Intro to Deep Learning

Why is Recognition Difficult?

This book has a lot of material:

K. Grauman and B. Leibe

Visual Object Recognition

Synthesis Lectures On Computer Vision, 2011

It was supposed to be solved 51 years ago

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

PROJECT MAC

Artificial Intelligence Group — July 7, 1966
Vision Memo. No. 100.

THE SUMMER VISION PROJECT

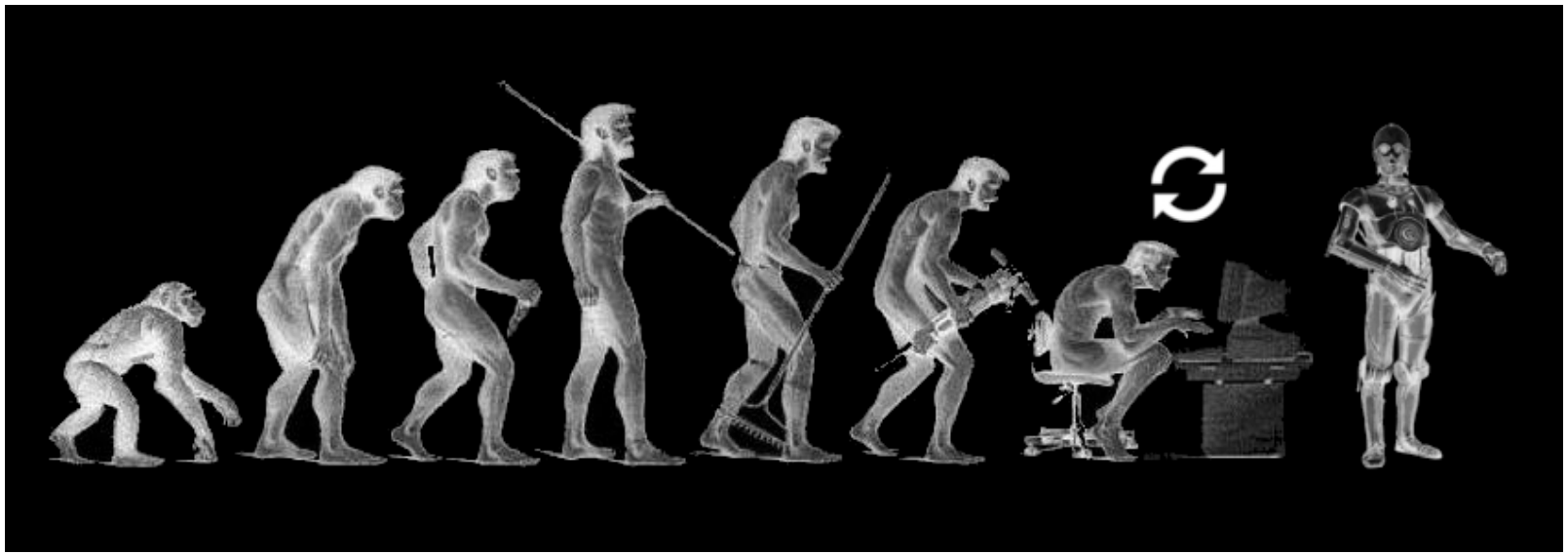
Seymour Papert

The summer vision project is an attempt to use our summer workers effectively in the construction of a significant part of a visual system. The particular task was chosen partly because it can be segmented into sub-problems which will allow individuals to work independently and yet participate in the construction of a system complex enough to be a real landmark in the development of "pattern recognition".

[Slide credit: A. Torralba]

- What's still missing?
- What happens if we solve it?

Figure: Singularity?



http://www.futurebuff.com/wp-content/uploads/2014/06/singularity-c3po.jpg

The Recognition Tasks

- Let's take some typical tourist picture. What all do we want to recognize?

[Adopted from S. Lazebnik]

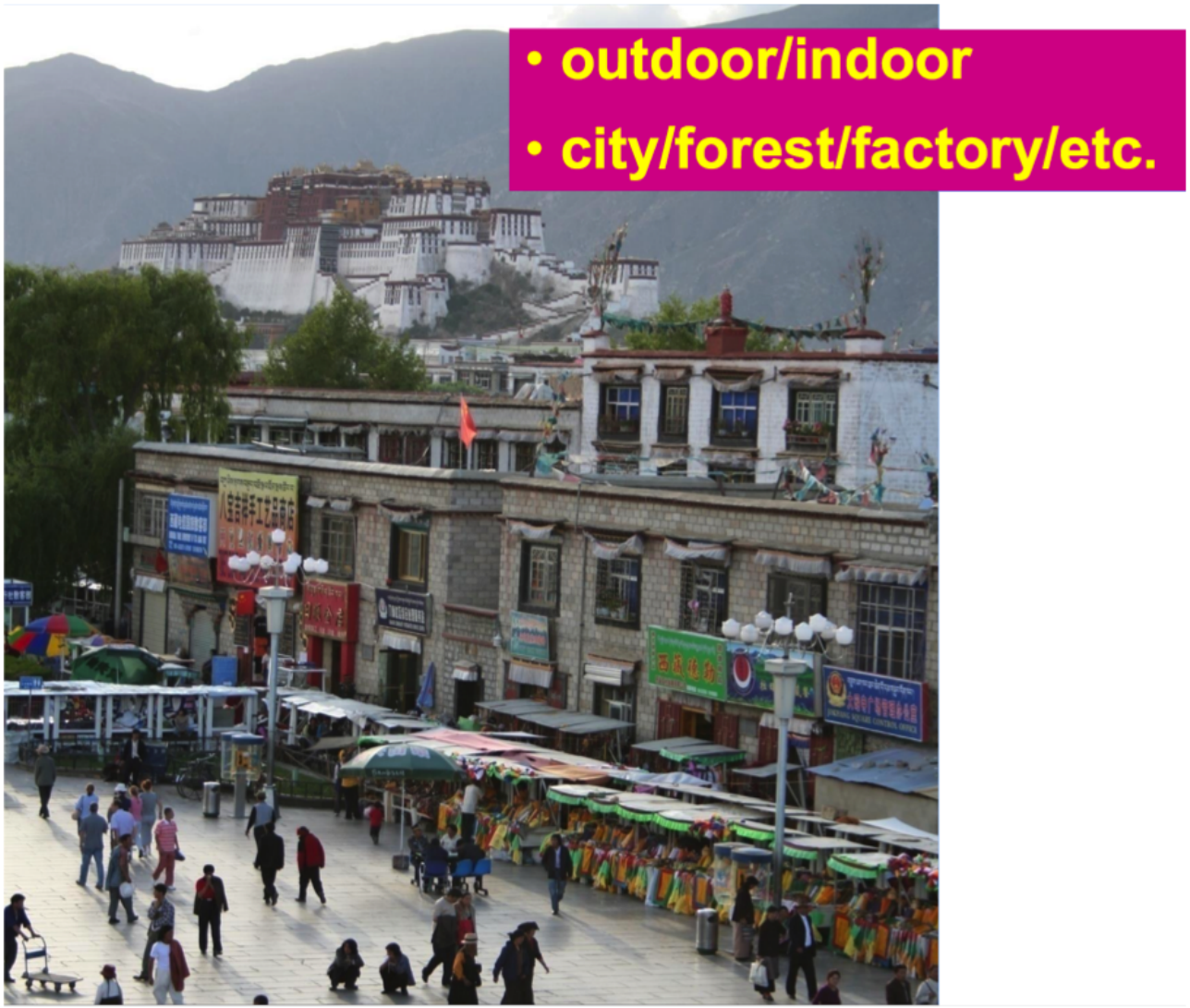
The Recognition Tasks

- Identification: we know this one (like our DVD recognition pipeline)

[Adopted from S. Lazebnik]

The Recognition Tasks

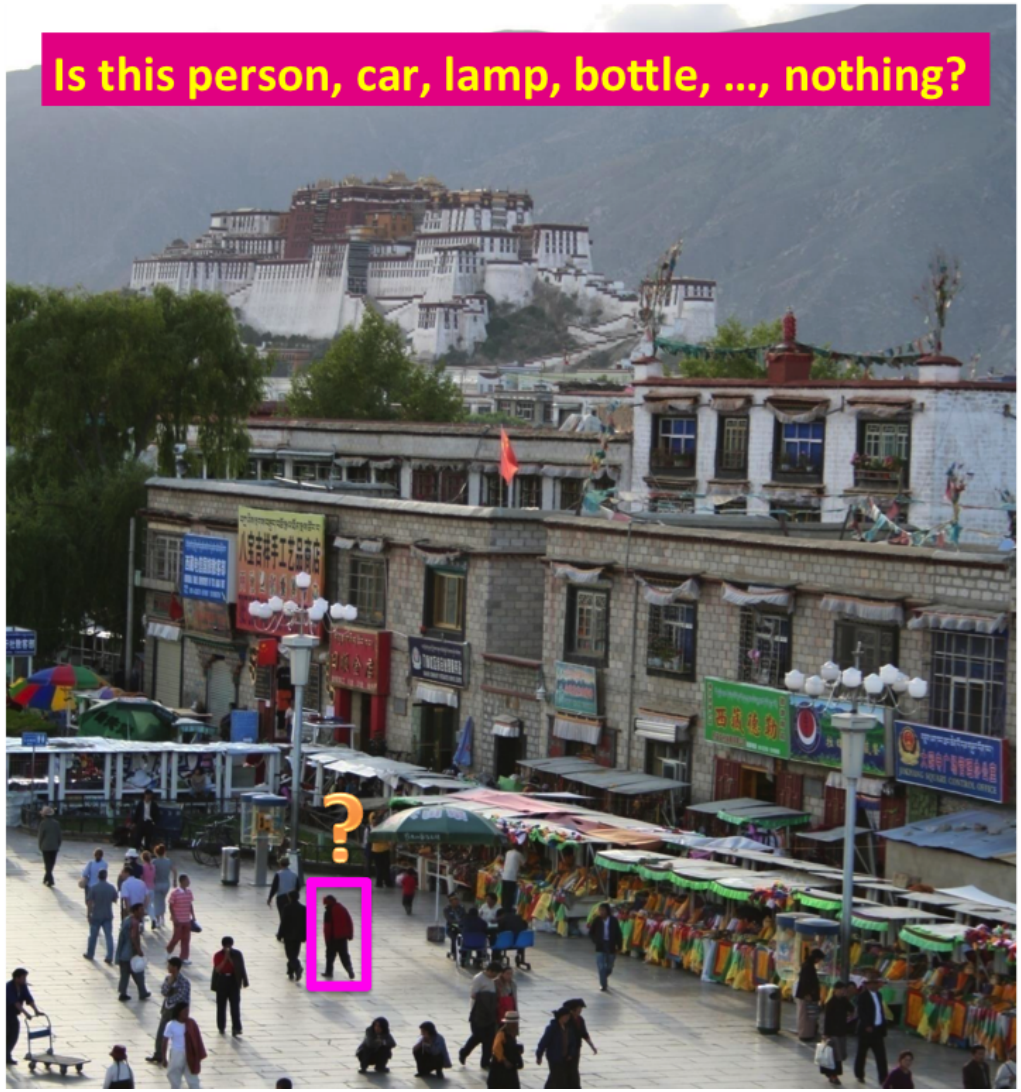
- Scene classification: what type of scene is the picture showing?



[Adopted from S. Lazebnik]

The Recognition Tasks

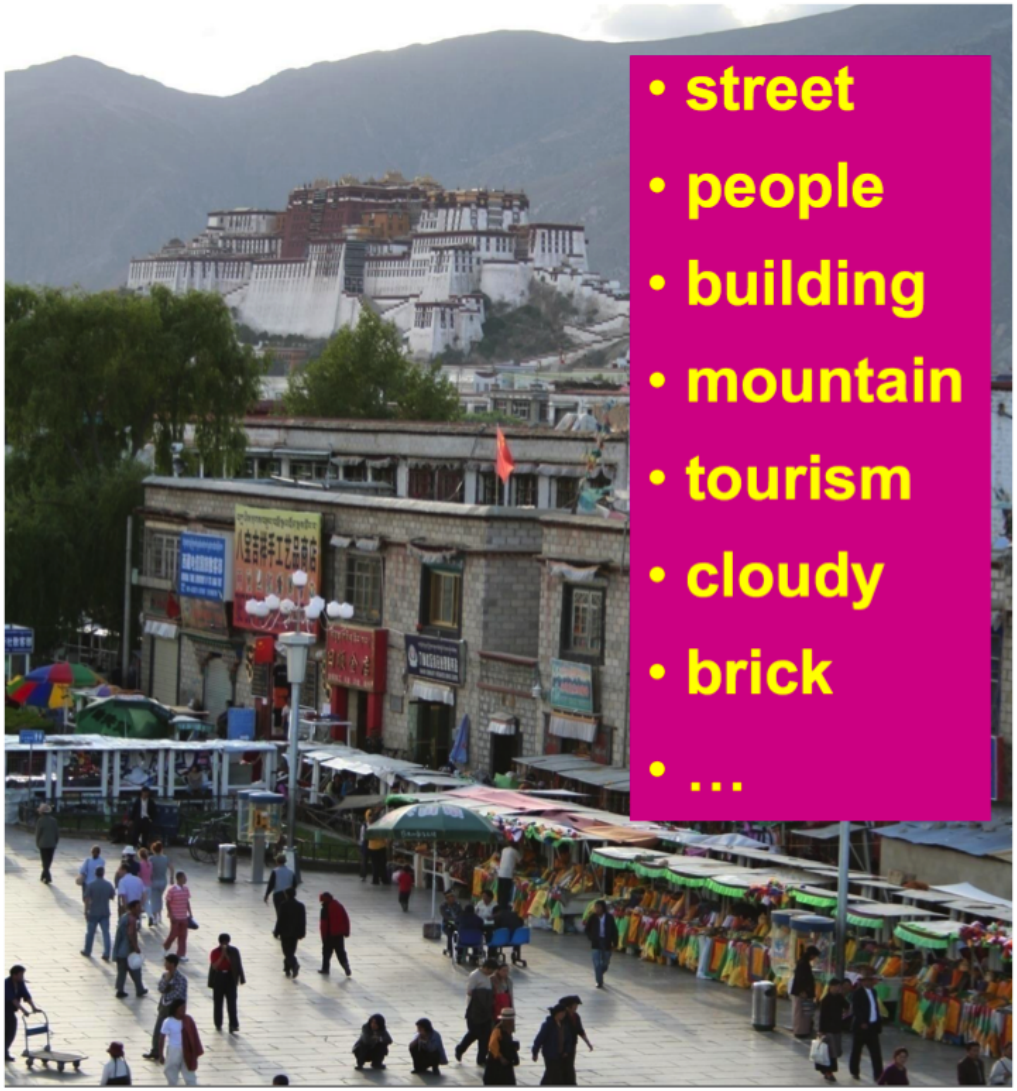
- Classification: Is the object in the window a person, a car, etc



[Adopted from S. Lazebnik]

The Recognition Tasks

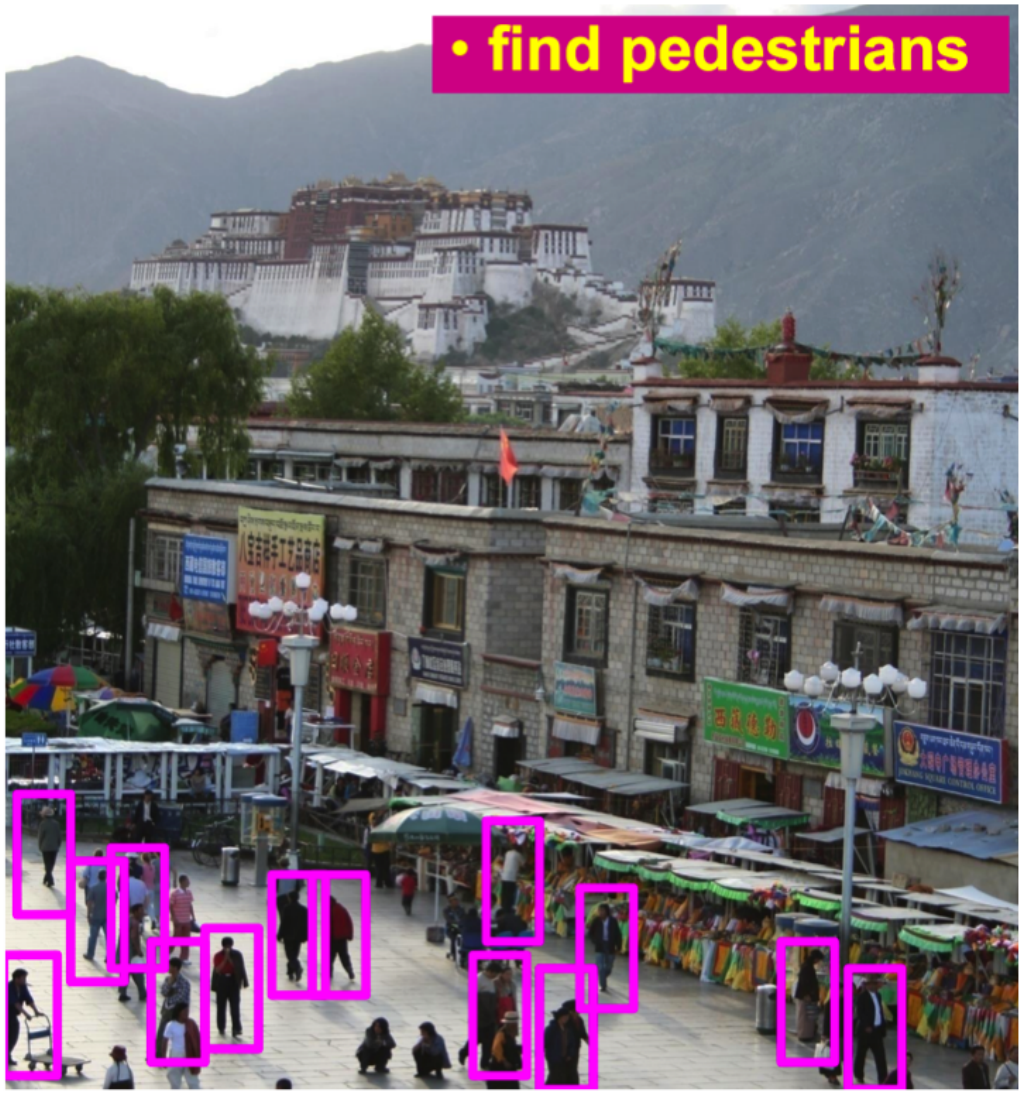
- Image Annotation: Which types of objects are present in the scene?



[Adopted from S. Lazebnik]

The Recognition Tasks

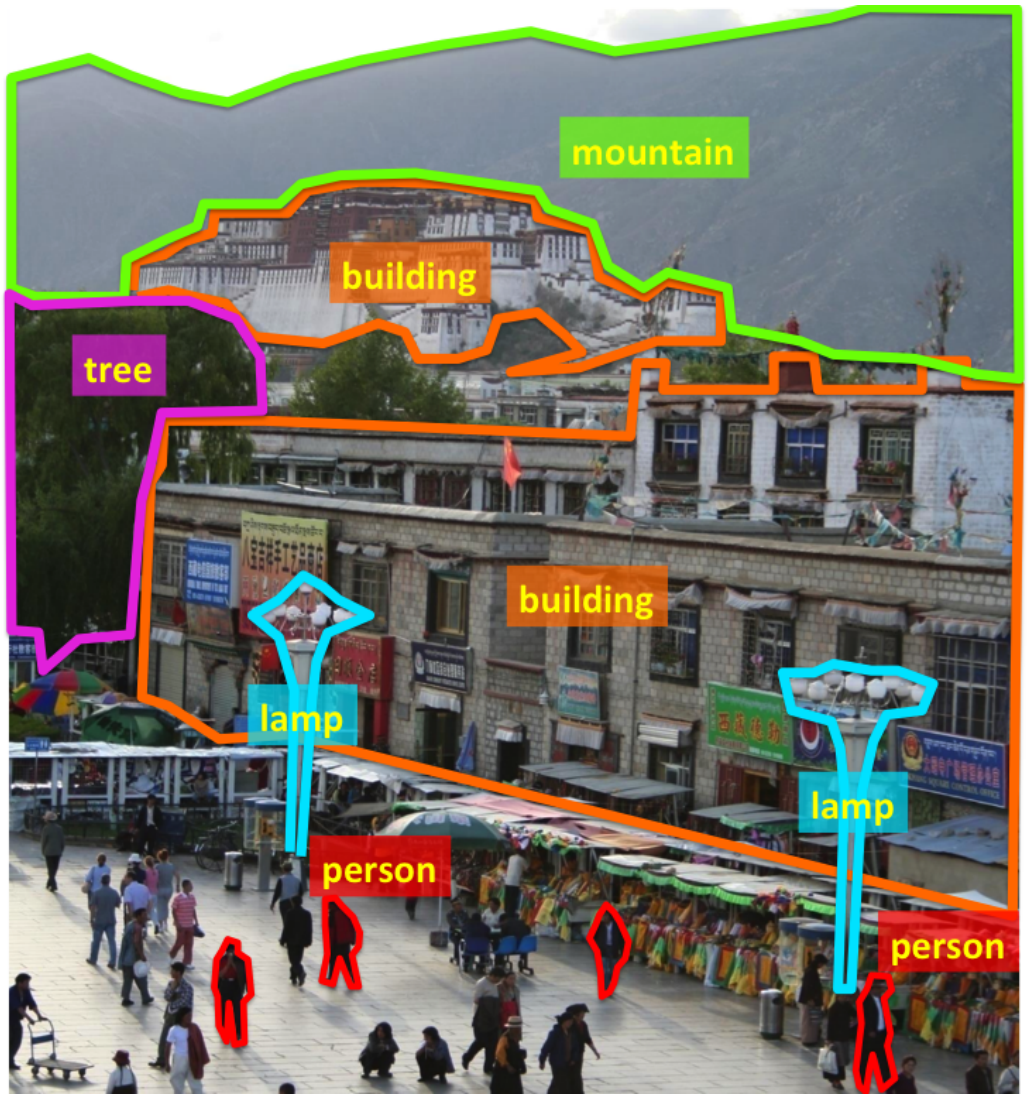
- Detection: Where are all objects of a particular class?



