li & Justin Johnson & Serena Yeung

Rather than synthesizing an image to maximize a specific neuron, instead try to **amplify** the neuron activations at some layer in the network





Choose an image and a layer in a CNN; repeat:

- 1. Forward: compute activations at chosen layer
- 2. Set gradient of chosen layer equal to its activation
- 3. Backward: Compute gradient on image
- 4. Update image

```
Equivalent to:
____ I* = arg max<sub>I</sub> \sum_{i} f_{i}(I)^{2}
```

Mordvintsev, Olah, and Tyka, "Inceptionism: Going Deeper into Neural Networks", <u>Google Research Blog</u>. Images are licensed under <u>CC-BY</u>

### Fei-Fei Li & Justin Johnson & Serena Yeung

Lecture 11 - 38 May 10, 2017

```
def objective L2(dst):
    dst.diff[:] = dst.data
def make step(net, step size=1.5, end='inception 4c/output',
              jitter=32, clip=True, objective=objective L2):
    '''Basic gradient ascent step.'''
    src = net.blobs['data'] # input image is stored in Net's 'data' blob
    dst = net.blobs[end]
   ox, oy = np.random.randint(-jitter, jitter+1, 2)
    src.data[0] = np.roll(np.roll(src.data[0], ox, -1), oy, -2) # apply jitter shift
   net.forward(end=end)
    objective(dst) # specify the optimization objective
   net.backward(start=end)
    g = src.diff[0]
    # apply normalized ascent step to the input image
    src.data[:] += step size/np.abs(g).mean() * g
    src.data[0] = np.roll(np.roll(src.data[0], -ox, -1), -oy, -2) # unshift image
    if clip:
        bias = net.transformer.mean['data']
        src.data[:] = np.clip(src.data, -bias, 255-bias)
```

# <u>Code</u> is very simple but it uses a couple tricks:

(Code is licensed under Apache 2.0)

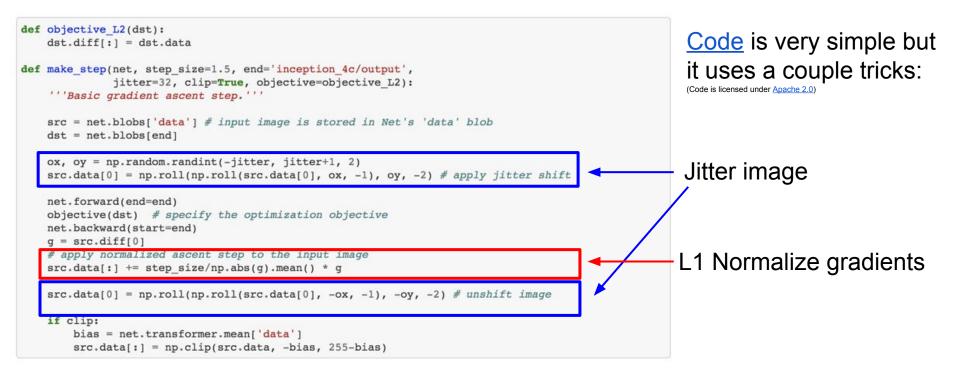
#### Fei-Fei Li & Justin Johnson & Serena Yeung

Lecture 11 - 39 May 10, 2017



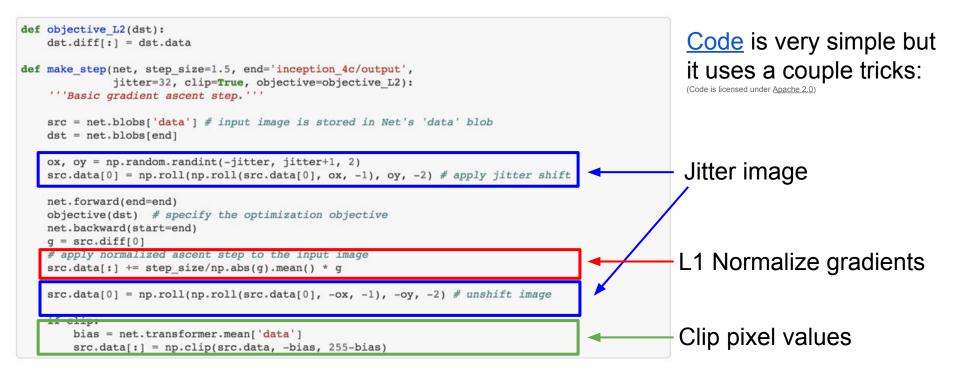
Fei-Fei Li & Justin Johnson & Serena Yeung

Lecture 11 - 40 May 10, 2017



Fei-Fei Li & Justin Johnson & Serena Yeung

Lecture 11 - 41 May 10, 2017



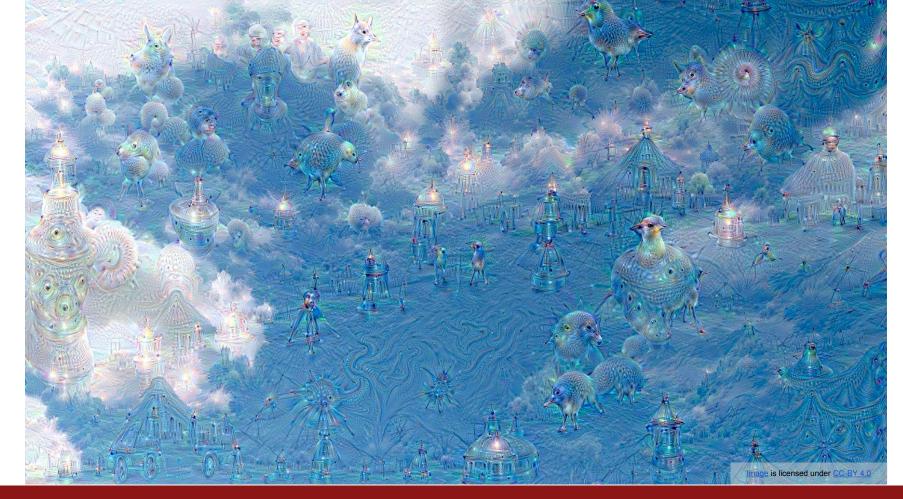
Also uses multiscale processing for a fractal effect (not shown)

