Computer Vision:
Foundations and Applications

Deepraj shukla

deeprajshukla@gmail.com

Convolution Neural networks

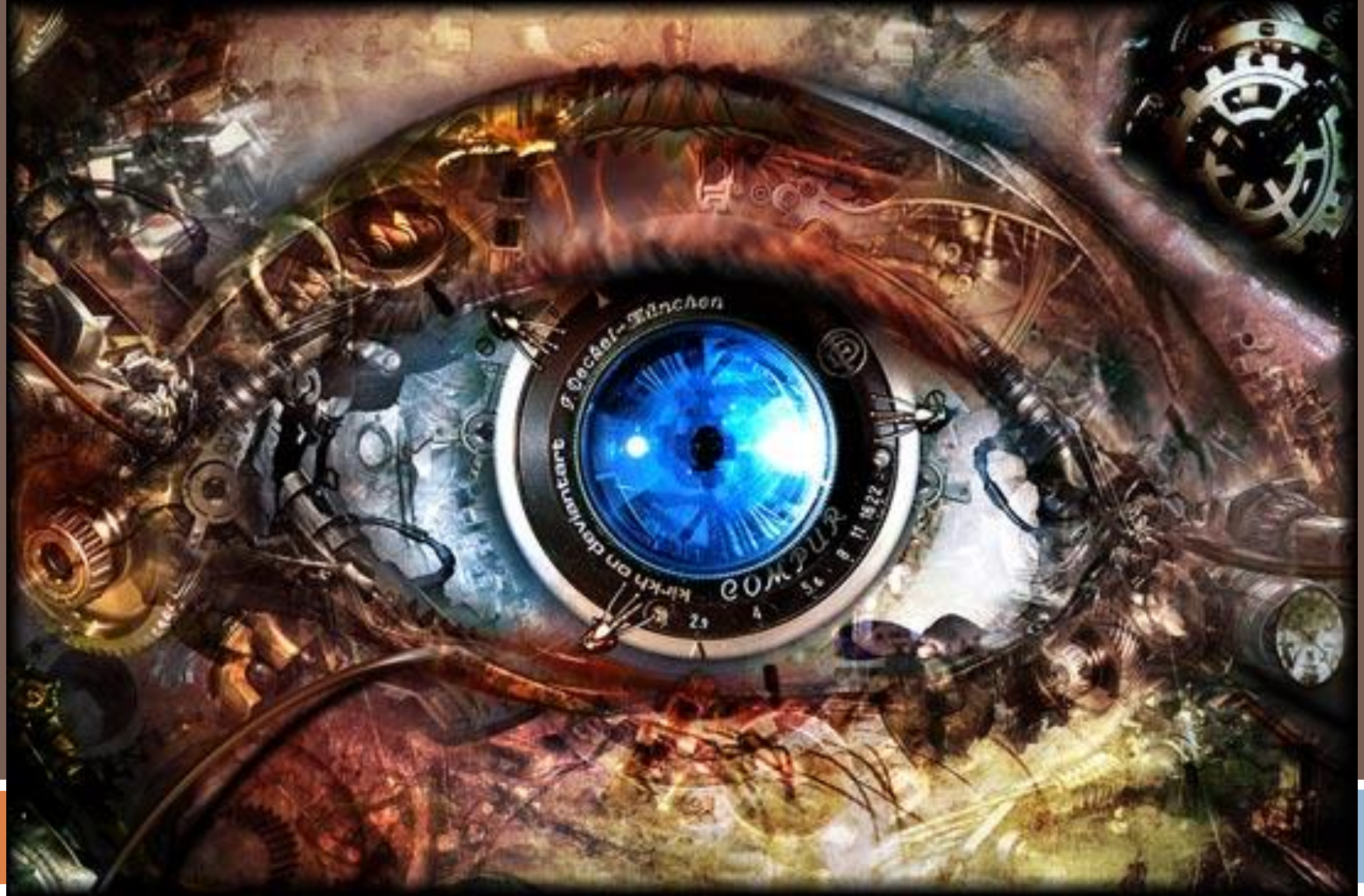
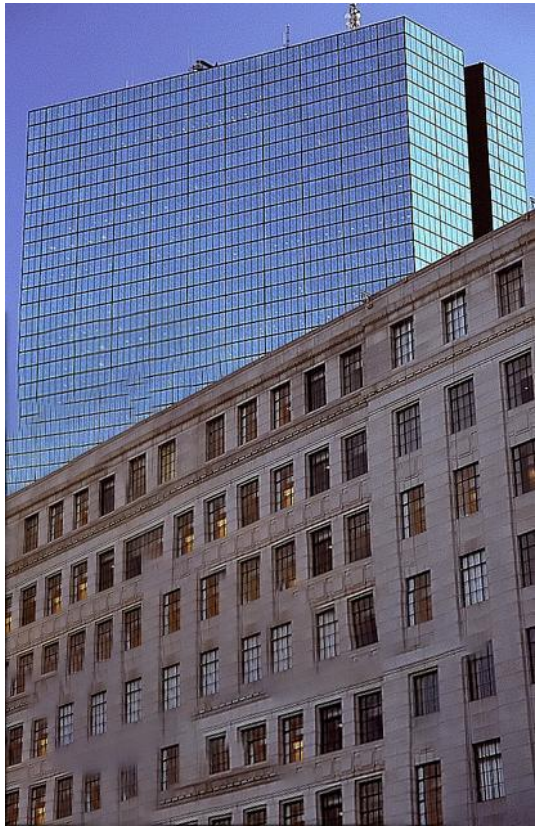

