Lecture 8:
Visualizing and Understanding

Administrative

- Assignment 1 grades will be released over the weekend
- Assignment 2 due Friday April 28th, 11:59pm
    - Assignment 3 has extra credit opportunities
- Project proposal due April 24th

- Will release model midterm question papers before recitation

- Zoom recordings are always available on Canvas

Today: What's going on inside ConvNets?

This image is CC0 public domain

Input Image:
3 x 224 x 224
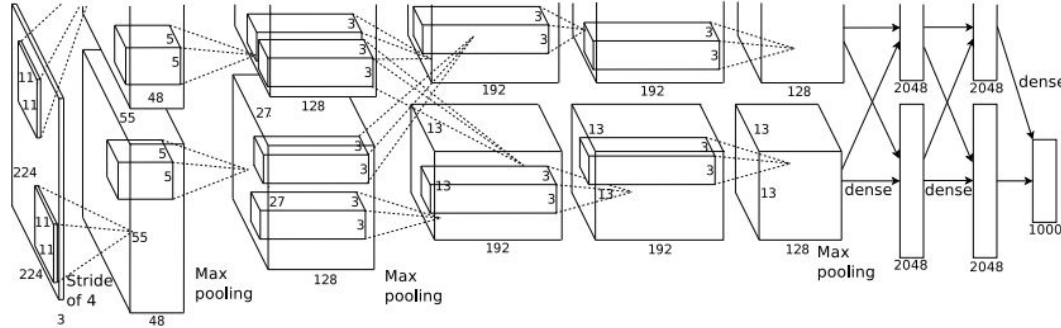
Class Scores:
1000 numbers

What are the intermediate features looking for?

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

Today's agenda
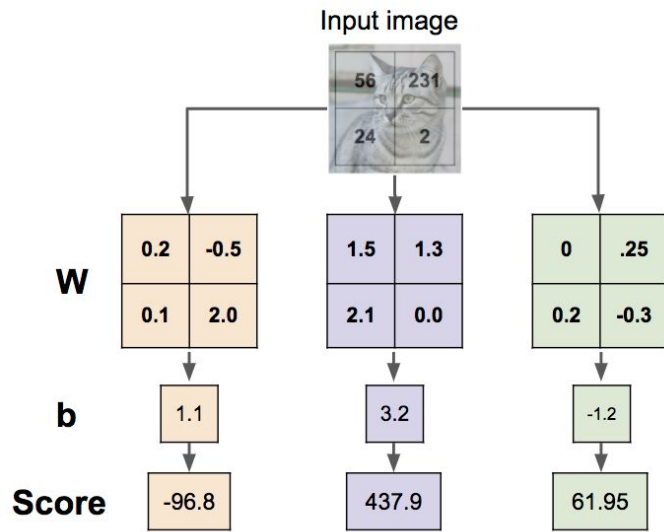
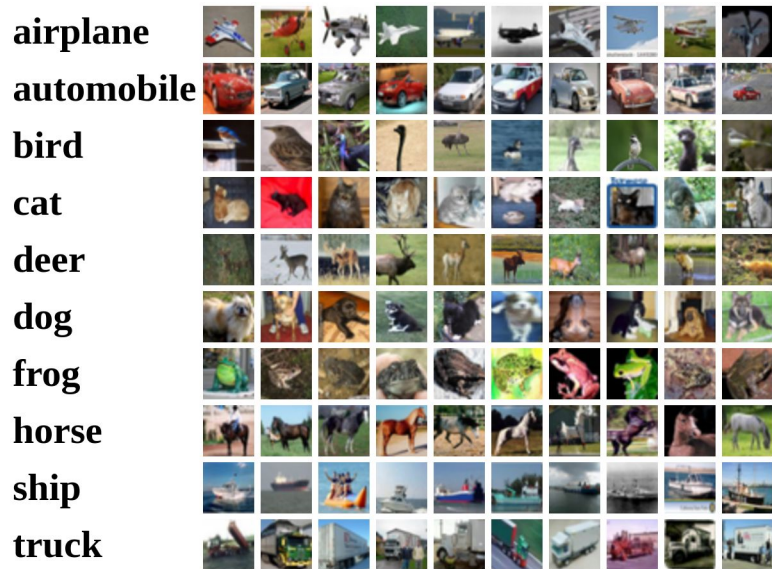**Visualizing what models have learned:**
- Visualizing filters
- Visualizing final layer features
- Visualizing activations
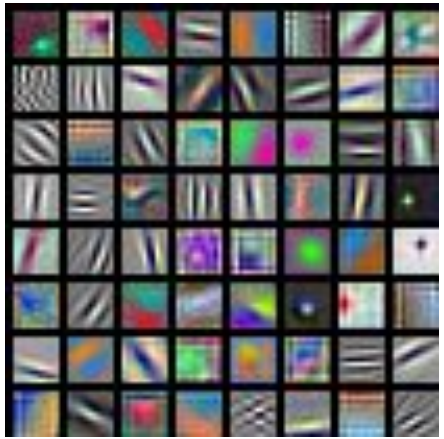
Understanding input pixels
- Identifying important pixels
- Saliency via backprop
- Guided backprop to generate images
- Gradient ascent to visualize features

Adversarial perturbations

Today's agenda

Visualizing what models have learned:
- Visualizing filters
- Visualizing final layer features
- Visualizing activations

Understanding input pixels
- Identifying important pixels
- Saliency via backprop
- Guided backprop to generate images
- Gradient ascent to visualize features
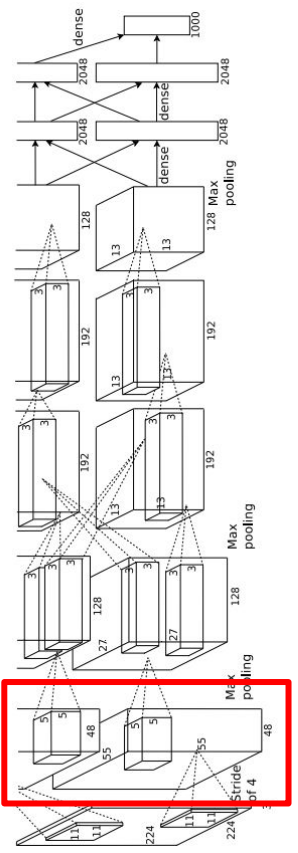
Adversarial perturbations

# Interpreting a Linear Classifier: <u>Visual Viewpoint</u>

airplane
automobile
bird
cat
deer
dog
frog
horse
ship
truck

Input image

| 56 | 231 |
| 24 | 2 |

W

| 0.2 | -0.5 |
| 0.1 | 2.0 |

| 1.5 | 1.3 |
| 2.1 | 0.0 |

| 0 | .25 |
| 0.2 | -0.3 |

b

1.1

3.2

-1.2

Score

-96.8

437.9

61.95

plane   car   bird   cat   deer   dog   frog   horse   ship   truck

First Layer: Visualize Filters

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

First Layer: Visualize Filters
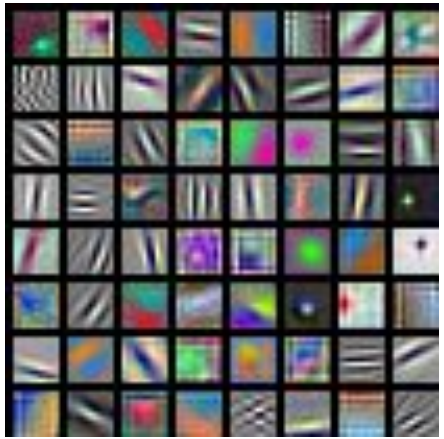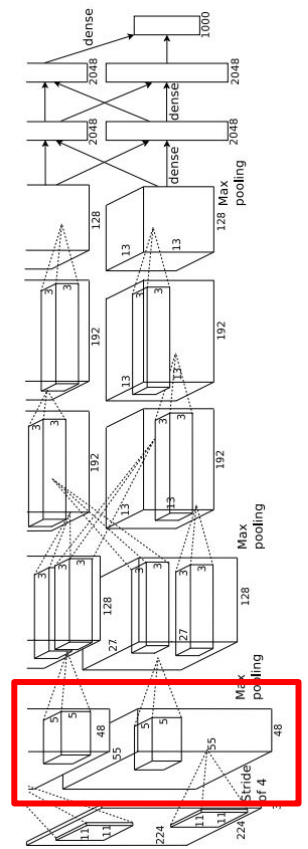