Fei-Fei Li & Justin Johnson & Serena Yeung

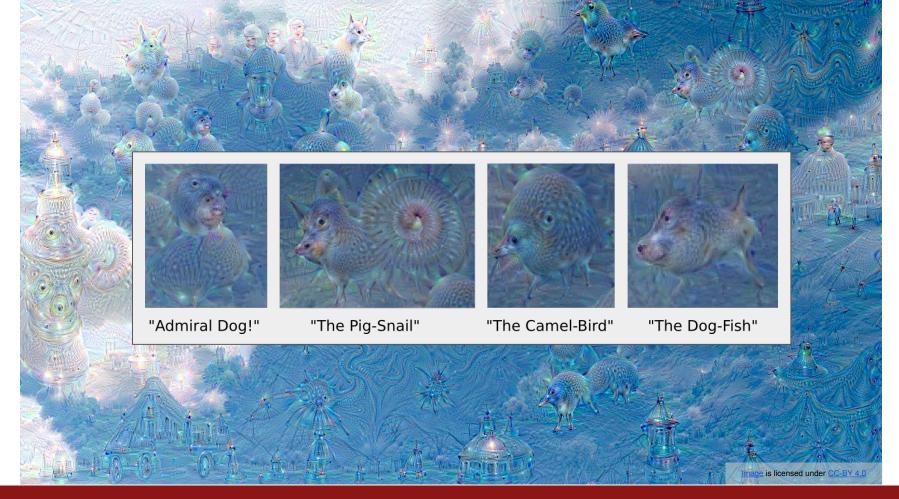
Lecture 11 - 42 May 10, 2017



Lecture 11 - 43 May 10, 2017



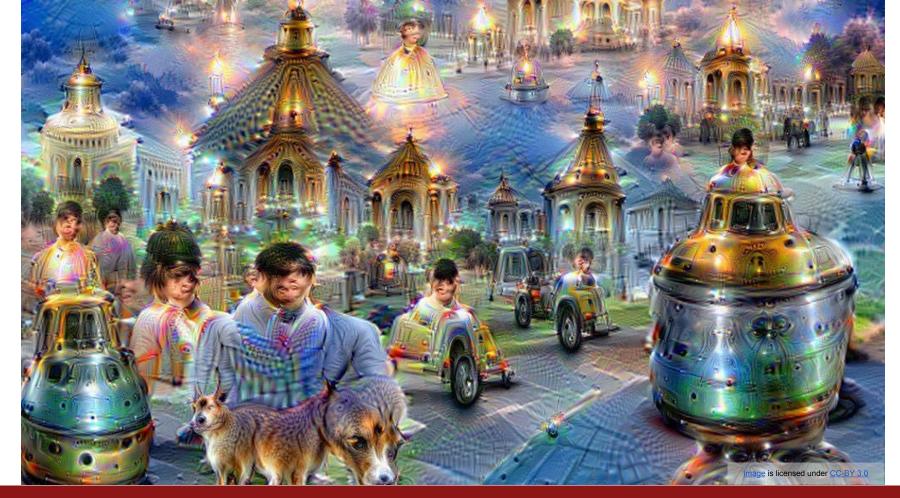
Lecture 11 - 44 May 10, 2017



Lecture 11 - 45 May 10, 2017



Lecture 11 - 46 May 10, 2017



Lecture 11 - 47 May 10, 2017



Image is licensed under CC-BY 4.0

#### Fei-Fei Li & Justin Johnson & Serena Yeung

Lecture 11 - 48 May 10, 2017

### **Feature Inversion**

Given a CNN feature vector for an image, find a new image that:

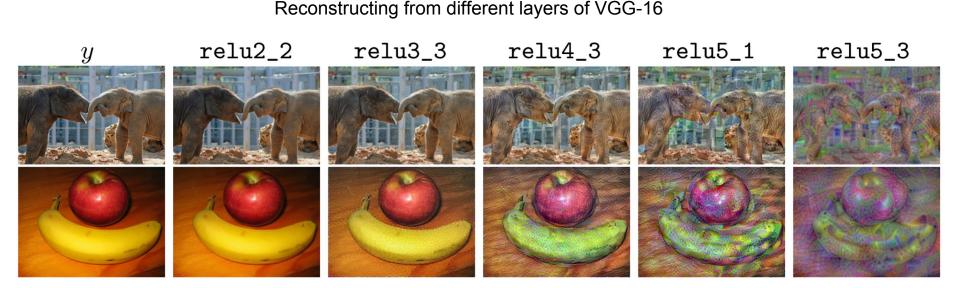
- Matches the given feature vector
- "looks natural" (image prior regularization)

Mahendran and Vedaldi, "Understanding Deep Image Representations by Inverting Them", CVPR 2015

Fei-Fei Li & Justin Johnson & Serena Yeung

Lecture 11 - 49 May 10, 2017

### **Feature Inversion**



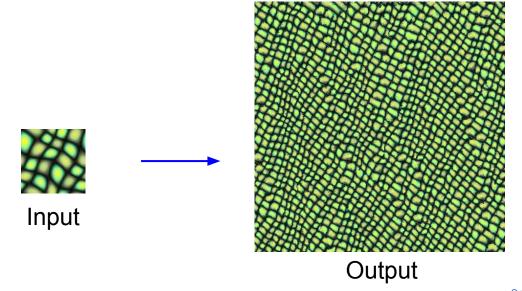
Mahendran and Vedaldi, "Understanding Deep Image Representations by Inverting Them", CVPR 2015 Figure from Johnson, Alahi, and Fei-Fei, "Perceptual Losses for Real-Time Style Transfer and Super-Resolution", ECCV 2016. Copyright Springer, 2016. Reproduced for educational purposes.

#### Fei-Fei Li & Justin Johnson & Serena Yeung

### Lecture 11 - 50 May 10, 2017

### **Texture Synthesis**

Given a sample patch of some texture, can we generate a bigger image of the same texture?