[Adopted from S. Lazebnik]

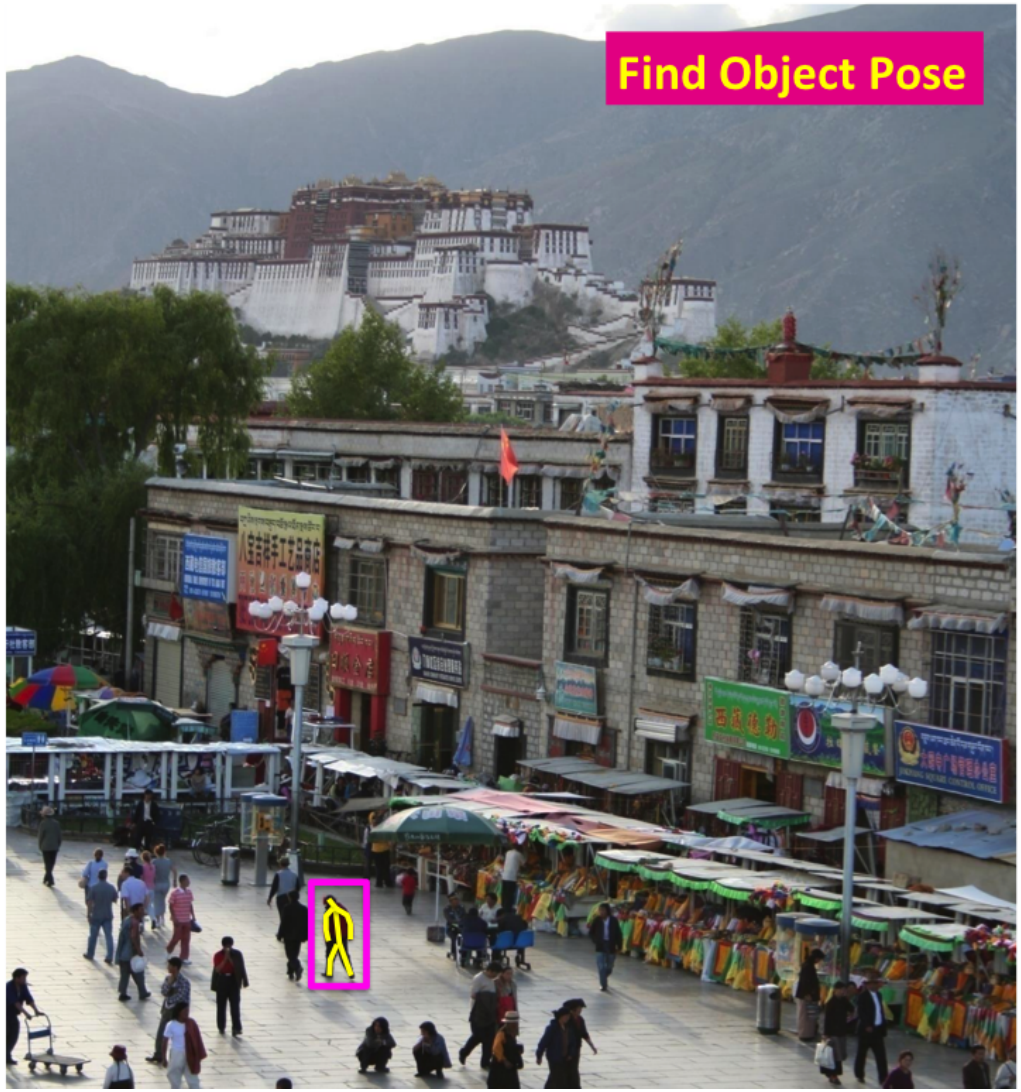
The Recognition Tasks

- Segmentation: Which pixels belong to each class of objects?



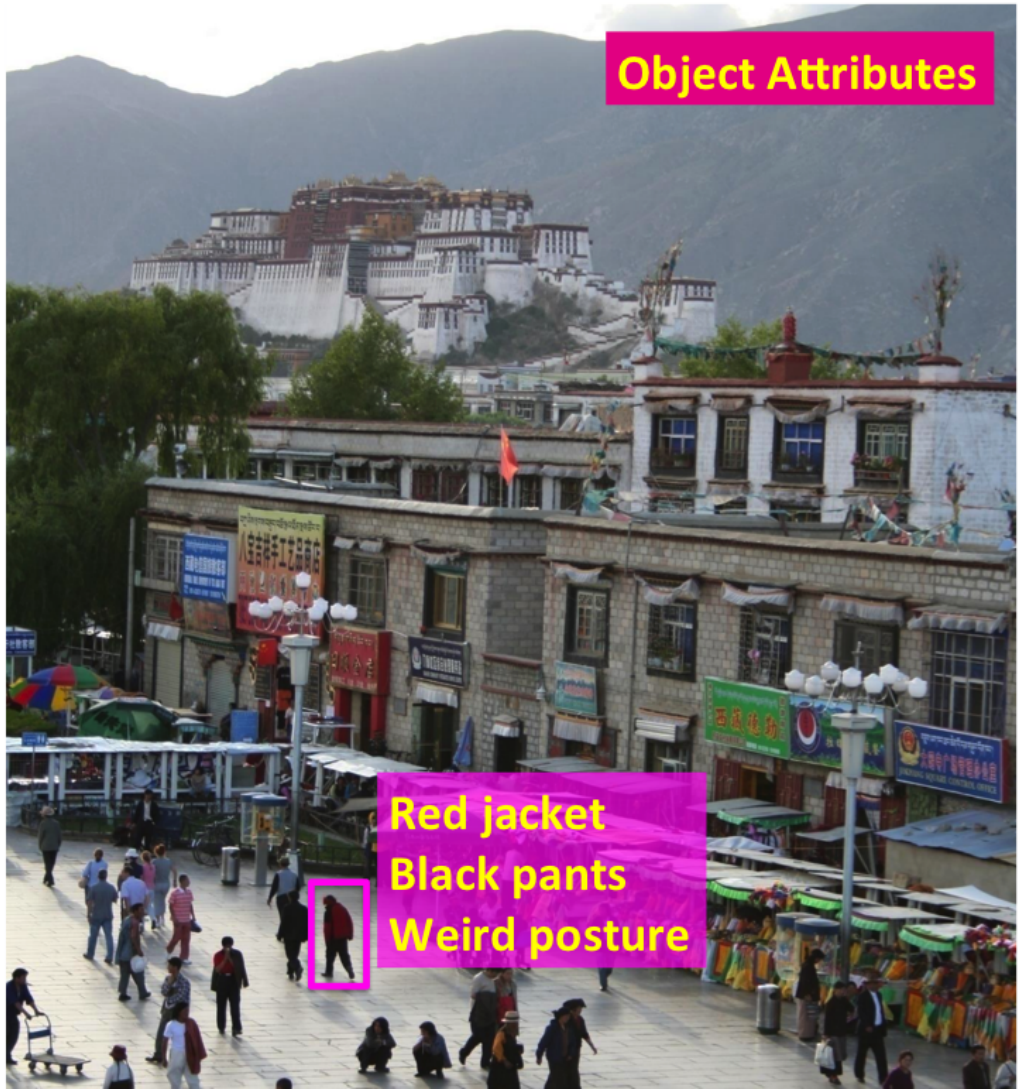
The Recognition Tasks

- Pose estimation: What is the pose of each object?



The Recognition Tasks

- Attribute recognition: Estimate attributes of the objects (color, size, etc)

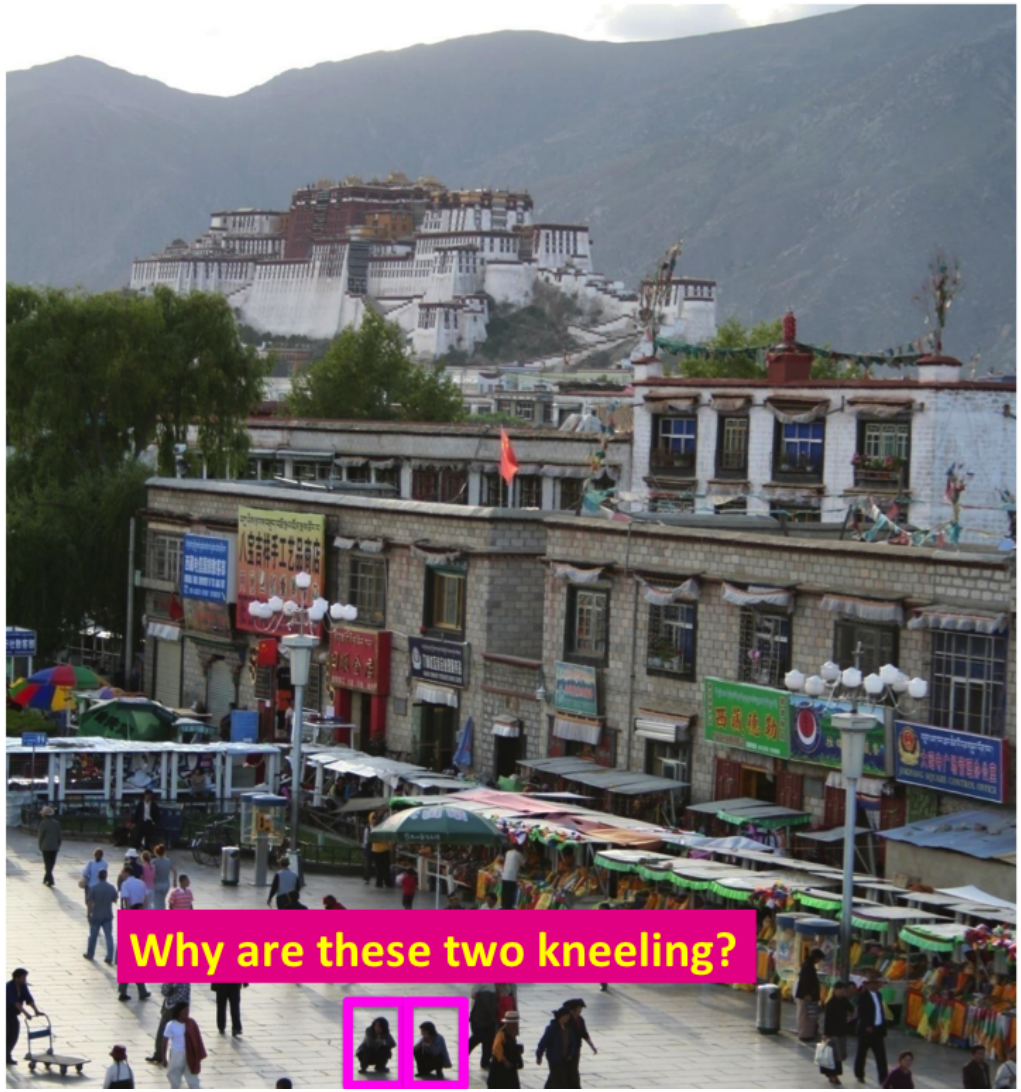


The Recognition Tasks

- Action recognition: What is happening in the image?

The Recognition Tasks

- Surveillance: Why is something happening?

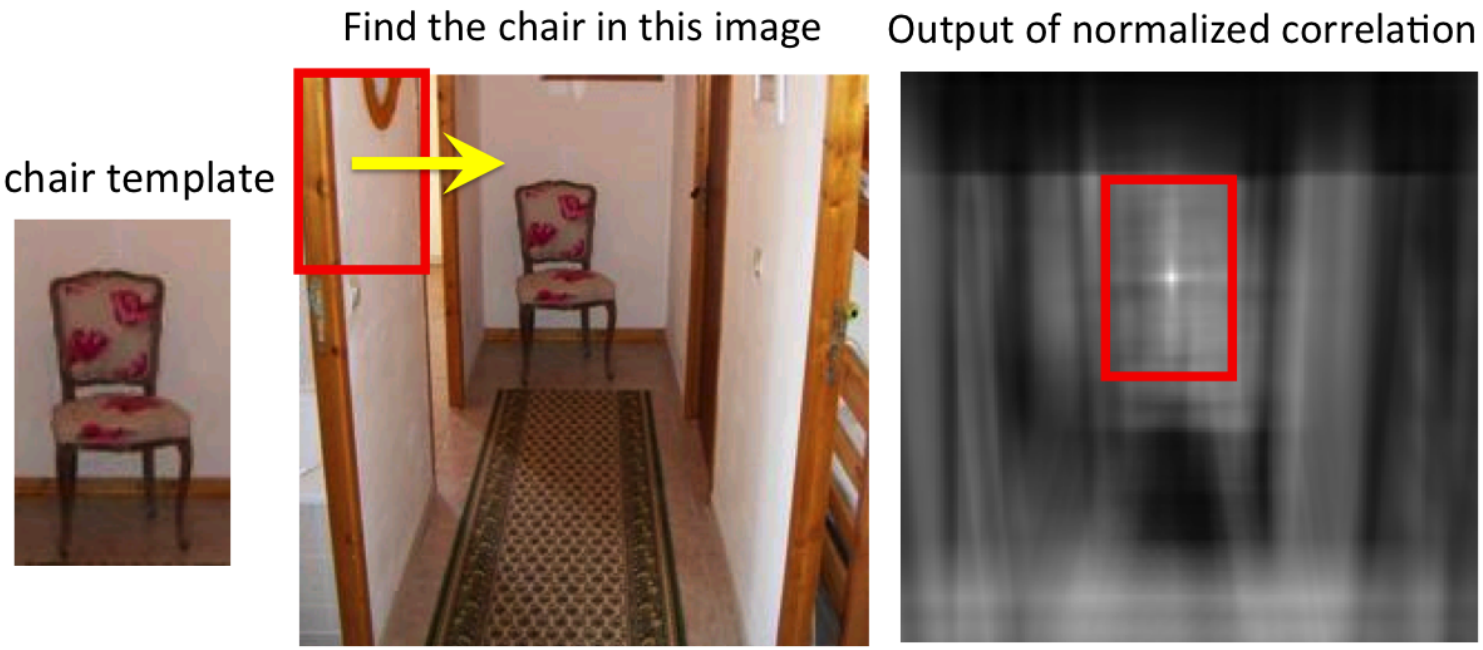


Have we encountered these things before?

- Before we proceed, let's first give a shot to the techniques we already know
- Let's try detection
- These techniques are:
  - Template matching (remember Waldo in Lecture 3-5?)
  - Large-scale retrieval: store millions of pictures, recognize new one by finding the most similar one in database. This is a Google approach.

Template Matching

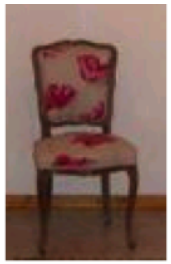
- Template matching: normalized cross-correlation with a template (filter)



[Slide from: A. Torralba]

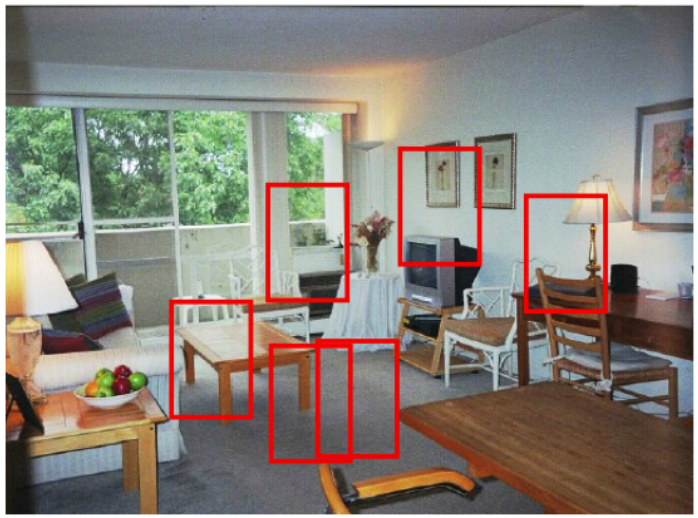
Template Matching

- Template matching: normalized cross-correlation with a template (filter)

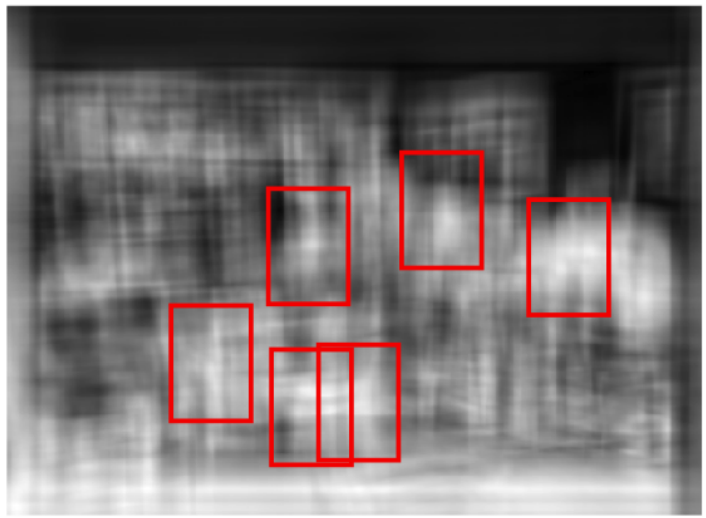


template

Find the chair in this image



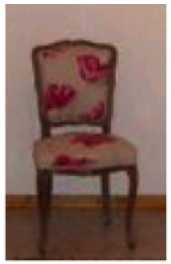
Pretty much garbage
Simple template matching is
not going to make it



[Slide from: A. Torralba]

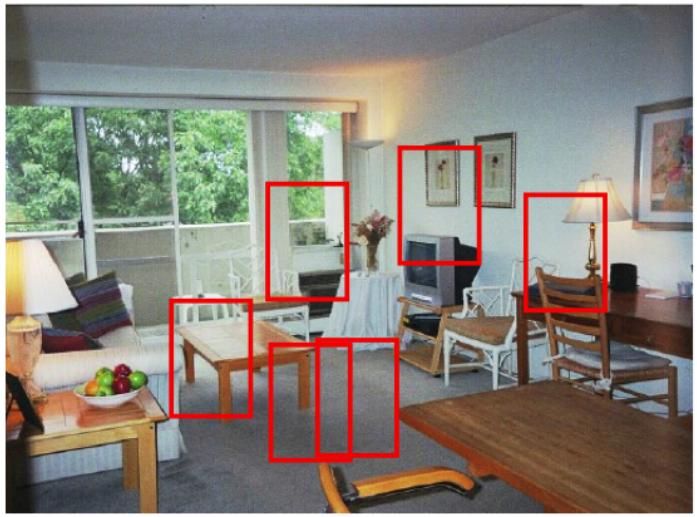
Template Matching

- Template matching: normalized cross-correlation with a template (filter)

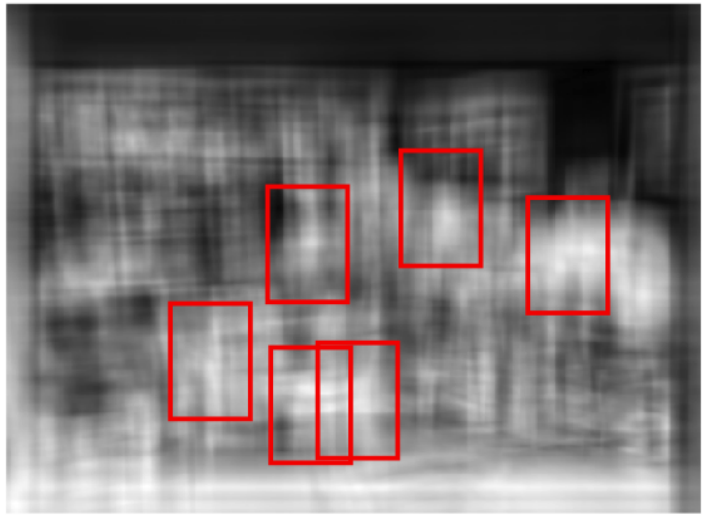


template

Find the chair in this image



Pretty much garbage
Simple template matching is
not going to make it

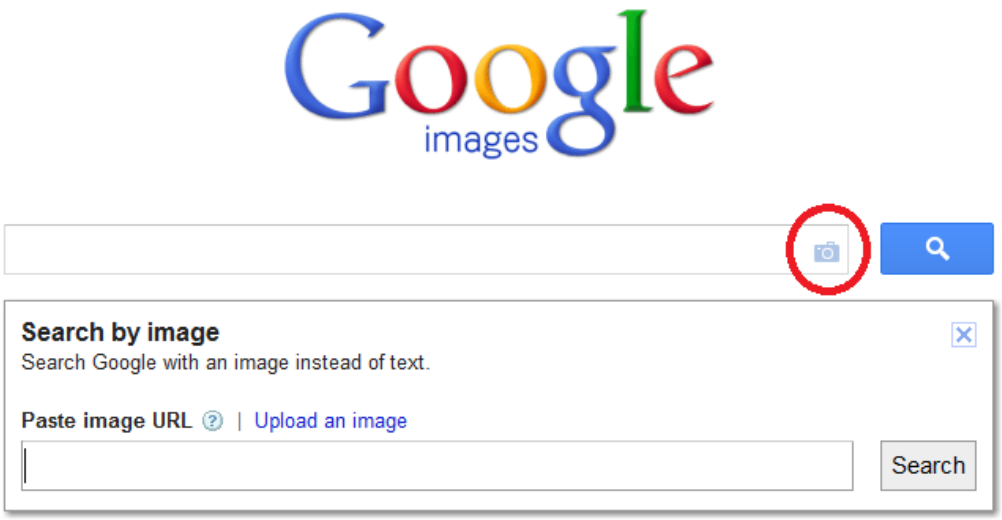


A "popular method is that of template matching, by point to point correlation of a model pattern with the image pattern. These techniques **are inadequate for three-dimensional scene analysis for many reasons, such as occlusion, changes in viewing angle, and articulation of parts.**" Nevatia & Binford, 1977.

