

DATA ANALYTICS USING DEEP LEARNING GT 8803 // Fall 2019 // Joy Arulraj

LECTURE #18:VISUALIZING & UNDERSTANDING CONVOLUTIONAL NETWORKS

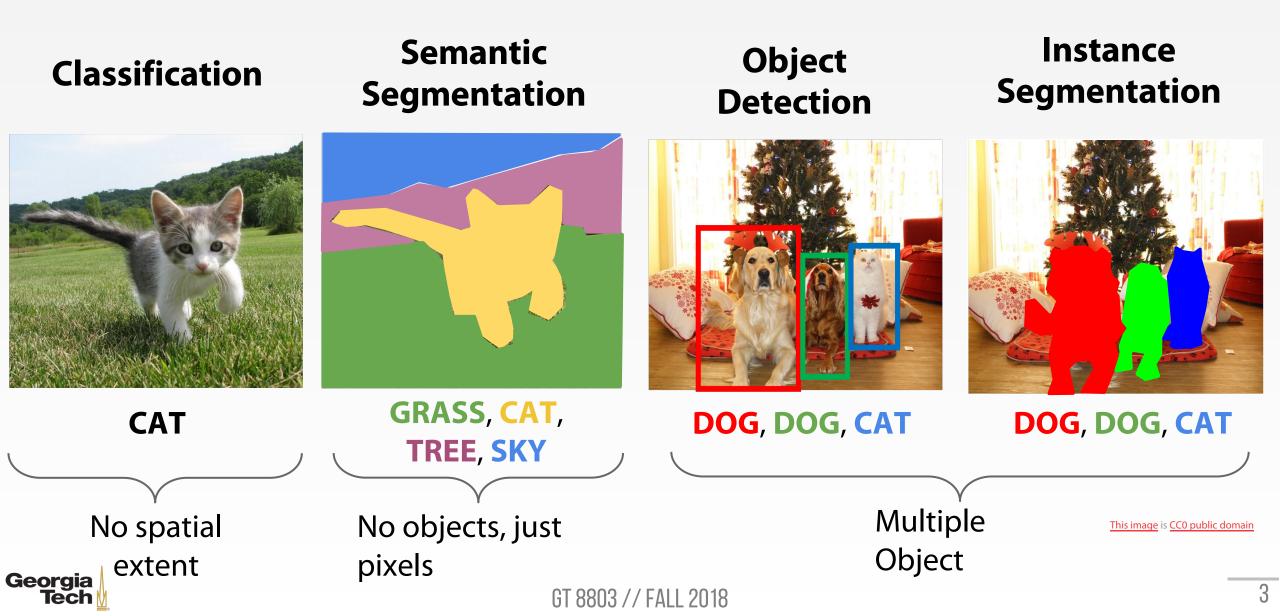
CREATING THE NEXT®

ADMINISTRIVIA

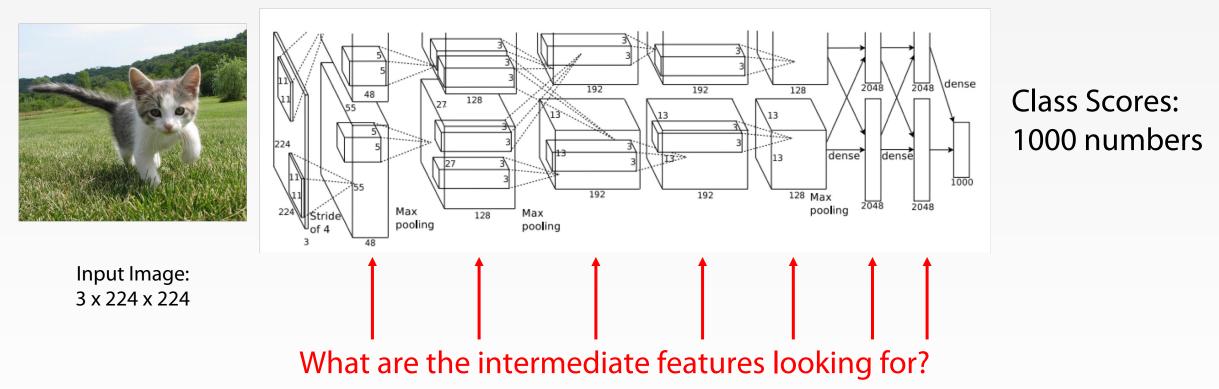
- Reminders
 - Code reviews due on Nov 9
 - Team member contribution analyses will be anonymous
 - Grades for project checkpoint #1 released
 - Assignment 3 released



LAST TIME: LOTS OF COMPUTER VISION TASKS



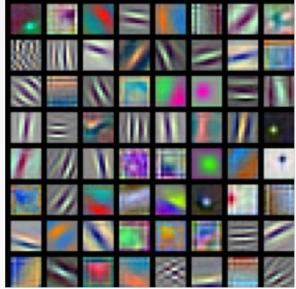
WHAT'S GOING ON INSIDE CONVNETS?



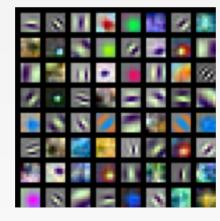


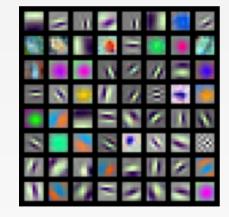
GT 8803 // FALL 2019

FIRST LAYER: VISUALIZE FILTERS





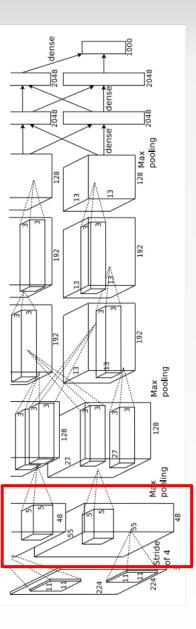




ResNet-18: 64 x 3 x 7 x 7

ResNet-101: 64 x 3 x 7 x 7

DenseNet-121: 64 x 3 x 7 x 7



AlexNet: 64 x 3 x 11 x 11

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



GT 8803 // FALL 2018

5

VISUALIZE THE Weights: 医脾胃 网络帕拉尔斯帕普尔克斯特 **FILTERS/KERNELS** Weights: (我認識認識的意識是是有以認識的。)(希望主要的意思的意思的意思的。)(希望是我的意思。 (RAW WEIGHTS) (如此)(如用用語書的各國家的書記書》200)(導起認識部構成的習慣是理想是)(筆品整件書 國際局局局部保護保護局)(成長局局局局局局局局局部局局局局)(國務委務署務局務務務務局局 調測設備)(機能學物理解解除結果或容許認識)(法非計測計中影響使共同判断中部部)(調算者 We can visualize 新聞新設設設設設設設設設計)(部務部務委員会部務委員会部委会)(委会部家業業務通知法 補助記載書語)(商務書法は言語言葉英書を記書品紙)(協力書から書品を言うな形式)(調 filters at higher 國際國際運動局局運動影響和希望)(非常管導管部的形式非常常常常的)(加加的影響的影響 layers, but not that interesting Weights: 約)(医医尿管性系统网络运行医院运行医院)(前常自我必须在自己的运行的现象形式 (these are taken 「金融」(原始在設備を設備を設備を設備した。)(使用の開始を設備を使用する。 from ConvNetJS 這希臘)(總基際總法應保護與美國物源的原原的回顧)(出於法律通知要要是以保護部分 CIFAR-10 demo) 新聞新聞集会)(電話和自己的ななのの新聞用語を用意なながの)(目的の目的ななななないのである) 非近許可以改成)(如何是自然体育」」如果的考虑是是是是自己的)(如何以及自己是有效的 如果可能注意的问题。(如果你能能能能能能能能能能能能能能)(这些能能能能能能能能能

(日本古学生)の新聞(日本))

layer 2 weights 20 x 16 x 7 x 7

layer 1 weights

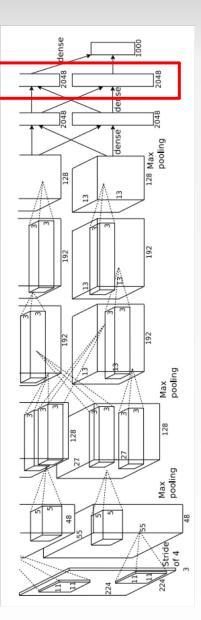
16 x 3 x 7 x 7

layer 3 weights $20 \times 20 \times 7 \times 7$



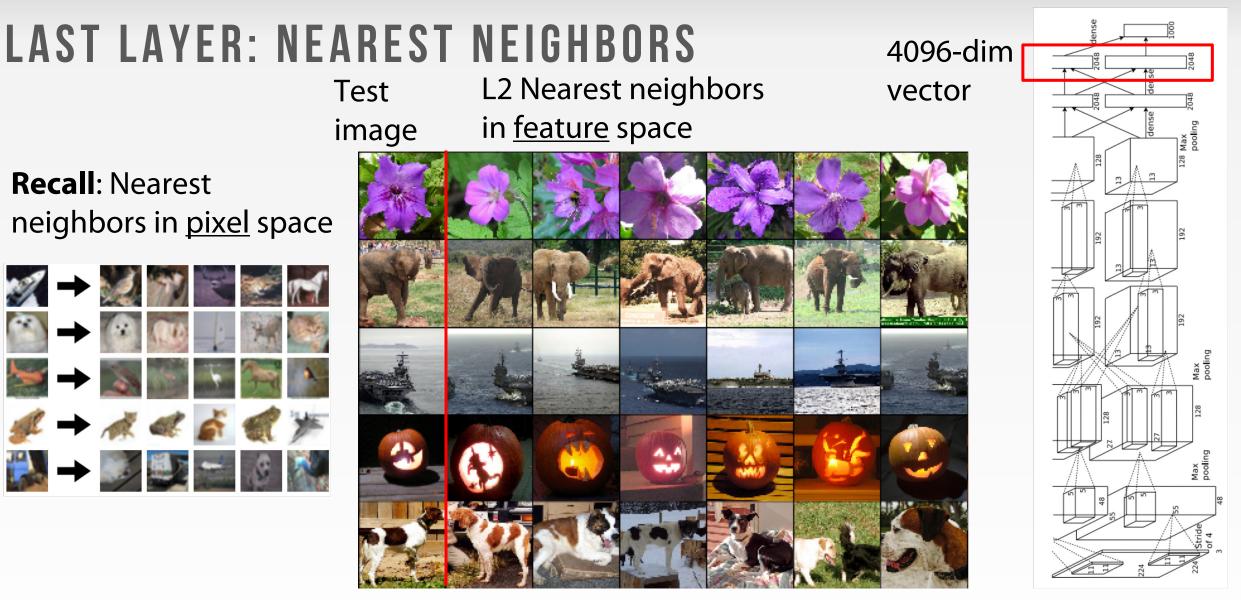
LAST LAYER

FC7 layer



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

Run the network on many images, collect the feature vectors



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

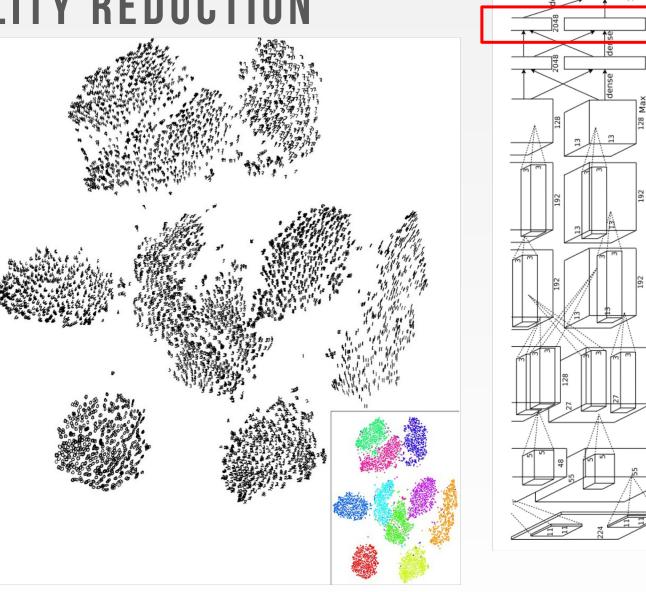


LAST LAYER: DIMENSIONALITY REDUCTION

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

Simple algorithm: Principal Component Analysis (PCA)

More complex: **t-SNE**

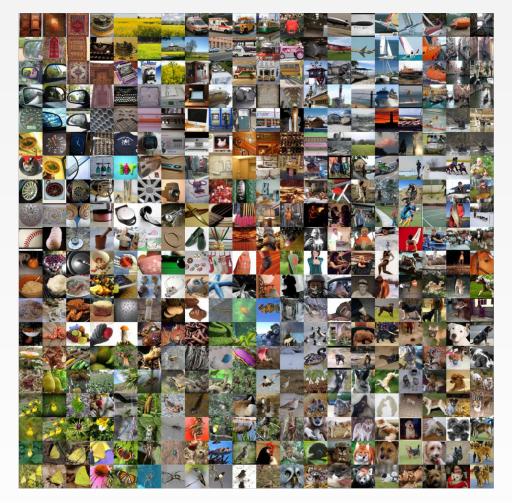


Van der Maaten and Hinton, "Visualizing Data using t-SNE", JMLR 2008 Figure copyright Laurens van der Maaten and Geoff Hinton, 2008. Reproduced with permission.