COMPUR

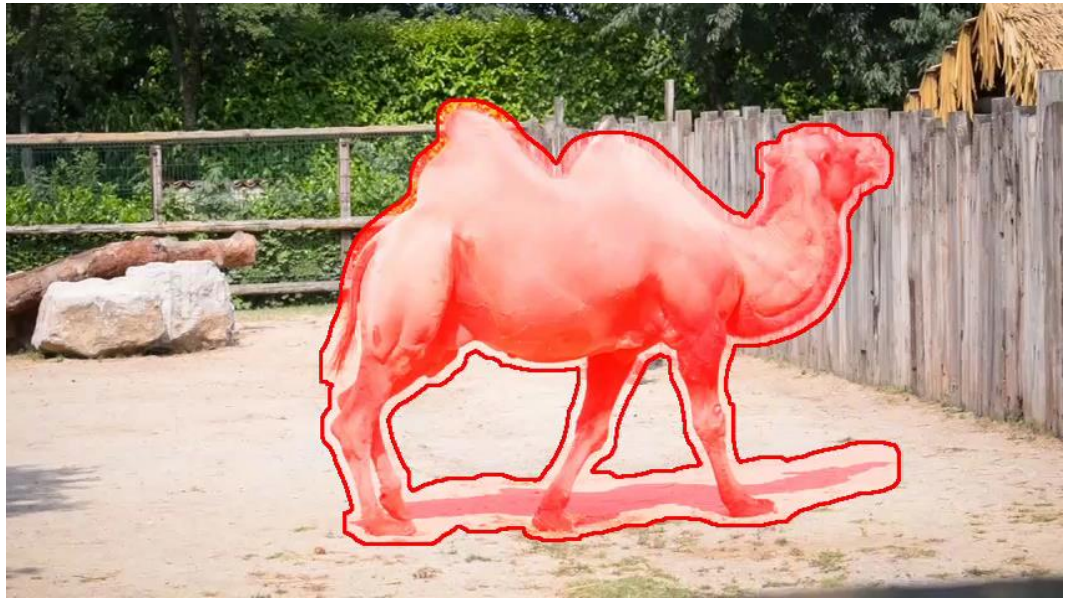



Image Completion [SIGGRAPH14]

- Revealing unseen pixels





Video Completion [SIGGRAPH Asia16]