[Slide from: A. Torralba]

Recognition via Retrieval by Similarity

- Upload a photo to Google image search and check if something reasonable comes out



query

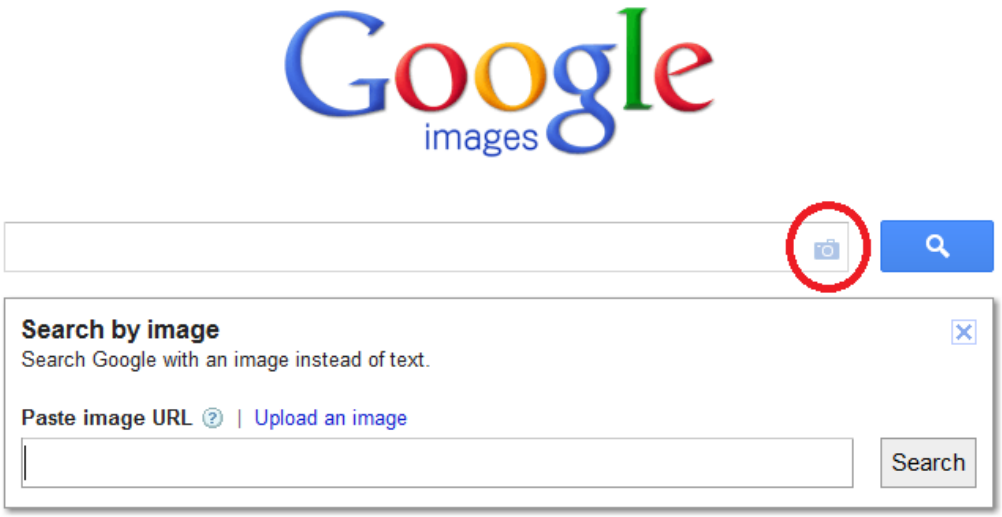
Recognition via Retrieval by Similarity

- Upload a photo to Google image search
- Pretty reasonable, both are Golden Gate Bridge

query

Recognition via Retrieval by Similarity

- Upload a photo to Google image search
- Let's try a typical bathtub object



query

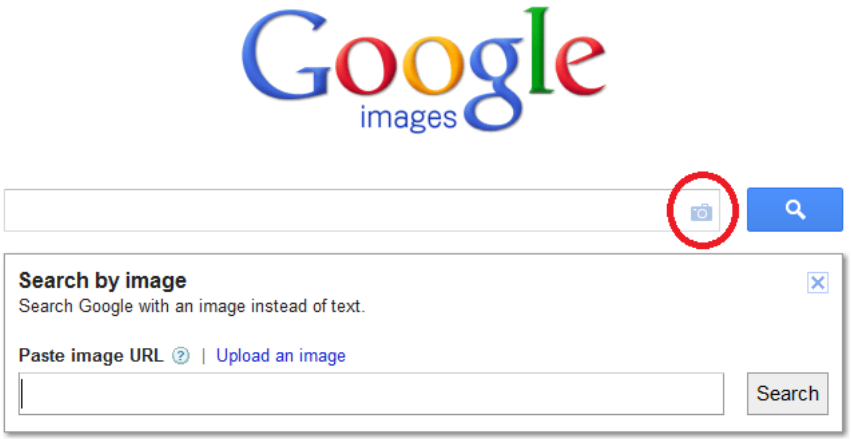
Recognition via Retrieval by Similarity

- Upload a photo to Google image search
- A bit less reasonable, but still some striking similarity

query

Recognition via Retrieval by Similarity

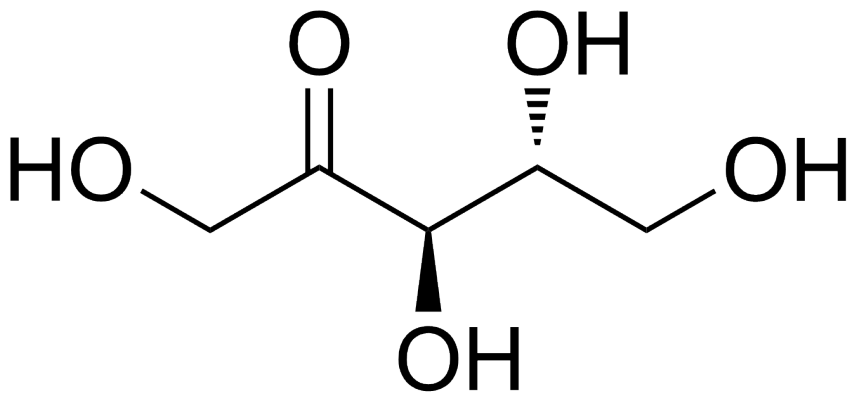
- Make a beautiful drawing and upload to Google image search
- Can you recognize this object?



query

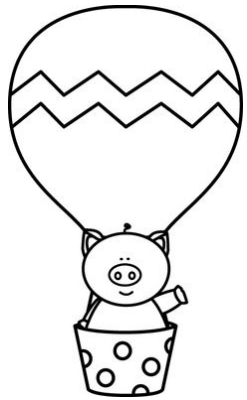
Recognition via Retrieval by Similarity

- Make a beautiful drawing and upload to Google image search
- Not a very reasonable result



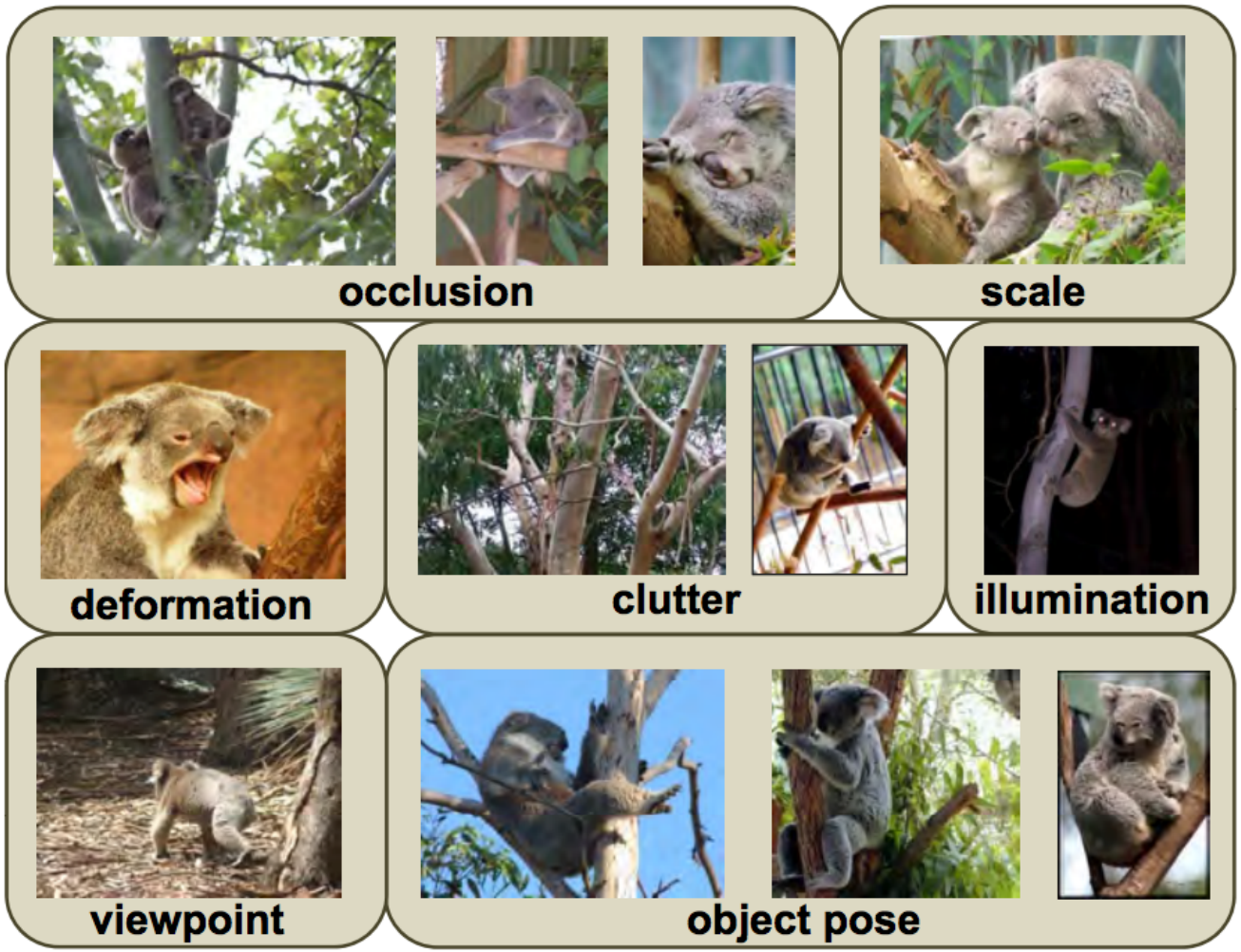
query

other retrieved results:



Why is it a Problem?

- Difficult scene conditions



[From: Grauman & Leibe]

Why is it a Problem?

- Huge within-class variations. Recognition is mainly about modeling variation.






[Pic from: S. Lazebnik]

Why is it a Problem?

- Tonnes of classes

Overview

- There does exist a single concept that helps us come closer to solving these issues
- And it is quite simple.

Overview

- There does exist a single concept that helps us come closer to solving these issues
- And it is quite simple.
- This concept is called **Neural Networks**

Overview

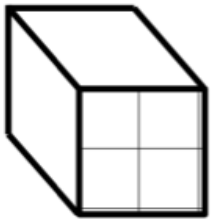
- There does exist a single concept that helps us come closer to solving these issues
- And it is quite simple.
- This concept is called **Neural Networks**

Convolutional Neural Networks (CNN)

- Remember our Lecture 2 about filtering?

Input “image”

Filter

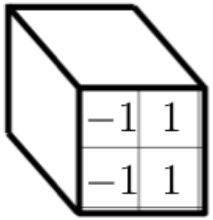


Convolutional Neural Networks (CNN)

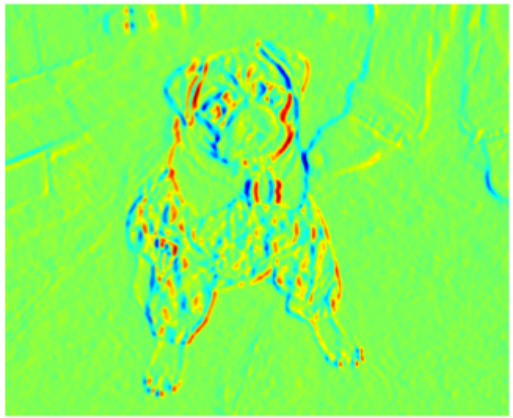
- If our filter was $[-1, 1]$, we got a vertical edge detector

Input "image"

Filter

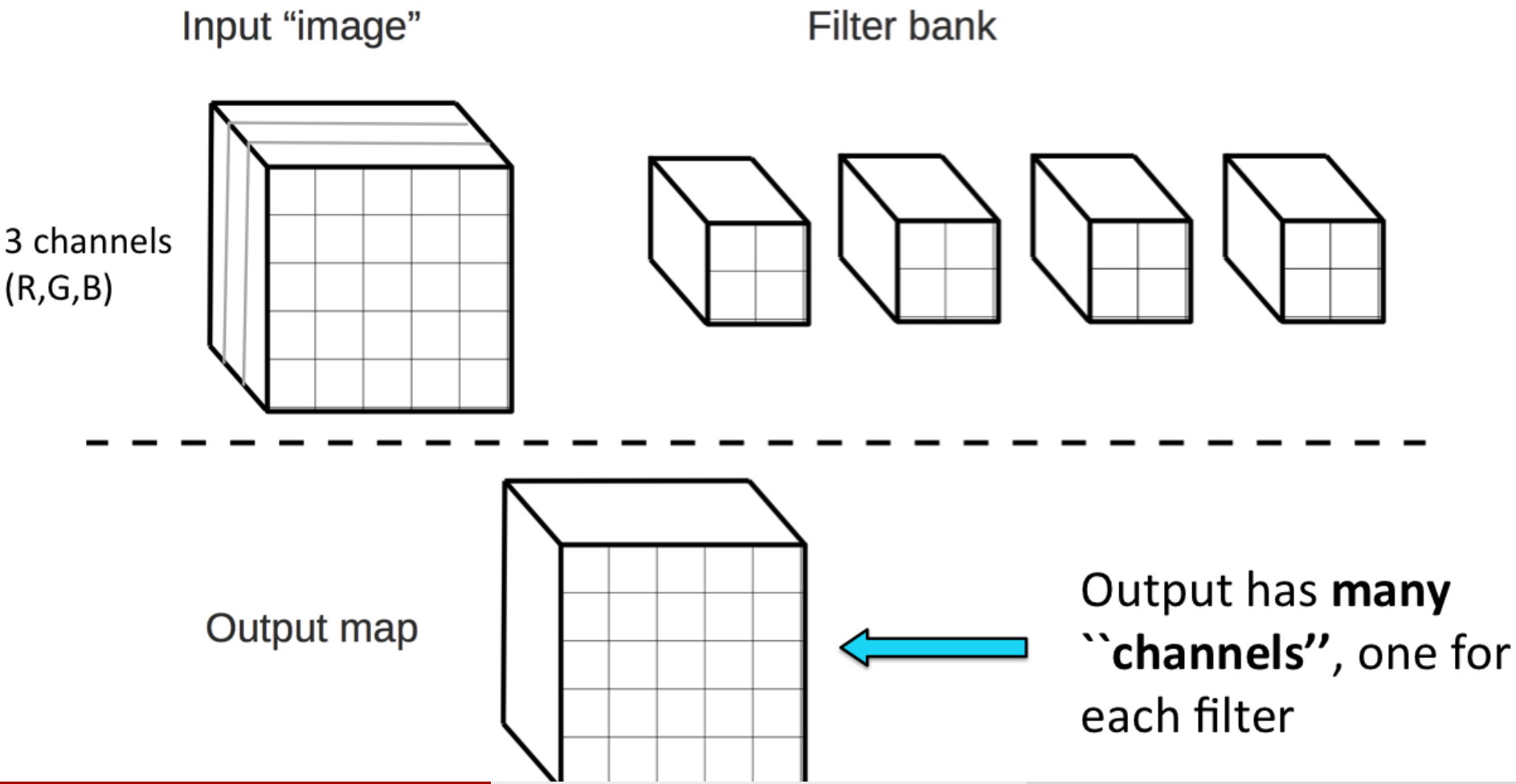


Output map



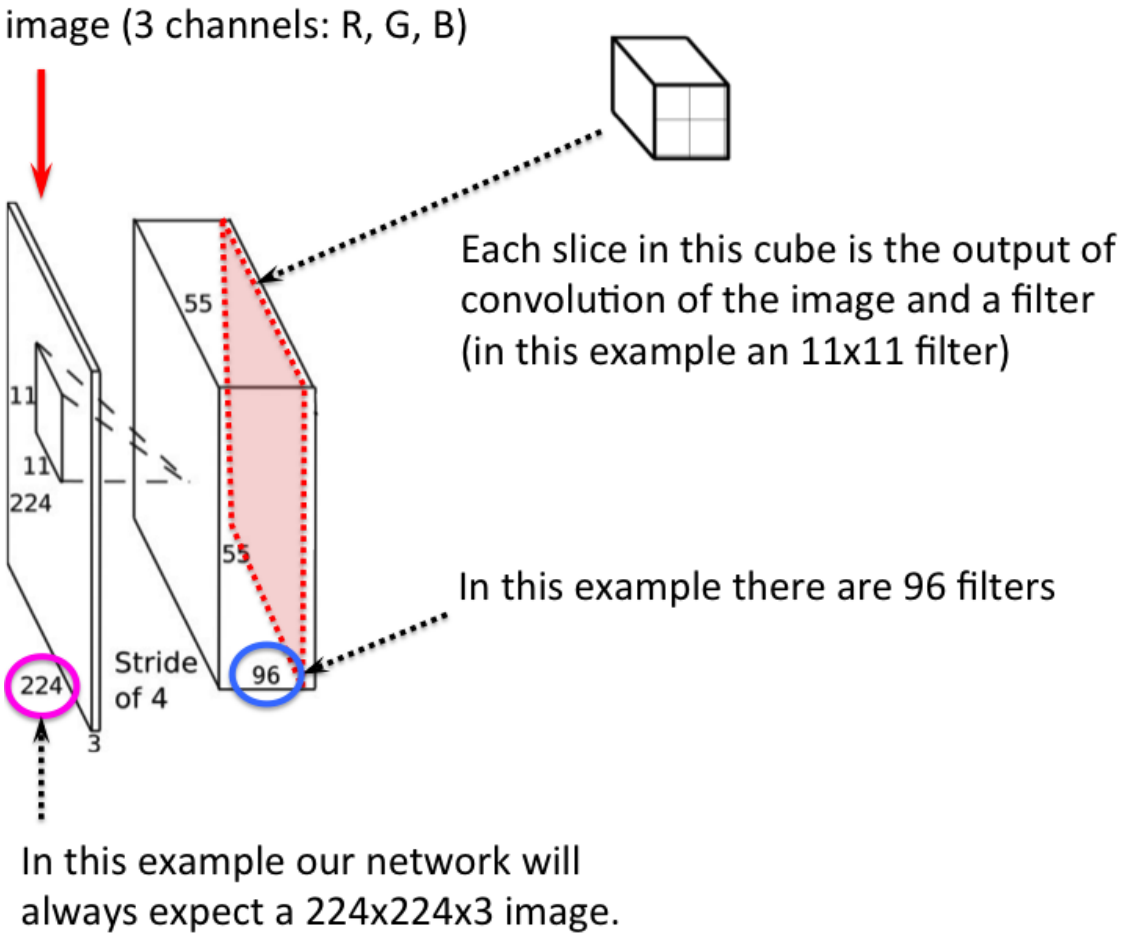
Convolutional Neural Networks (CNN)

- Now imagine we didn't only want a vertical edge detector, but also a horizontal one, and one for corners, one for dots, etc. We would need to take many filters. A **filterbank**.