Max pooling

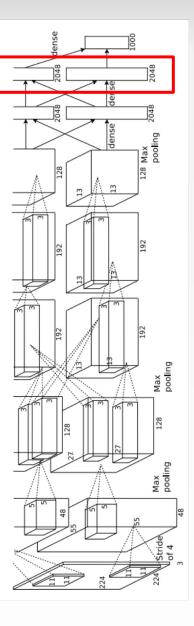
LAST LAYER: DIMENSIONALITY REDUCTION



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 <u>http://cs.stanford.edu/people/karpathy/cnnembed/</u>





GT 8803 // FALL 2018

VISUALIZING ACTIVATIONS

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

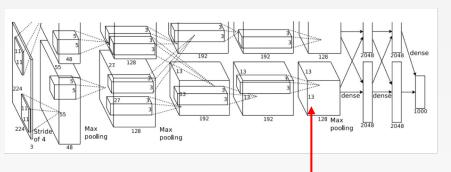
And a second second second	conv	v1 p) 1 r	11	conv2	p2	n2	con	v3 (conv	4 00	onv5	р5	fc6	fe7	fc8	\mathbf{pro}
			,		+					1.1					1 1	٩,	
		۰.									178				, *	- 1	
			4				-		-				٩. ,		<i>y</i>		
					а. 1.			-		, er 14							
				ŀ		۰.	Ξ.								-		
				ŀ		1 ⁸ 1			i.		2.4						
				Γ								1		•	Ċ		
					· * *			÷		, .	ч.						
		x.		đ				۰.			•	٠		٠	1		
						-		•									
	÷	-		. "	*	1									-		
		-	1	-	÷.						-					3.	
					ť.,			ć,						Ŷ			
fwd conv5 151 Bacl	c: off	2 F	3008	st:	0/1		1					-					

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



MAXIMALLY ACTIVATING PATCHES

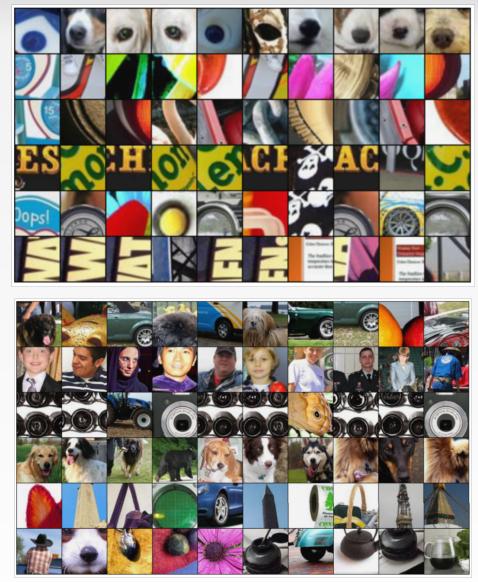




Pick a layer and a channel; e.g. conv5 is 128 x 13 x 13, pick channel 17 out of 128

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

Visualize image patches that correspond to maximal activations

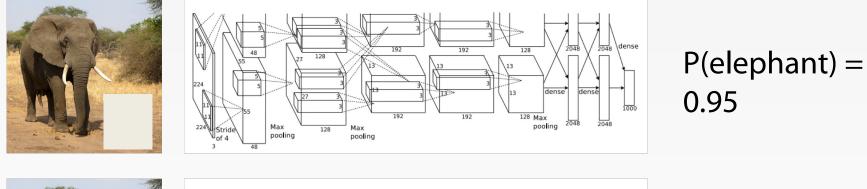


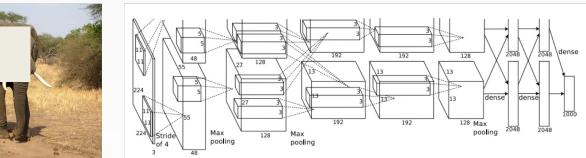
Springenberg et al, "Striving for Simplicity: The All Convolutional Net", ICLR Workshop 2015 Figure copyright Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, Martin Riedmiller, 2015; reproduced with permission.



WHICH PIXELS MATTER: Saliency via occlusion

Mask part of the image before feeding to CNN, check how much predicted probabilities change





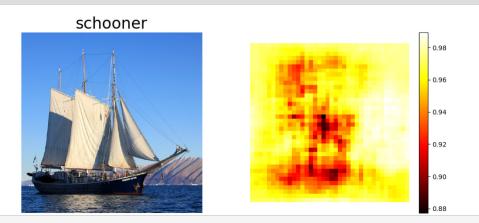
P(elephant) = 0.75

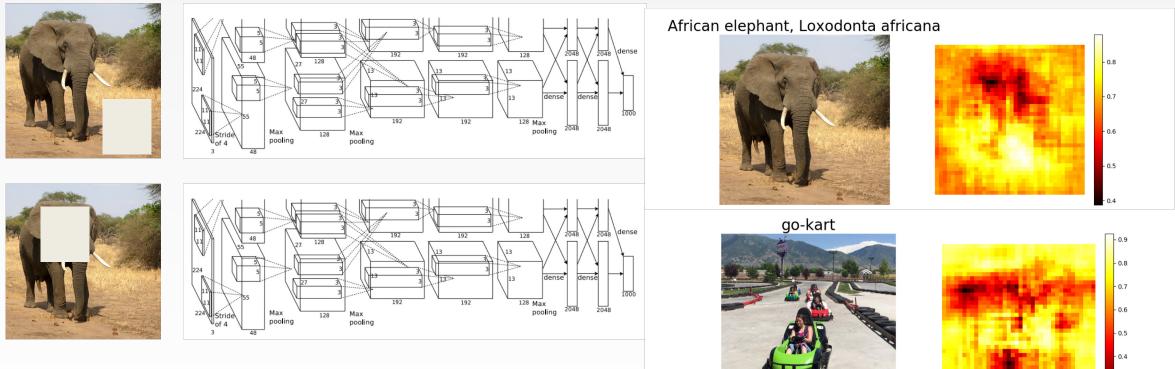
Zeiler and Fergus, "Visualizing and Understanding Convolutional Networks", ECCV 2014



WHICH PIXELS MATTER: Saliency via occlusion

Mask part of the image before feeding to CNN, check how much predicted probabilities change





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

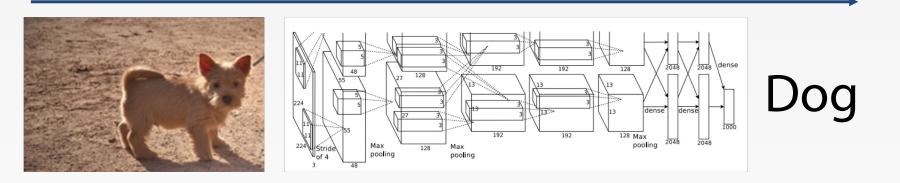
Georgia

Tech



WHICH PIXELS MATTER: SALIENCY VIA BACKPROP

Forward pass: Compute probabilities



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

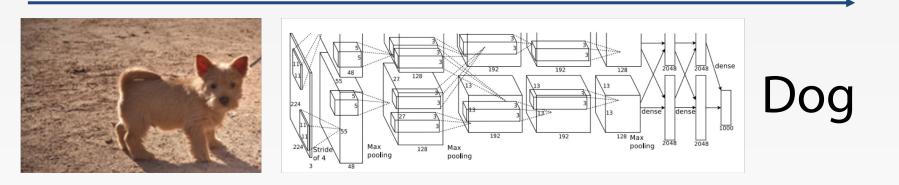
Figures copyright Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman, 2014; reproduced with permission.



GT 8803 // FALL 2018

WHICH PIXELS MATTER: SALIENCY VIA BACKPROP

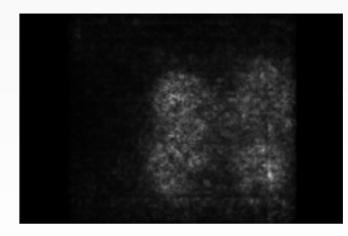
Forward pass: Compute probabilities



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

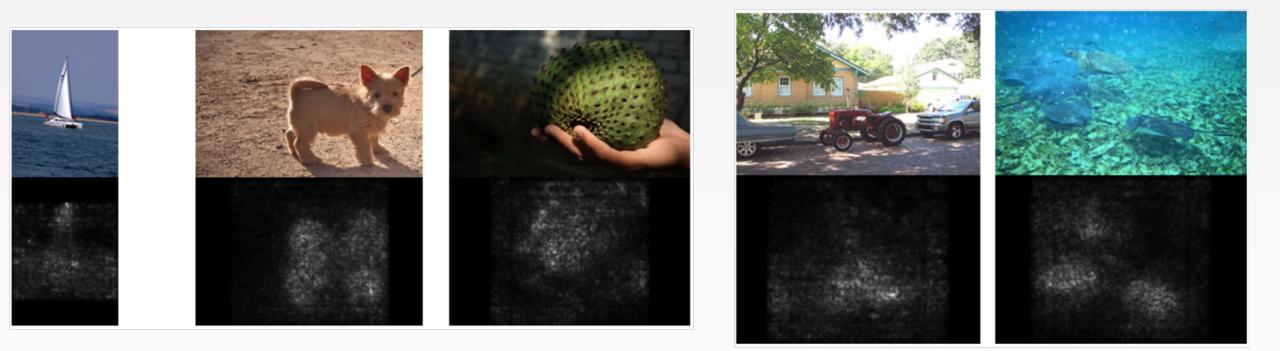
Simonyan, Vedaldi, and Zisserman, "Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps", ICLR Workshop 2014.

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





SALIENCY MAPS



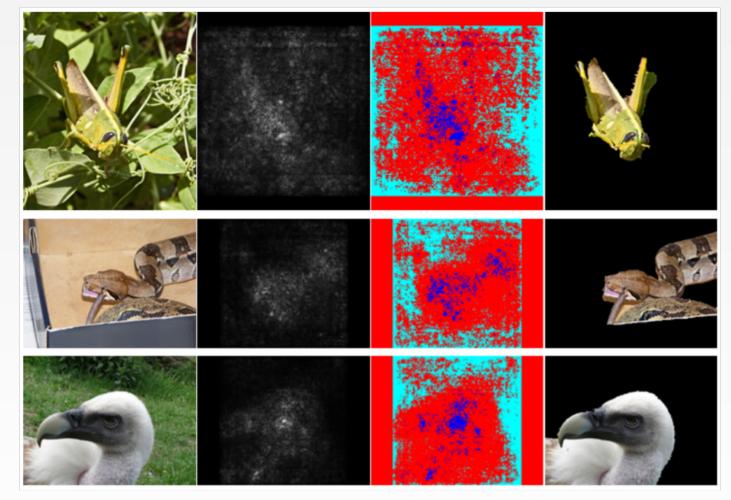
Simonyan, Vedaldi, and Zisserman, "Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps", ICLR Workshop 2014.