- Revealing temporally coherent pixels

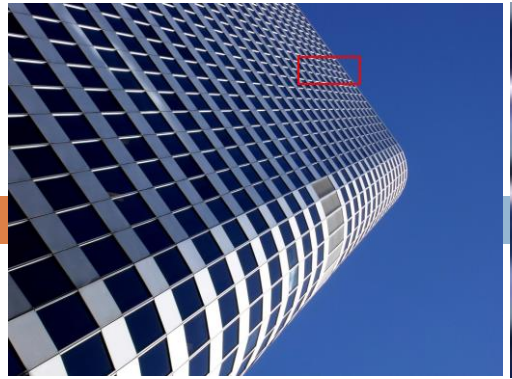
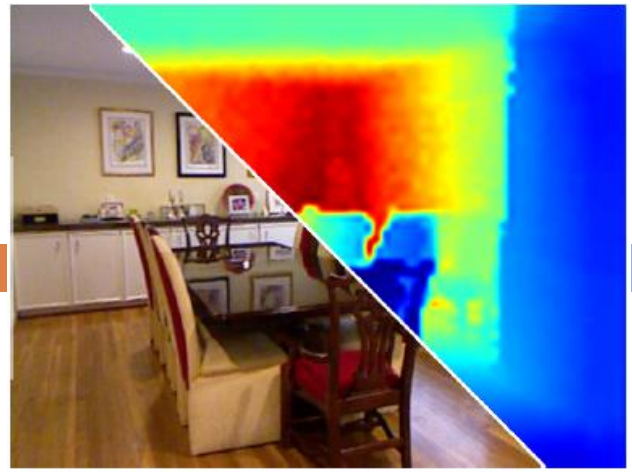




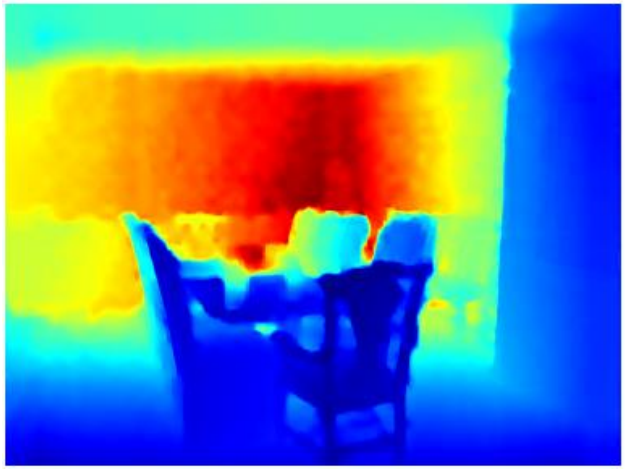



Image super-resolution [CVPR15]