AlexNet:
64 x 3 x 11 x 11

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

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

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)

layer 1 weights

16 x 3 x 7 x 7

layer 2 weights

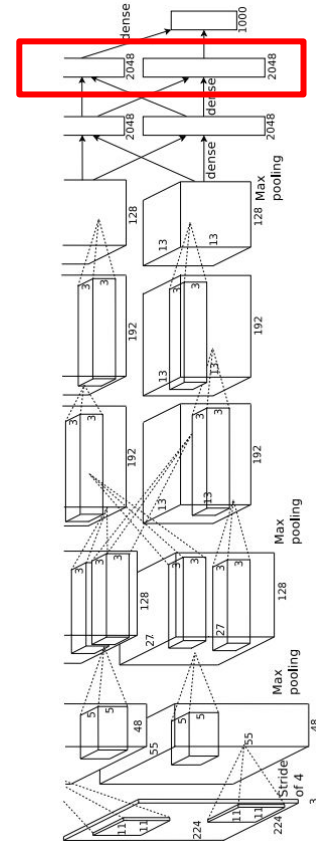20 x 16 x 7 x 7

layer 3 weights

20 x 20 x 7 x 7

Last Layer
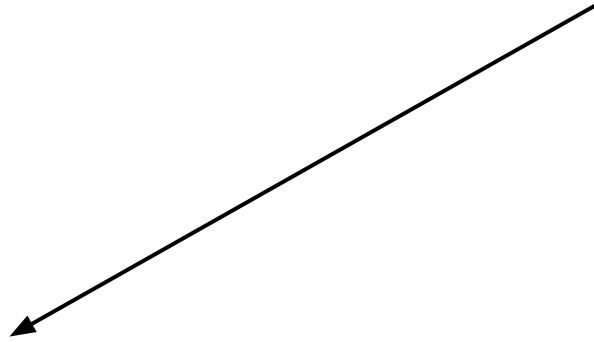
FC7 layer

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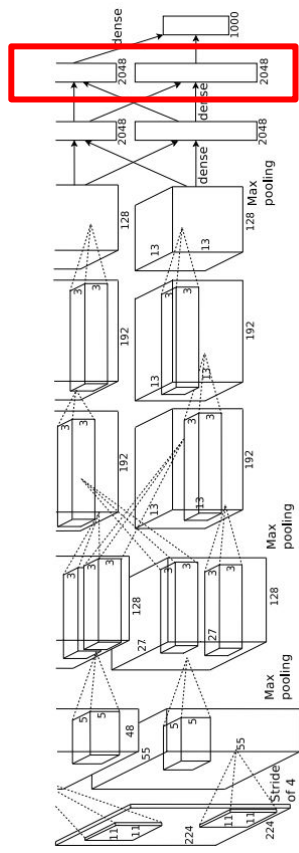(layer immediately before the classifier)

Run the network on many images, collect the
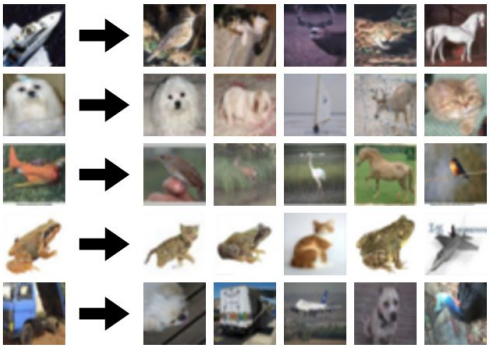feature vectors

Last Layer: Nearest Neighbors

4096-dim vector

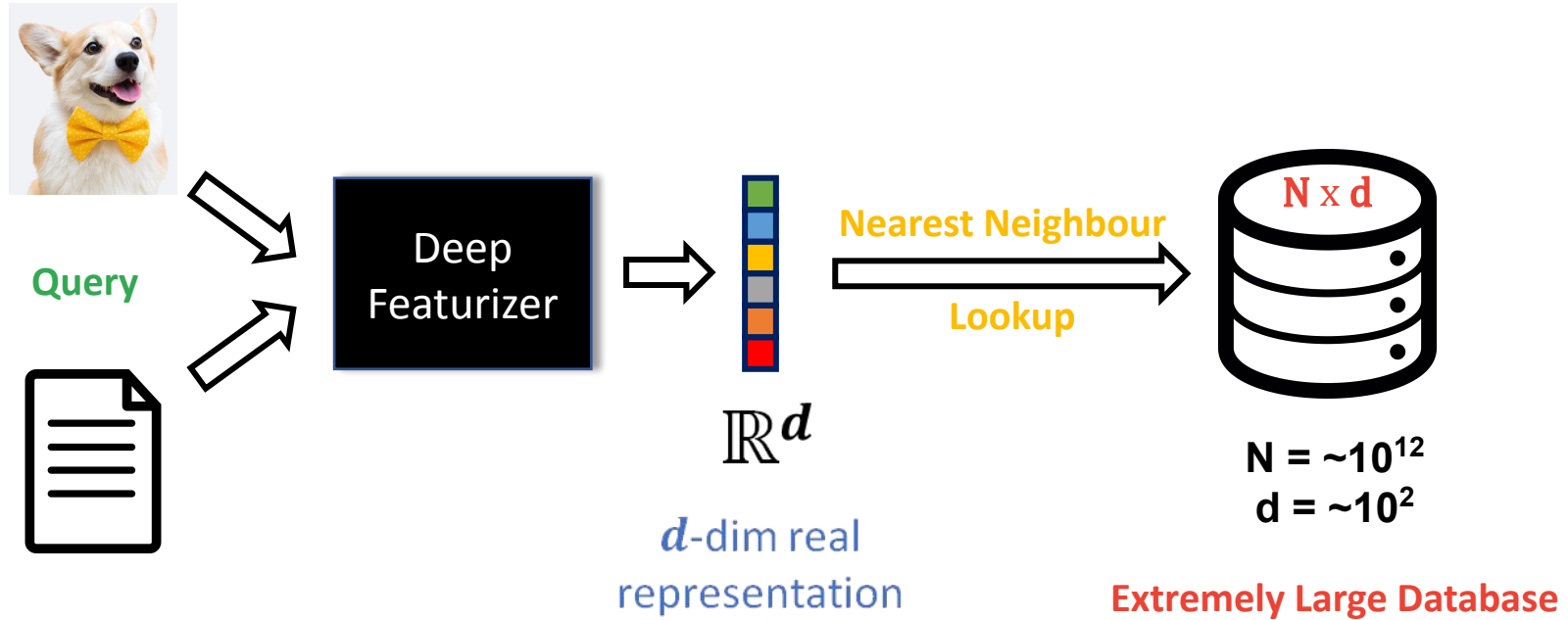Test image   L2 Nearest neighbors in <u>feature</u> space
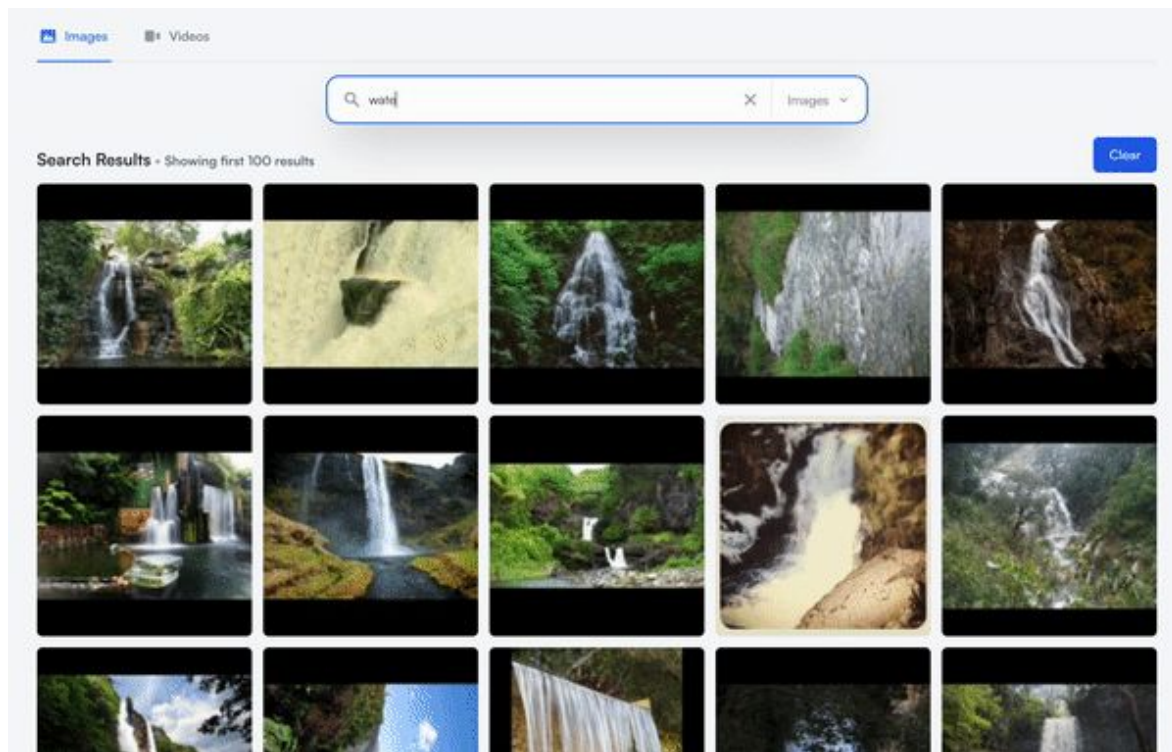
Recall: Nearest neighbors in <u>pixel</u> space