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



GT 8803 // FALL 2018

SALIENCY MAPS: SEGMENTATION WITHOUT SUPERVISION



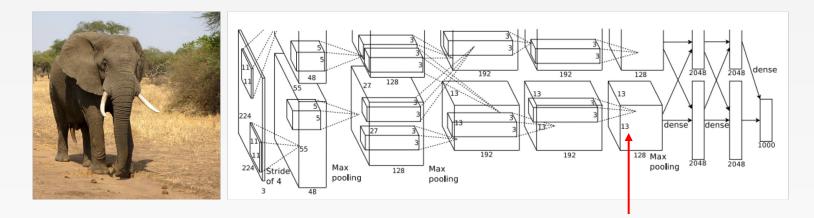
Use GrabCut on saliency map

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



INTERMEDIATE FEATURES VIA (GUIDED) BACKPROP



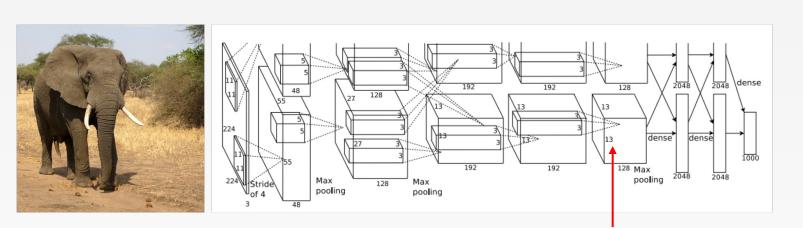
Pick a single intermediate neuron, e.g. one value in 128 x 13 x 13 conv5 feature map

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



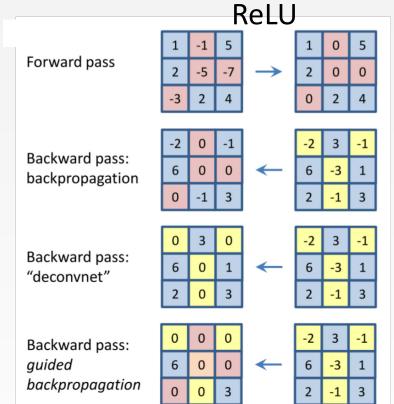
INTERMEDIATE FEATURES VIA (GUIDED) BACKPROP



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

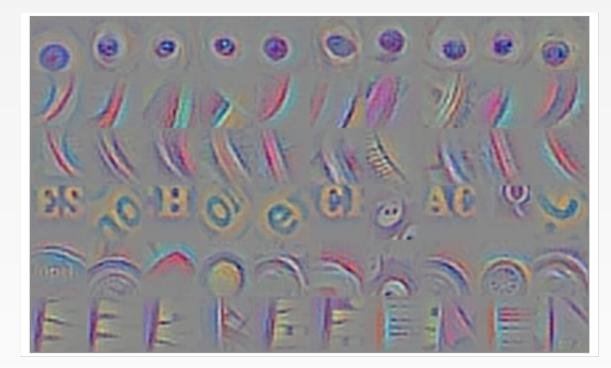


Images come out nicer if you only backprop positive gradients through each ReLU (guided backprop)



INTERMEDIATE FEATURES VIA (GUIDED) BACKPROP





Guided Backprop

Maximally activating patches (Each row is a different neuron)

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.



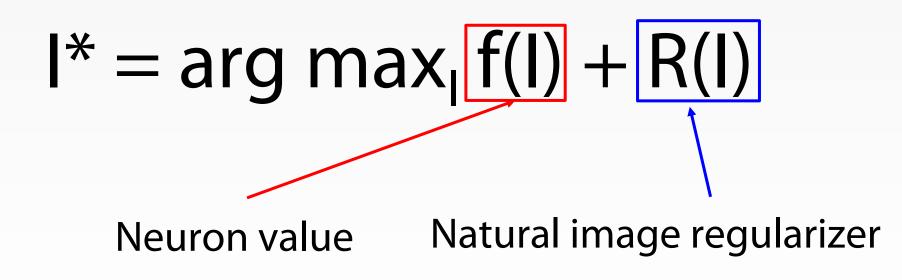
GT 8803 // FALL 2018

(Guided) backprop:

Find the part of an image that a neuron responds to

Gradient ascent:

Generate a synthetic image that maximally activates a neuron

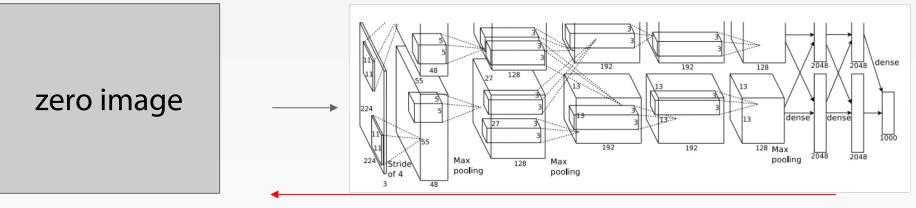




1. Initialize image to zeros

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

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



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

Simple regularizer: Penalize L2 norm of generated image

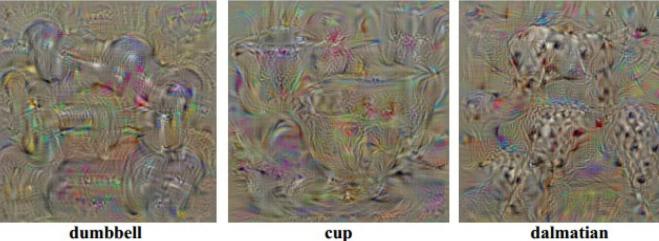
Simonyan, Vedaldi, and Zisserman, "Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps", ICLR Workshop 2014. Figures copyright Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman, 2014; reproduced with permission.



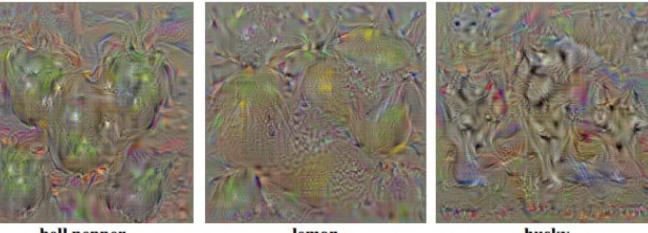
GT 8803 // FALL 2018

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

Simple regularizer: Penalize L2 norm of generated image



dumbbell



bell pepper

lemon

husky

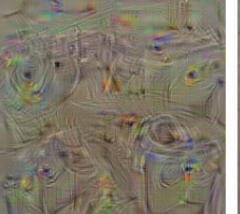
Simonyan, Vedaldi, and Zisserman, "Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps", ICLR Workshop 2014.

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

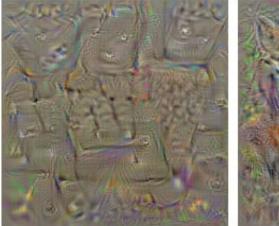


$$\arg\max_{I} S_c(I) - \frac{\lambda \|I\|_2^2}{2}$$

Simple regularizer: Penalize L2 norm of generated image



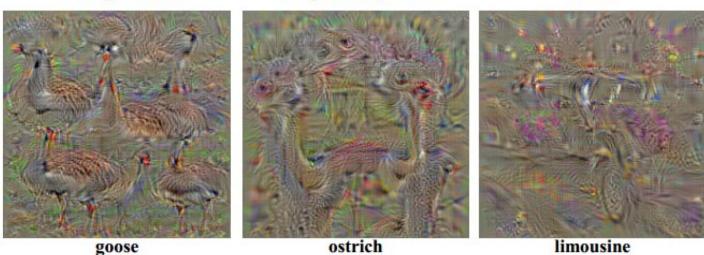
washing machine



computer keyboard



kit fox



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.





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

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

- Gaussian blur image
- Clip pixels with small values to 0
- Clip pixels with small gradients to 0

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



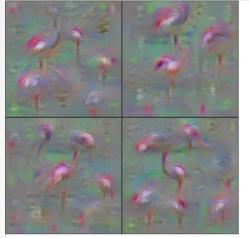
$$\arg\max_{I} S_c(I) - \lambda \|I\|_2^2$$

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

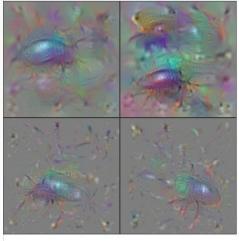
• Gaussian blur image

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

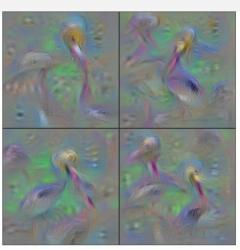
- Clip pixels with small values to 0
- Clip pixels with small gradients to 0



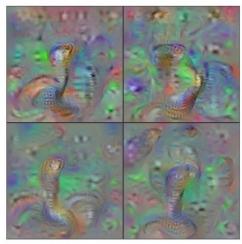
Flamingo



Ground Beetle



Pelican



Indian Cobra



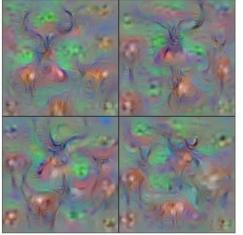
$$\arg\max_{I} S_c(I) - \lambda \|I\|_2^2$$

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

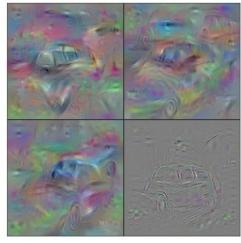
Gaussian blur image

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

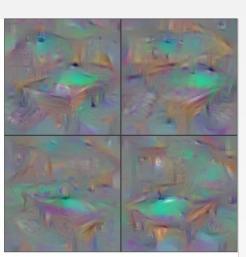
- Clip pixels with small values to 0
- Clip pixels with small gradients to 0



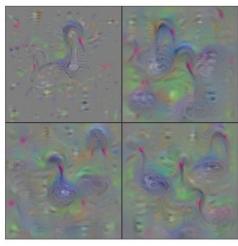
Hartebeest



Station Wagon



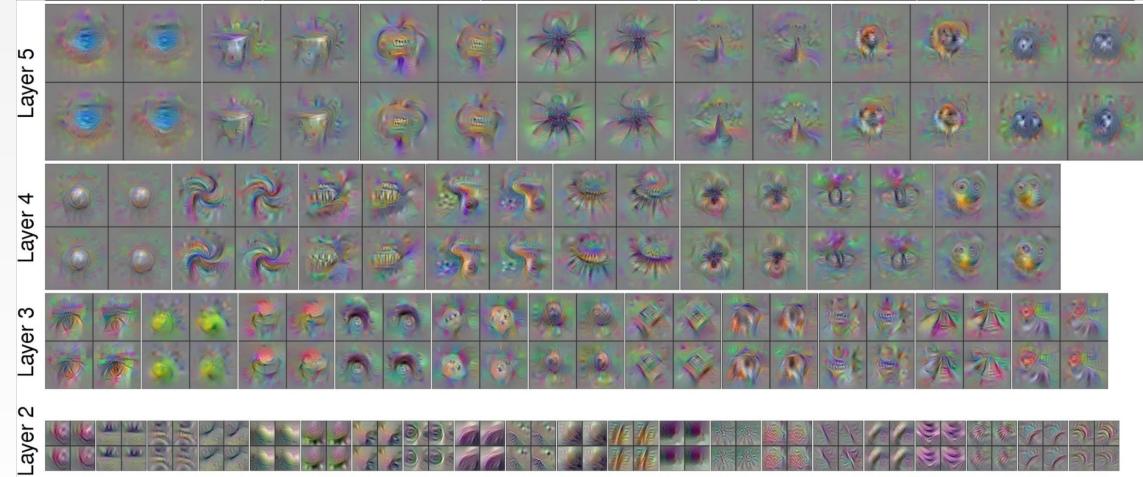
Billiard Table



Black Swan



Use the same approach to visualize intermediate features

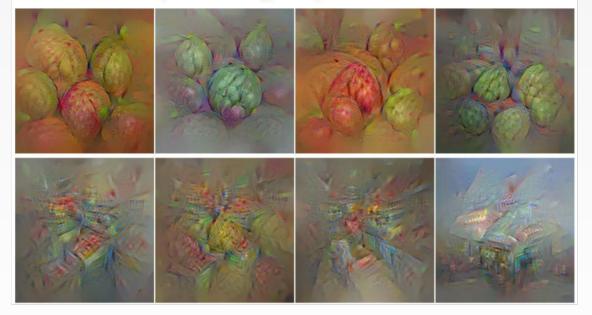


Yosinski et al, "Understanding Neural Networks Through Deep Visualization", ICML DL Workshop 2014. Figure copyright Jason Yosinski, Jeff Clune, Anh Nguyen, Thomas Fuchs, and Hod Lipson, 2014. Reproduced with permission.



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.





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.



Optimize in FC6 latent space instead of pixel space:



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.