- Revealing unseen high frequency details







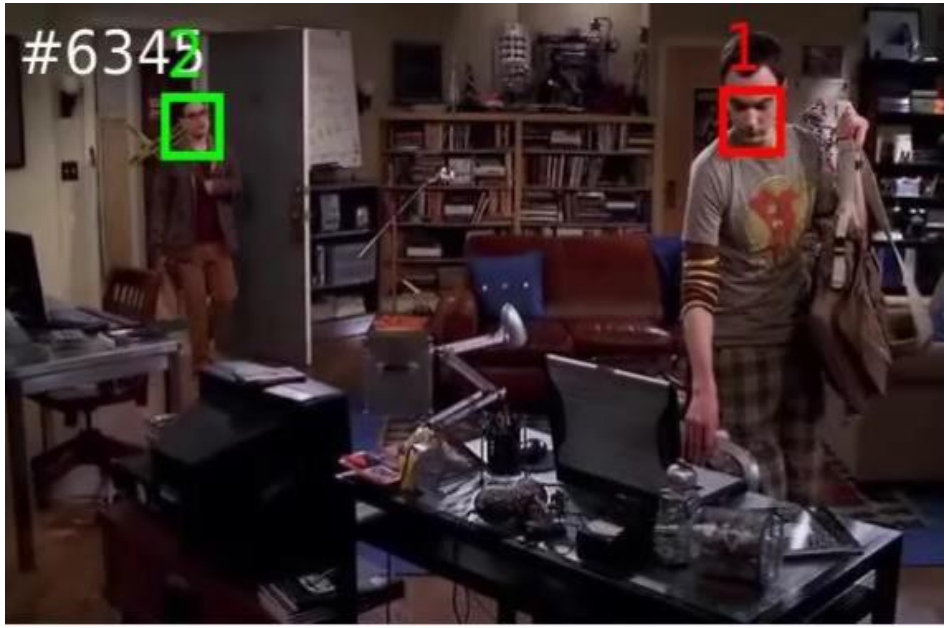

Depth upsampling | Noise reduction | Inverse halftoning | Texture removal

Deep Joint Image Filtering

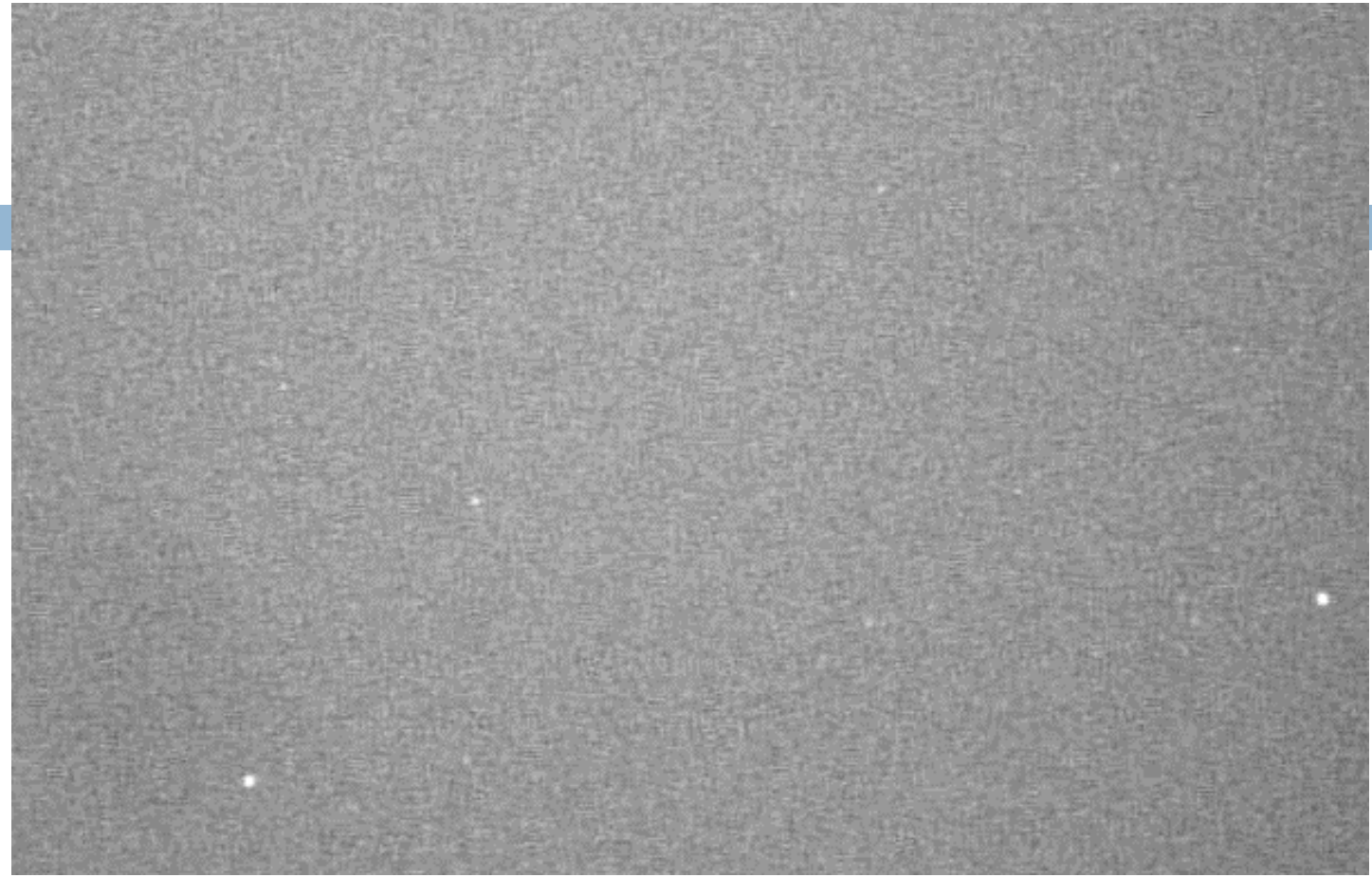
[ECCV16]