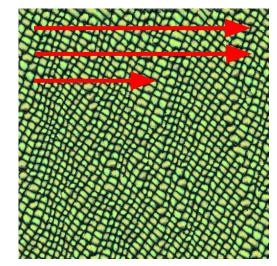
Output image is licensed under the MIT license

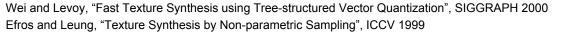
Fei-Fei Li & Justin Johnson & Serena Yeung

Lecture 11 - 51 May 10, 2017

# **Texture Synthesis: Nearest Neighbor**

Generate pixels one at a time in scanline order; form neighborhood of already generated pixels and copy nearest neighbor from input



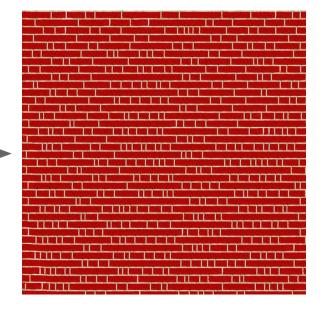


Fei-Fei Li & Justin Johnson & Serena Yeung

Lecture 11 - 52 May 10, 2017



### **Texture Synthesis: Nearest Neighbor**



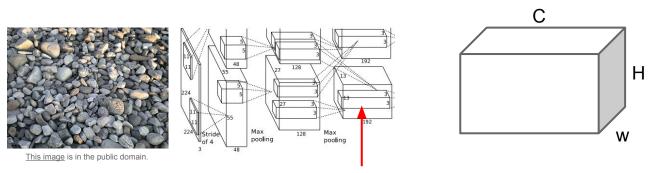
it ndateears coune Tring rooms," as Heft he fast nd it ars dat notars ortseas ribed it last nt hest bedian Al. econicalHomd ith Al. Heftars of as da Lewindailf I tian Al Ths," as Lewing questies last aticarstical 1. He is dian Al last fal counda Lew; at "this dailyears d ily edianicall. Hoorewing rooms," as House De fale f De und itical counsestscribed it last fall. He fall. Hefft rs oroheoned it nd it he left a ringing questica Lewin icars coecoms," astore years of Monica Lewinow see a Thas Fring roome stooniscat nowea re left a roouse bouestof Mfe lelft a Lést fast ngine làuuesticars He nd it rip?" TrHouself, a ring ind itsonestid it a ring que astical cois ore years of Moung fall. He ribof Mouse ore years of and a Tripp?" That hedian Al Lest fasee yea ada Tripp?' folitical comedian Alet he few se ring que olitical cone re years of the storears ofas 1 Frat nica L res Lew se lest a rime 1 He fas quest nging of, at beou

sero of the wei someone when in lasoness do falsofit int, A'se' a living of latti of entressoon while sed in Fof Lofar twoff the sel Lusinudari the getics if experiences seen of the selection of the second of the second s Lievou res ong 1 Hauwelei ques II. Hilleral confai cout far asterate her issa aquus daiortorius s'i Firs comprecing big ica <sup>23</sup>) locphicaswater out i figurations so i libitation set and gaining to dist, at ye collect HaroApseed I Frigging ogang out ribitation <sup>23</sup> and higher zionindo streg for yet cloude to soverooungel can one of <sup>3</sup> and market at which it unifing so of with fractionation. Here is 104, Arenous Lesir, "amansusia ander diondes aufors à hareana Fria<sup>01 ro</sup>t roqué l'94, stadiatet f Mohdat the diale dialege avenou avet al it a sof, aw seeds Len Hoeves as is not a that noeveer to are La movie year to a fallence in our days, i wonwinquiton and af thing lat nousineerst it in the moratasta ing as willy F. Exact a we make a reconducarfality roticarolicity lighterests of PHotessin attessing of the solution there are a constructed and from the start the start of the succession of the succes ac junys da n'w quanch many sal back Horyeas Leewfand 1 of fur, 2 Cabuyenties," dea Léura fa catthich to ubmériu hy jucus as us hoting comments that the store of the second way a second way a start the store of the sto 1 a tas concessed its of each process the lararse ris dioit brack ince Lts, even glock whoes glaicalt the rit familia <sup>stle</sup>iff nexistenes La laurewerse inwawn michaizabus or estis, <sup>10</sup>far 2001 latar s dasum isse<sup>pens</sup> (0da et il. Levinandra e yin<sub>j</sub>uze ysgigt be a rfatenti gilyg omarsies weal Hat yars nev, refit used ng though Holourfathe infostival ot malfo it aulear qual decorring of ars Ded frame, Ar moonue riprative, ere Ledeforde Heitse source: HeAst, 't utbose oreater your thisie rataineus Eetor: ofies flasmhams micas est off.et obravov su moder franz failaths Eougi lasts yest corought sein Hodin, vyslaest loon: Thatie, ast, of nist's see As ing t f w sis a cartourous see is cookilier fairlt dies capoos Heeding is night welling in Harded tobuls far and instance purchains rithed Fhare Thaines de as Thasorseeing ag order dia arrss Broom send success and Himseft anit anit anit of a Lite. A Leadars used as dit assessibles aquiptus itages of acaeas, on Protect Ishans qualast support a sites attirts ved it id liftst a at he's ridz frite luisi ft". siptismit sous I I enasive; Hing: Atom ing ins ridk foit I w qu to the set of the tan inatis consumptances as farmers in parsies inginger nos deis innu zouestowest, and ut at michinkous e til de Frs is over i Ar mity iburstans , sHe?" Ichis environed anna attionigal Hase ficaid stit rwiteda, " zonis j'at inga aut rob. H Aoforinaus, dailof haqu stucoeffa t que une y que de sere d'inter fait nit Thinatlaipber intéré dijaster la initiaté sere la sere la sere at sere none FormauHifthdairse inea "intestidated, at n Alexandal estic," ensert a ynow ng pore vertgiserigicou fquee s erst stort rus (tod it it The Fulls rog rysocial statisticais' af daiburg inter : ylameautic que Hotses it Lewic fatiolary est ryonaling noras st as Al are out interesting of the stand of the standard of t gireolailfailt se noaiusresou oft ious faat reneurs cuers, dia er la le rears, vaaffakine tilpre sershiw unvitsoione is or un ndew: De. Husá va Ho zusiveje valjtat te visevezavorus stizzooiridan las gat igit ofa De as That gik he fal inevarais solw my far of Haas confing our ag queritance a confir leders Hass dred forse fars on fare ais new fast cour cas Levelars a dreng at e e notedue mes utes géstid. A Pouse s'eles récours de grimere fl'é ridises trittérésenstices este "A b Maisson disentition de the needing z along one Leg rig fast fair reases sws input daielft's itsts of with a rm. Leg with dal de dail. Theous forming r ore : yearb string of Maoniby ibringft. Iws a way not Haginself and isat cases of a anticcofanition 1se. hear etars out ring ne anomai ("in a state"). Hot is sal of words is a solution of a state of the state e yals usig in daice evelaving heistass Lones, arone viewelus lainomin's Eorinistieng dra offices Pist la frietort. Lievindained b qomilisione iniciata stillowing data detector and vectorized and state of the Sistooce 22 up a vitat ricoicistorijes ile fija Aiof: fallmonarritvlagig rearris dacountirsead Pranise damofat ind antifre T gunnable in "genorfinder Freezenses" 19 gunnable in "genorfinder Freezenses" 19 gunnable in "genorfinder Freezenses" 19 gunnable in "f hat caobigunties," sequee queue ific lest avant nei. Lab Diese gecoal hada' gunnabilitations leus dagstof HI que le un Honeruist, secina rey roces rorts a Lesinginearkister re : destructivita Fores assanta He str Litorez ers othop?