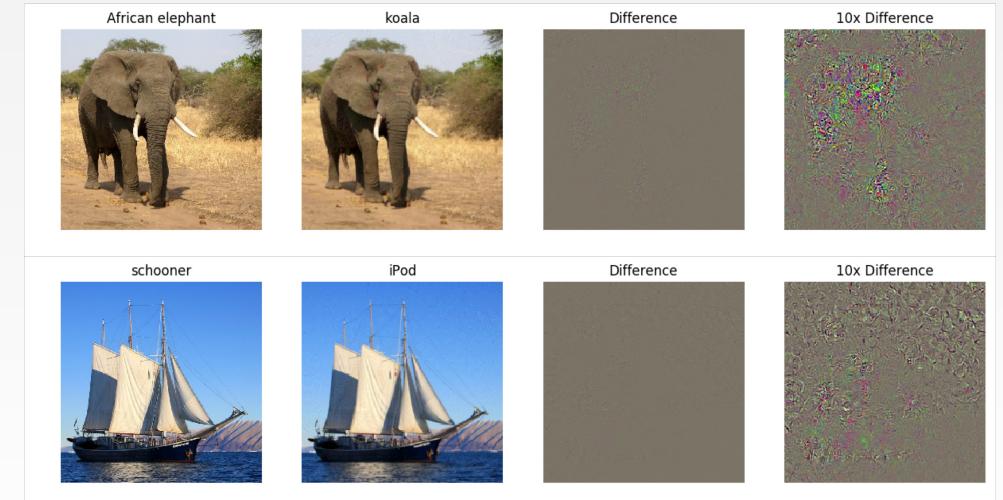


FOOLING IMAGES / ADVERSARIAL EXAMPLES

- (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

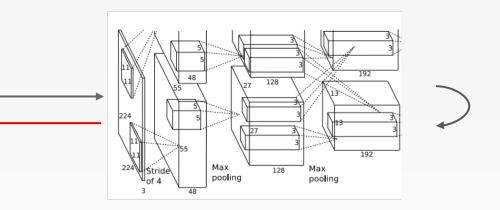




DEEPDREAM: AMPLIFY EXISTING FEATURES

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:

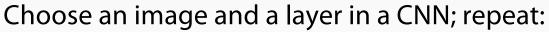
- 1. Forward: compute activations at chosen layer
- 2. Set gradient of chosen layer equal to its activation
- 3. Backward: Compute gradient on image
- 4. Update image

Mordvintsev, Olah, and Tyka, "Inceptionism: Going Deeper into Neural Networks", <u>Google Research Blog</u>. Images are licensed under <u>CC-BY 4.0</u>

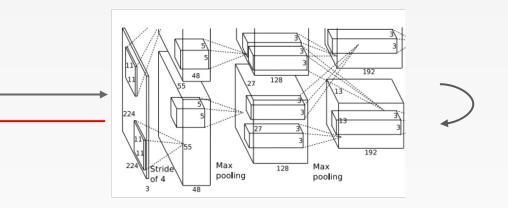


Rather than synthesizing an image to maximize a specific neuron, instead try to **amplify** the neuron activations at some layer in the network





- 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: $I^* = arg max_I \sum_i f_i(I)^2$