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

# Last Layer: Learned Metric for "Semantic" Search

Query

Deep Featurizer

$\mathbb{R}^{\boldsymbol{d}}$

$\boldsymbol{d}$-dim real representation

Nearest Neighbour Lookup

N x d

N = ~10^{12}
d = ~10^{2}

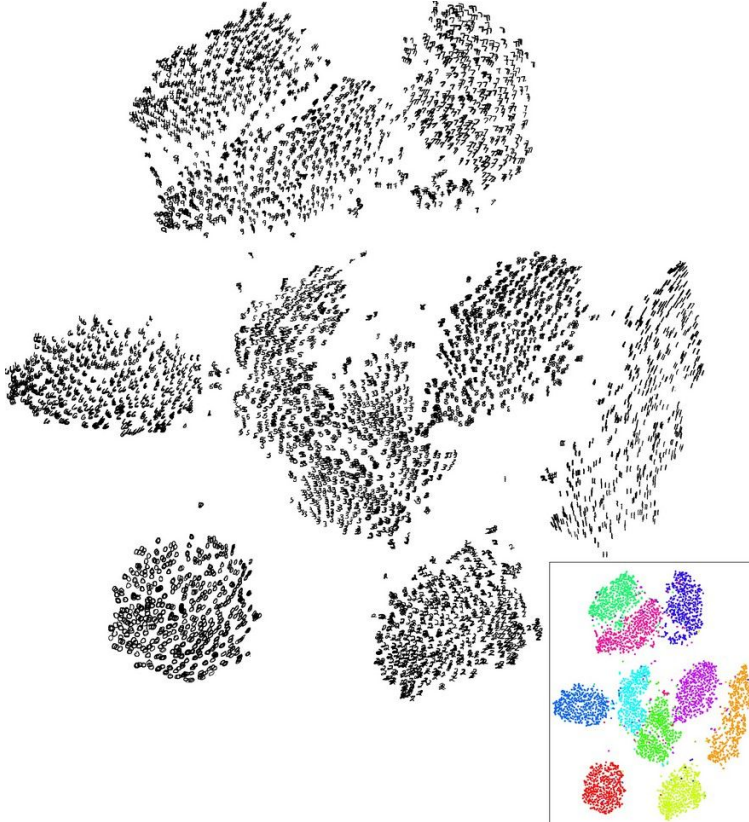**Extremely Large Database**

Last Layer: Modern Day Search

coactive.ai

Last Layer: Dimensionality Reduction

Visualize the "space" of FC7 feature vectors by reducing dimensionality of vectors from 4096 to 2 dimensions
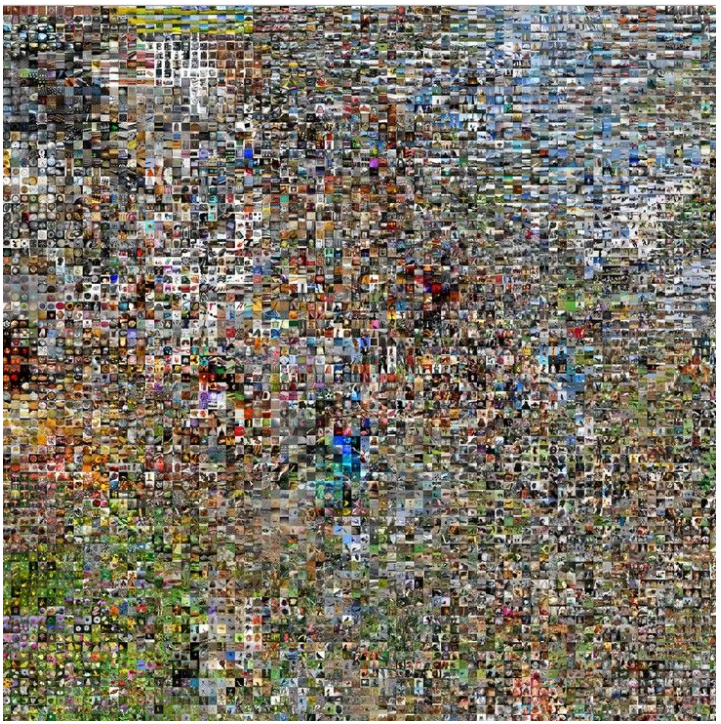
Simple algorithm: Principal Component Analysis (PCA)

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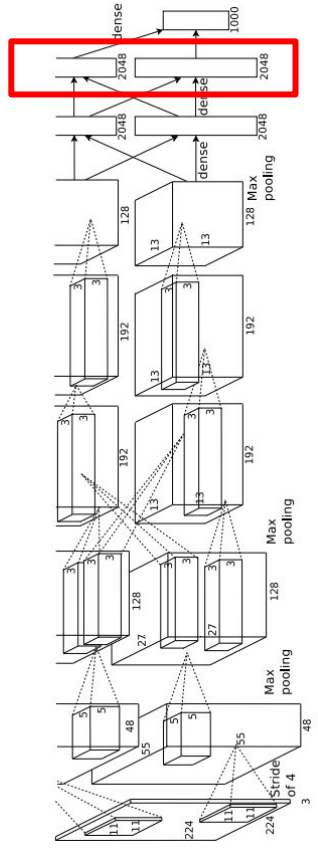
Van der Maaten and Hinton, "Visualizing Data using t-SNE", JMLR 2008
Figure copyright Laurens van der Maaten and Geoff Hinton, 2008. Reproduced with permission.

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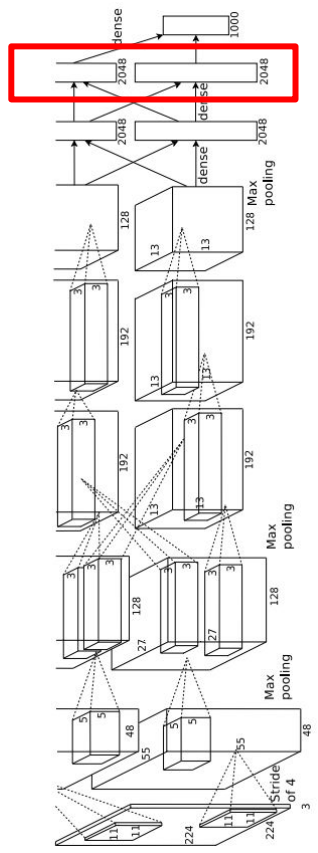
Van der Maaten and Hinton, "Visualizing Data using t-SNE", JMLR 2008
Krizhevsky et al, "ImageNet Classification with Deep Convolutional Neural Networks", NIPS 2012.
Figure reproduced with permission.

See high-resolution versions at
http://cs.stanford.edu/people/karpathy/cnnembed/

Visualizing Activations

conv5 feature map is
128x13x13; visualize
as 128 13x13
grayscale images
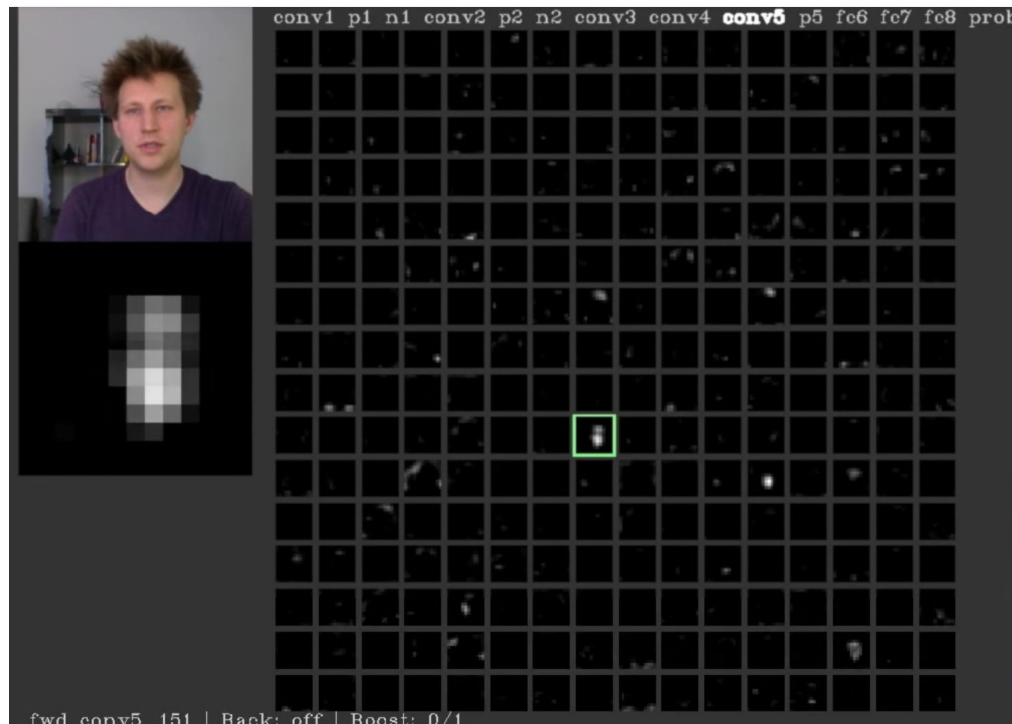
Today's agenda