Convolutional Neural Networks (CNN)

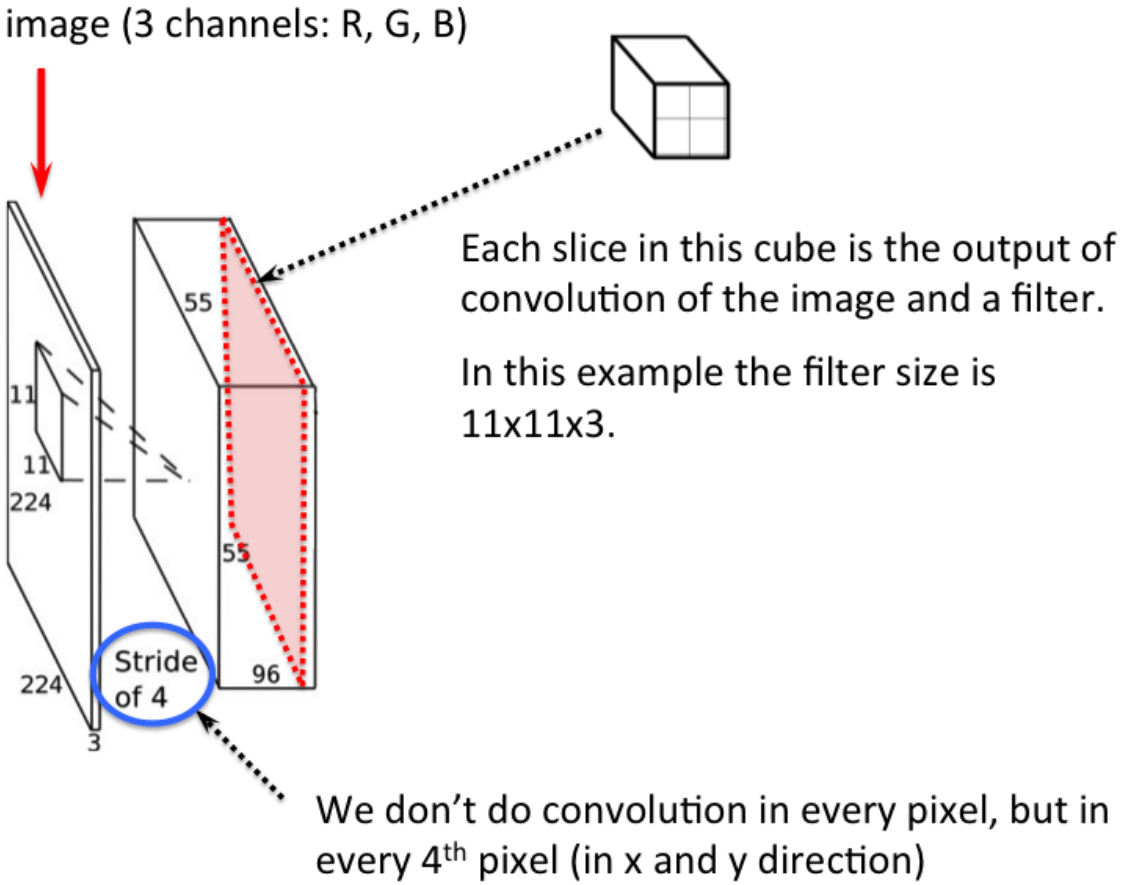
- So applying a filterbank to an image yields a cube-like output, a 3D matrix in which each slice is an output of convolution with one filter.



[Pic adopted from: A. Krizhevsky]

Convolutional Neural Networks (CNN)

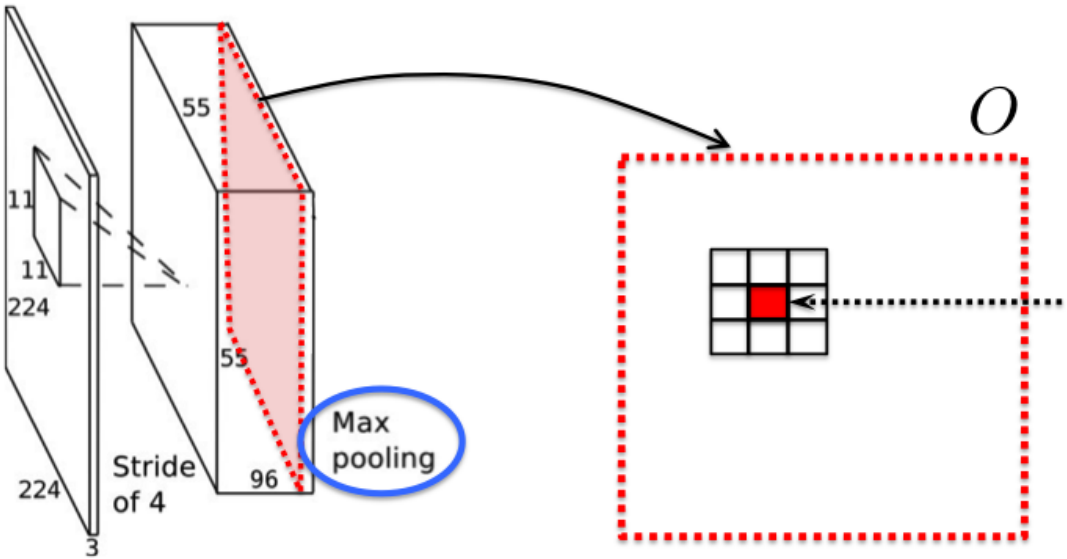
- So applying a filterbank to an image yields a cube-like output, a 3D matrix in which each slice is an output of convolution with one filter.



[Pic adopted from: A. Krizhevsky]

Convolutional Neural Networks (CNN)

- Do some additional tricks. A popular one is called **max pooling**. Any idea why you would do this?



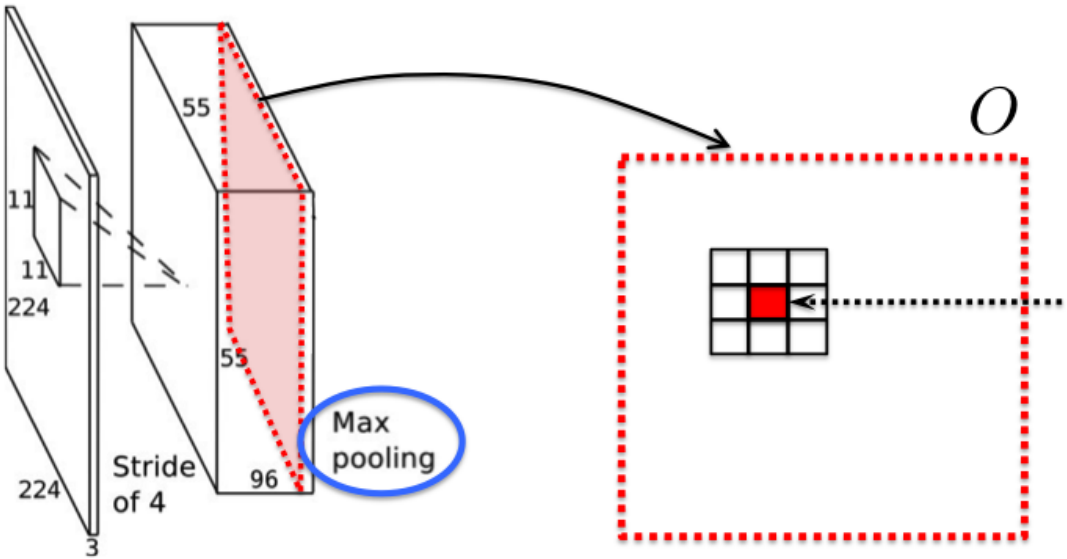
$$O(i,j) = \max_{\substack{k\in\{i-1,i,i+1\}\\ l\in\{j-1,j,j+1\}}} O(k,l)$$

Take each slice in the output cube, and in each pixel compute a max over a small patch around it. This is called **max pooling**.

[Pic adopted from: A. Krizhevsky]

Convolutional Neural Networks (CNN)

- Do some additional tricks. A popular one is called **max pooling**. Any idea why you would do this? To get **invariance to small shifts in position**.



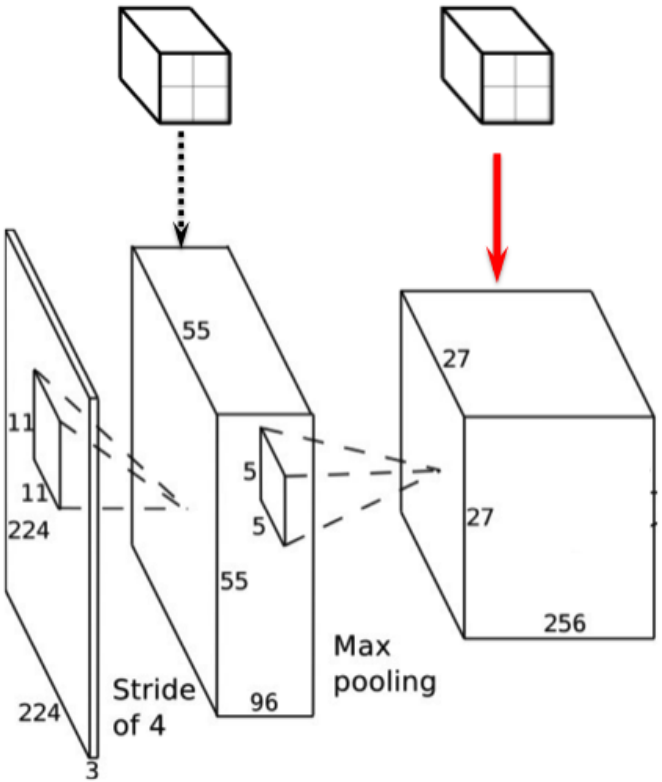
$$O(i,j) = \max_{\substack{k\in\{i-1,i,i+1\}\\ l\in\{j-1,j,j+1\}}} O(k,l)$$

Take each slice in the output cube, and in each pixel compute a max over a small patch around it. This is called **max pooling**.

[Pic adopted from: A. Krizhevsky]

Convolutional Neural Networks (CNN)

- Now add another "layer" of filters. For each filter again do convolution, but this time with the output cube of the previous layer.



Add one more layer of filters

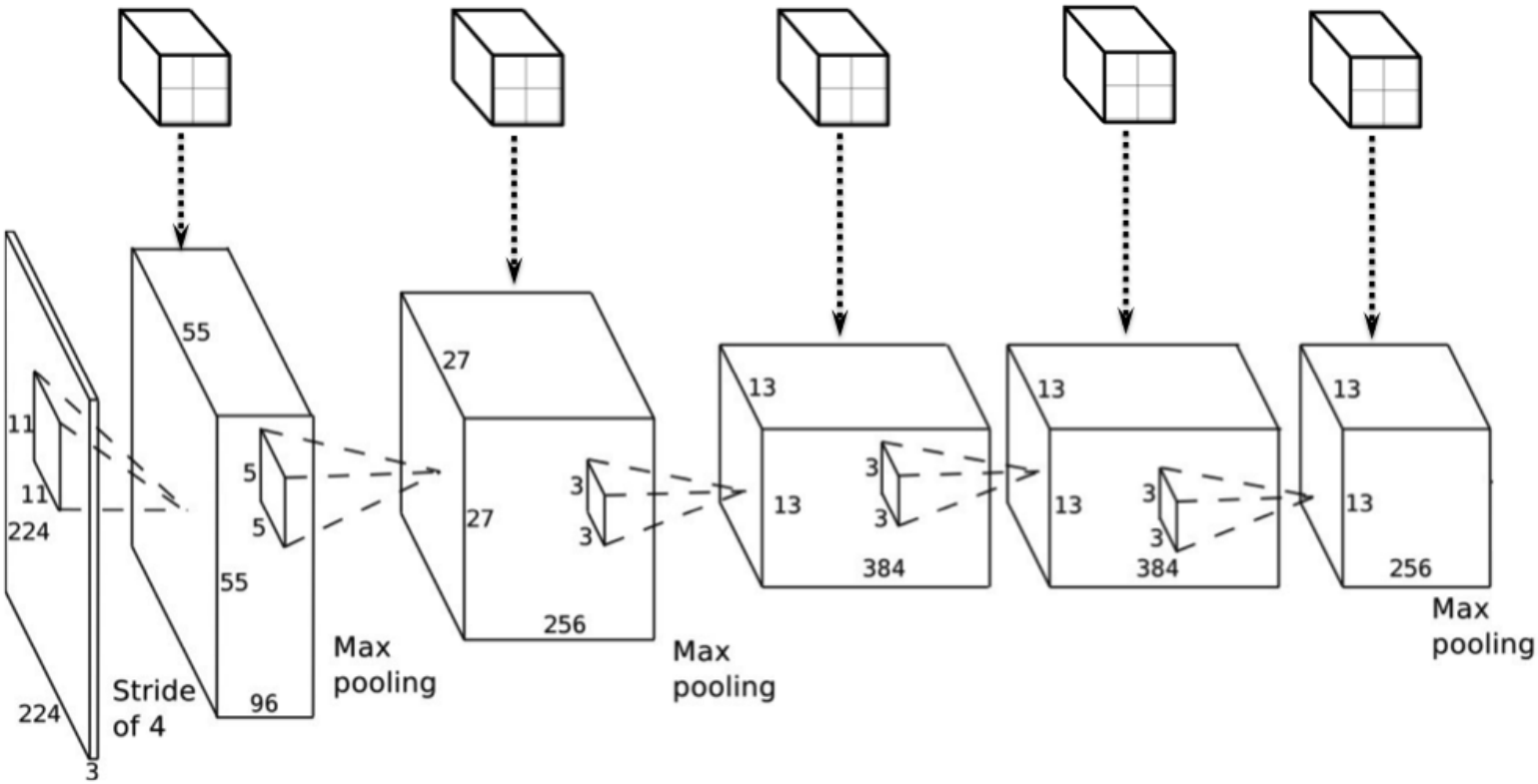
These filters are convolved with the output of the previous layer. The results of each convolution is again a slice in the cube on the right.

What is the dimension of each of these filters?

[Pic adopted from: A. Krizhevsky]

Convolutional Neural Networks (CNN)

- Keep adding a few layers.



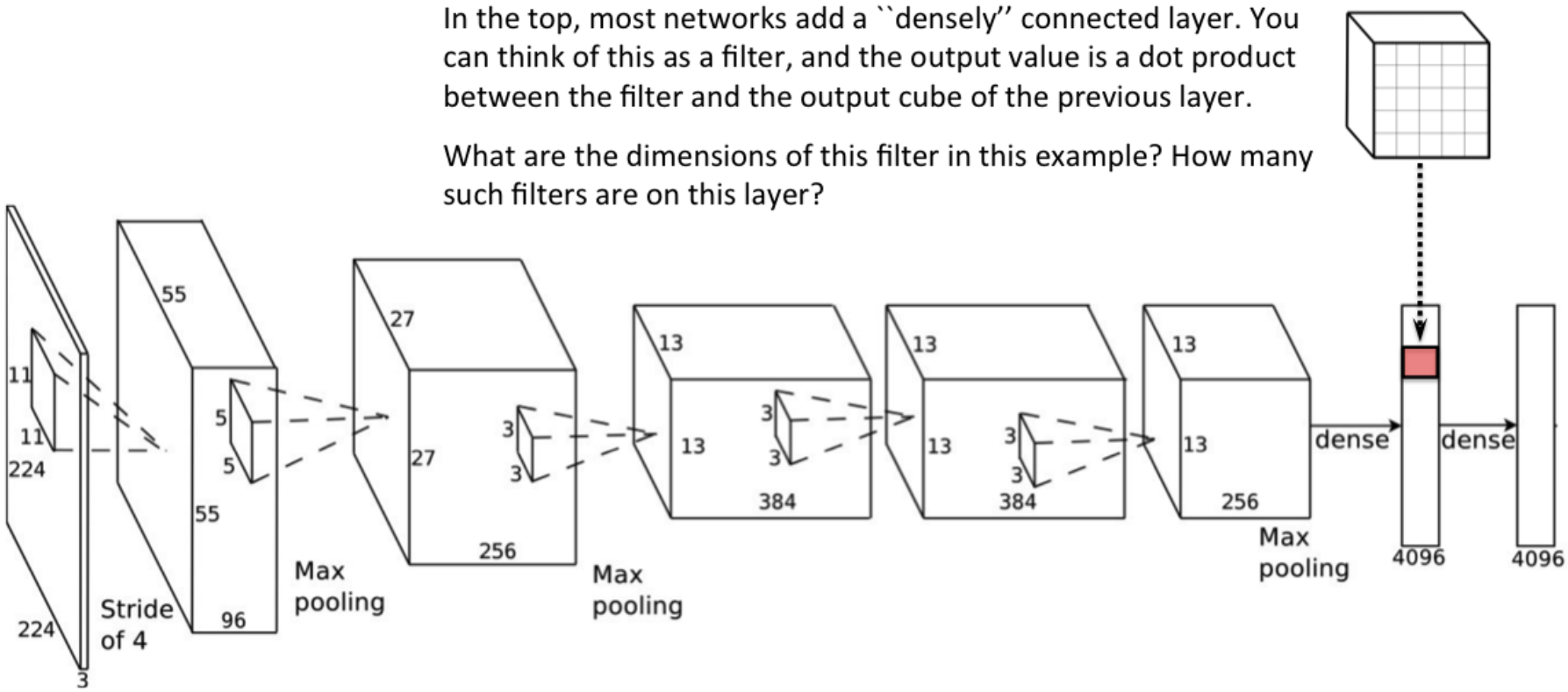
Do it repeatedly

Have multiple ``layers''

[Pic adopted from: A. Krizhevsky]

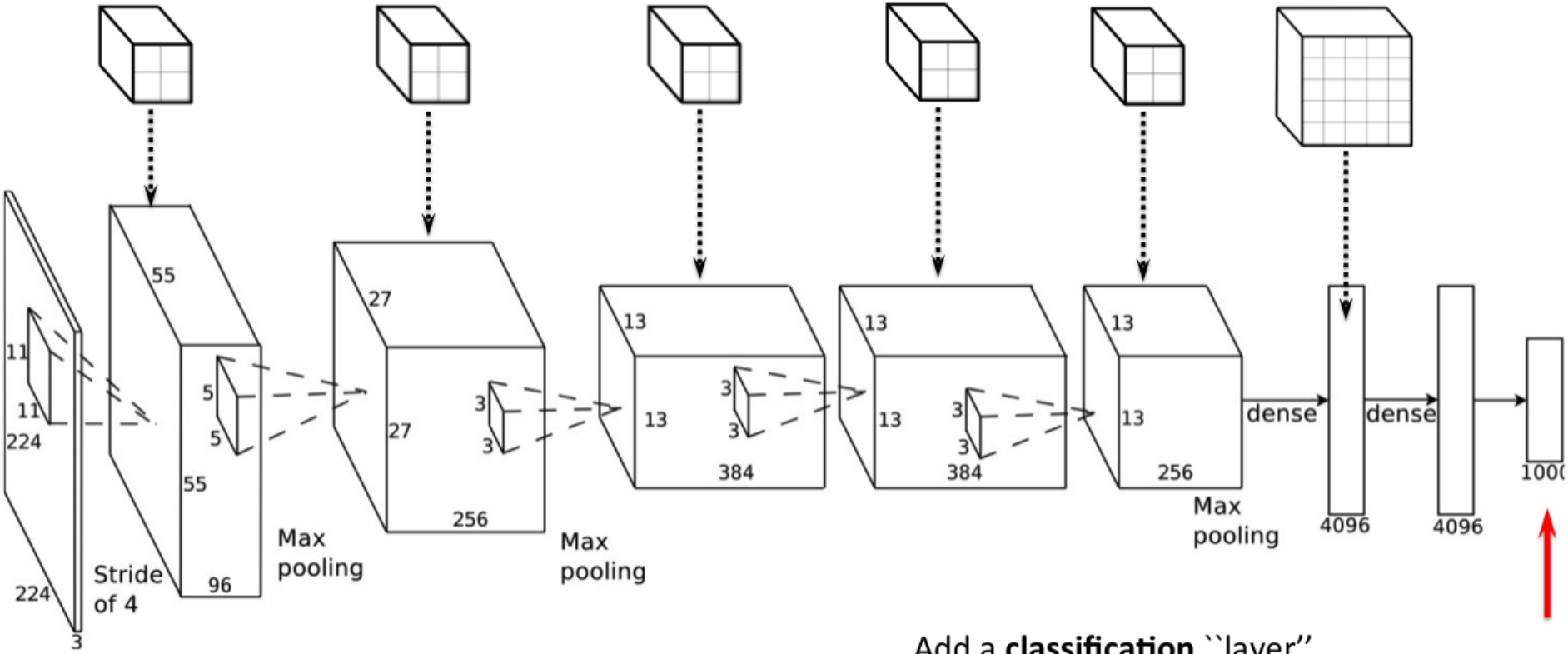
Convolutional Neural Networks (CNN)

- In the end add one or two **fully** (or **densely**) connected layers. In this layer, we don't do convolution we just do a dot-product between the "filter" and the output of the previous layer.