- *Transferring structural details*

Object tracking [ICCV15]



Multi-face tracking [ECCV16]



Detecting migrating birds [CVPR16]

Visual Tracking

- Locating moving objects across video frames

What is Computer Vision?

- Make computers understand images and videos.

- What kind of scene?
- Where are the cars?
- How far is the building?

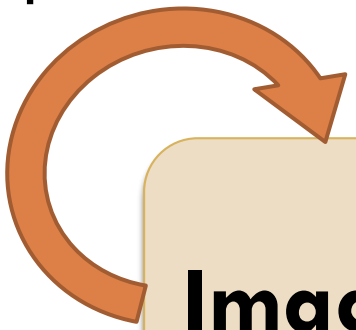
What is Computer Vision?

- Make computers understand images and videos.

- What are they doing?
- Why is this happening?
- What is important?
- What will I see?

Computer Vision and Nearby Fields

Digital Image Processing
Computational Photography



Images (2D)

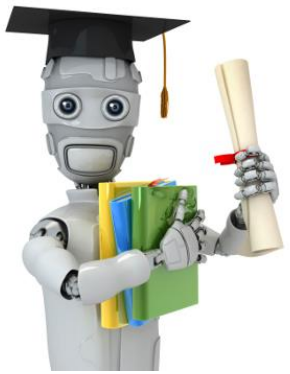
Computer Vision

Computer Graphics

Geometry (3D)
Shape

+

Photometry
Appearance



Machine learning:
Vision = Machine learning applied to visual data

Visual data on the Internet

- Flickr
  - 10+ billion photographs
  - 60 million images uploaded a month
- Facebook
  - 250 billion+
  - 300 million a day
- Instagram
  - 55 million a day
- YouTube
  - 100 hours uploaded every minute

CISCO **90%** of net traffic will be visual!

Mostly about cats

Too big for humans

http://www.petittube.com/

- Need automatic tools to access and analyze visual data!

Vision is Really Hard

- Vision is an amazing feature of natural intelligence
  - Visual cortex occupies about 50% of Macaque brain
  - More human brain devoted to vision than anything else

Why is Computer Vision Hard?

Why is Computer Vision Hard?

What did you see?