Images licensed under the MIT license

Fei-Fei Li & Justin Johnson & Serena Yeung

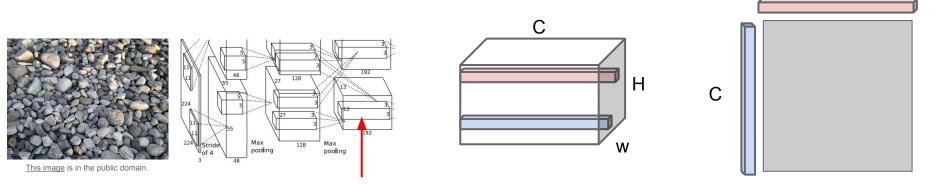
Lecture 11 - 53 May 10, 2017



Each layer of CNN gives C x H x W tensor of features; H x W grid of C-dimensional vectors

Fei-Fei Li & Justin Johnson & Serena Yeung

Lecture 11 - 54 May 10, 2017

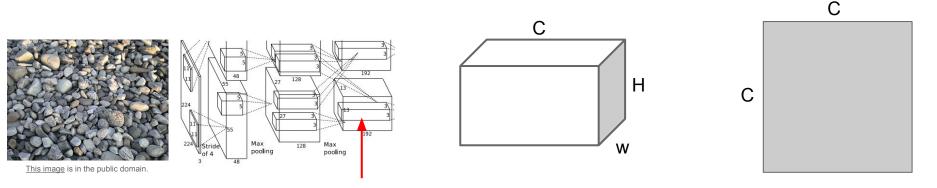


Each layer of CNN gives C x H x W tensor of features; H x W grid of C-dimensional vectors

Outer product of two C-dimensional vectors gives C x C matrix measuring co-occurrence

#### Fei-Fei Li & Justin Johnson & Serena Yeung

Lecture 11 - 55 May 10, 2017



Each layer of CNN gives C x H x W tensor of features; H x W grid of C-dimensional vectors

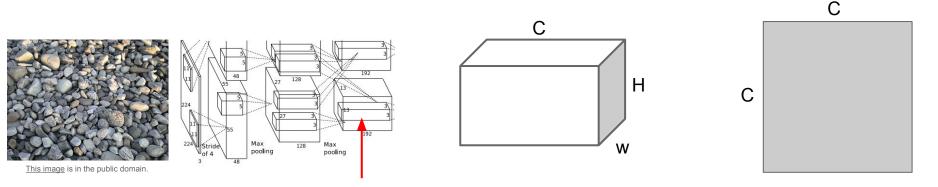
Outer product of two C-dimensional vectors gives C x C matrix measuring co-occurrence

Average over all HW pairs of vectors, giving **Gram matrix** of shape C x C

Gram Matrix

### Fei-Fei Li & Justin Johnson & Serena Yeung

Lecture 11 - 56 May 10, 2017



Each layer of CNN gives C x H x W tensor of features; H x W grid of C-dimensional vectors

Outer product of two C-dimensional vectors gives C x C matrix measuring co-occurrence

Average over all HW pairs of vectors, giving **Gram matrix** of shape C x C

Efficient to compute; reshape features from

 $C \times H \times W$  to  $=C \times HW$ 

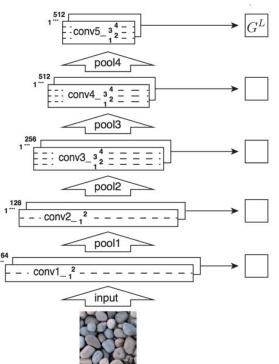
then compute  $G = FF^T$ 

#### Fei-Fei Li & Justin Johnson & Serena Yeung

Lecture 11 - 57 May 10, 2017

- 1. Pretrain a CNN on ImageNet (VGG-19)
- Run input texture forward through CNN, record activations on every layer; layer i gives feature map of shape C<sub>i</sub> × H<sub>i</sub> × W<sub>i</sub>
- 3. At each layer compute the *Gram matrix* giving outer product of features:

$$G_{ij}^{l} = \sum_{k} F_{ik}^{l} F_{jk}^{l} \text{ (shape } \mathbf{C_{i}} \times \mathbf{C_{i}})$$



Gatys, Ecker, and Bethge, "Texture Synthesis Using Convolutional Neural Networks", NIPS 2015 Figure copyright Leon Gatys, Alexander S. Ecker, and Matthias Bethge, 2015. Reproduced with permission.