Visualizing what models have learned:
- Visualizing filters
- Visualizing final layer features
- Visualizing activations

Understanding input pixels
- Identifying important pixels
- Saliency via backprop
- Guided backprop to generate images
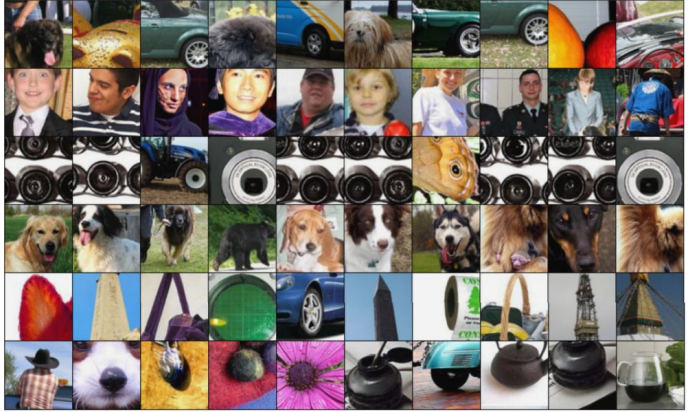- Gradient ascent to visualize features

Adversarial perturbations

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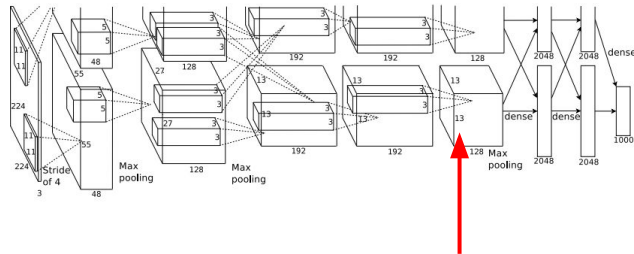
Pick a layer and a channel; e.g. conv5 is 128 x 13 x 13, pick channel 17/128

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.

Which pixels matter: Saliency via Occlusion

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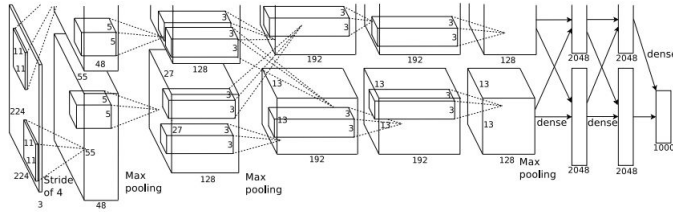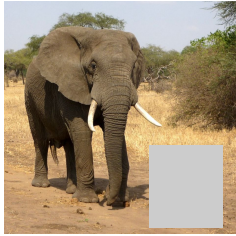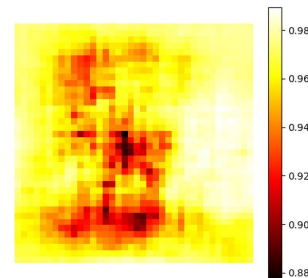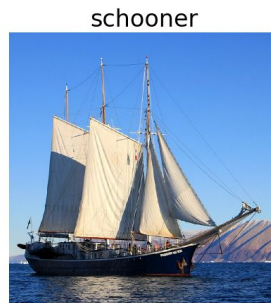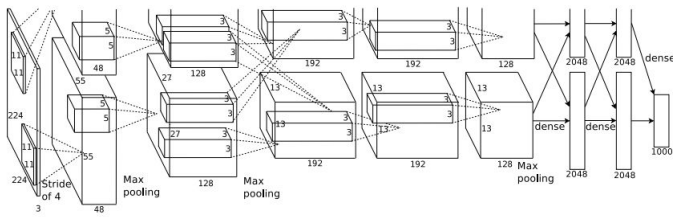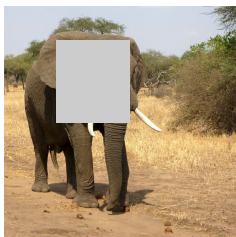

P(elephant) = 0.95
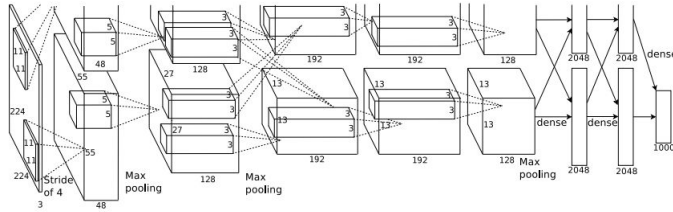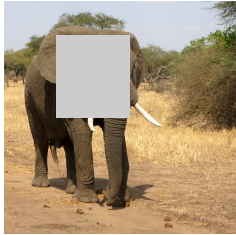
P(elephant) = 0.75

Zeiler and Fergus, "Visualizing and Understanding Convolutional Networks", ECCV 2014
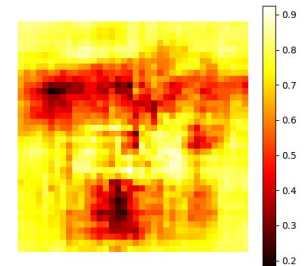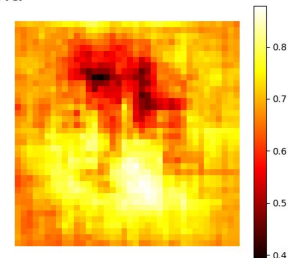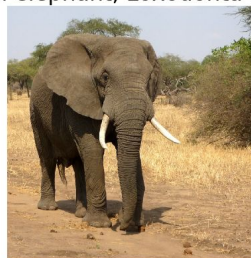
Which pixels matter: Saliency via Occlusion

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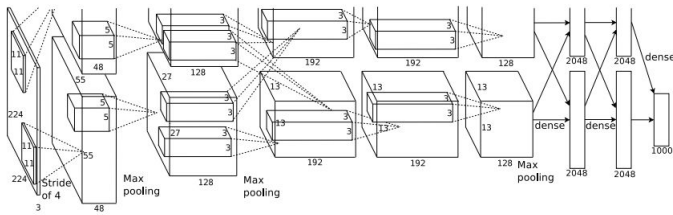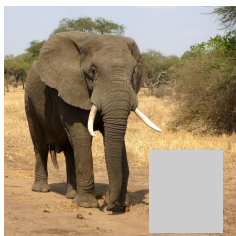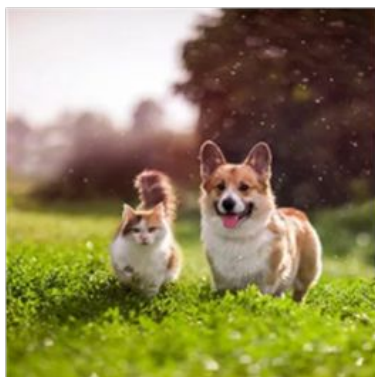

Zeiler and Fergus, "Visualizing and Understanding Convolutional Networks", ECCV 2014

Saliency via Occlusion: Shapley Values

P(corgi) = 0.99
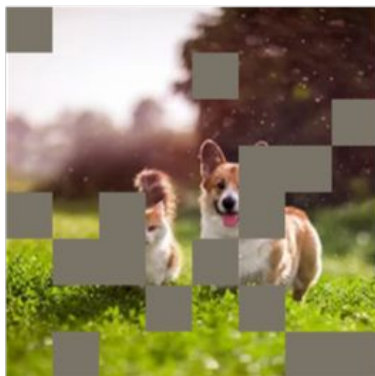
P(corgi) = 0.8

Credit: Ian Covert; Lundberg & Lee 2017

Which pixels matter: Saliency via Backprop

Forward pass: Compute probabilities

Dog

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

Which pixels matter: Saliency via Backprop

Forward pass: Compute probabilities

Dog

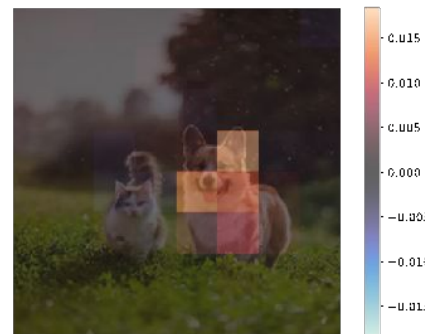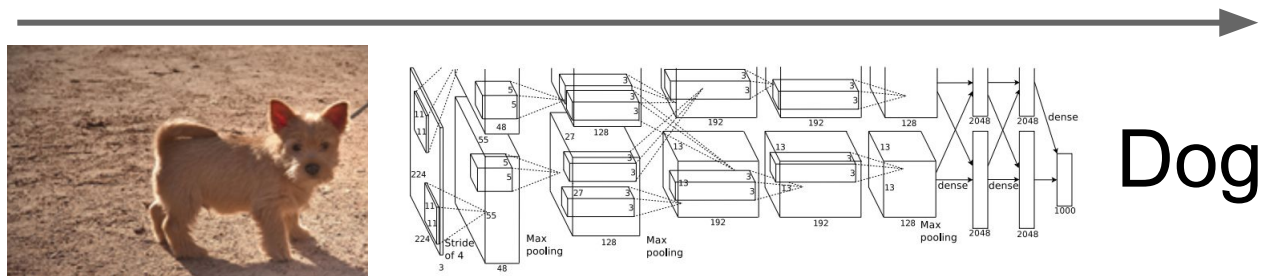Compute gradient of (unnormalized) class score with respect to image pixels, take absolute value and max over RGB channels