Mordvintsev, Olah, and Tyka, "Inceptionism: Going Deeper into Neural Networks", <u>Google Research Blog</u>. Images are licensed under <u>CC-BY 4.0</u>



```
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)
    q = 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 <u>Apache 2.0</u>)

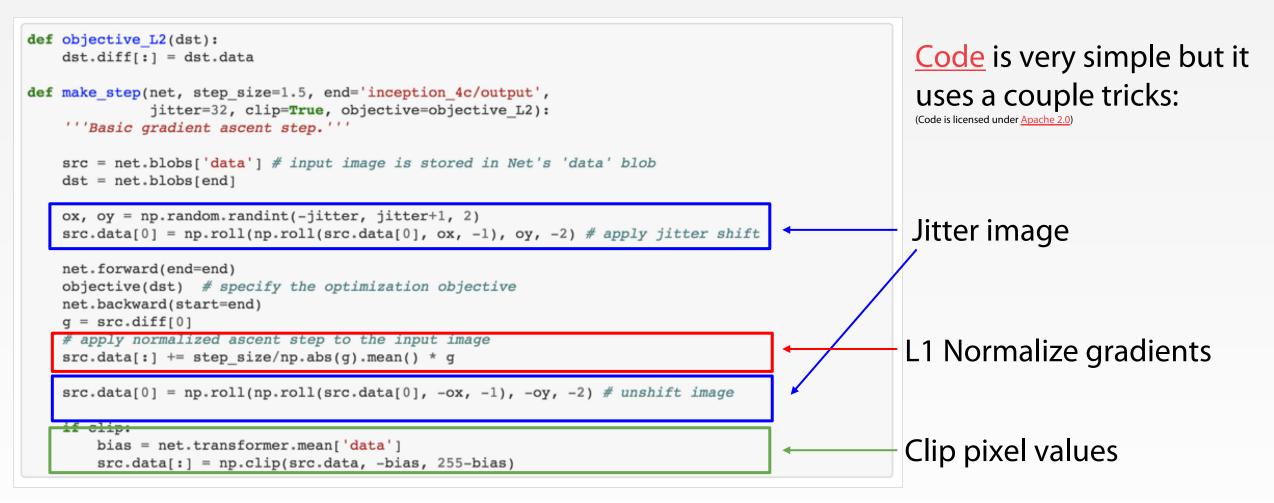






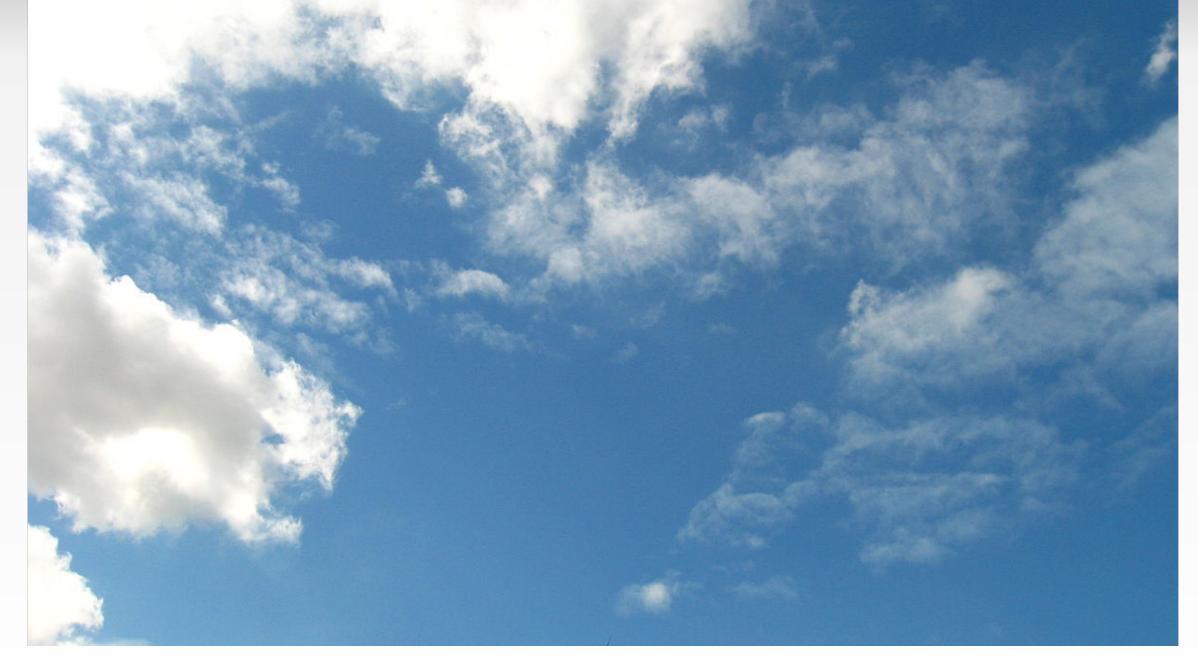






Also uses multiscale processing for a fractal effect (not shown)





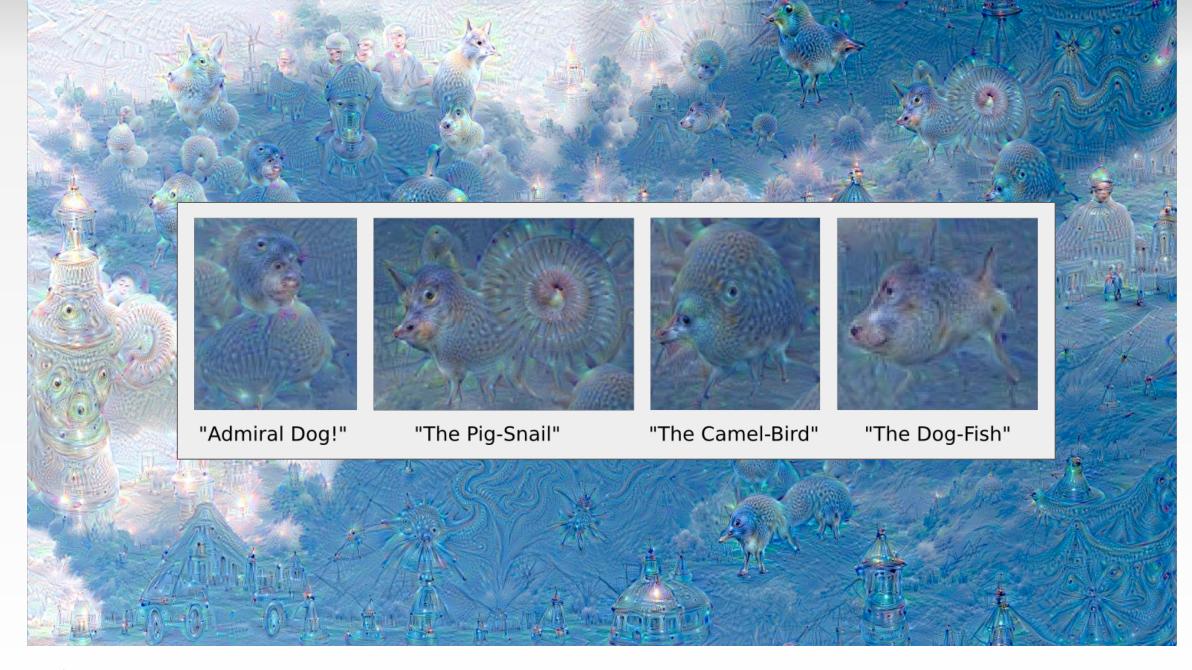








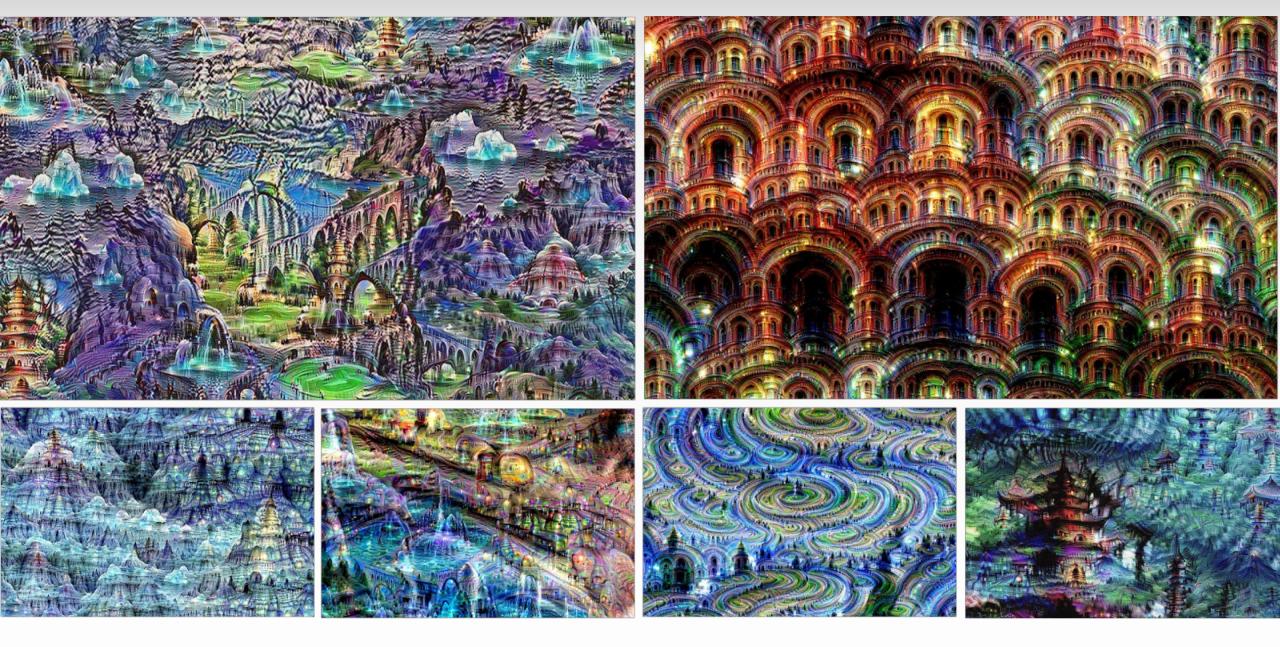














FEATURE INVERSION

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

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

$$\mathbf{x}^{*} = \underset{\mathbf{x} \in \mathbb{R}^{H \times W \times C}}{\operatorname{argmin}} \underbrace{\ell(\Phi(\mathbf{x}), \Phi_{0}) + \lambda \mathcal{R}(\mathbf{x})}_{\mathbf{x} \in \mathbb{R}^{H \times W \times C}} \xrightarrow{\mathsf{Given feature}}_{\substack{\text{vector}}} \operatorname{Features of new}}_{\substack{\text{image}}}$$

$$\frac{\ell(\Phi(\mathbf{x}), \Phi_{0}) = \|\Phi(\mathbf{x}) - \Phi_{0}\|^{2}}{\mathcal{R}_{V^{\beta}}(\mathbf{x}) = \sum_{i,j} \left((x_{i,j+1} - x_{ij})^{2} + (x_{i+1,j} - x_{ij})^{2} \right)^{\frac{\beta}{2}}}_{\stackrel{\text{Total Variation regularizer}}}$$
Mahendran and Vedaldi, "Understanding Deep Image Representations by Inverting Them", CVPR 2015



FEATURE INVERSION

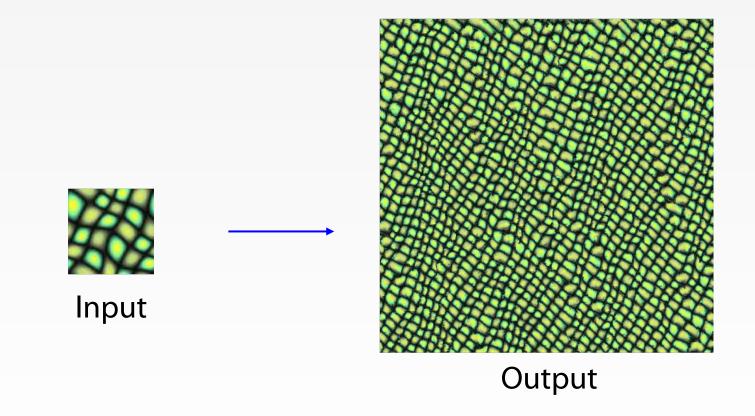
Reconstructing from different layers of VGG-16 relu2_2 relu3_3 relu4_3 relu5_1 relu5_3 y

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.



TEXTURE SYNTHESIS

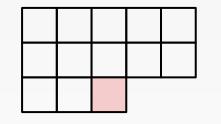
Given a sample patch of some texture, can we generate a bigger image of the same texture?

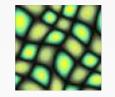


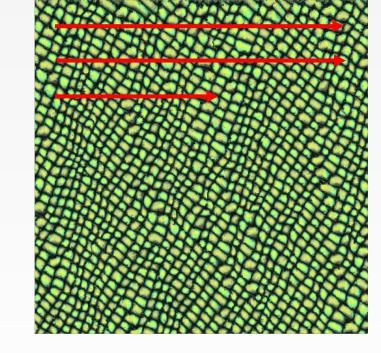


TEXTURE SYNTHESIS: NEAREST NEIGHBOR

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



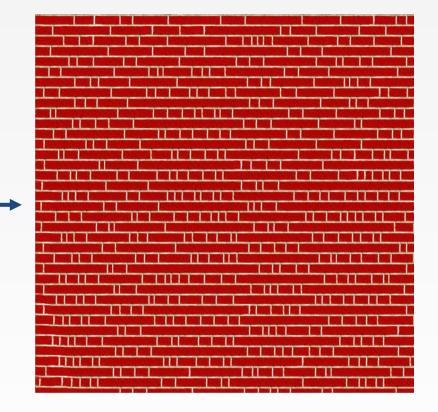




Wei and Levoy, "Fast Texture Synthesis using Tree-structured Vector Quantization", SIGGRAPH 2000 Efros and Leung, "Texture Synthesis by Non-parametric Sampling", ICCV 1999



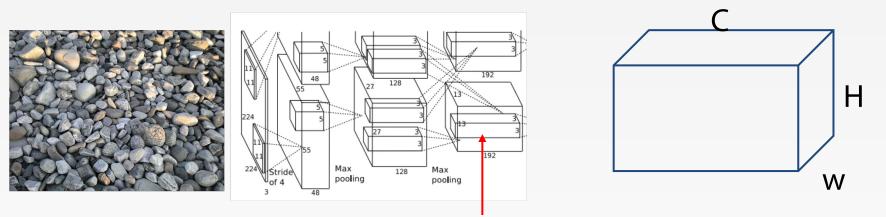
TEXTURE SYNTHESIS: NEAREST NEIGHBOR



RETORIACIO ICTALCOORD (ISELLAC, UNS OF LEW ACDE) it ndateears coune Tring rooms," as Heft he fast nd it. ars dat notars ortseas ribed it last n# hest bedian A1. I econicalHomd it h Al. Heft ars of as da Lewindailf l lian Al Ths," as Lewing questies last aticarsticall. H is dian Al last fal counda Lew; at "this dailyears d ily edianicall. Hoorewing rooms," as House De fale f De und itical councestscribed 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 MHe lelft a Lést fast ngine làuuesticars Hef nd it rip?" TrHouself, a ringind itsonestid it a ring que astical cois ore years of Moung fall. He ribof Mouse ere years ofanda Tripp?" That hedian Al Lest fasee yea nda Tripp?' Iolitical comedian Alét he few se ring que olitical cone re years of the storears ofas 1 Frat nica L ras Lew se lest a rime l He fas quest nging of, at beou

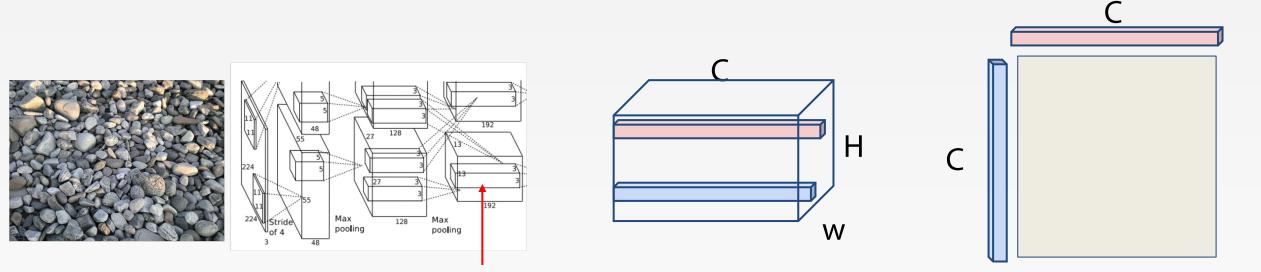
the self Lusinustarf the gittics in comparisons are ripp? " top? classif nate training ib stiff reservices I 2" lest coeff broad damas I bt i Lie wir res ong 1 Hauwellie auss II. HHieral conful courtas: "asterate herrs" agous date to sure st. Firs our reing lag ica ¹ by out ressons in the work of the data set of the second set o Hornestan zellevesiller at Lene Leders duerans et leder deve Frist rine if t des carsof he sucandersans divine vat red of Hospie at. n^{e,} jung dia mwa quanchi m_{any s}eal baol Houyeas Lectrifani l'offisiⁿ ^{a C}abuyenties," dea Lémes fa cl^{atthúch} ca _bbheith hy sincus us het je group wat at 1 as noodiwind wore qual at ming ting a lalas "doogustwy root ran His sour a work the rates Hes l ^{e la}tási gyneswétlitt G¹ isrómoroou ftasvtla lararse rik deloi tasvi^{at} inne Lós, 'en sv' gigodriwn⁵⁰⁰⁵ glaicaH: He rit faswiuae sticiff providences La la denoise sex sexwart of the same sex of the sex of the sex of