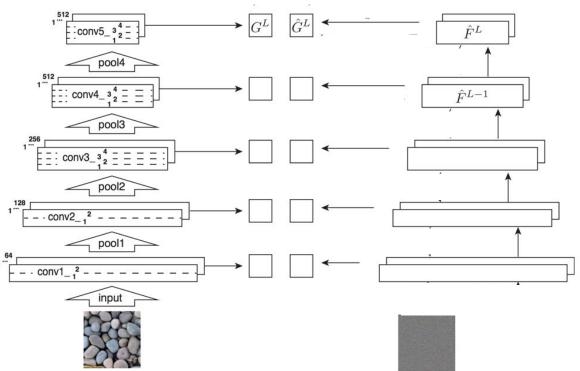
Fei-Fei Li & Justin Johnson & Serena Yeung

### Lecture 11 - 58 May 10, 2017

- 1. Pretrain a CNN on ImageNet (VGG-19)
- Run input texture forward through CNN, record activations on every layer; layer i gives feature map of shape C<sub>i</sub> × H<sub>i</sub> × W<sub>i</sub>
- 3. At each layer compute the *Gram matrix* giving outer product of features:

$$G_{ij}^{l} = \sum_{k} F_{ik}^{l} F_{jk}^{l}$$
 (shape C<sub>i</sub> × C<sub>i</sub>)

- 4. Initialize generated image from random noise
- 5. Pass generated image through CNN, compute Gram matrix on each layer



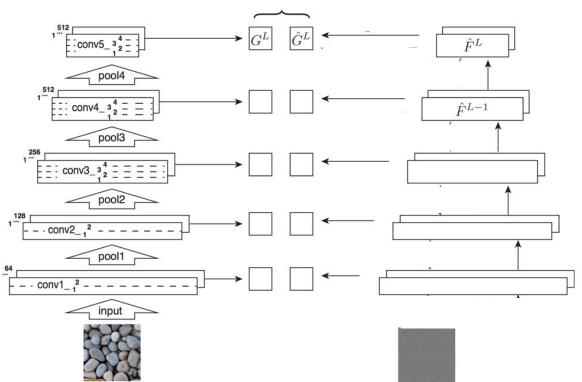
Lecture 11 - 59 May 10, 2017

Gatys, Ecker, and Bethge, "Texture Synthesis Using Convolutional Neural Networks", NIPS 2015 Figure copyright Leon Gatys, Alexander S. Ecker, and Matthias Bethge, 2015. Reproduced with permission.

Fei-Fei Li & Justin Johnson & Serena Yeung

$$E_l = rac{1}{4N_l^2 M_l^2} \sum_{i,j} \left( G_{ij}^l - \hat{G}_{ij}^l 
ight)^2 \qquad \mathcal{L}(ec{x}, \hat{ec{x}}) = \sum_{l=0}^L w_l E_l$$

- 1. Pretrain a CNN on ImageNet (VGG-19)
- Run input texture forward through CNN, record activations on every layer; layer i gives feature map of shape C<sub>i</sub> × H<sub>i</sub> × W<sub>i</sub>
- 3. At each layer compute the *Gram matrix* giving outer product of features:
- $G_{ij}^{l} = \sum_{k} F_{ik}^{l} F_{jk}^{l} \text{ (shape C}_{\rm i} \times {\rm C}_{\rm i}\text{)}$
- 4. Initialize generated image from random noise
- 5. Pass generated image through CNN, compute Gram matrix on each layer
- 6. Compute loss: weighted sum of L2 distance between Gram matrices



Lecture 11 - 60 May 10, 2017

Gatys, Ecker, and Bethge, "Texture Synthesis Using Convolutional Neural Networks", NIPS 2015 Figure copyright Leon Gatys, Alexander S. Ecker, and Matthias Bethge, 2015. Reproduced with permission.

### Fei-Fei Li & Justin Johnson & Serena Yeung

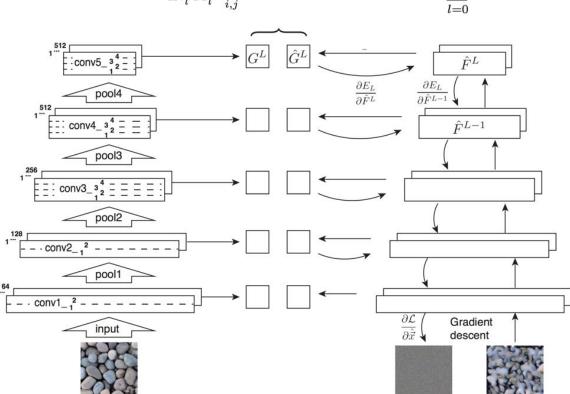
$$E_l = rac{1}{4N_l^2 M_l^2} \sum_{i,j} \left( G_{ij}^l - \hat{G}_{ij}^l 
ight)^2 \qquad \mathcal{L}(ec{x}, \hat{ec{x}}) = \sum_{l=0}^L w_l E_l$$

- 1. Pretrain a CNN on ImageNet (VGG-19)
- Run input texture forward through CNN, record activations on every layer; layer i gives feature map of shape C<sub>i</sub> × H<sub>i</sub> × W<sub>i</sub>
- 3. At each layer compute the *Gram matrix* giving outer product of features:
- $G_{ij}^{l} = \sum_{k} F_{ik}^{l} F_{jk}^{l}$  (shape C<sub>i</sub> × C<sub>i</sub>)
- 4. Initialize generated image from random noise
- 5. Pass generated image through CNN, compute Gram matrix on each layer
- 6. Compute loss: weighted sum of L2 distance between Gram matrices
- 7. Backprop to get gradient on image
- 8. Make gradient step on image
- 9. GOTO 5

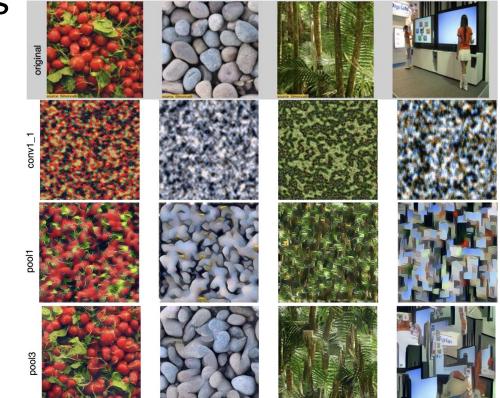
Gatys, Ecker, and Bethge, "Texture Synthesis Using Convolutional Neural Networks", NIPS 2015 Figure copyright Leon Gatys, Alexander S. Ecker, and Matthias Bethge, 2015. Reproduced with permission.

Fei-Fei Li & Justin Johnson & Serena Yeung

### Lecture 11 - 61 May 10, 2017



Reconstructing texture from higher layers recovers larger features from the input texture



Gatys, Ecker, and Bethge, "Texture Synthesis Using Convolutional Neural Networks", NIPS 2015 Figure copyright Leon Gatys, Alexander S. Ecker, and Matthias Bethge, 2015. Reproduced with permission.