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

Saliency Maps: Segmentation without supervision

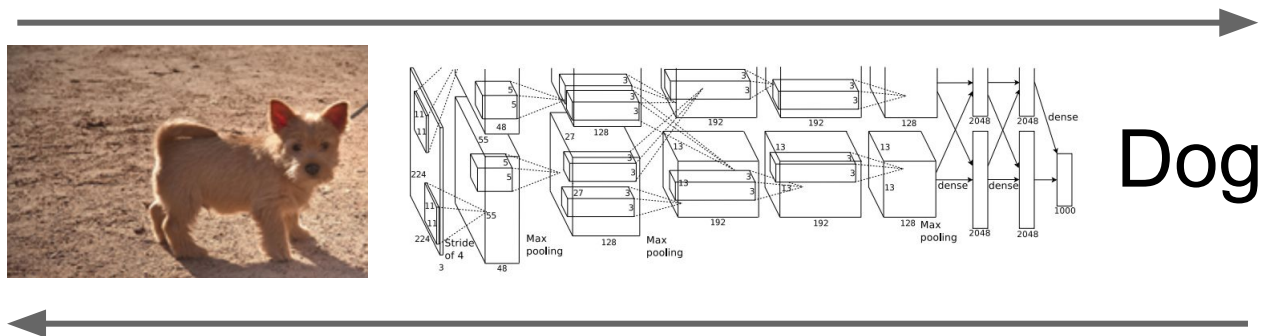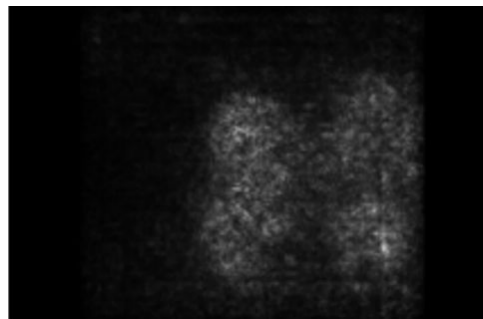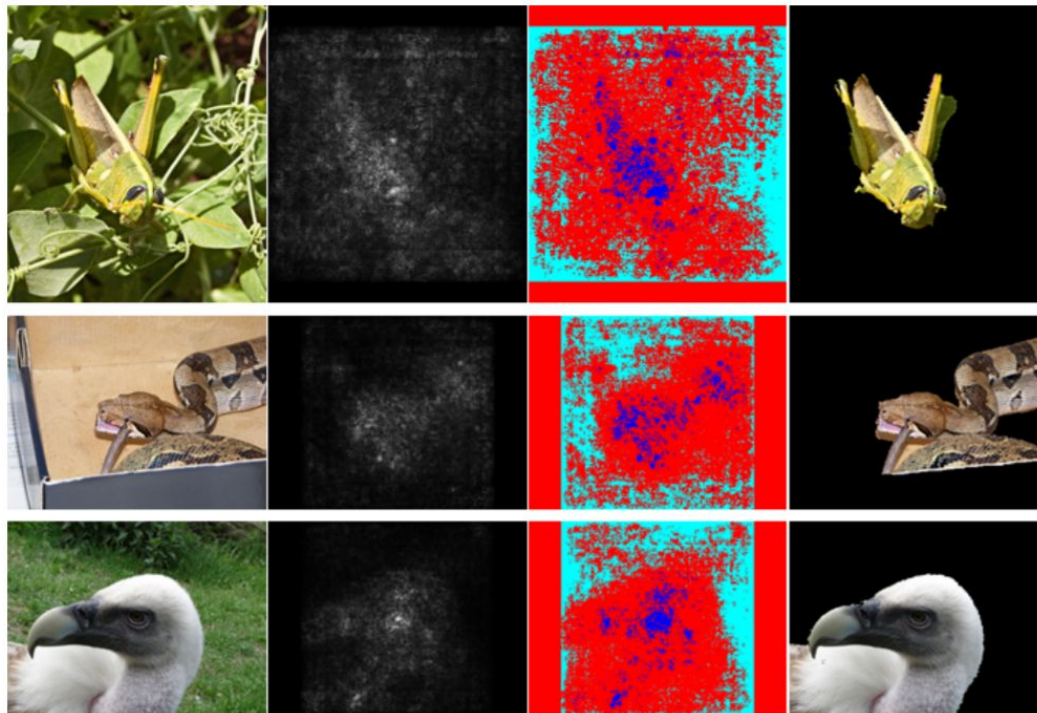Use GrabCut on saliency map

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

Saliency maps: Uncovers biases

Such methods also find biases

wolf vs dog classifier looks is actually a snow vs no-snow classifier
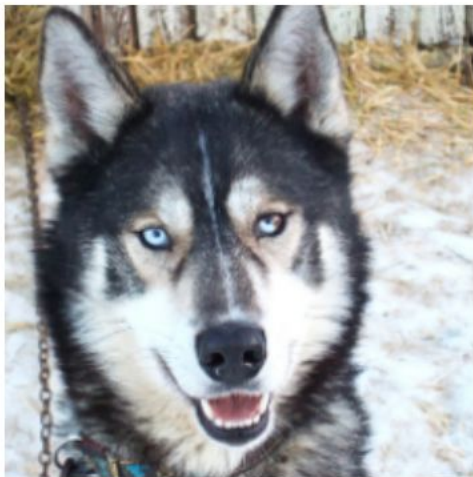
(a) Husky classified as wolf
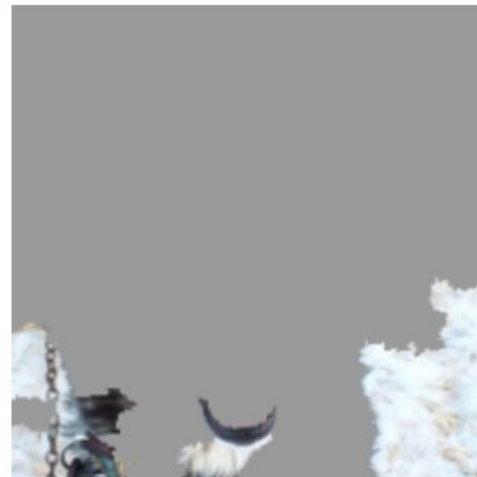
(b) Explanation

Ribeiro et al, ""Why Should I Trust You?" Explaining the Predictions of Any Classifier", ACM KDD 2016

Intermediate Features via (guided) backprop
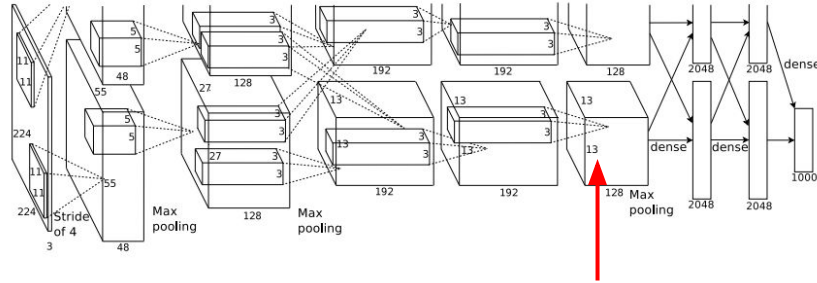
Pick a single intermediate channel, e.g. one value in 128 x 13 x 13 conv5 feature map

Compute gradient of activation 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
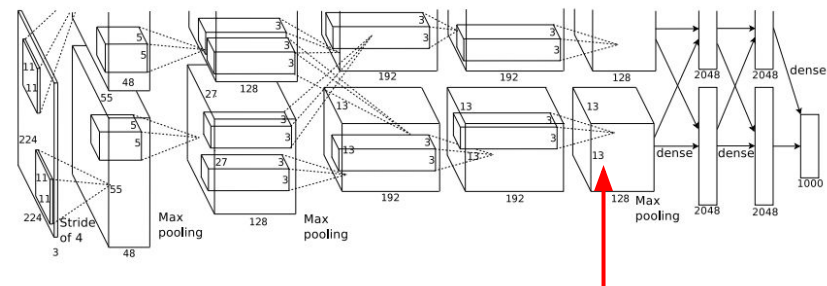
ReLU

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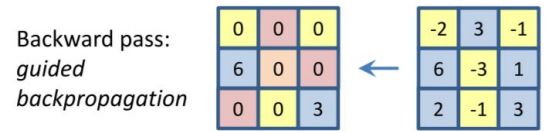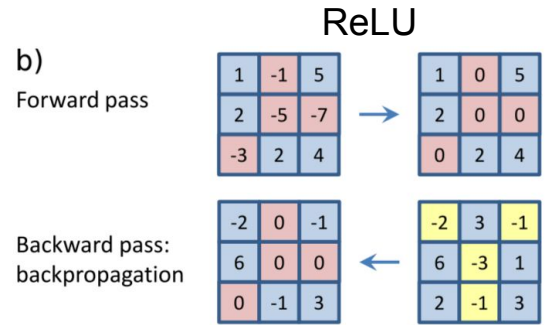
Images come out nicer if you only backprop positive gradients through each ReLU (guided backprop)

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

Intermediate features via (guided) backprop

Maximally activating patches
(Each row is a different neuron)

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.