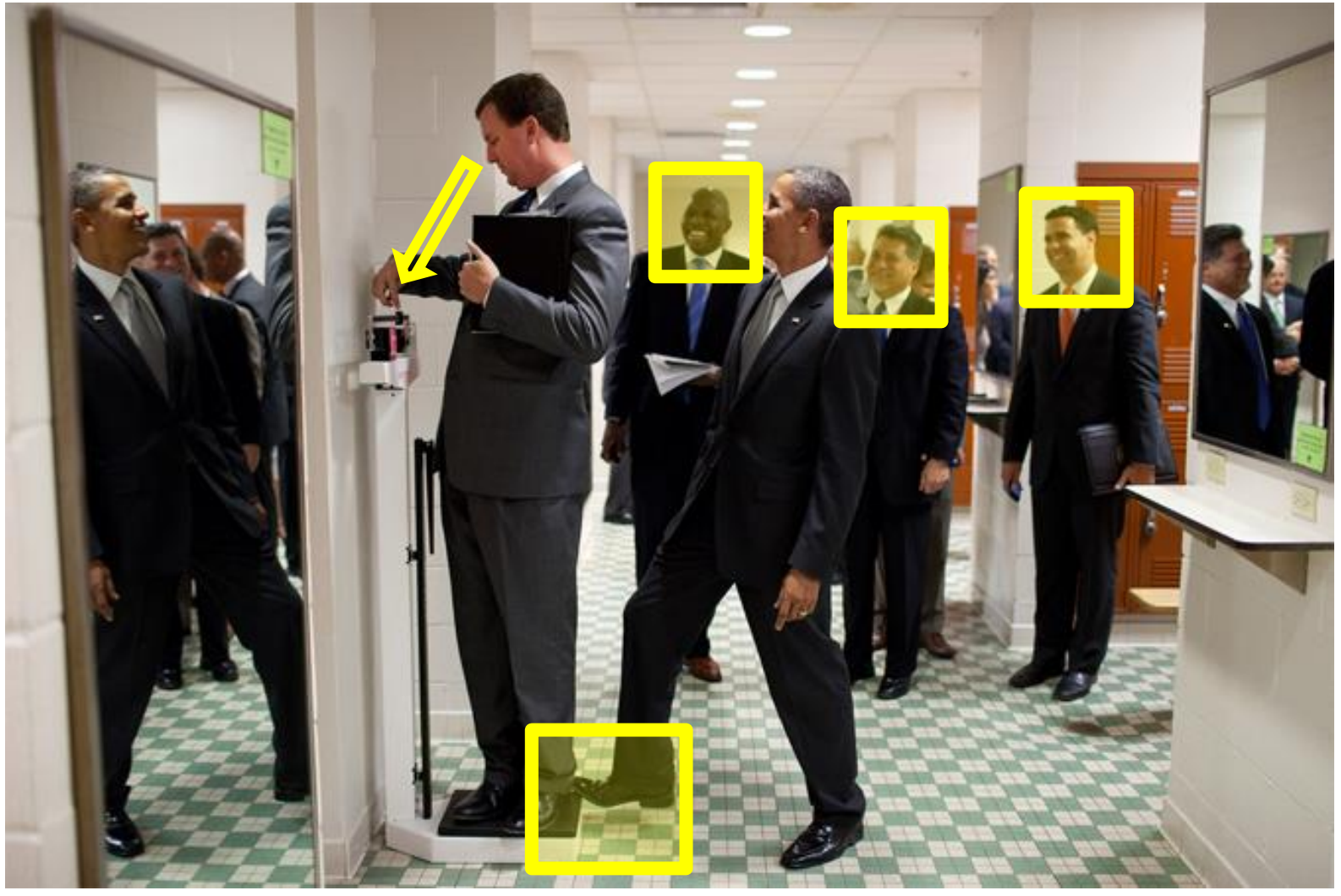
- Where this picture was taken?
- How many people are there?
- What are they doing?
- What object the person on the left standing on?
- Why this is a funny picture?

Why is Computer Vision Hard?

Why is Computer Vision Hard?

Why is Computer Vision Hard?

Why is Computer Vision Hard?



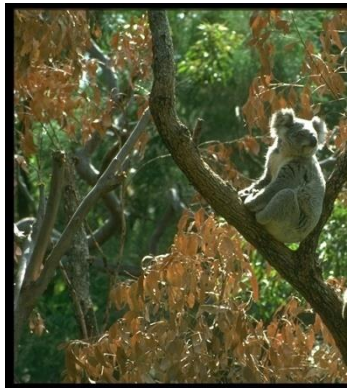
Why is Computer Vision Hard?

Why is Computer Vision Hard?