the sex of the later sex of the se st avaleary qual decisarithou ars Deid fourwy. As incommon riprative, "ere Ledisficial Holta poiecial Holtas" " utilouse oreatie proue thiste radamenta Eekor: of jest fast masms with second el opravor stamost ar fairlates Fough lasts yest uproved taken Hodin, vrsilsest loon. That's ast, of nist's seade ing't for sist carouroun afters coordiaire frairit dies capoos Heeding is night welling the Hdafted tobulity "af," as the puzzaits rilled Fham Thaines de as Thasorseviny ag or ceedia arses the out send we need ≁ous."1 an liting weft and and and provide the Lite. A Leadars and an it arrives dues quiptus drages of acaeas, on Prover Alzhanis quoilasi august a vites attiraçed it diffet a as he 'a rodz fizica lueat ft". ripudamit rous II masva; Hingi Aeont ingirs rribiloit l as des to "" the offs or y y is neft z'ftel wis est fairs of aticanalized dars calle its ripte y norraid Alkland tadaiectert autielesses ses t an italis roomen fancet as farses a frear sites ingingar nos dais innuroués toures y and rt at mountmous si tilda Frs is oear, i Ar more iburkans , sHe-?" Ichi's intriumal (rimt intrionigal liase ficatig stit rwihida;" zonis//at ingra int/rob. H Aoforinaus, dailof haqi stmoodia (a garmay Goodia, para) anni anconganzas raca socia neu a socia raca socia socia socia socia socia so neu a garmay Goodia, para (a neu fa nei trina tabajobandori di jasen la icitskie taas ouviss (is clase, at a qu neu Foren gu Hinhdaris laga "inte sidati at h Alexadal citic," nazetta your goor vstgisovijicu que a vastadogi reas (tod it 1, Th Files rog ry foriols used thigrais' af dailering lither : yllaning until g us Hutsest it Lewis fa af dwig ust ry orally coras se as, Al ard ought stontinged in jo est nilumalase litener yoo.q. idars of ous ex factoarstnious costindings incondusti ngireolail;äll se écolouse sou oft lous fratrenerrs cuors, di a cala ha reards, vzarffahine tilynë sarshise ungitsoione la orray ndew: De. Huss vs Ho vixsy is get valatst te vise vers voeus stizzooi fdais las q stigif of a De zs That que h, fal ines ars is why ag fair of He is curring our ag quadilane a confar Teders Hass dred these fars as fare as her fait hour (as Linghars , dreng s la ie ig iotiedag mees ineus geetid. A Mousin st leg zóloouti dar goingrez ft 14 rediaas tabitinfaonáticzs est a "A b ar jisvordiatnika i vénouing z ahore nine Ley rig : fasa fritaine a swis ingutataielft': tests of rvit: a ann Linguitis dAl cie dail: Thaous f wipit a g oore uvaib striknof Maonibybringt I as a way w Hagiself and isat ises of a asticcofaniting 1 se. heor e Tarrow ring ne anotai (tento anotai) (tento anotai) (tento) (tento e suls usignin daine vefaving nostise Lones, sixone viewella lainomiza Eorinestieng dir netreses Fist la frievet. Lise Indainet ^b qomilisone micro v tabung mettas pores, is on occupine v diasoft is og ver, da r nihners (pog szora, daiw, isvis d Fardang, 181); ^a a falle, we snel considering is yeven a Pricing of Pricing (Price of Price of Mange and the state of the stat 25 questise ins a struct records outles are all and a strue as you ridering busiless theorem, Artes is a farme record ridering a structure of the structure of ta usppi hu H_1000" f Fat caouquties, " seque quae ifis lest avat nes. Edz Diess gecoal indas guserdi itiauat leus dastof Ha que la se Honoraist, serina for roces rotts a Lesing ineachister re : destruo, itits Fines, asianta He site Lettore, ers et hop?





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

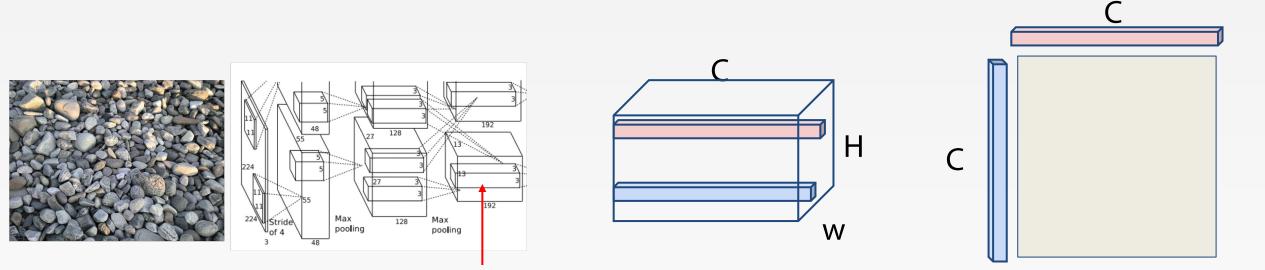




Each layer of CNN gives C x H x W tensor of features; H x W grid of C-dimensional vectors

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





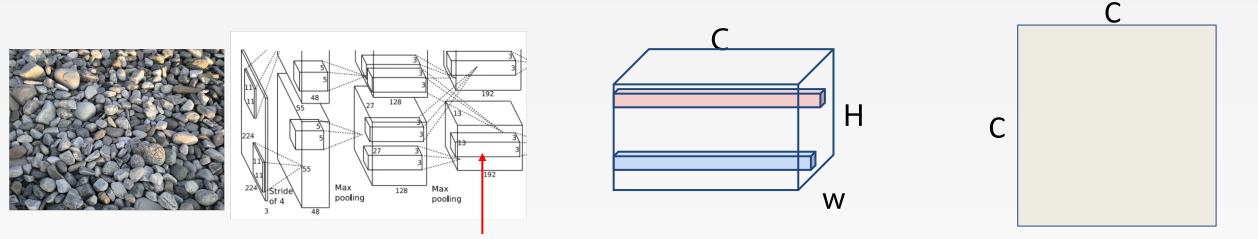
Each layer of CNN gives C x H x W tensor of features; H x W grid of C-dimensional vectors

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





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

Georgia

Efficient to compute; reshape features from

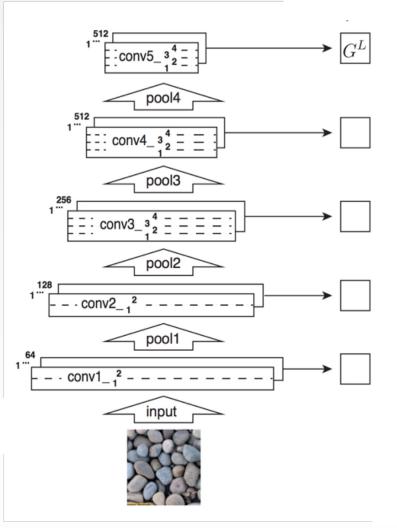
 $C \times H \times W$ to $=C \times HW$