Intermediate features via (guided) backprop

Maximally activating patches
(Each row is a different neuron)

Guided Backprop

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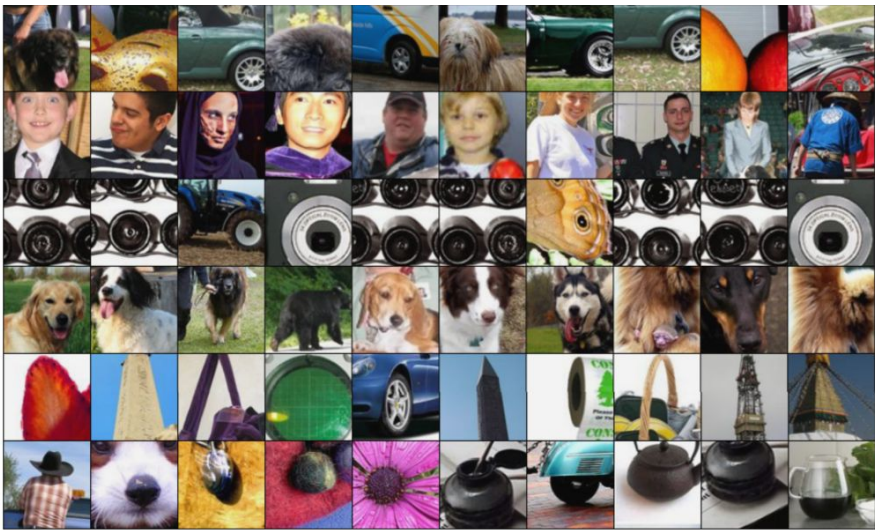Springenberg et al, "Striving for Simplicity: The All Convolutional Net", ICLR Workshop 2015
Figure copyright Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, Martin Riedmiller, 2015; reproduced with permission.

Visualizing CNN features: Gradient Ascent

(Guided) backprop:
Find the part of an
image that a neuron
responds to

Gradient ascent:
Generate a synthetic
image that maximally
activates a neuron

$$I^* = \arg\max_I f(I) + R(I)$$

Neuron value

Natural image regularizer

Visualizing CNN features: Gradient Ascent

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

score for class c (before Softmax)

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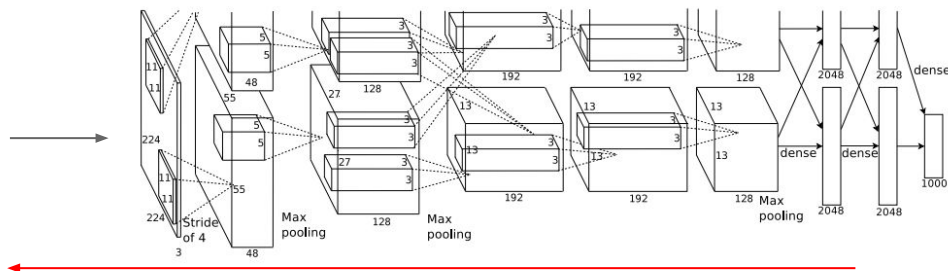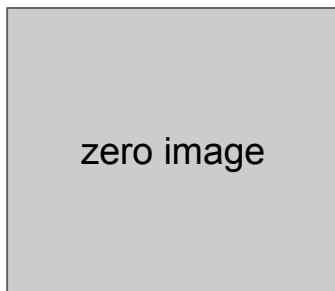
Repeat:
2. Forward image to compute current scores
3. Backprop to get gradient of neuron value with respect to image pixels
4. Make a small update to the image

Visualizing CNN features: Gradient Ascent

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

Simple regularizer: Penalize L2
norm of generated image

Visualizing CNN features: Gradient Ascent

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

Simple regularizer: Penalize L2
norm of generated image

dumbbell   cup   dalmatian

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.

Visualizing CNN features: Gradient Ascent

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

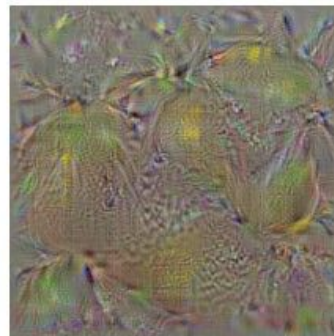Simple regularizer: Penalize L2
norm of generated image
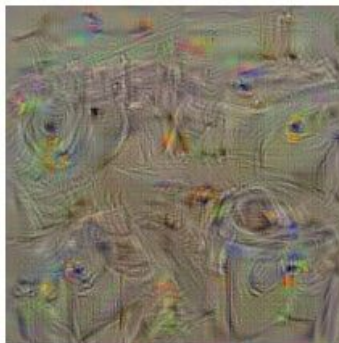
washing machine   computer keyboard   kit fox

goose   ostrich   limousine

Visualizing CNN features: Gradient Ascent

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

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

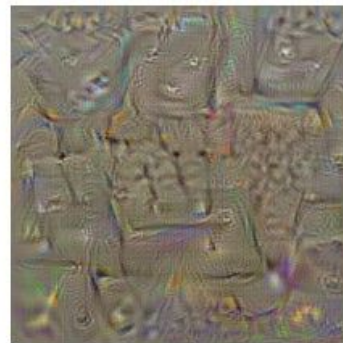Yosinski et al, "Understanding Neural Networks Through Deep Visualization", ICML DL Workshop 2014.

Visualizing CNN features: Gradient Ascent

$$\arg\max_I S_c(I) - \lambda\|I\|_2^2$$

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

Ground Beetle

Indian Cobra

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.

Visualizing CNN features: Gradient Ascent

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

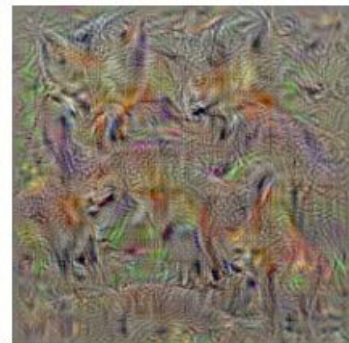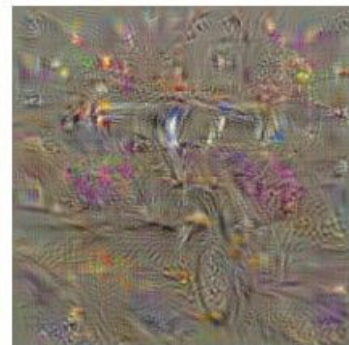Better regularizer: Penalize L2 norm of image; also during optimization periodically

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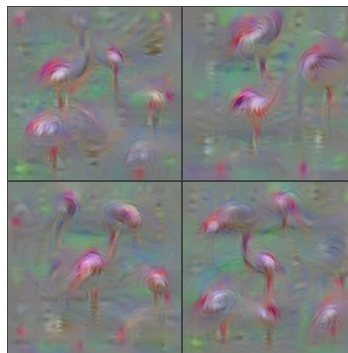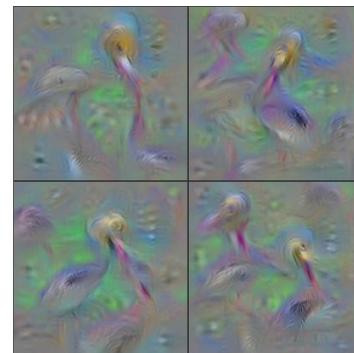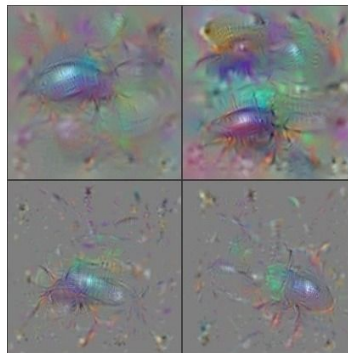(3)   Clip pixels with small gradients to 0

Hartebeest
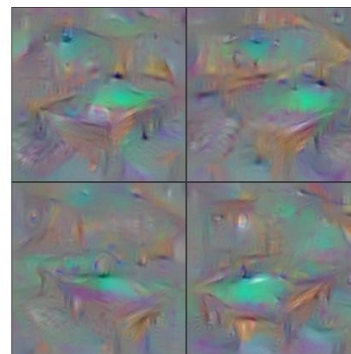
Billiard Table
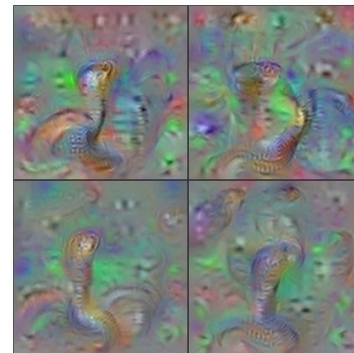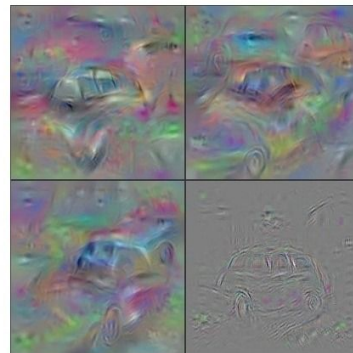
Station Wagon
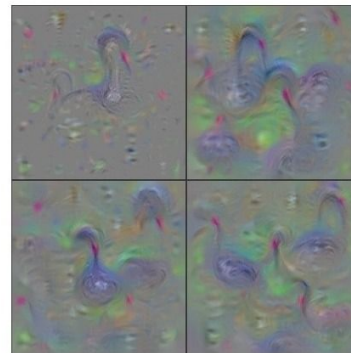
Black Swan

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.