Convolutional Neural Networks (CNN)

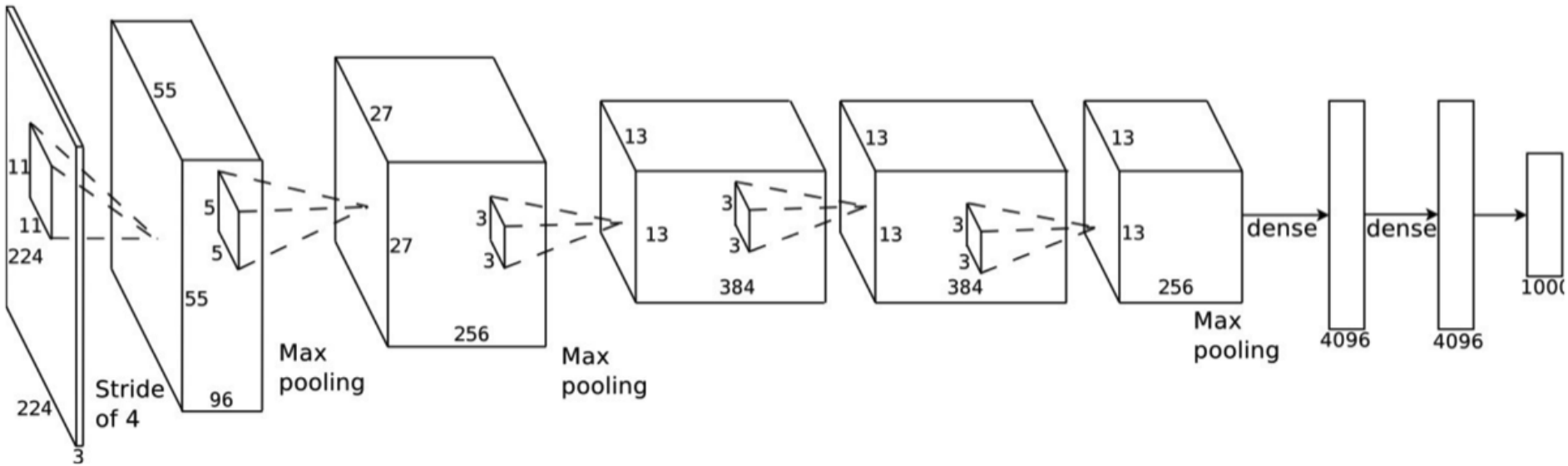
- Add one final layer: a **classification** layer. Each dimension of this vector tells us the probability of the input image being of a certain class.



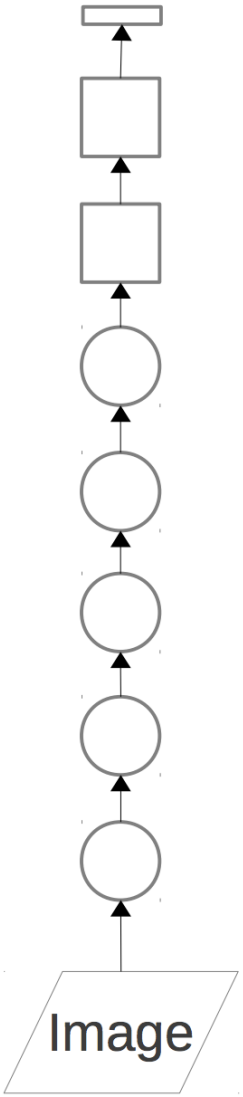
[Pic adopted from: A. Krizhevsky]

Convolutional Neural Networks (CNN)

- This fully specifies a network. The one below has been a popular choice in the last few years. It was proposed by UofT guys: A. Krizhevsky, I. Sutskever, G. E. Hinton, *ImageNet Classification with Deep Convolutional Neural Networks*, NIPS 2012. This network won the Imagenet Challenge of 2012, and revolutionized computer vision.
- How many parameters (weights) does this network have?



Convolutional Neural Networks (CNN)



- Trained with stochastic gradient descent on two NVIDIA GPUs for about a week
- 650,000 neurons
- 60,000,000 parameters
- 630,000,000 connections
- **Final feature layer:** 4096-dimensional

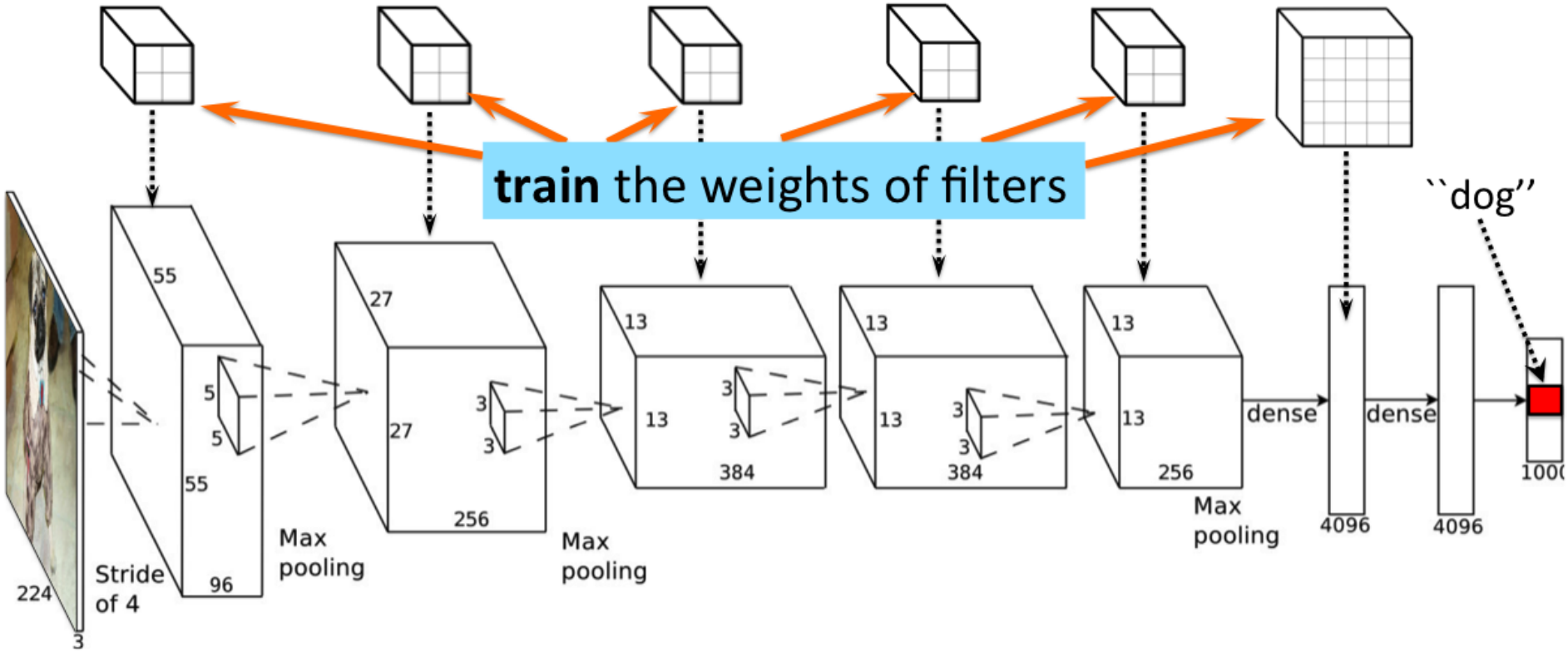
Convolutional layer: convolves its input with a bank of 3D filters, then applies point-wise non-linearity

Fully-connected layer: applies linear filters to its input, then applies point-wise non-linearity

Figure: From: http://www.image-net.org/challenges/LSVRC/2012/supervision.pdf

Convolutional Neural Networks (CNN)

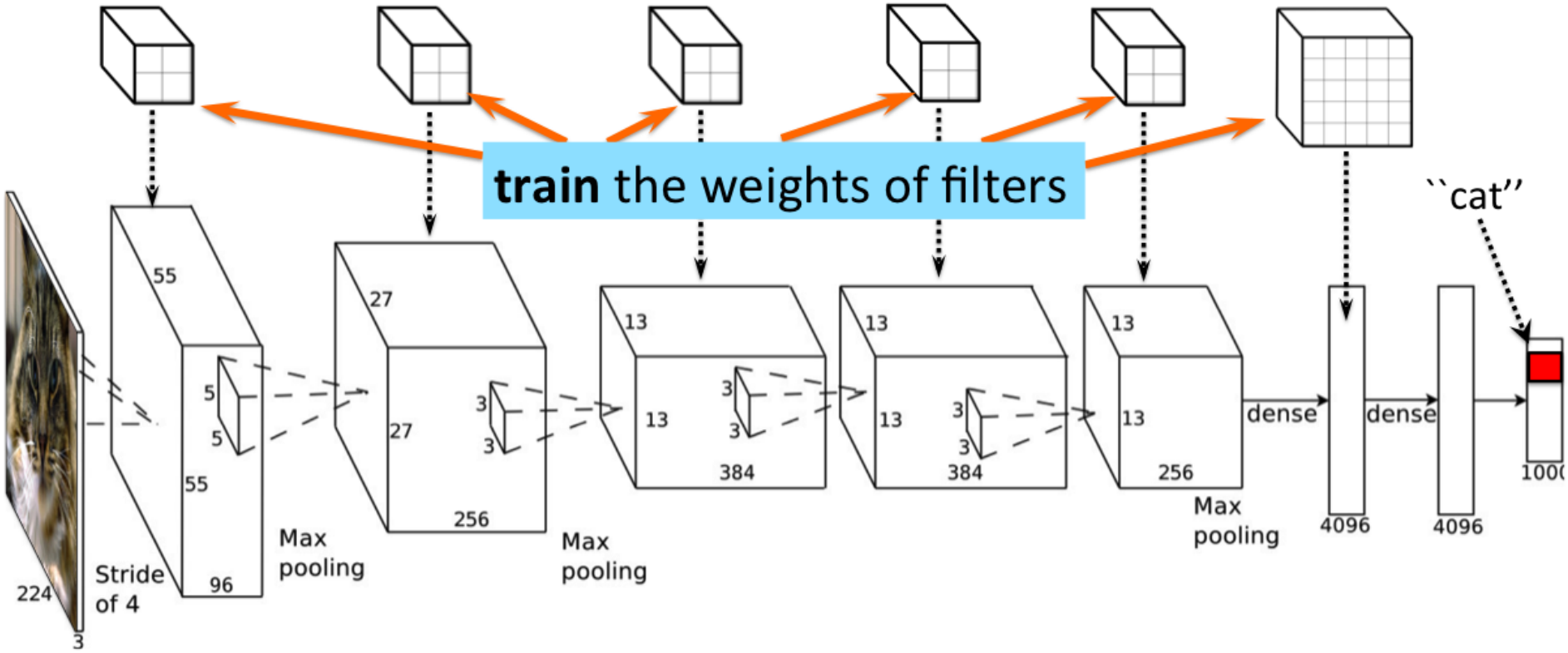
- The trick is to not hand-fix the weights, but to **train** them. Train them such that when the network sees a picture of a dog, the last layer will say "dog".



[Pic adopted from: A. Krizhevsky]

Convolutional Neural Networks (CNN)

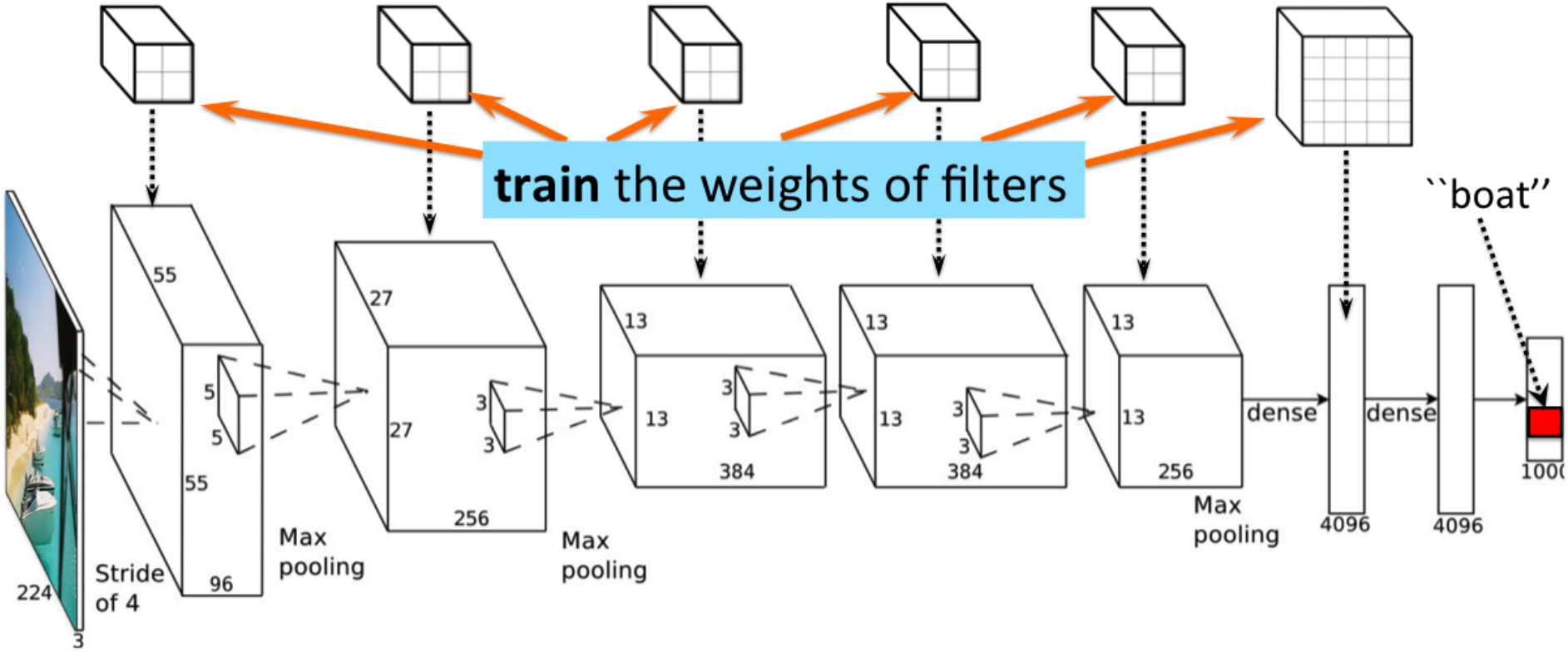
- Or when the network sees a picture of a cat, the last layer will say "cat".



[Pic adopted from: A. Krizhevsky]

Convolutional Neural Networks (CNN)

- Or when the network sees a picture of a boat, the last layer will say "boat"... The more pictures the network sees, the better.



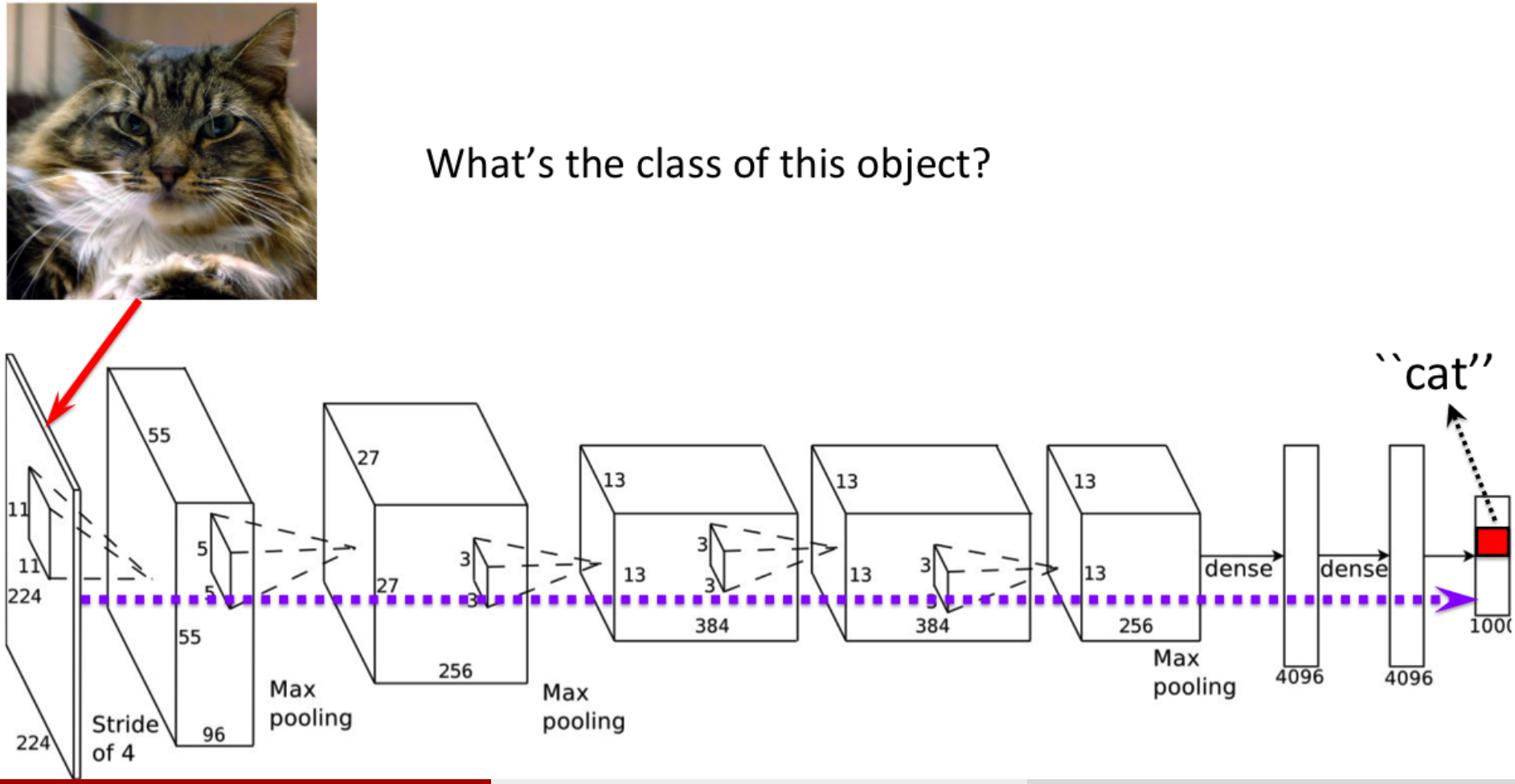
Train on **lots** of examples. Millions. Tens of millions. Wait a week for training to finish.

Share your network (the weights) with others who are not fortunate enough with GPU power.