Fei-Fei Li & Justin Johnson & Serena Yeung

Lecture 11 - 62 May 10, 2017

### Neural Texture Synthesis: Texture = Artwork

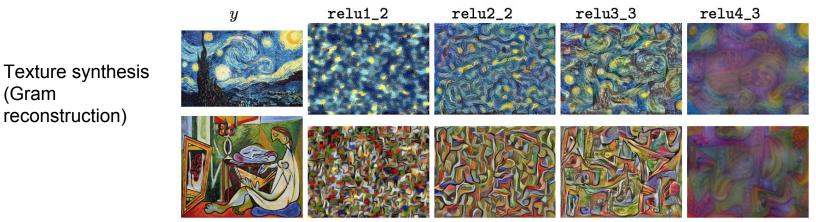


Figure from Johnson, Alahi, and Fei-Fei, "Perceptual Losses for Real-Time Style Transfer and Super-Resolution", ECCV 2016. Copyright Springer, 2016. Reproduced for educational purposes

(Gram

#### Fei-Fei Li & Justin Johnson & Serena Yeung

Lecture 11 - 63 May 10, 2017

# Neural Style Transfer: Feature + Gram

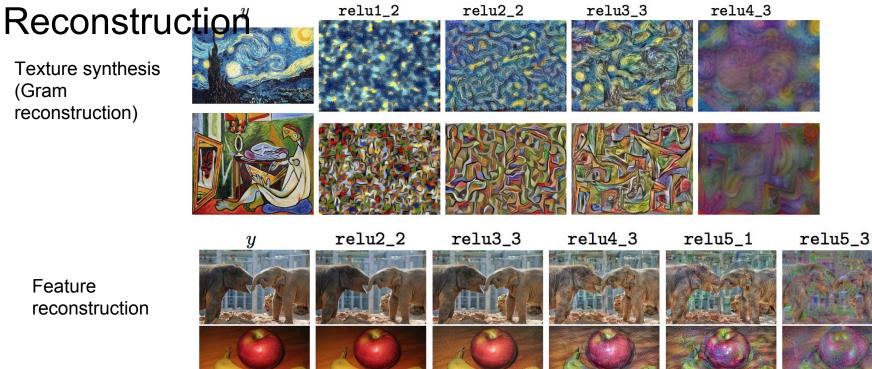


Figure from Johnson, Alahi, and Fei-Fei, "Perceptual Losses for Real-Time Style Transfer and Super-Resolution", ECCV 2016. Copyright Springer, 2016. Reproduced for educational purposes.

Feature

(Gram

Fei-Fei Li & Justin Johnson & Serena Yeung

Lecture 11 - 64 May 10, 2017

#### Content Image



This image is licensed under CC-BY 3.0

#### Style Image



Starry Night by Van Gogh is in the public domain

Gatys, Ecker, and Bethge, "Texture Synthesis Using Convolutional Neural Networks", NIPS 2015

Fei-Fei Li & Justin Johnson & Serena Yeung

### Lecture 11 - 65 May 10, 2017

#### Content Image



This image is licensed under CC-BY 3.0

#### Style Image



Starry Night by Van Gogh is in the public domain

Style Transfer!

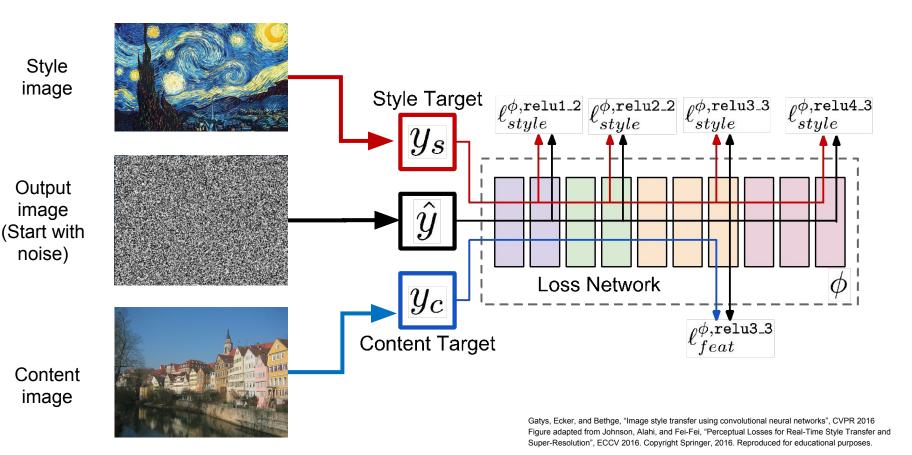


This image copyright Justin Johnson, 2015. Reproduced with permission.

Gatys, Ecker, and Bethge, "Image style transfer using convolutional neural networks", CVPR 2016

#### Fei-Fei Li & Justin Johnson & Serena Yeung

### Lecture 11 - 66 May 10, 2017



### Lecture 11 - 67 May 10, 2017

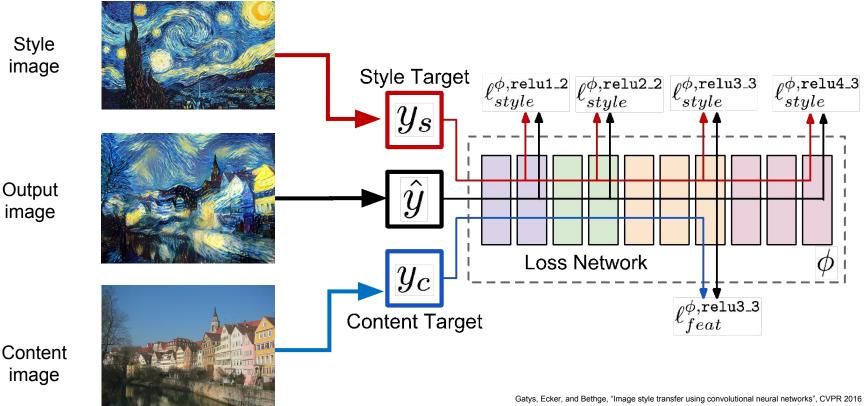


Figure adapted from Johnson, Alahi, and Fei-Fei, "Perceptual Losses for Real-Time Style Transfer and Super-Resolution", ECCV 2016. Copyright Springer, 2016. Reproduced for educational purposes.