Visualizing CNN features: Gradient Ascent

Use the same approach to visualize intermediate features

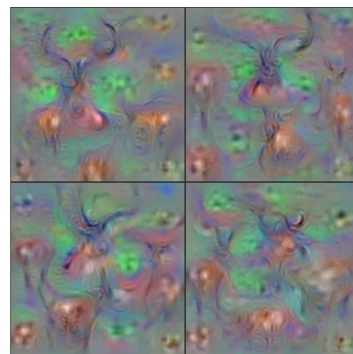Layer 5

Layer 4

Layer 3

Layer 2

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.

Visualizing CNN features: Gradient Ascent

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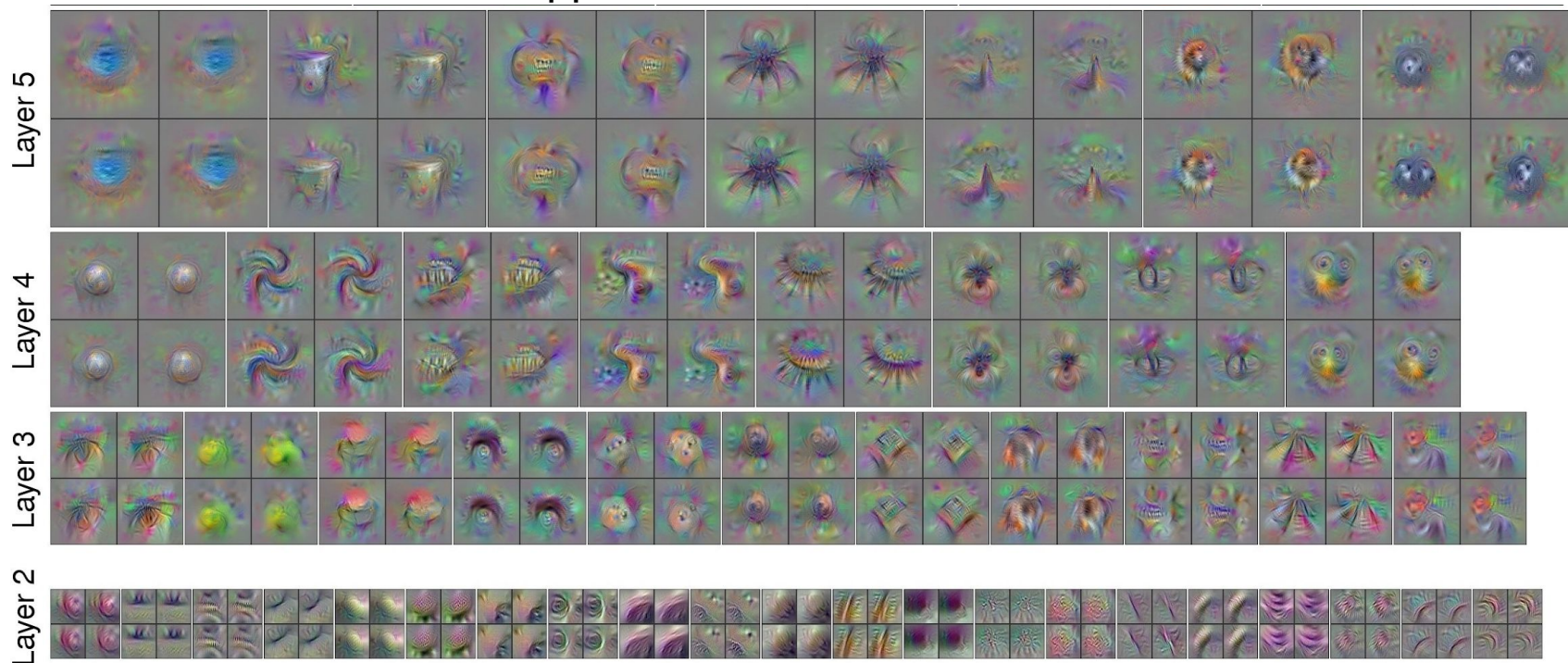
Corresponding example training set images recognized by the same neuron as in the "grocery store" class

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.

Visualizing CNN features: Gradient Ascent

bell pepper, cardoon, strawberry, orange, pineapple, hay, alp, bubble, cliff, beer bottle, birdhouse, breakwater, breastplate, broom, caldron, candle, cinema, cowboy boot, entertainment, gasmask, golf ball, golfcart, gown, grand piano, hourglass, jack-o'-lantern, knot, lampshade, monitor, mosque, motor scooter, pirate, planetarium, radio, sarong, schooner
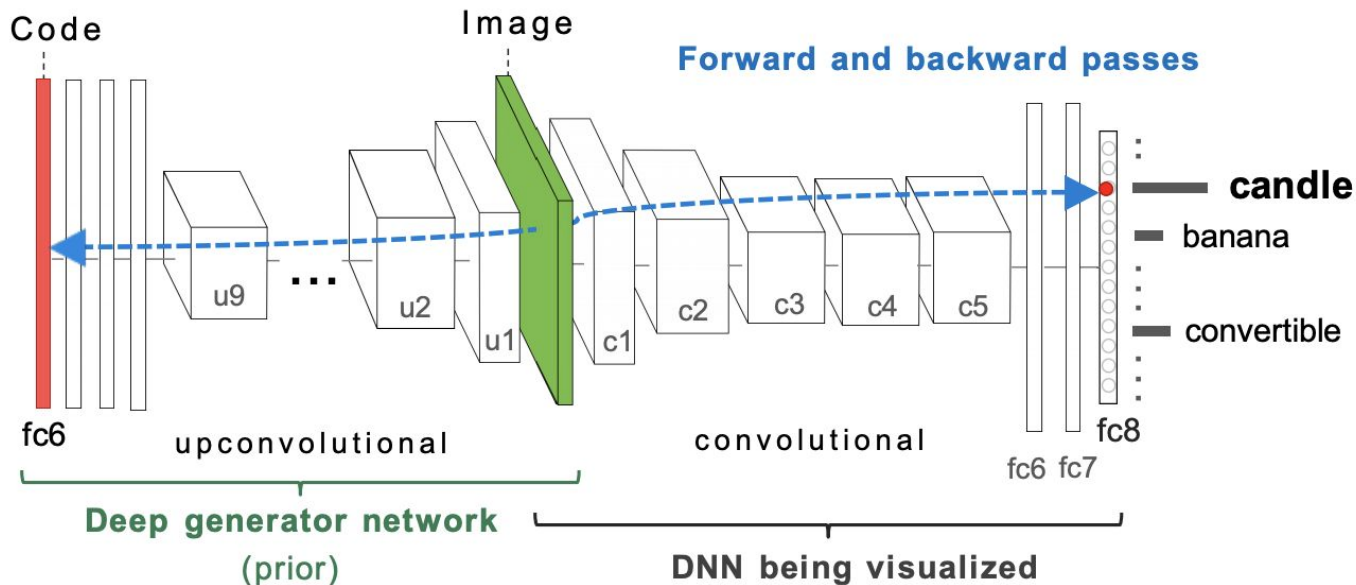
Visualizing CNN features: Gradient Ascent