[Pic adopted from: A. Krizhevsky]

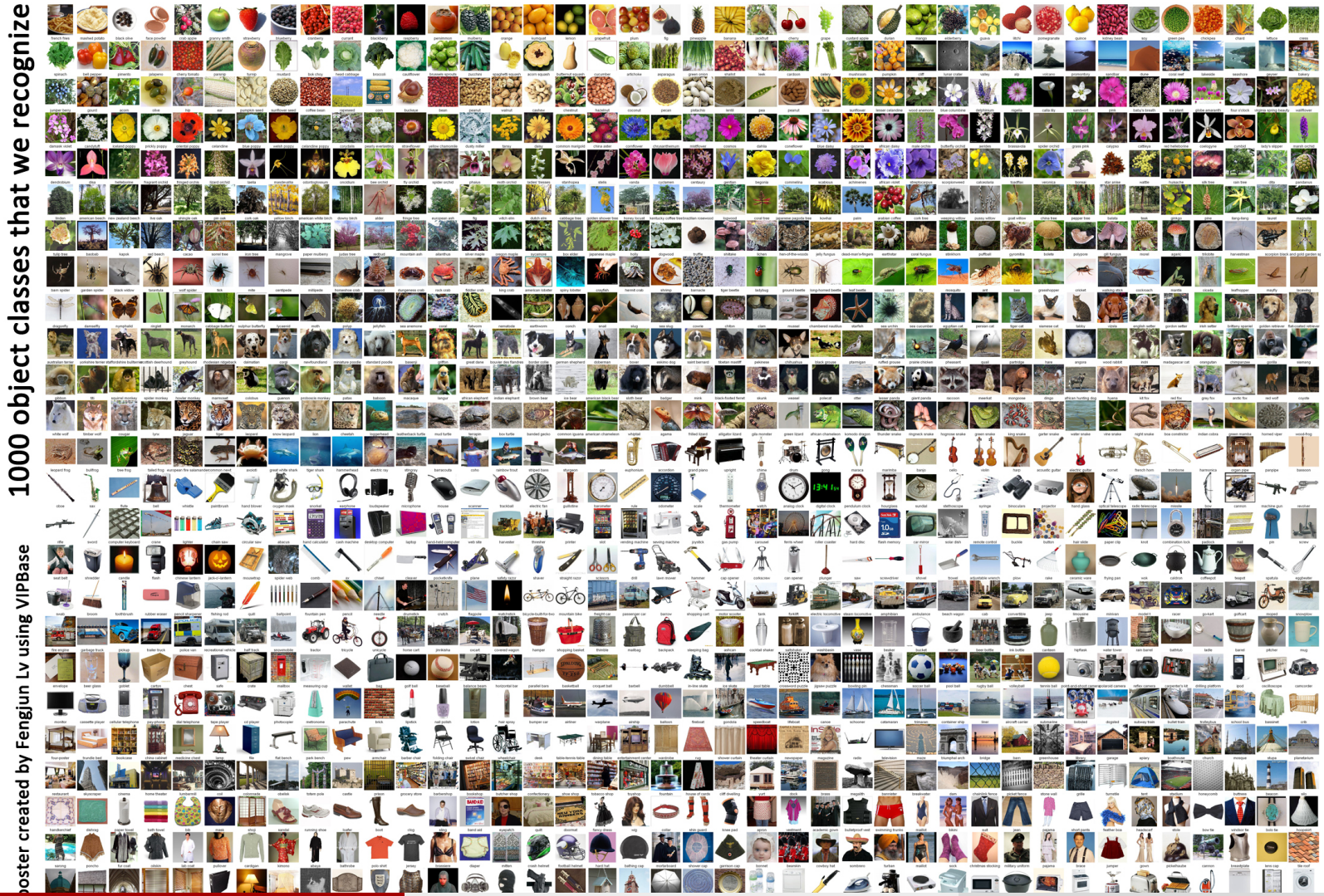
Classification

- Once trained we can do classification. Just feed in an image or a crop of the image, run through the network, and read out the class with the highest probability in the last (classification) layer.



Classification Performance

- Imagenet, main challenge for object classification: http://image-net.org/
- 1000 classes, 1.2M training images, 150K for test



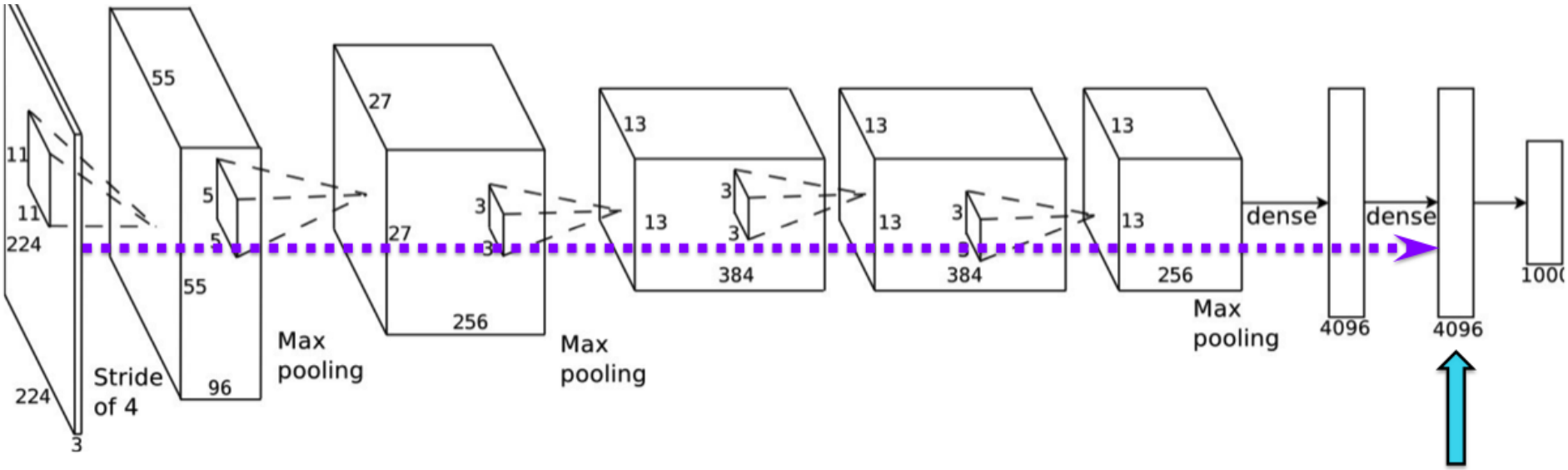
Classification Performance (2012)

- A. Krizhevsky, I. Sutskever, and G. E. Hinton won the Imagenet Challenge

Team name	Filename	Error (5 guesses)	Description
SuperVision	test-preds-141-146.2009-131-137-145-146.2011-145f.	0.15315	Using extra training data from ImageNet Fall 2011 release
SuperVision	test-preds-131-137-145-135-145f.txt	0.16422	Using only supplied training data
ISI	pred_FVs_wLACs_weighted.txt	0.26172	Weighted sum of scores from each classifier with SIFT+FV, LBP+FV, GIST+FV, and CSIFT+FV, respectively.
ISI	pred_FVs_weighted.txt	0.26602	Weighted sum of scores from classifiers using each FV.
ISI	pred_FVs_summed.txt	0.26646	Naive sum of scores from classifiers using each FV.
ISI	pred_FVs_wLACs_summed.txt	0.26952	Naive sum of scores from each classifier with SIFT+FV, LBP+FV, GIST+FV, and

Neural Networks as Descriptors

- What vision people like to do is take the already trained network (avoid one week of training), and remove the last classification layer. Then take the top remaining layer (the 4096 dimensional vector here) and use it as a descriptor (feature vector).

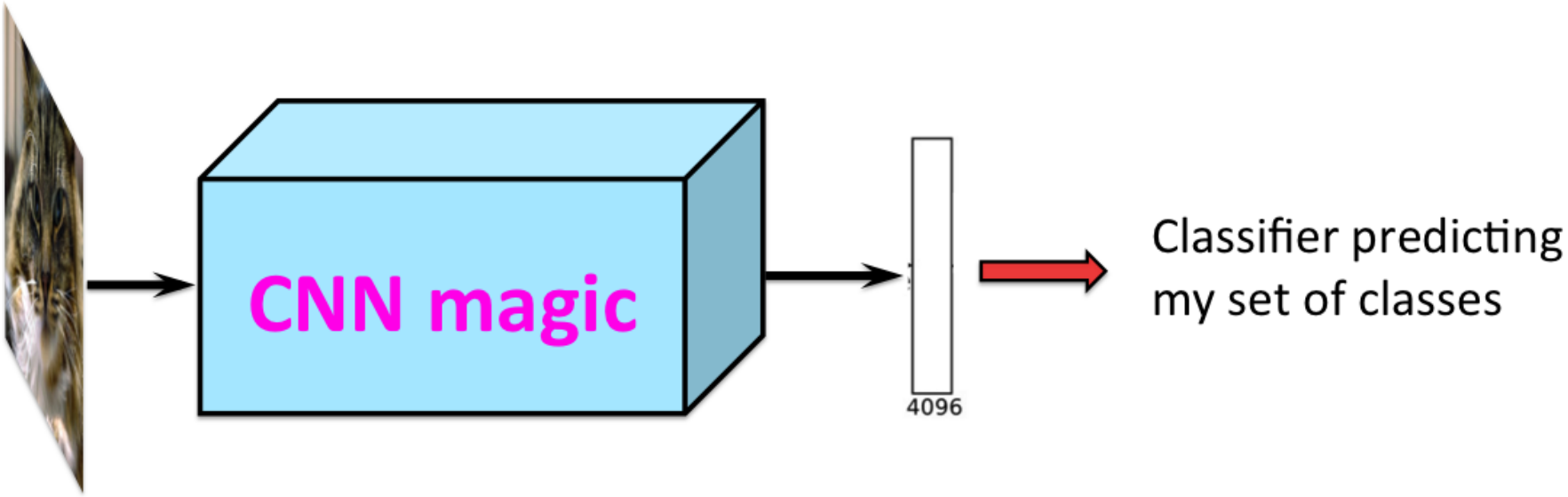


Vision people are mainly interested in this vector. **You can use it as a descriptor**. A much better descriptor than SIFT, etc.

Train your own classifier on top for your choice of classes.

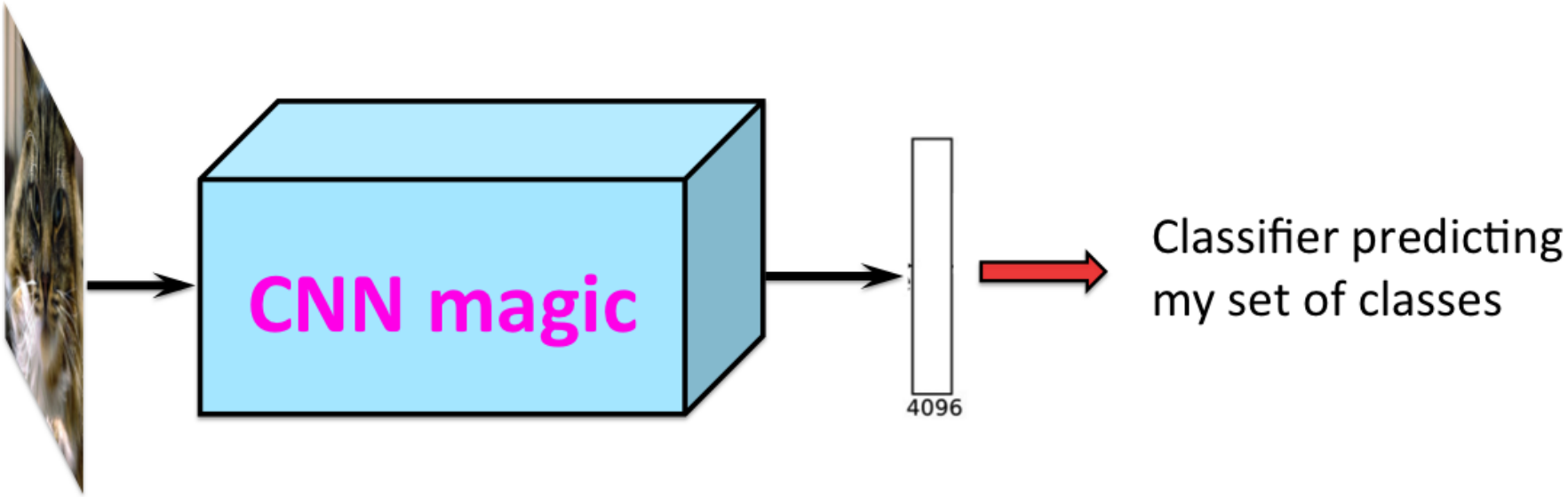
Neural Networks as Descriptors

- What vision people like to do is take the already trained network, and remove the last classification layer. Then take the top remaining layer (the 4096 dimensional vector here) and use it as a descriptor (feature vector).
- Now train your own classifier on top of these features for arbitrary classes.



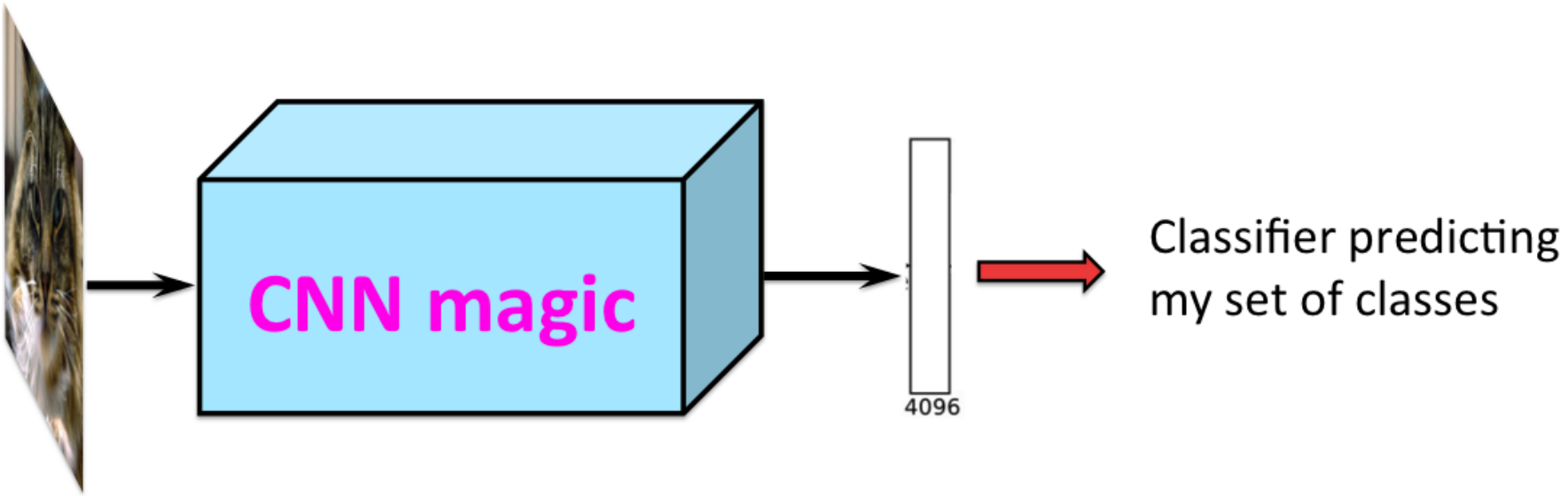
Neural Networks as Descriptors

- What vision people like to do is take the already trained network, and remove the last classification layer. Then take the top remaining layer (the 4096 dimensional vector here) and use it as a descriptor (feature vector).
- Now train your own classifier on top of these features for arbitrary classes.
- This is quite hacky, but works miraculously well.



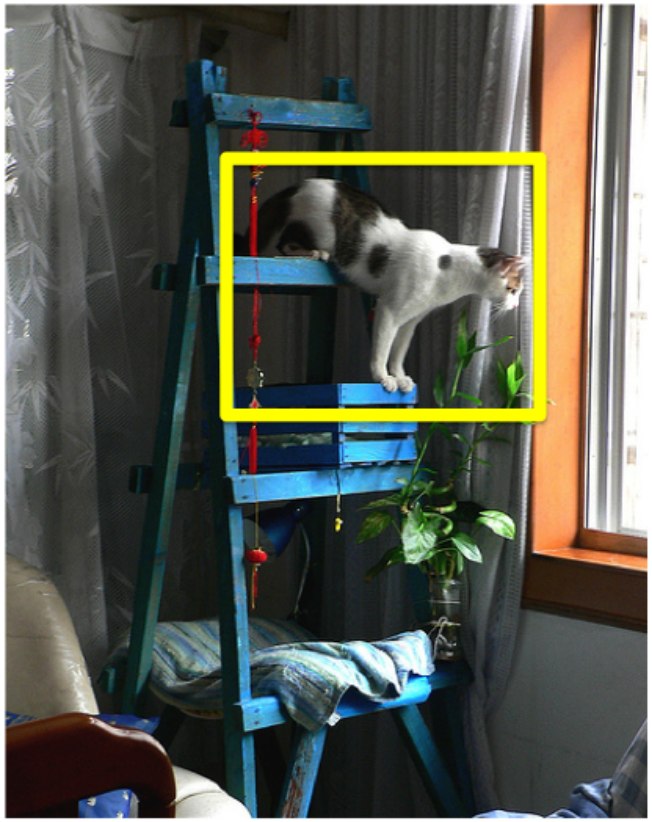
Neural Networks as Descriptors

- What vision people like to do is take the already trained network, and remove the last classification layer. Then take the top remaining layer (the 4096 dimensional vector here) and use it as a descriptor (feature vector).
- Now train your own classifier on top of these features for arbitrary classes.
- This is quite hacky, but works miraculously well.
- Everywhere where we were using SIFT (or anything else), you can use NNs.



And Detection?

- For classification we feed in the full image to the network. But how can we perform detection?



Find all objects of interest in this image!

And Detection?

- Generate lots of proposal bounding boxes (rectangles in image where we think any object could be)
- Each of these boxes is obtained by grouping similar clusters of pixels

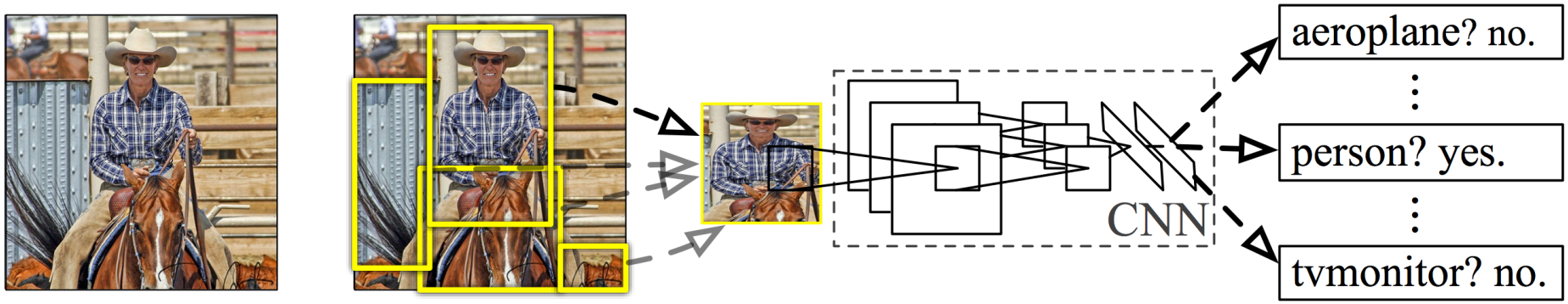


Figure: R. Girshick, J. Donahue, T. Darrell, J. Malik, Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation, CVPR'14

And Detection?

- Generate lots of proposal bounding boxes (rectangles in image where we think any object could be)
- Each of these boxes is obtained by grouping similar clusters of pixels
- Crop image out of each box, warp to fixed size (224×224) and run through the network

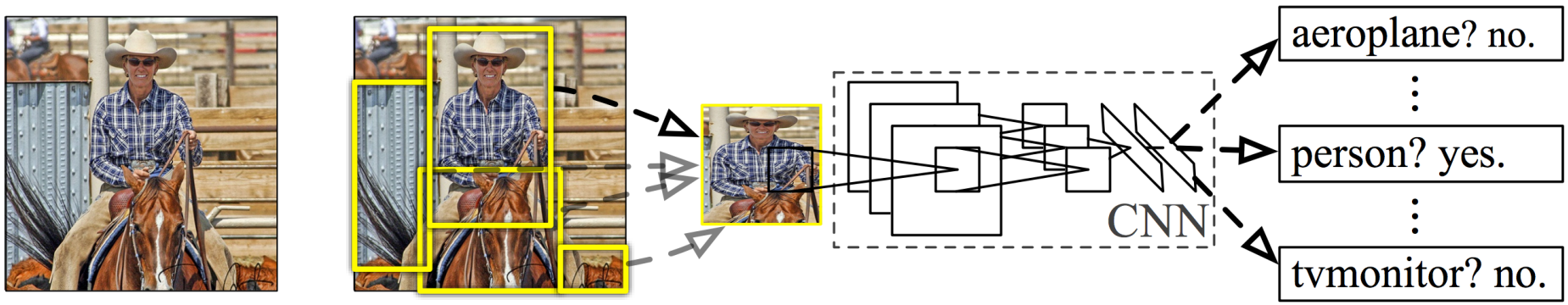


Figure: R. Girshick, J. Donahue, T. Darrell, J. Malik, Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation, CVPR'14

And Detection?

- Generate lots of proposal bounding boxes (rectangles in image where we think any object could be)
- Each of these boxes is obtained by grouping similar clusters of pixels
- Crop image out of each box, warp to fixed size (224×224) and run through the network.
- If the warped image looks weird and doesn't resemble the original object, don't worry. Somehow the method still works.
- This approach, called R-CNN, was proposed in 2014 by Girshick et al.