Challenges: Many nuisance parameters

Illumination

Object pose



Clutter

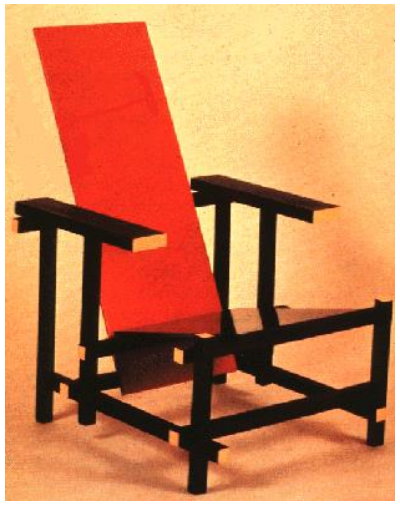
Occlusions

Intra-class appearance

Viewpoint

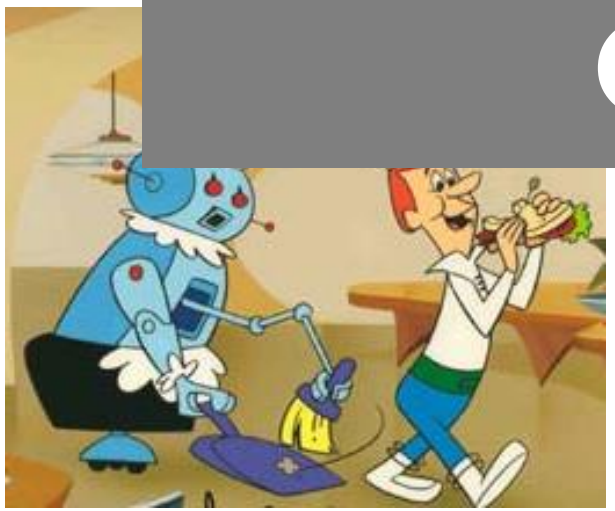
Slide credit: Kristen Grauman

Challenges: Intra-class variation



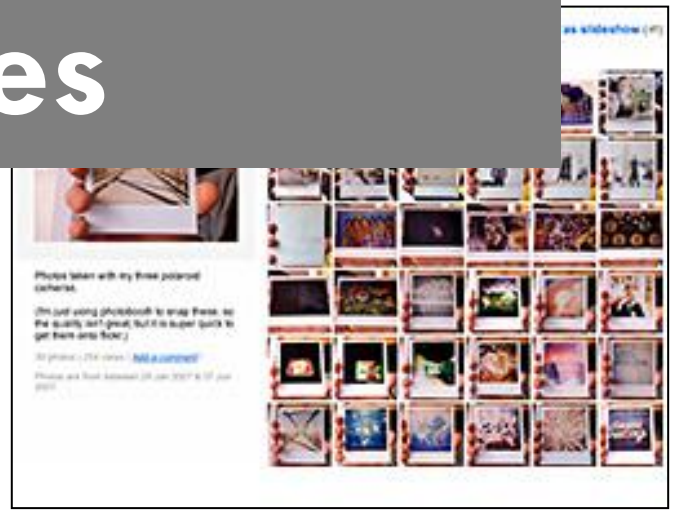
Slide credit: Fei-Fei, Fergus & Torralba

Computer Vision Technology Can Better Our Lives



Comfort

Fun



Access

History of Computer Vision

Marvin Minsky, MIT
Turing award, 1969

"In 1966, Minsky hired a first-year undergraduate student and assigned him a problem to solve over the summer:

connect a camera to a computer and get the machine to describe what it sees."

Crevier 1993, pg. 88

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

PROJECT MAC

Artificial Intelligence Group
Vision Memo. No. 100.

July 7, 1966

Half a century later,
we're still working on it.

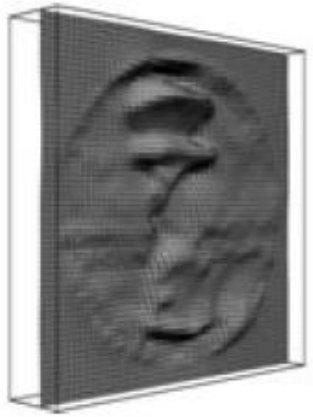
THE SUMMER VISION PROJECT

Seymour Papert

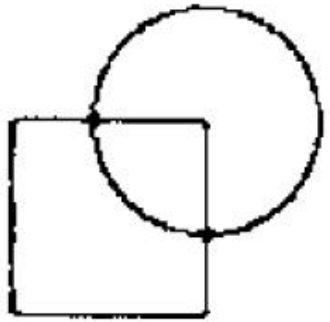
The summer vision project is an attempt to use our summer workers effectively in the construction of a significant part of a visual system. The particular task was chosen partly because it can be segmented into sub-problems which will allow individuals to work independently and yet participate in the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