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
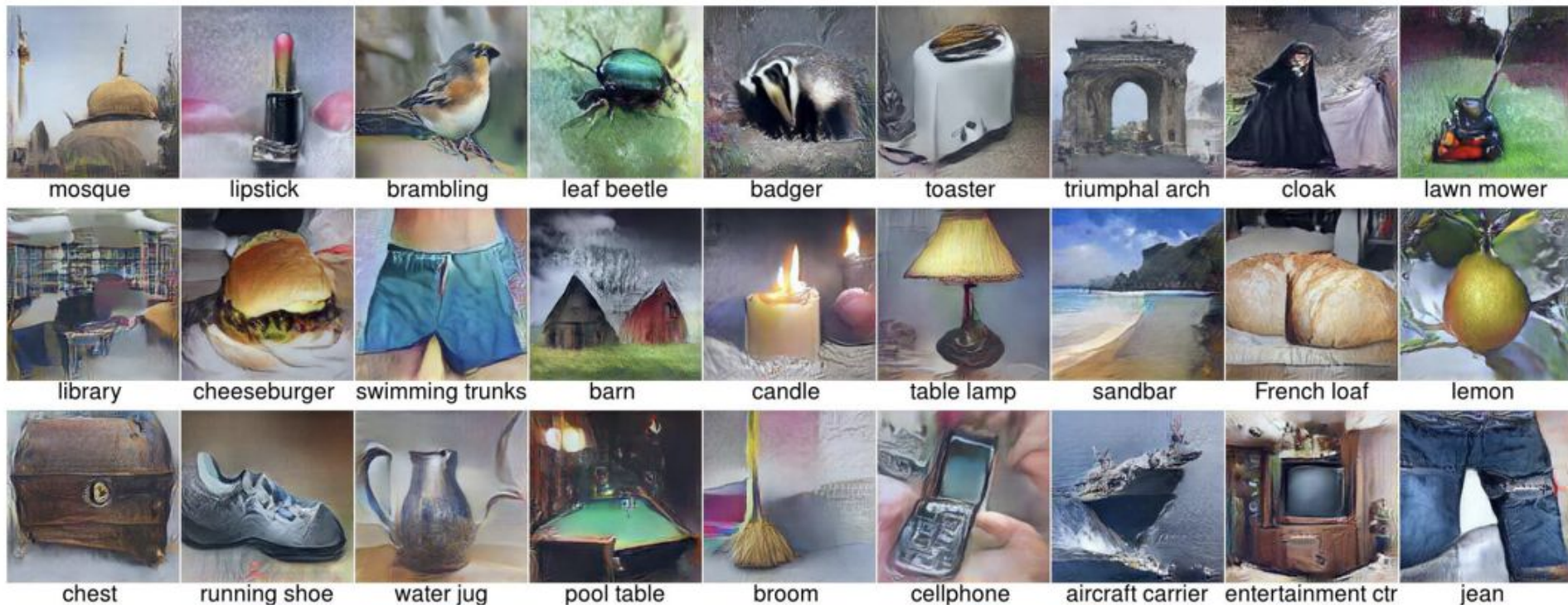Figure copyright Nguyen et al, 2016; reproduced with permission.

Visualizing CNN features: Gradient Ascent

Optimize in FC6 latent space instead of pixel space:

mosque   lipstick   brambling   leaf beetle   badger   toaster   triumphal arch   cloak   lawn mower

library   cheeseburger   swimming trunks   barn   candle   table lamp   sandbar   French loaf   lemon

chest   running shoe   water jug   pool table   broom   cellphone   aircraft carrier   entertainment ctr   jean

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.

Today's agenda

Visualizing what models have learned:
- Visualizing filters
- Visualizing final layer features
- Visualizing activations

Understanding input pixels
- Identifying important pixels
- Saliency via backprop
- Guided backprop to generate images
- Gradient ascent to visualize features

Adversarial perturbations

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
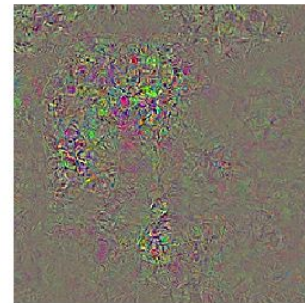
Fooling Images / Adversarial Examples

Fooling Images / Adversarial Examples

African elephant | koala | Difference | 10x Difference
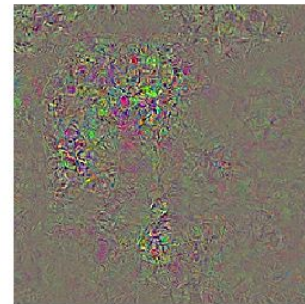
schooner | iPod | Difference | 10x Difference
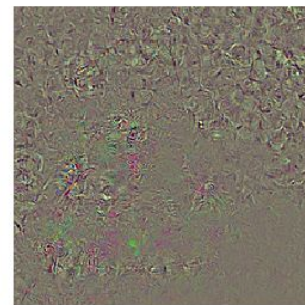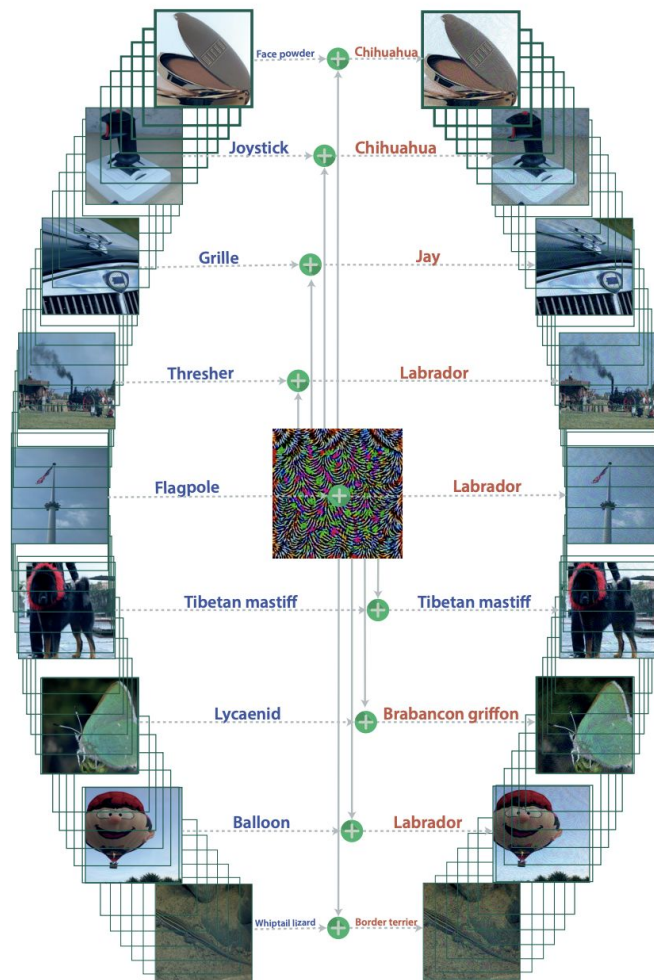
Check out Ian Goodfellow's lecture from 2017

Fooling Person Detectors and Self-driving Cars
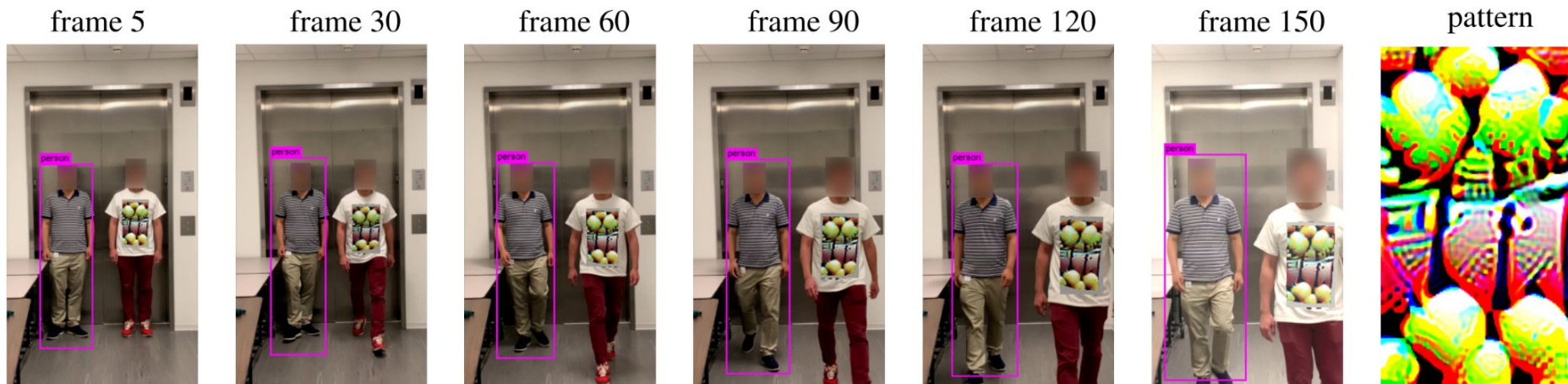
Xu et al., 2019; Eykholt et al., 2018

Fooling Images / Adversarial Exan

Universal perturbations

Moosavi-Dezfooli, Seyed-Mohsen, et al. "Universal adversarial perturbations." Proceedings of the IEEE conference on computer vision and pattern recognition. 2017.
Figure reproduced with permission

Summary

Many methods for understanding CNN representations

Activations: Nearest neighbors, Dimensionality reduction, maximal patches, occlusion
Gradients: Saliency maps, class visualization, fooling images, feature inversion

Adversarial Examples: To confuse the models

Next time: **Self-supervised Learning**