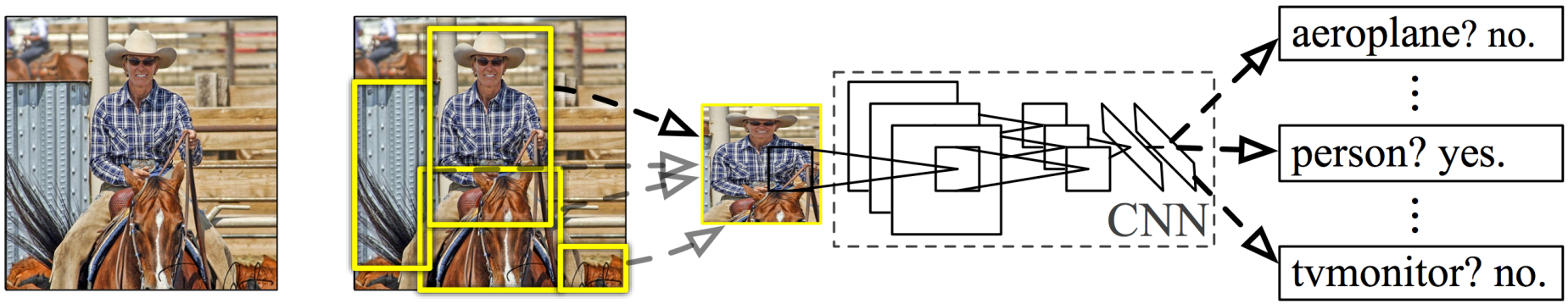


Figure: R. Girshick, J. Donahue, T. Darrell, J. Malik, Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation, CVPR'14

And Detection?

- One way of getting the proposal boxes is by hierarchical merging of regions. This particular approach, called Selective Search, was proposed in 2011 by Uijlings et al. We will talk more about this later in class.

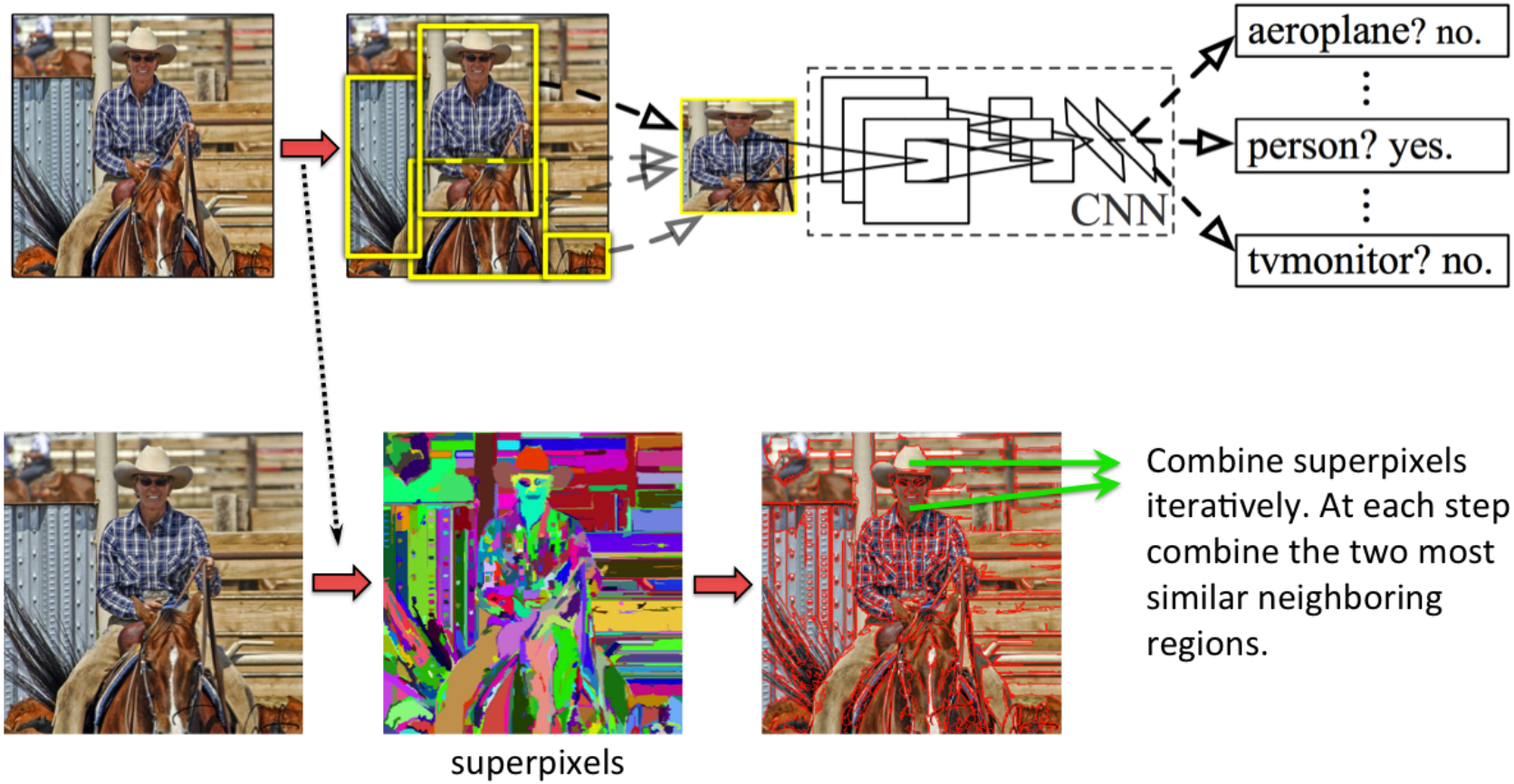


Figure: Bottom: J. R. R. Uijlings, K. E. A. van de Sande, T. Gevers, A. W. M. Smeulders, Selective Search for Object Recognition, IJCV 2013

And Detection?

- One way of getting the proposal boxes is by hierarchical merging of regions. This particular approach, called Selective Search, was proposed in 2011 by Uijlings et al. We will talk more about this later in class.

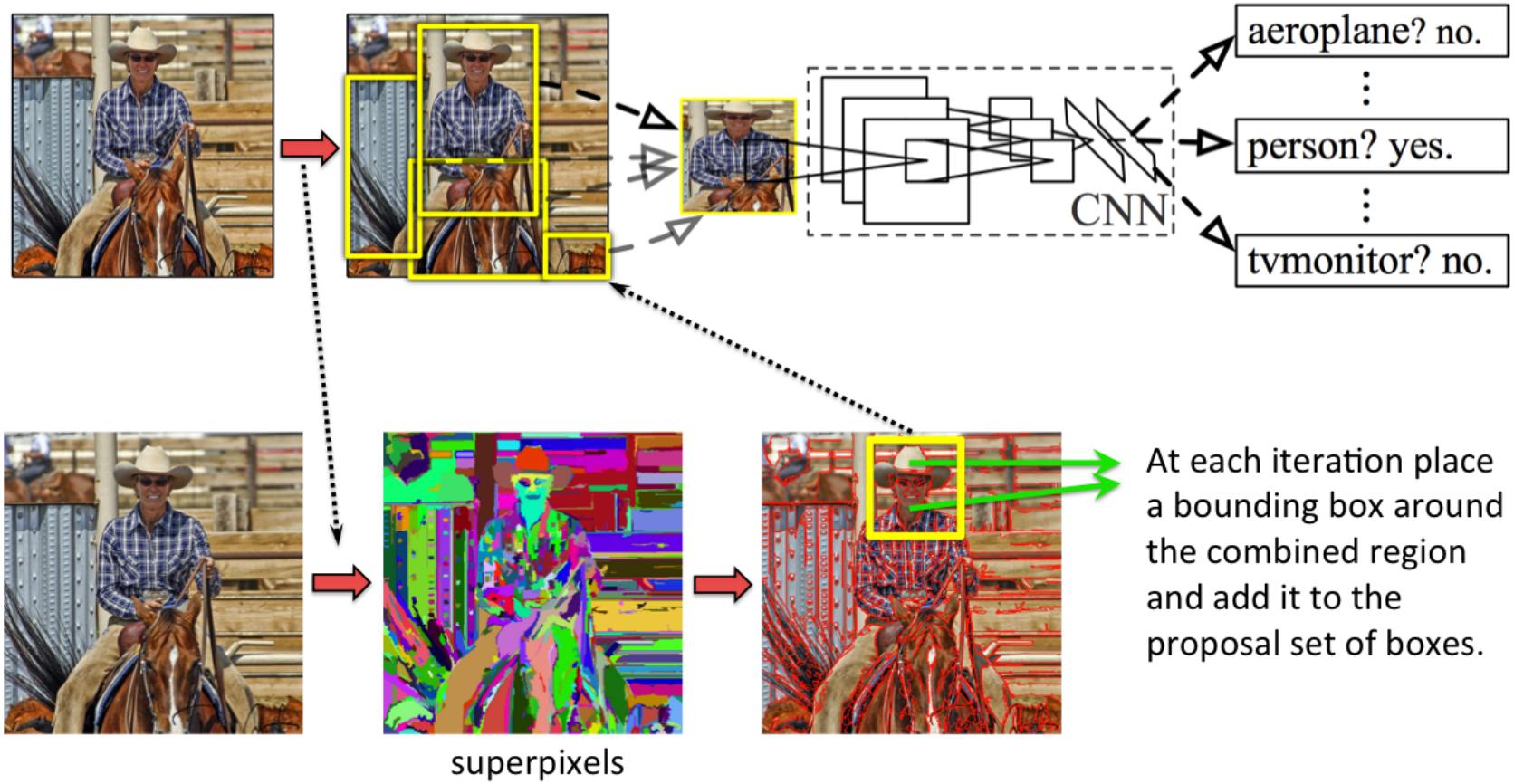


Figure: Bottom: J. R. R. Uijlings, K. E. A. van de Sande, T. Gevers, A. W. M. Smeulders, Selective Search for Object Recognition, IJCV 2013

Detection Datasets

- **PASCAL VOC challenge**: http://pascallin.ecs.soton.ac.uk/challenges/VOC/.

Figure: PASCAL has 20 object classes, 10K images for training, 10K for test

Detection Performance 2011-2012: 40.4%

2011-2012, (a year before the resurgence of neural networks

- Results on the main recognition benchmark, the **PASCAL VOC challenge**.

	mean	aero plane	bicycle	bird	boat	bottle	bus	car	cat	chair	cow	dining table	dog	horse	motor bike	person	potted plant	sheep	sofa	train	tv/ monitor	submission date
segDPM [?]	40.4	**61.4**	53.4	25.6	25.2	**35.5**	51.7	50.6	50.8	19.3	33.8	26.8	40.4	48.3	54.4	47.1	14.8	38.7	35.0	52.8	43.1	24-Feb-2014
Boosted HOG-LBP and multi-context (LC, EGC, HLC) [?]	36.8	53.3	**55.3**	19.2	21.0	30.0	54.5	46.7	41.2	**20.0**	31.5	20.8	30.3	48.6	55.3	46.5	10.2	34.4	26.6	50.3	40.3	29-Aug-2010
MITUCLA_Hierarchy [?]	36.0	54.3	48.5	15.7	19.2	29.2	55.6	43.5	41.7	16.9	28.5	26.7	30.9	48.3	55.0	41.7	9.7	35.8	30.8	47.2	40.8	30-Aug-2010
HOGLBP_context_classification_rescore_v2 [?]	34.2	49.1	52.4	17.8	12.0	30.6	53.5	32.8	37.3	17.7	30.6	27.7	29.5	**51.9**	56.3	44.2	9.6	14.8	27.9	49.5	38.4	30-Aug-2010
LSVM-MDPM [?]	33.7	52.4	54.3	13.0	15.6	35.1	54.2	49.1	31.8	15.5	26.2	13.5	21.5	45.4	51.6	47.5	9.1	35.1	19.4	46.6	38.0	26-Aug-2010
UOCTTI_LSVM_MDPM [?]	33.4	49.2	53.8	13.1	15.3	**35.5**	53.4	49.7	27.0	17.2	28.8	14.7	17.8	46.4	51.2	**47.7**	10.8	34.2	20.7	43.8	38.3	21-May-2012
Detection Monkey [?]	32.9	56.7	39.8	16.8	12.2	13.8	44.9	36.9	47.7	12.1	26.9	26.5	37.2	42.1	51.9	25.7	12.1	37.8	33.0	41.5	41.7	30-Aug-2010
RMA2C [?]	32.8	49.8	50.6	15.1	15.5	28.5	51.1	42.2	30.5	17.3	28.3	12.4	26.0	45.6	51.8	41.4	12.6	30.4	26.1	44.0	37.6	29-Oct-2013
UOCTTI_LSVM_MDPM [?]	32.2	48.2	52.2	14.8	13.8	28.7	53.2	44.9	26.0	18.4	24.4	13.7	23.1	45.8	50.5	43.7	9.8	31.1	21.5	44.4	35.7	11-May-2012
GroupLoc [?]	31.9	58.4	39.6	18.0	13.3	11.1	46.4	37.8	43.9	10.3	27.5	20.8	36.0	39.4	48.5	22.9	13.0	36.9	30.5	41.2	41.9	30-Aug-2010
UOCTTI_LSVM_MDPM [?]	29.6	45.6	49.0	11.0	11.6	27.2	50.5	43.1	23.6	17.2	23.2	10.7	20.5	42.5	44.5	41.3	8.7	29.0	18.7	40.0	34.5	21-May-2012
Bonn_FGT_Segm [?]	26.1	52.7	33.7	13.2	11.0	14.2	43.2	31.9	35.6	5.8	25.4	14.4	20.6	38.1	41.7	25.0	5.8	26.3	18.1	37.6	28.1	30-Aug-2010
HOG-LBP + DHOG bag of words, SVM [?]	23.5	40.4	34.7	2.7	8.4	26.0	43.1	33.8	17.2	11.2	14.3	14.5	14.9	31.8	37.3	30.0	6.4	25.2	11.6	30.0	35.7	30-Aug-2010
Svr-Segm [?]	23.4	50.5	24.5	17.1	13.3	10.9	39.5	32.9	36.5	5.6	16.0	6.6	22.3	24.9	29.0	29.8	6.7	28.4	13.3	32.1	27.2	30-Aug-2010
HOG-LBP Linear SVM [?]	22.1	37.9	33.7	2.7	6.5	25.3	37.5	33.1	15.5	10.9	12.3	12.5	13.7	29.7	34.5	33.8	7.2	22.9	9.9	28.9	34.1	29-Aug-2010
HOG+LBP+LTP+PLS2ROOTS [?]	17.5	32.7	29.7	0.8	1.1	19.9	39.4	27.5	8.6	4.5	8.1	6.3	11.0	22.9	34.1	24.6	3.1	24.0	2.0	23.5	27.0	31-Aug-2010
RandomParts [?]	14.2	23.8	31.7	1.2	3.4	11.1	29.7	19.5	14.2	0.8	11.1	7.0	4.7	16.4	31.5	16.0	1.1	15.6	10.2	14.7	21.0	25-Aug-2010
SIFT-GMM-MKL2 [?]	8.3	20.0	14.5	3.8	1.2	0.5	17.6	8.1	28.5	0.1	2.9	3.1	17.5	7.2	18.8	3.3	0.8	2.9	6.3	7.6	1.1	30-Aug-2010
UC3M_Generative_Discriminative [?]	6.3	15.8	5.5	5.6	2.3	0.3	10.2	5.4	12.6	0.5	5.6	4.5	7.7	11.3	12.6	5.3	1.5	2.0	5.9	9.1	3.2	30-Aug-2010
SIFT-GMM-MKL [?]	2.3	10.6	1.6	1.2	0.9	0.1	2.8	1.6	6.7	0.1	2.0	0.4	3.0	2.0	4.4	2.0	0.3	1.1	1.2	2.1	1.9	30-Aug-2010

Detection Performance 2013-14: 53.7%

2013-14, one year after the resurgence of neural networks

- Results on the main recognition benchmark, the **PASCAL VOC challenge**.

	mean	aero plane	bicycle	bird	boat	bottle	bus	car	cat	chair	cow	dining table	dog	horse	motor bike	person	potted plant	sheep	sofa	train	tv/ monitor	submission date
R-CNN (bbox reg)	53.7	71.8	65.8	53.0	36.8	35.9	59.7	60.0	69.9	27.9	50.6	41.4	70.0	62.0	69.0	58.1	29.5	59.4	39.3	61.2	52.4	2014-Mar-13
R-CNN	50.2	67.1	64.1	46.7	32.0	30.5	56.4	57.2	65.9	27.0	47.3	40.9	66.6	57.8	65.9	53.6	26.7	56.5	38.1	52.8	50.2	2014-Jan-30

Figure: Leading method R-CNN is by Girshick et al.

R. Girshick, J. Donahue, T. Darrell, J. Malik, Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation, CVPR'14

So Neural Networks are Great

- So networks turn out to be great.
- At this point Google, Facebook, Microsoft, Baidu "steal" most neural network professors from academia.

So Neural Networks are Great

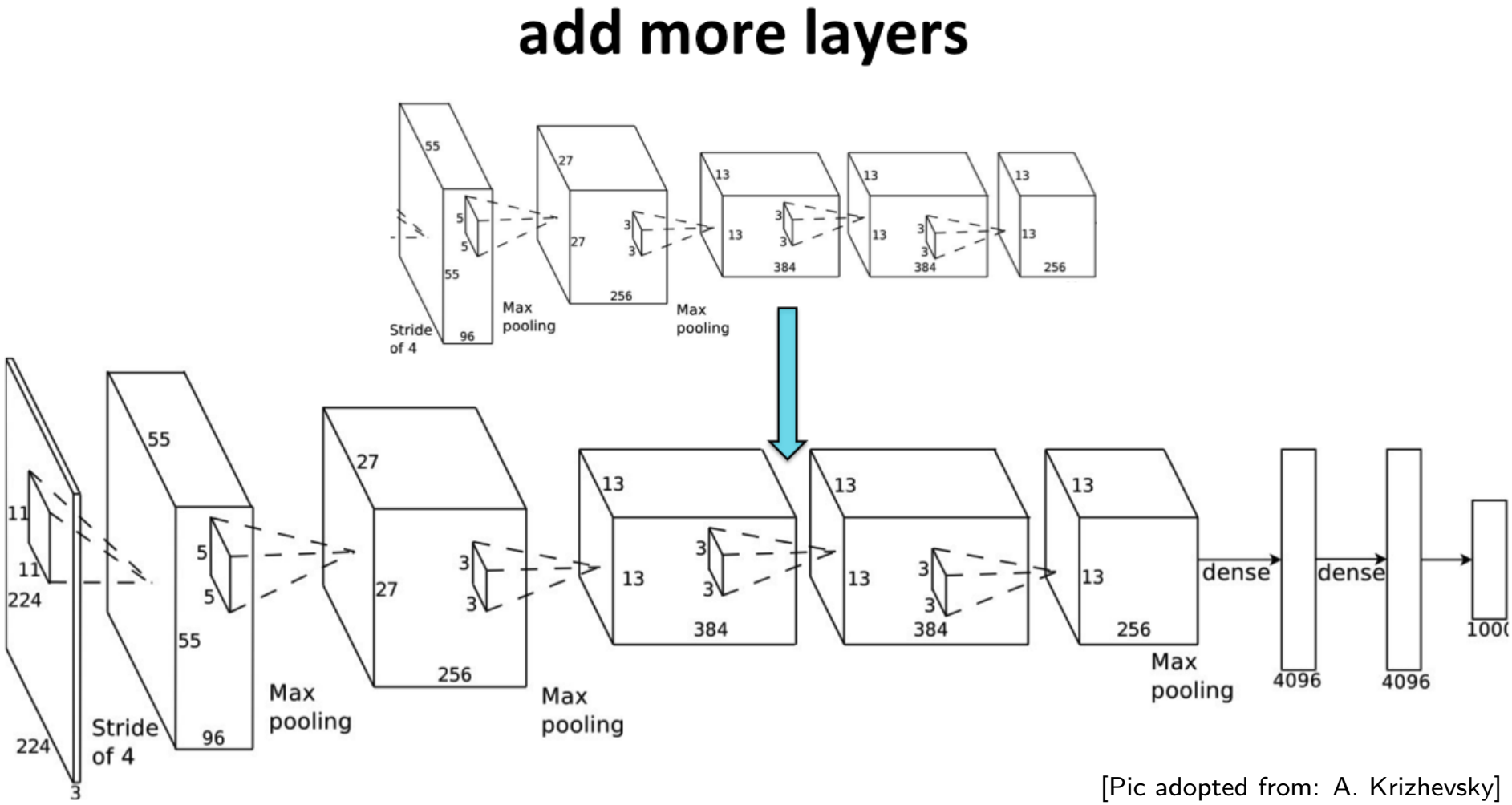
- But to train the networks you need quite a bit of computational power. So what do you do?

So Neural Networks are Great

- Buy even more.

So Neural Networks are Great

- And train **more layers**. 16 instead of 7 before. 144 million parameters.