#### Fei-Fei Li & Justin Johnson & Serena Yeung

### Lecture 11 - 68 May 10, 2017

Example outputs from <u>my implementation</u> (in Torch)



Gatys, Ecker, and Bethge, "Image style transfer using convolutional neural networks", CVPR 2016 Figure copyright Justin Johnson, 2015.

Fei-Fei Li & Justin Johnson & Serena Yeung

Lecture 11 - 69 May 10, 2017



Fei-Fei Li & Justin Johnson & Serena Yeung

Lecture 11 - 70 May 10, 2017

Resizing style image before running style transfer algorithm can transfer different types of features



Gatys, Ecker, and Bethge, "Image style transfer using convolutional neural networks", CVPR 2016 Figure copyright Justin Johnson, 2015.

Fei-Fei Li & Justin Johnson & Serena Yeung

### Lecture 11 - 71 May 10, 2017

# Neural Style Transfer: Multiple Style Images

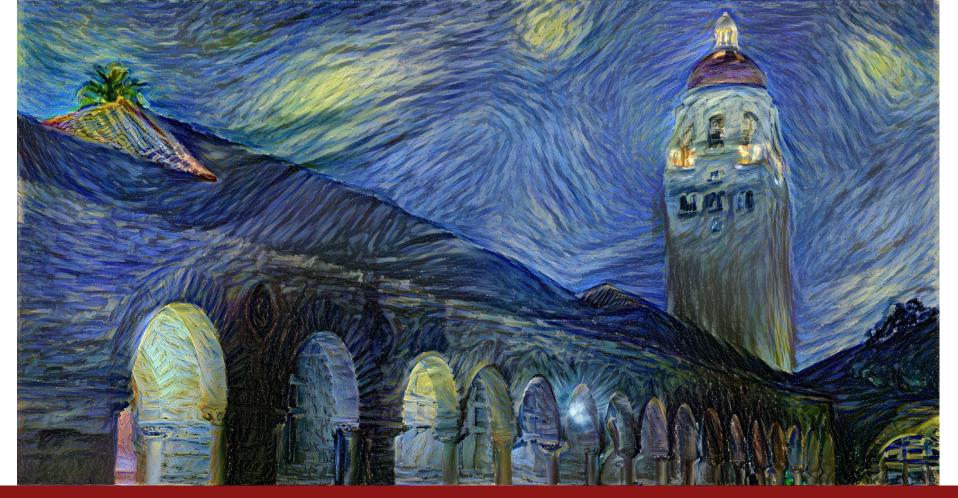
Mix style from multiple images by taking a weighted average of Gram matrices



Gatys, Ecker, and Bethge, "Image style transfer using convolutional neural networks", CVPR 2016 Figure copyright Justin Johnson, 2015.

#### Fei-Fei Li & Justin Johnson & Serena Yeung

### Lecture 11 - 72 May 10, 2017



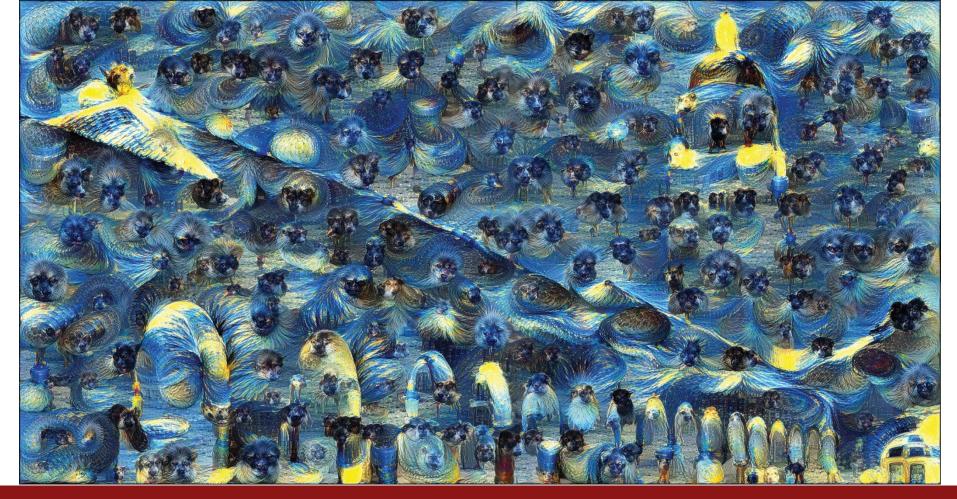
Fei-Fei Li & Justin Johnson & Serena Yeung

Lecture 11 - 73 May 10, 2017



Fei-Fei Li & Justin Johnson & Serena Yeung

Lecture 11 - 74 May 10, 2017



Fei-Fei Li & Justin Johnson & Serena Yeung

Lecture 11 - 75 May 10, 2017

## Neural Style Transfer

**Problem:** Style transfer requires many forward / backward passes through VGG; very slow!

Fei-Fei Li & Justin Johnson & Serena Yeung

Lecture 11 - 76 May 10, 2017

## Neural Style Transfer

**Problem:** Style transfer requires many forward / backward passes through VGG; very slow!

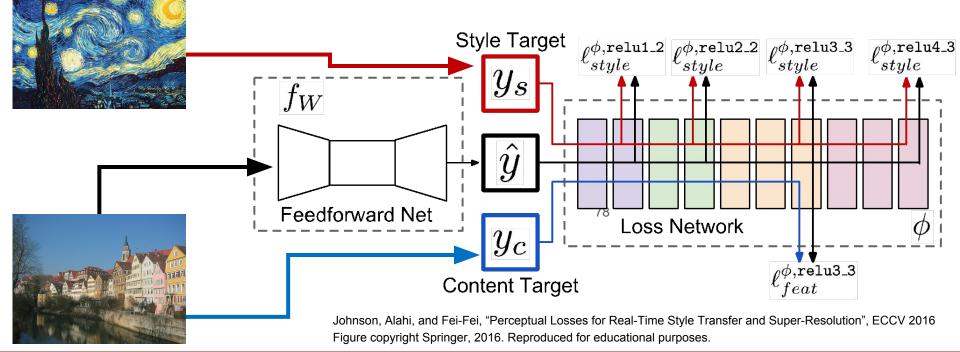
**Solution**: Train <u>another</u> neural network to perform style transfer for us!