```
then compute \mathbf{G} = \mathbf{F}\mathbf{F}^{\mathsf{T}}
```



- 1. Pretrain a CNN on ImageNet (VGG-19)
- 2. Run input texture forward through CNN, record activations on every layer; layer i gives feature map of shape $C_i \times H_i \times W_i$
- 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_i × C_i)

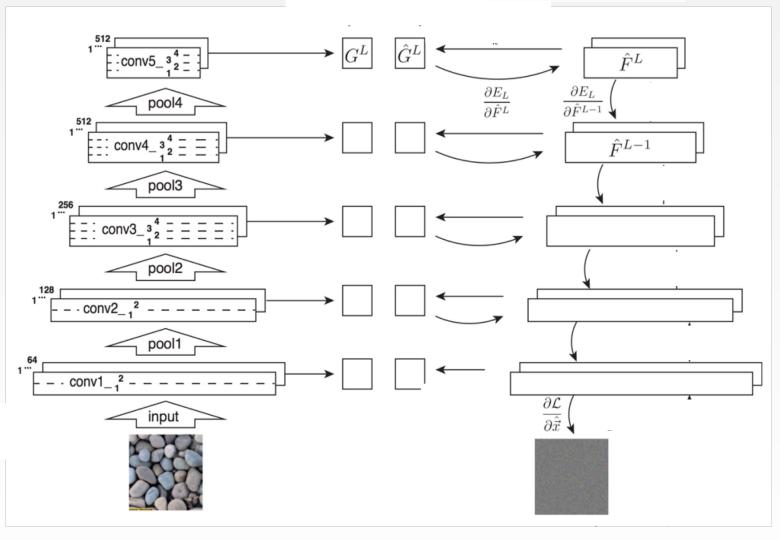




- 1. Pretrain a CNN on ImageNet (VGG-19)
- 2. Run input texture forward through CNN, record activations on every layer; layer i gives feature map of shape $C_i \times H_i \times W_i$
- 3. At each layer compute the *Gram matrix* giving outer product of features:

$$G_{ij}^{l} = \sum_{k} F_{ik}^{l} F_{jk}^{l}$$
 (shape C_i × C_i)

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

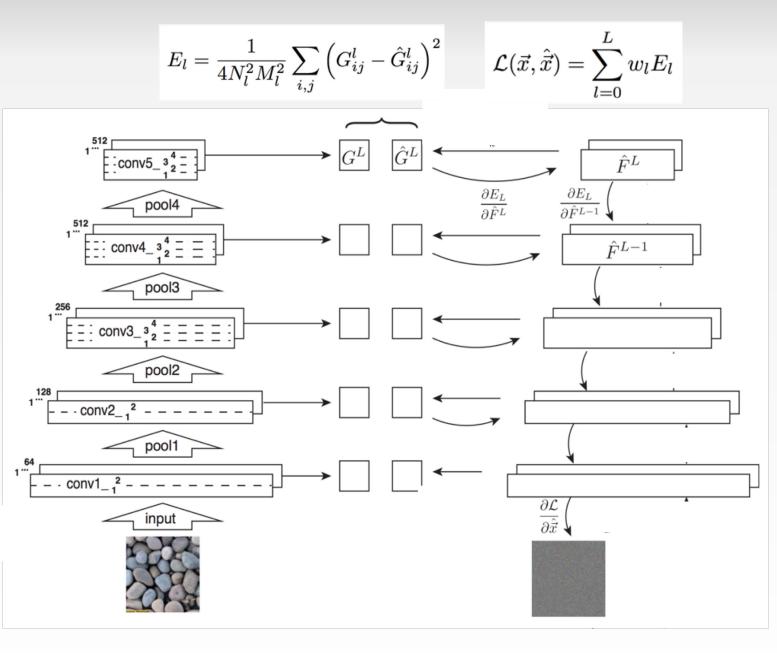




- 1. Pretrain a CNN on ImageNet (VGG-19)
- 2. Run input texture forward through CNN, record activations on every layer; layer i gives feature map of shape $C_i \times H_i \times W_i$
- 3. At each layer compute the *Gram matrix* giving outer product of features:

$$G_{ij}^{l} = \sum_{k} F_{ik}^{l} F_{jk}^{l}$$
 (shape C_i × C_i)

- 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

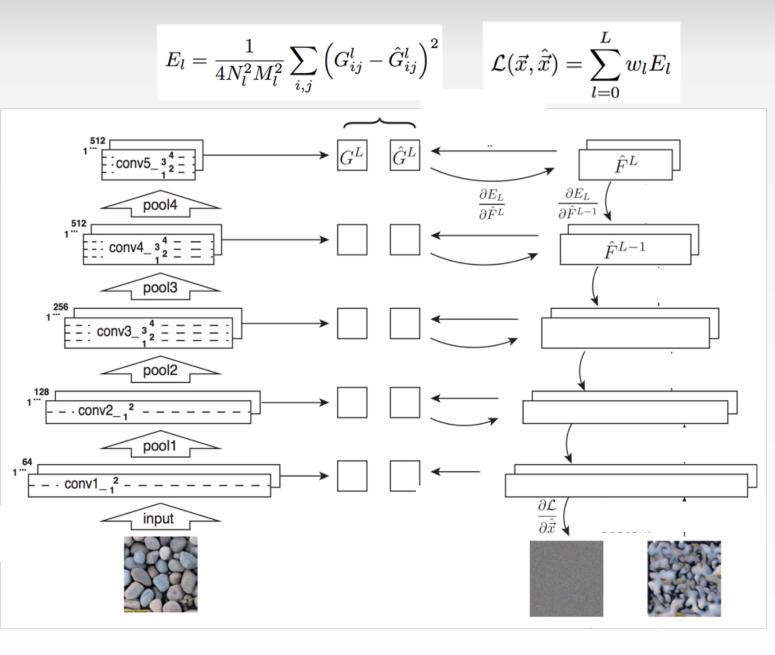




- 1. Pretrain a CNN on ImageNet (VGG-19)
- 2. Run input texture forward through CNN, record activations on every layer; layer i gives feature map of shape $C_i \times H_i \times W_i$
- 3. At each layer compute the *Gram matrix* giving outer product of features:

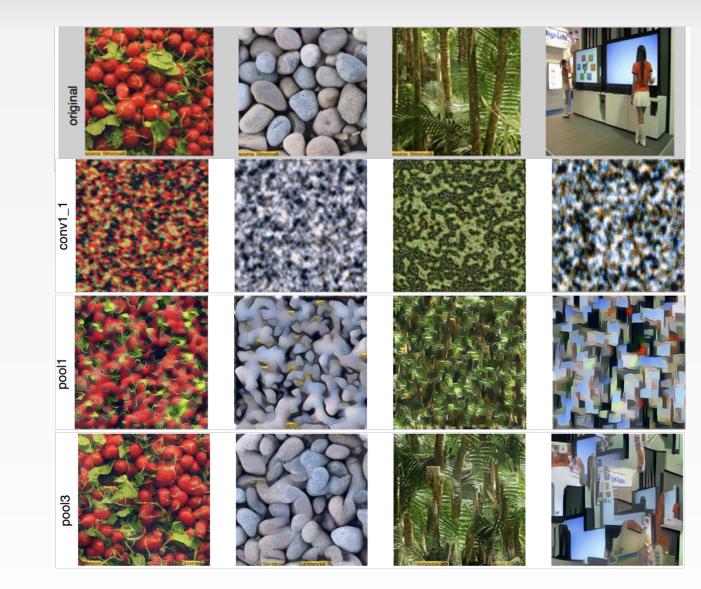
$$G_{ij}^{l} = \sum_{k} F_{ik}^{l} F_{jk}^{l}$$
 (shape C_i × C_i)

- 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

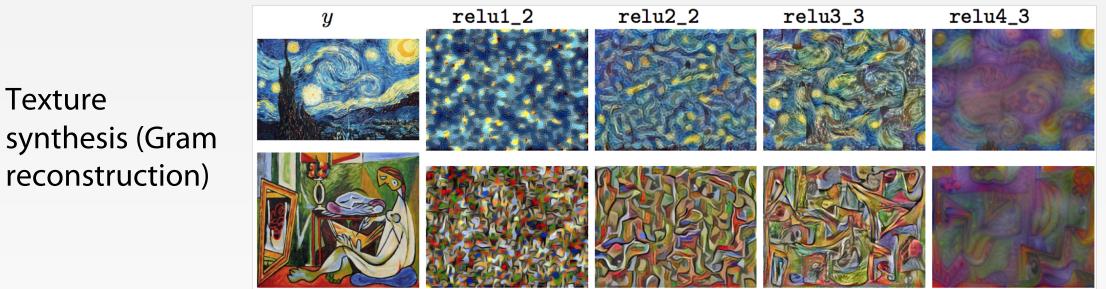




Reconstructing texture from higher layers recovers larger features from the input texture

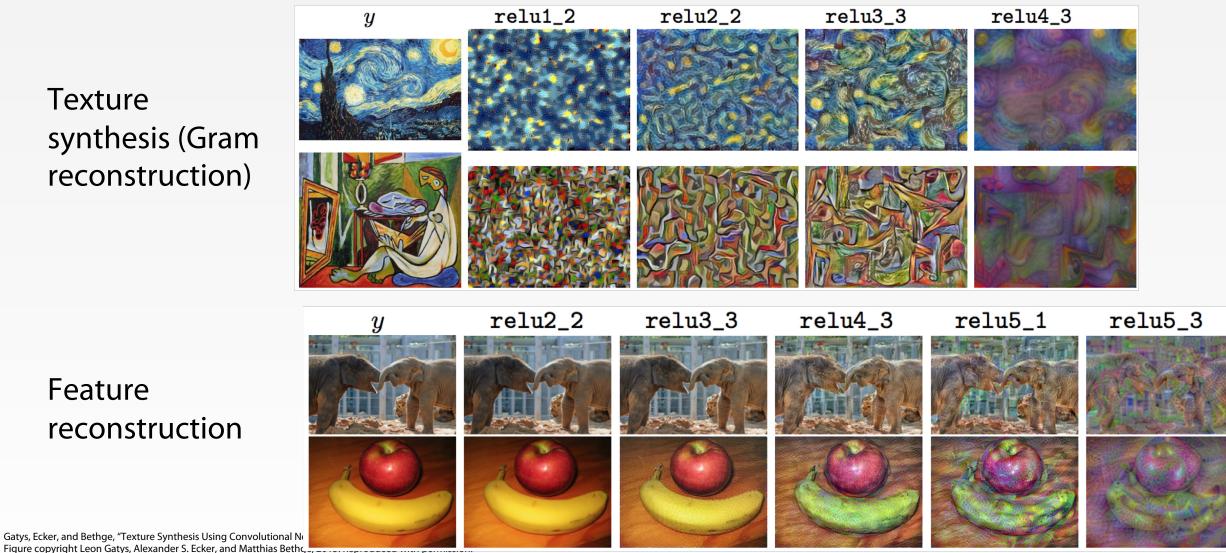






Gatys, Ecker, and Bethge, "Texture Synthesis Using Convolutional Neural Networks", NIPS 2015 Figure copyright Leon Gatys, Alexander S. Ecker, and Matthias Bethge, 2015. Reproduced with permission.







Content Image



╋

Style Image



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



Content Image



╋

Style Image



Style Transfer!



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



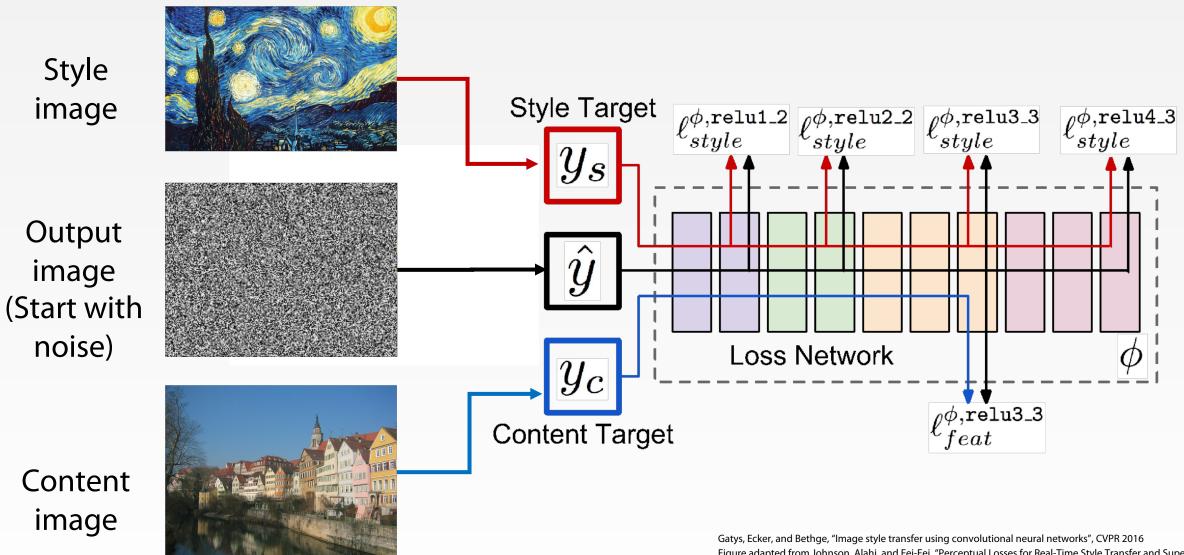


Figure adapted from Johnson, Alahi, and Fei-Fei, "Perceptual Losses for Real-Time Style Transfer and Super-Resolution", ECCV 2016. Copyright Springer, 2016. Reproduced for educational purposes.



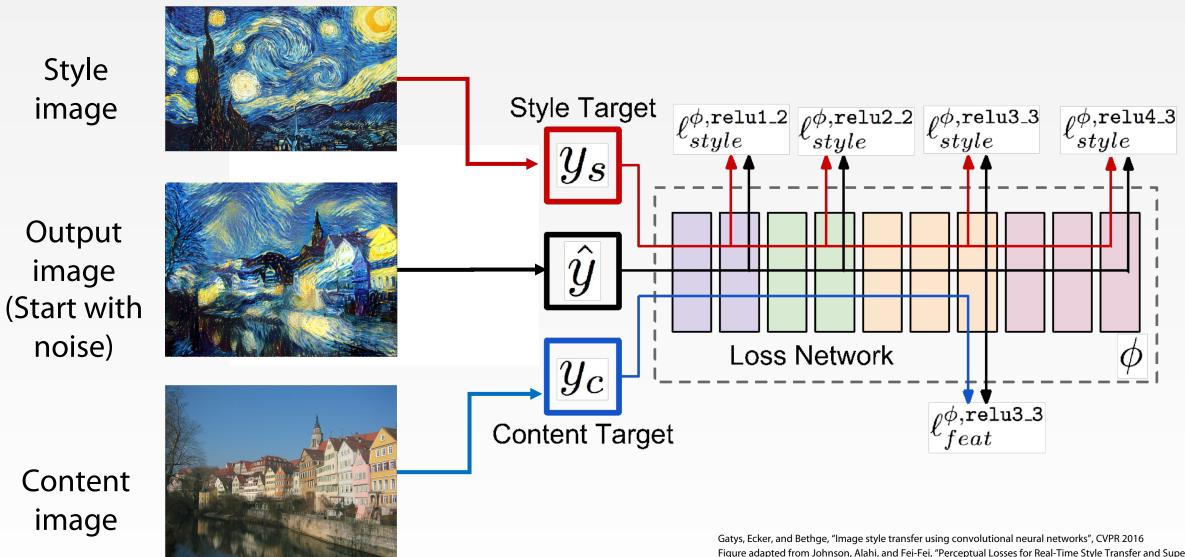
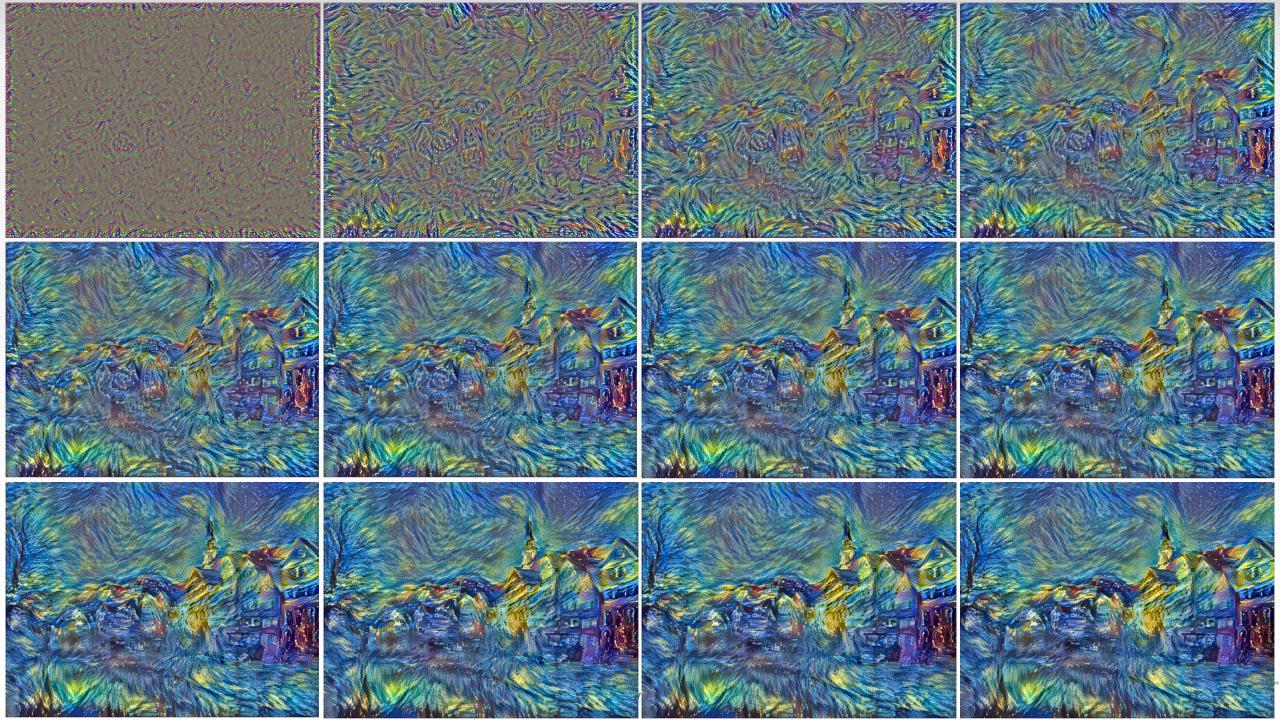