[Pic adopted from: A. Krizhevsky]

Detection Performance 2014-15: 62.9%

even bigger networks:

- Results on the main recognition benchmark, the **PASCAL VOC challenge**

	mean	plane	bicycle	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	bike	person	plant	sheep	sofa	train	monitor	date
R-CNN (bbox reg) [?]	62.9	79.3	72.4	63.1	44.0	44.4	64.6	66.3	84.9	38.8	67.3	48.4	82.3	75.0	76.7	65.7	35.8	66.2	54.8	69.1	58.8	27-Oct-2014
R-CNN [?]	59.8	76.5	70.4	58.0	40.2	39.6	61.8	63.7	81.0	36.2	64.5	45.7	80.5	71.9	74.3	60.6	31.5	64.7	52.5	64.6	57.2	27-Oct-2014
Feature Edit [?]	56.4	74.8	69.2	55.7	41.9	36.1	64.7	62.3	69.5	31.3	53.3	43.7	69.9	64.0	71.8	60.5	32.7	63.0	44.1	63.6	56.6	04-Sep-2014
R-CNN (bbox reg) [?]	53.7	71.8	65.8	53.0	36.8	35.9	59.7	60.0	69.9	27.9	50.6	41.4	70.0	62.0	69.0	58.1	29.5	59.4	39.3	61.2	52.4	13-Mar-2014

Figure: Leading method R-CNN is by Girshick et al.

R. Girshick, J. Donahue, T. Darrell, J. Malik, Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation, CVPR'14

Detection Performance 1 year ago: 70.8%

1 year ago, networks:

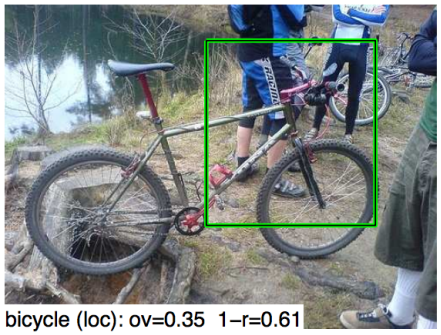
- Results on the main recognition benchmark, the **PASCAL VOC challenge**.

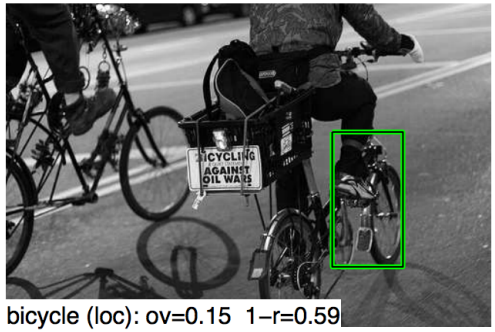
	mean	aero plane	bicycle	bird	boat	bottle	bus	car	cat	chair	cow	dining table	dog	horse	motor bike	person	potted plant	sheep	sofa	train	tv/ monitor	submission date
Fast R-CNN + YOLO [?]	**70.8**	**82.7**	77.7	**74.3**	**59.1**	47.1	**78.0**	**73.1**	89.2	**49.6**	**74.3**	**55.9**	87.4	79.8	82.2	**75.3**	**43.1**	71.4	**67.8**	**81.9**	**65.6**	05-Jun-2015
Fast R-CNN VGG16 extra data [?]	68.8	82.0	**77.8**	71.6	55.3	42.4	77.3	71.7	**89.3**	44.5	72.1	53.7	**87.7**	**80.0**	**82.5**	72.7	36.6	68.7	65.4	81.1	62.7	18-Apr-2015
segDeepM [?]	67.2	82.3	75.2	67.1	50.7	**49.8**	71.1	69.6	88.2	42.5	71.2	50.0	85.7	76.6	81.8	69.3	41.5	**71.9**	62.2	73.2	64.6	29-Jan-2015
BabyLearning [?]	63.8	77.7	73.8	62.3	48.8	45.4	67.3	67.0	80.3	41.3	70.8	49.7	79.5	74.7	78.6	64.5	36.0	69.9	55.7	70.4	61.7	12-Nov-2014
R-CNN (bbox reg) [?]	62.9	79.3	72.4	63.1	44.0	44.4	64.6	66.3	84.9	38.8	67.3	48.4	82.3	75.0	76.7	65.7	35.8	66.2	54.8	69.1	58.8	27-Oct-2014
R-CNN [?]	59.8	76.5	70.4	58.0	40.2	39.6	61.8	63.7	81.0	36.2	64.5	45.7	80.5	71.9	74.3	60.6	31.5	64.7	52.5	64.6	57.2	27-Oct-2014
Feature Edit [?]	56.4	74.8	69.2	55.7	41.9	36.1	64.7	62.3	69.5	31.3	53.3	43.7	69.9	64.0	71.8	60.5	32.7	63.0	44.1	63.6	56.6	04-Sep-2014
YOLO [?]	55.3	72.3	64.8	56.0	38.8	26.5	61.4	54.8	78.2	35.6	58.1	41.2	72.7	68.9	70.9	62.9	27.1	52.6	47.4	68.8	46.7	05-Jun-2015
R-CNN (bbox reg) [?]	53.7	71.8	65.8	53.0	36.8	35.9	59.7	60.0	69.9	27.9	50.6	41.4	70.0	62.0	69.0	58.1	29.5	59.4	39.3	61.2	52.4	13-Mar-2014
R-CNN [?]	50.2	67.1	64.1	46.7	32.0	30.5	56.4	57.2	65.9	27.0	47.3	40.9	66.6	57.8	65.9	53.6	26.7	56.5	38.1	52.8	50.2	30-Jan-2014
poselets [?]	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	59.3	-	-	-	-	-	06-Jun-2014
Head-Detect-Segment [?]	-	-	-	-	-	-	-	-	41.7	-	-	-	-	-	-	-	-	-	-	-	-	30-Aug-2010
BERKELEY POSELETS [?]	-	33.2	51.9	8.5	8.2	34.8	39.0	48.8	22.2	-	20.6	-	18.5	48.2	44.1	48.5	9.1	28.0	13.0	22.5	33.0	29-Aug-2010
** UCI_LSVM-MDPM-10X ** [?]	-	-	48.1	-	-	-	54.7	-	-	-	25.1	6.0	-	46.7	41.1	-	-	31.2	17.7	-	32.3	30-Aug-2010

Figure: Leading method Fast R-CNN is by Girshick et al.

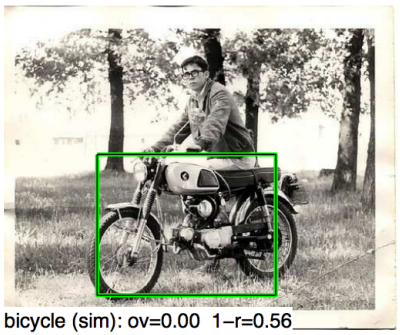
Neural Networks – Detections

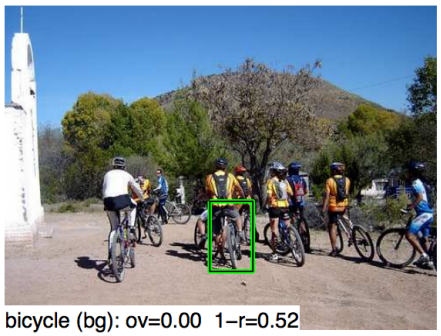

bicycle (loc): ov=0.41 1−r=0.64

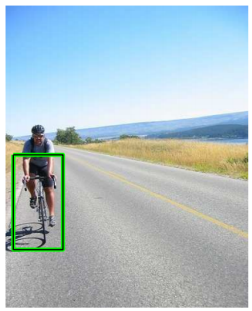

bicycle (loc): ov=0.35 1−r=0.61


bicycle (loc): ov=0.15 1−r=0.59

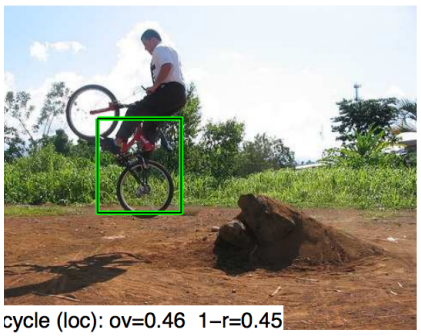

bicycle (loc): ov=0.44 1−r=0.57

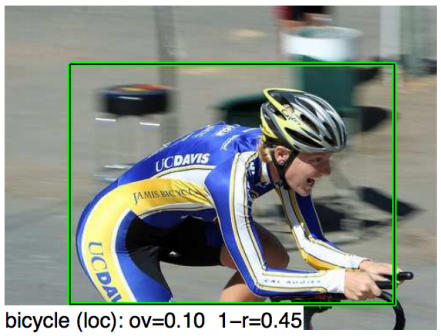

bicycle (sim): ov=0.00 1−r=0.56


bicycle (bg): ov=0.00 1−r=0.52

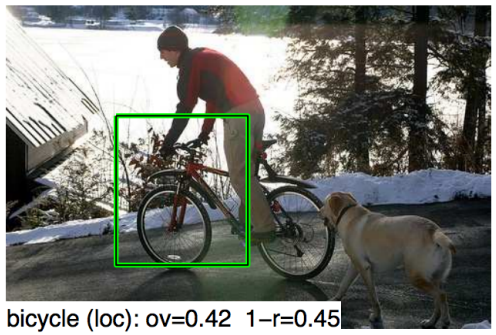

bicycle (loc): ov=0.55 1−r=0.47


bicycle (bg): ov=0.00 1−r=0.47


cycle (loc): ov=0.46 1−r=0.45



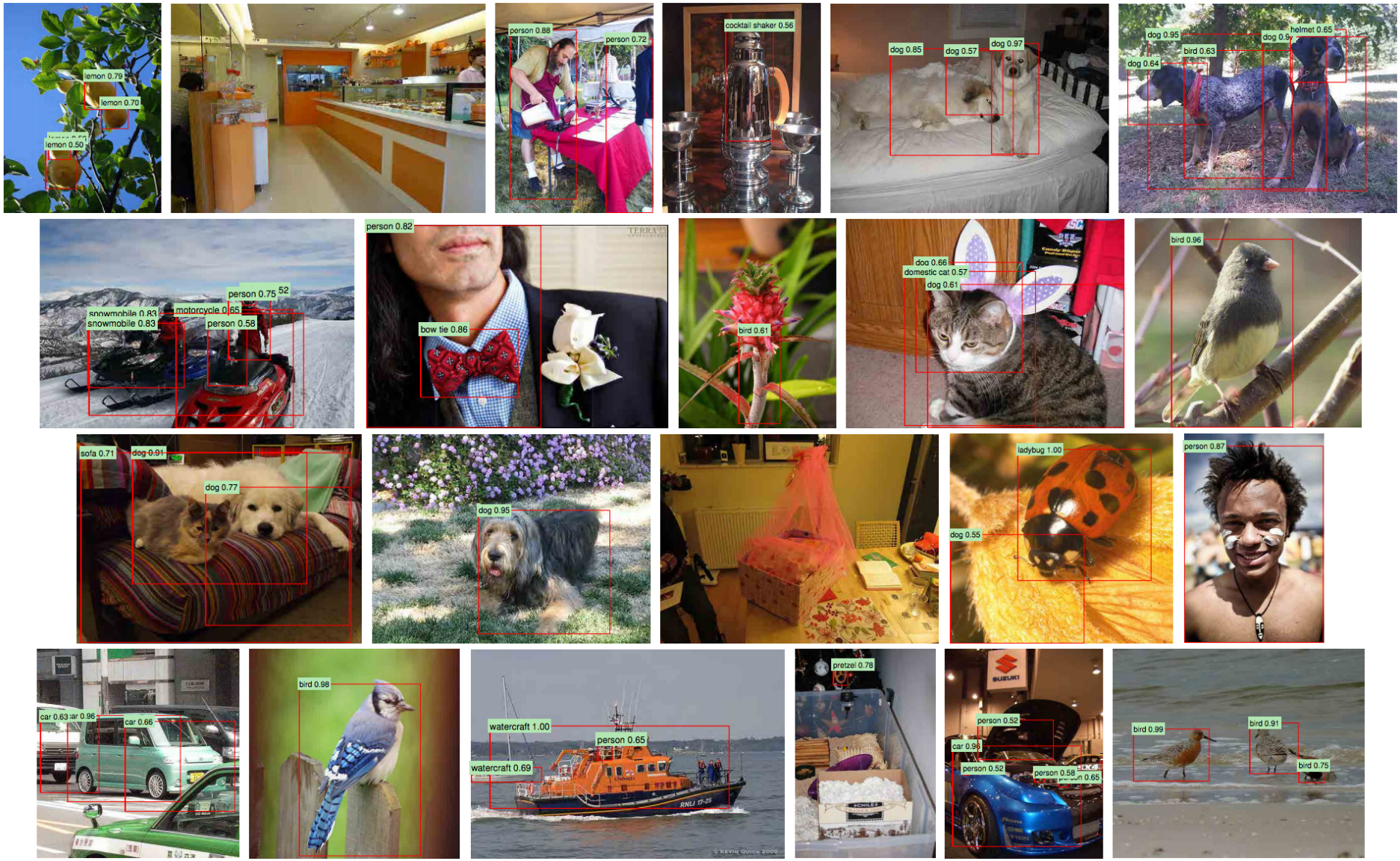
bicycle (loc): ov=0.10 1−r=0.45


bicycle (loc): ov=0.42 1−r=0.45


bicycle (bg): ov=0.00 1−r=0.44

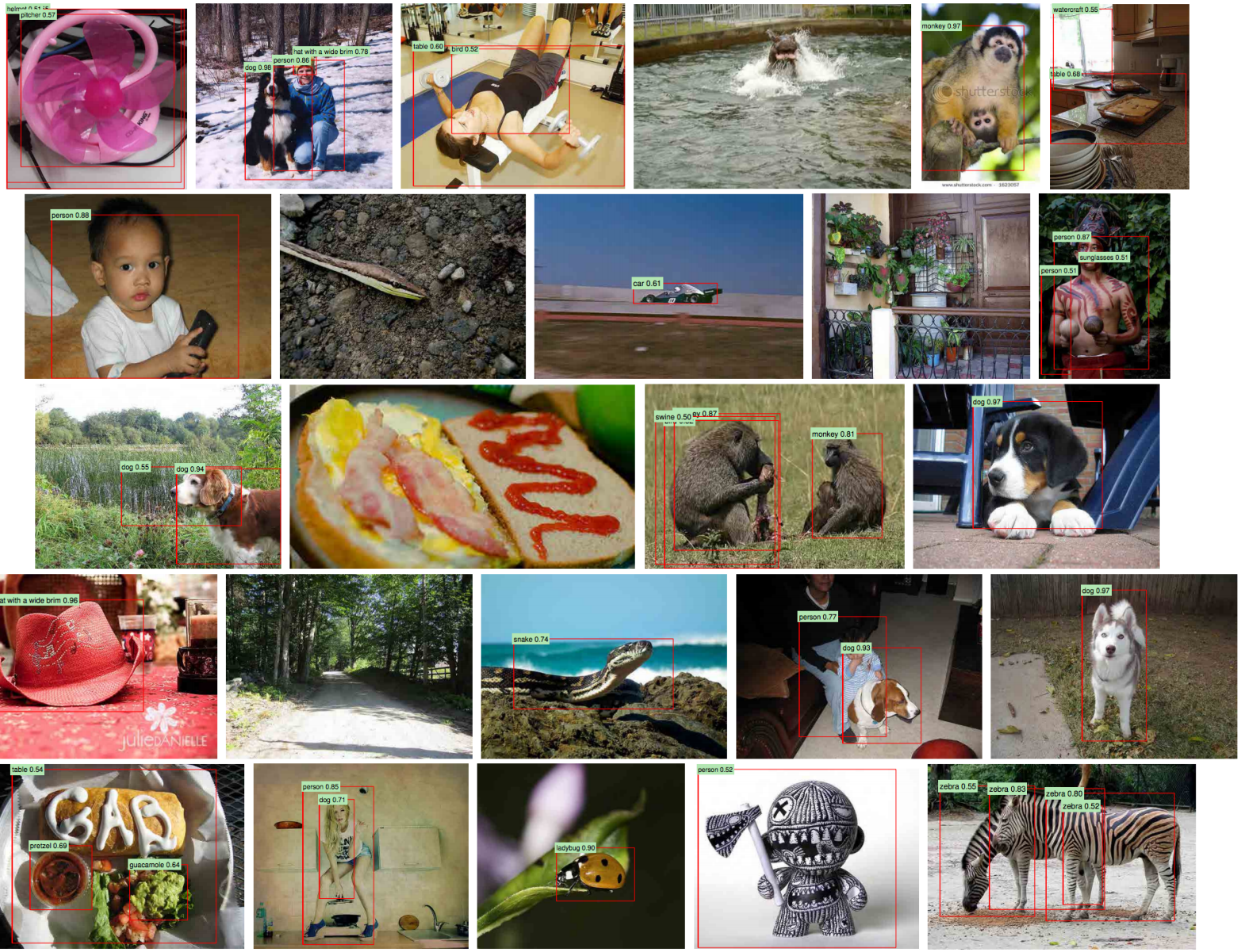
[Source: Girshick et al.]

Neural Networks – Detections



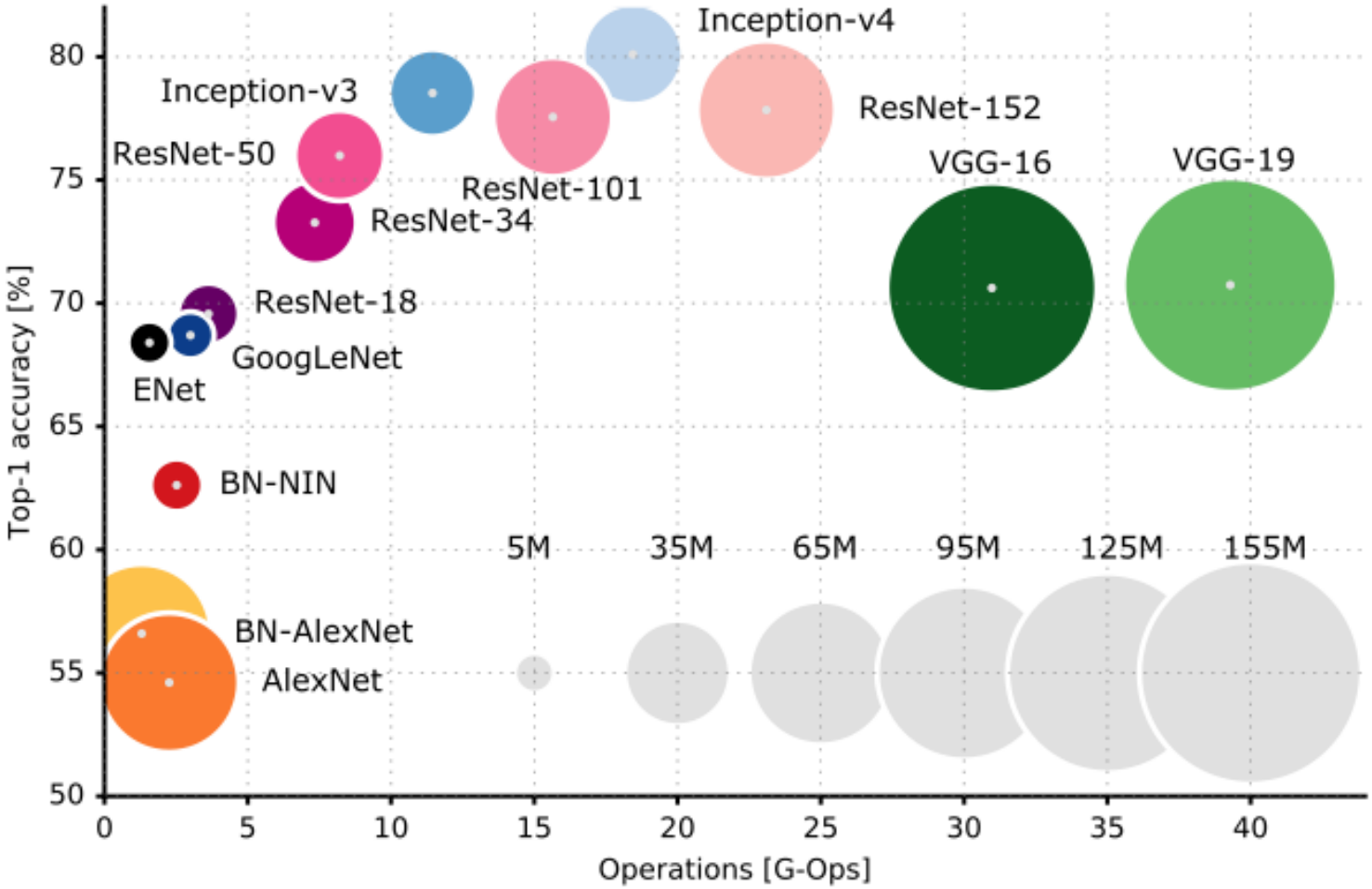
[Source: Girshick et al.]

Neural Networks – Detections



[Source: Girshick et al.]

An Analysis of Deep Neural Network Models for Practical Applications



[Canziani, Paszke, Culurciello, 2016]

Next Time: Basics of Training Neural Nets