Fei-Fei Li & Justin Johnson & Serena Yeung

Lecture 11 - 77 May 10, 2017

## **Fast Style Transfer**

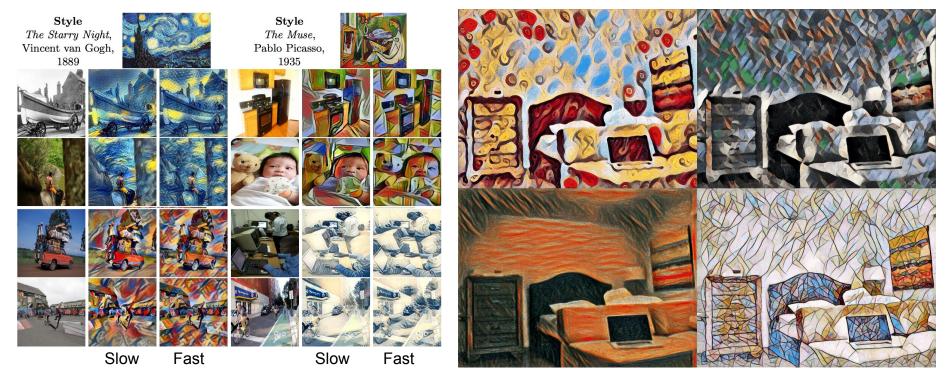
- (1) Train a feedforward network for each style
- (2) Use pretrained CNN to compute same losses as before
- (3) After training, stylize images using a single forward pass



Fei-Fei Li & Justin Johnson & Serena Yeung

### Lecture 11 - 78 May 10, 2017

# Fast Style Transfer

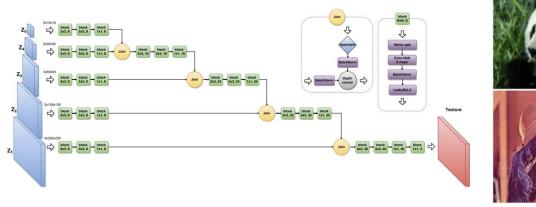


Johnson, Alahi, and Fei-Fei, "Perceptual Losses for Real-Time Style Transfer and Super-Resolution", ECCV 2016 Figure copyright Springer, 2016. Reproduced for educational purposes. https://github.com/jcjohnson/fast-neural-style

#### Fei-Fei Li & Justin Johnson & Serena Yeung

Lecture 11 - 79 May 10, 2017

## Fast Style Transfer



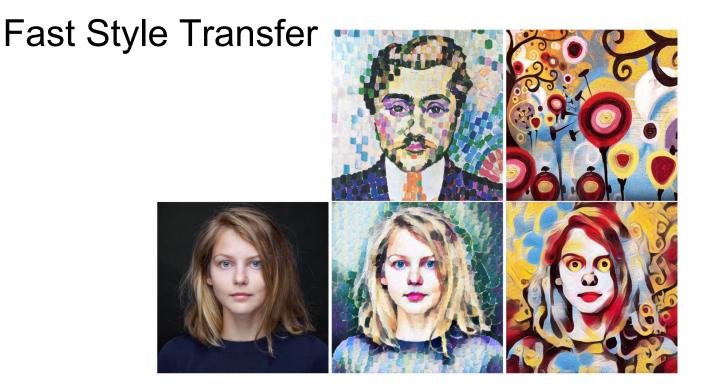


#### Concurrent work from Ulyanov et al, comparable results

Ulyanov et al, "Texture Networks: Feed-forward Synthesis of Textures and Stylized Images", ICML 2016 Ulyanov et al, "Instance Normalization: The Missing Ingredient for Fast Stylization", arXiv 2016 Figures copyright Dmitry Ulyanov, Vadim Lebedev, Andrea Vedaldi, and Victor Lempitsky, 2016. Reproduced with

Fei-Fei Li & Justin Johnson & Serena Yeung

#### Lecture 11 - 80 May 10, 2017



#### Replacing batch normalization with Instance Normalization improves results

Ulyanov et al, "Texture Networks: Feed-forward Synthesis of Textures and Stylized Images", ICML 2016 Ulyanov et al, "Instance Normalization: The Missing Ingredient for Fast Stylization", arXiv 2016 Figures copyright Dmitry Ulyanov, Vadim Lebedev, Andrea Vedaldi, and Victor Lempitsky, 2016. Reproduced with

Fei-Fei Li & Justin Johnson & Serena Yeung

### Lecture 11 - 81 May 10, 2017

## One Network, Many Styles



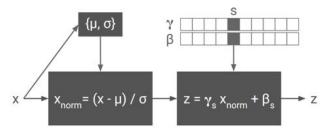
Dumoulin, Shlens, and Kudlur, "A Learned Representation for Artistic Style", ICLR 2017. Figure copyright Vincent Dumoulin, Jonathon Shlens, and Manjunath Kudlur, 2016; reproduced with permission.

Fei-Fei Li & Justin Johnson & Serena Yeung

Lecture 11 - 82 May 10, 2017

# One Network, Many Styles

Use the same network for multiple styles using <u>conditional instance</u> <u>normalization</u>: learn separate scale and shift parameters per style





Single network can blend styles after training

Dumoulin, Shlens, and Kudlur, "A Learned Representation for Artistic Style", ICLR 2017. Figure copyright Vincent Dumoulin, Jonathon Shlens, and Manjunath Kudlur, 2016; reproduced with permission.

Fei-Fei Li & Justin Johnson & Serena Yeung

### Lecture 11 - 83 May 10, 2017



Many methods for understanding CNN representations

Activations: Nearest neighbors, Dimensionality reduction, maximal patches, occlusion Gradients: Saliency maps, class visualization, fooling images, feature inversion Fun: DeepDream, Style Transfer.

Fei-Fei Li & Justin Johnson & Serena Yeung

Lecture 11 - 84 May 10, 2017

# Next time: **Unsupervised Learning** Autoencoders Variational Autoencoders Generative Adversarial Networks

Fei-Fei Li & Justin Johnson & Serena Yeung

Lecture 11 - 85 May 10, 2017