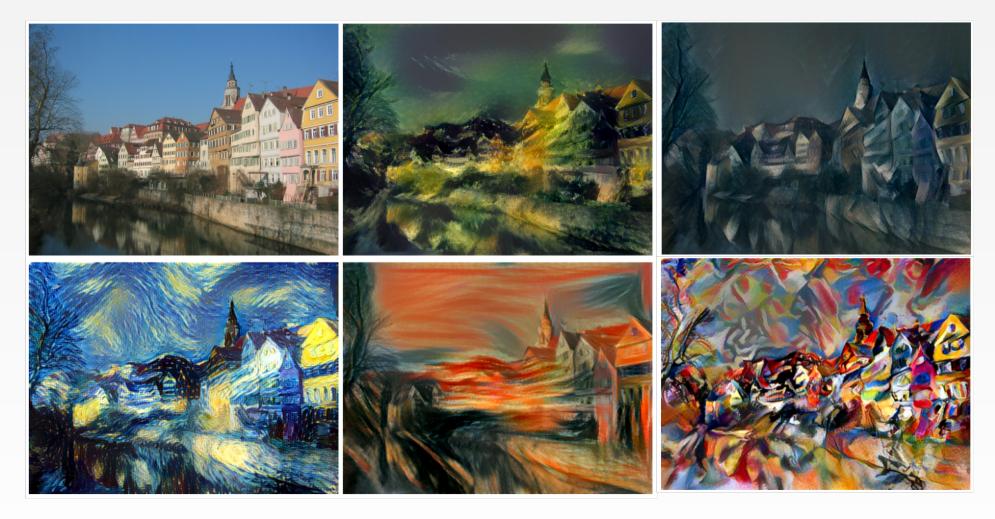


Figure adapted from Johnson, Alahi, and Fei-Fei, "Perceptual Losses for Real-Time Style Transfer and Super-Resolution", ECCV 2016. Copyright Springer, 2016. Reproduced for educational purposes.





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



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





More weight to content loss

More weight to style loss



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



Larger style image

Smaller style image

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



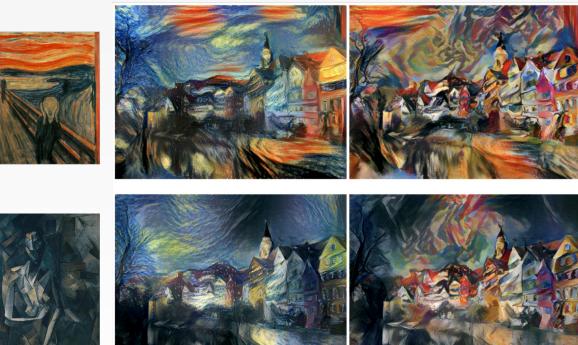


NEURAL STYLE TRANSFER: MULTIPLE STYLE IMAGES

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





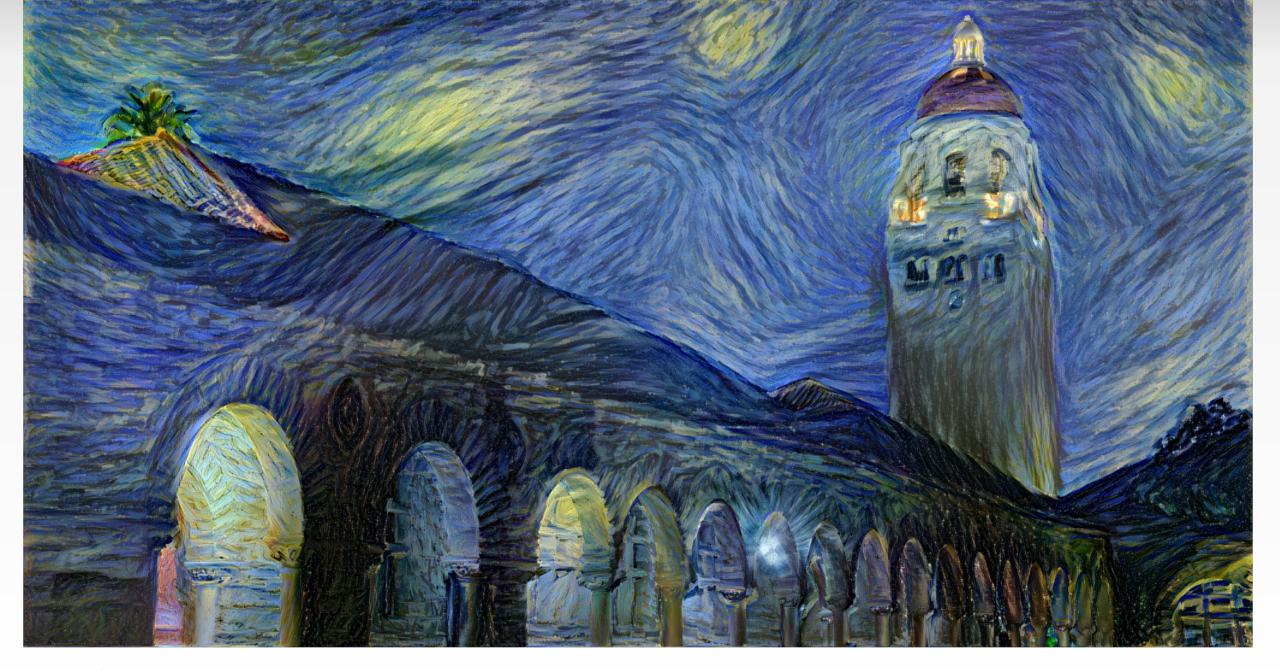


Gatys, Ecker, and Bethge, "Image style transfer using convolutional neural networks", CVPR 2016 Figure copyright Justin Johnson, 2015.





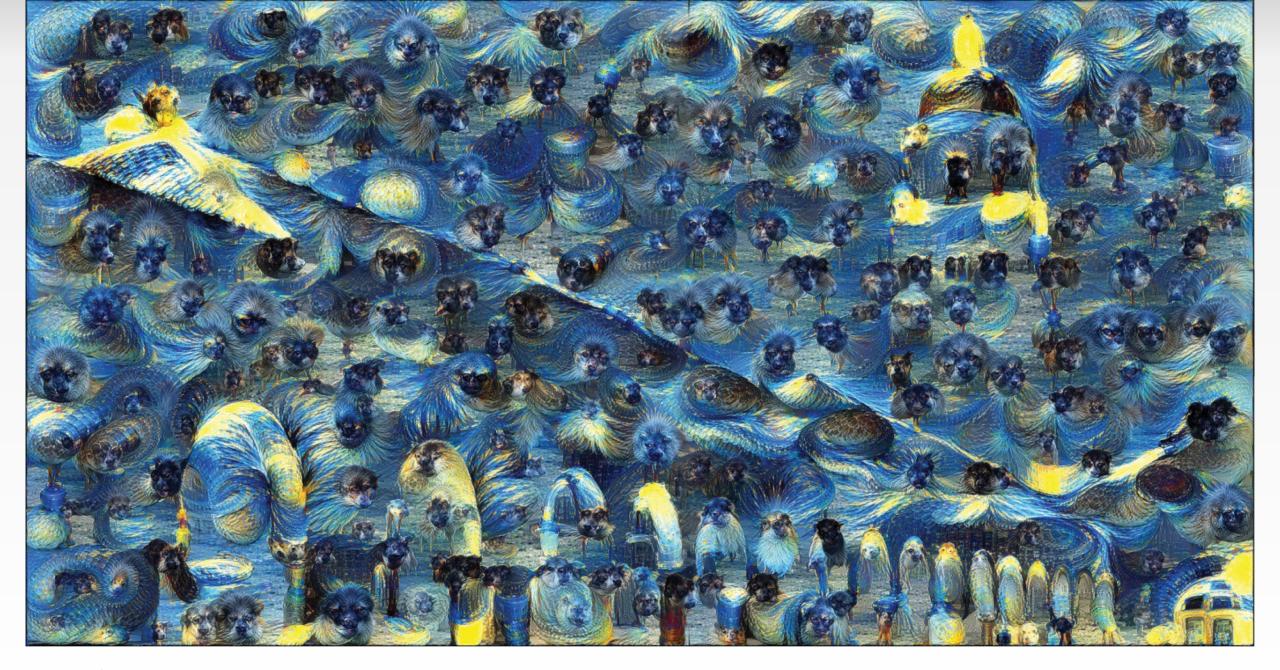














NEURAL STYLE TRANSFER

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



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!



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

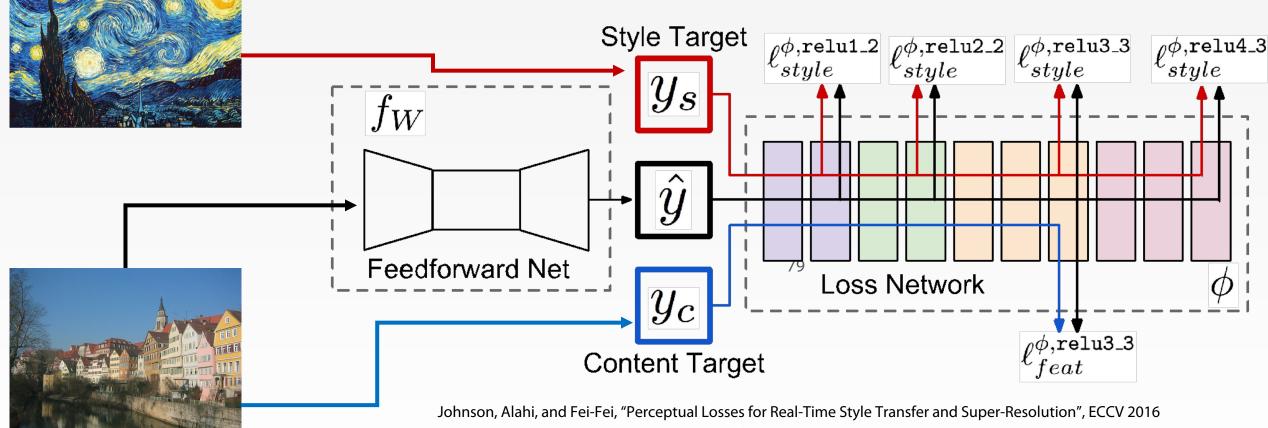
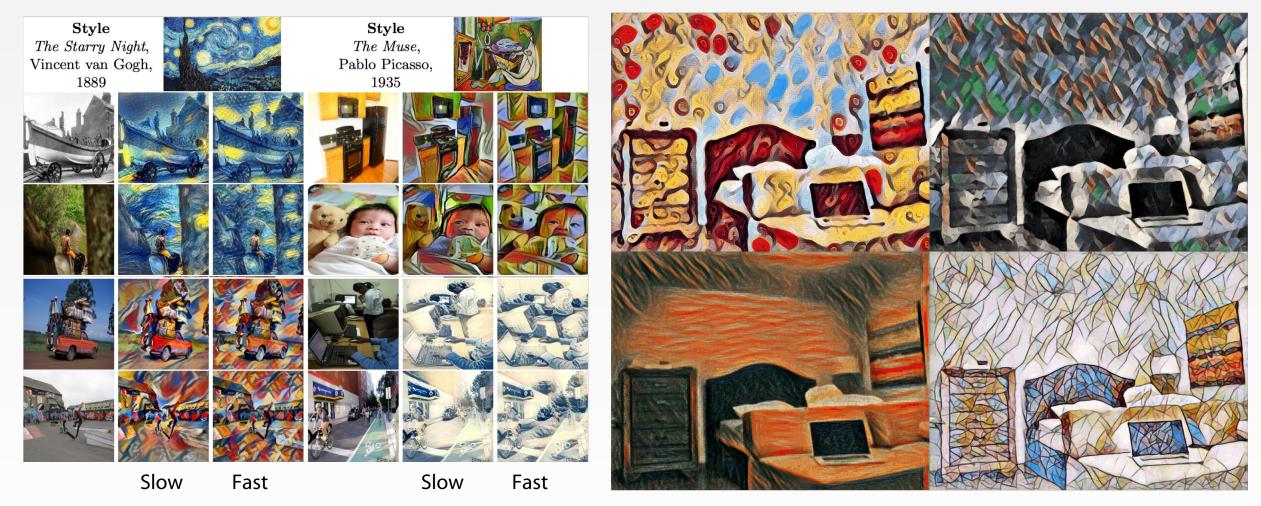


Figure copyright Springer, 2016. Reproduced for educational purposes.



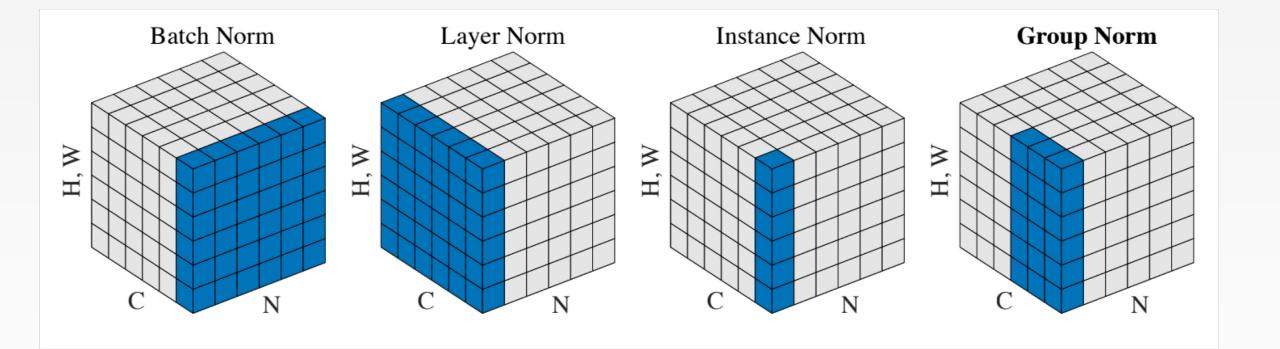
FAST STYLE TRANSFER



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



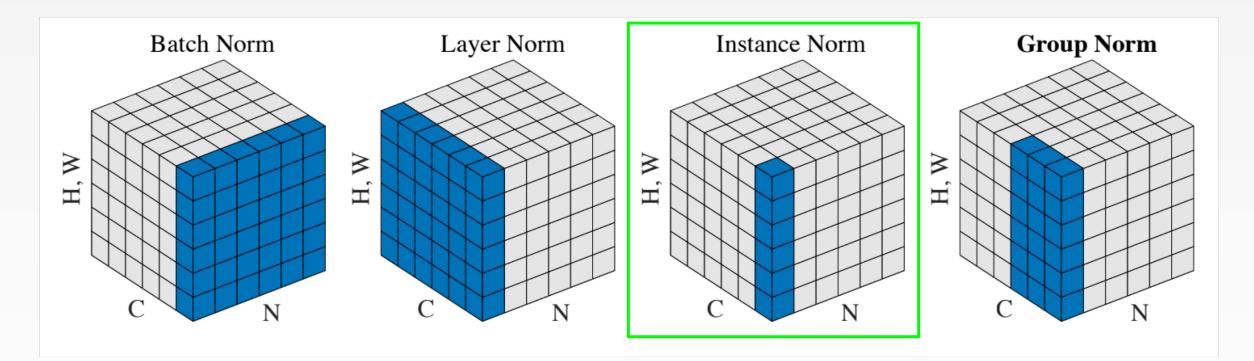
REMEMBER NORMALIZATION METHODS?





REMEMBER NORMALIZATION METHODS?

Instance Normalization was developed for style transfer!





FAST STYLE TRANSFER



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 permission.



GT 8803 // FALL 2018

ONE NETWORK, MANY STYLES

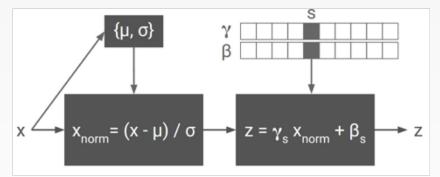


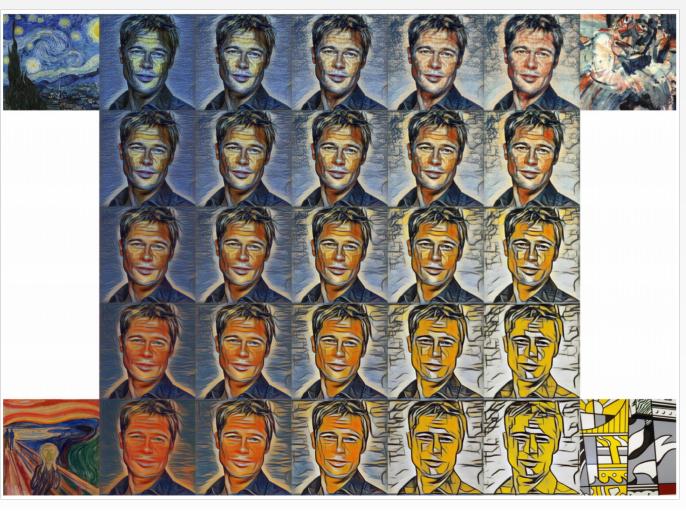
Dumoulin, Shlens, and Kudlur, "A Learned Representation for Artistic Style", ICLR 2017. Figure copyright Vincent Dumoulin, Jonathon Shlens, and Manjunath Kudlur, 2016; reproduced with permission.



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





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. Single network can blend styles after training



GT 8803 // FALL 2018

SUMMARY

- Many methods for understanding CNN representations
- Activation-based Methods: Nearest neighbors, Dimensionality reduction, maximal patches, occlusion
- **Gradient-based Methods**: Saliency maps, class visualization, fooling images, feature inversion
- Fun: DeepDream, Style Transfer.



NEXT TIME: (DEEP) REINFORCEMENT LEARNING



GT 8803 // FALL 2018