1980's: ANNs come and go; shift toward geometry and increased mathematical rigor

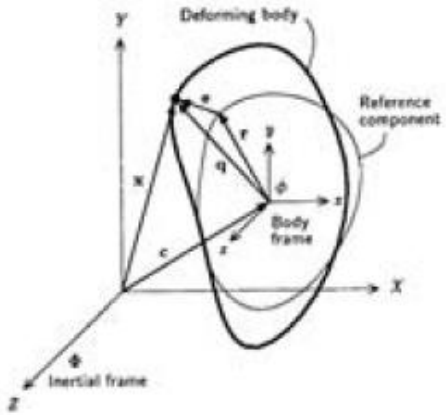
(a)



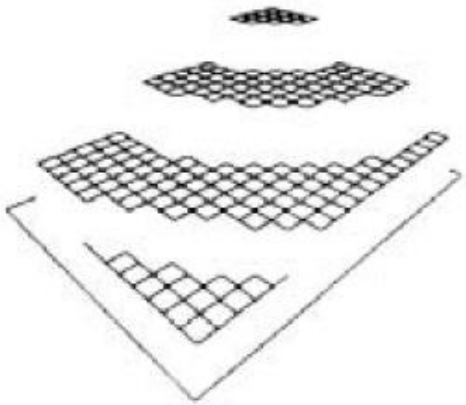
(b)



(c)



(d)



(e)

(f)

Image credit: Rick Szeliski

1990's: face recognition; statistical analysis in vogue

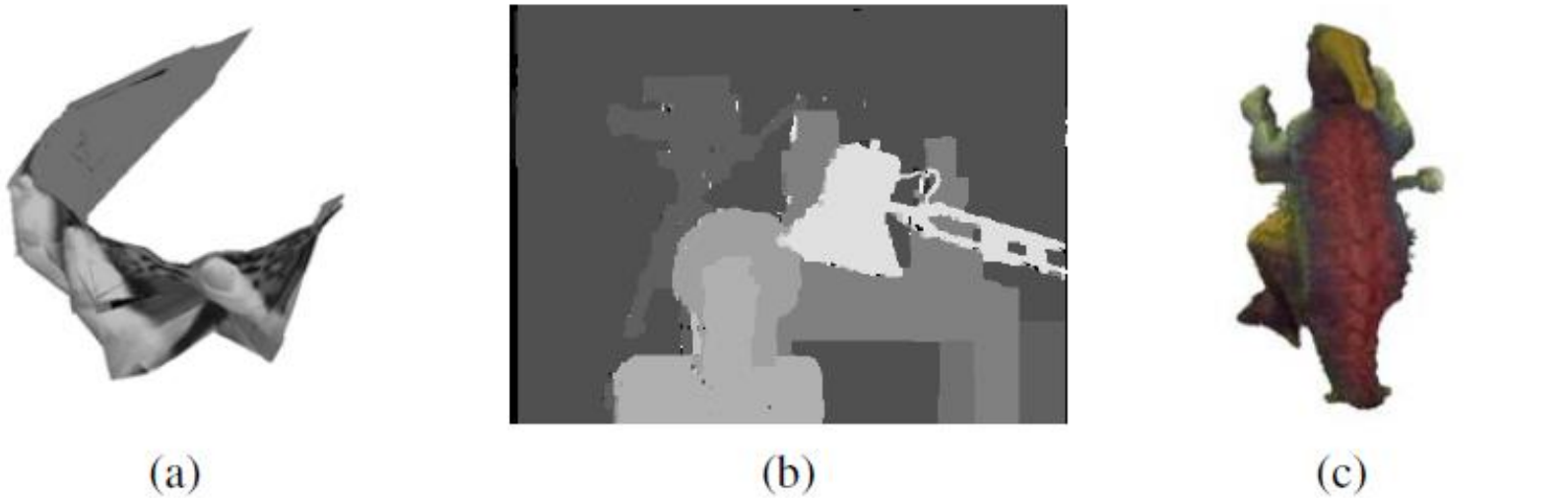


Image credit: Rick Szeliski

2000's: broader recognition; large annotated datasets available; video processing starts

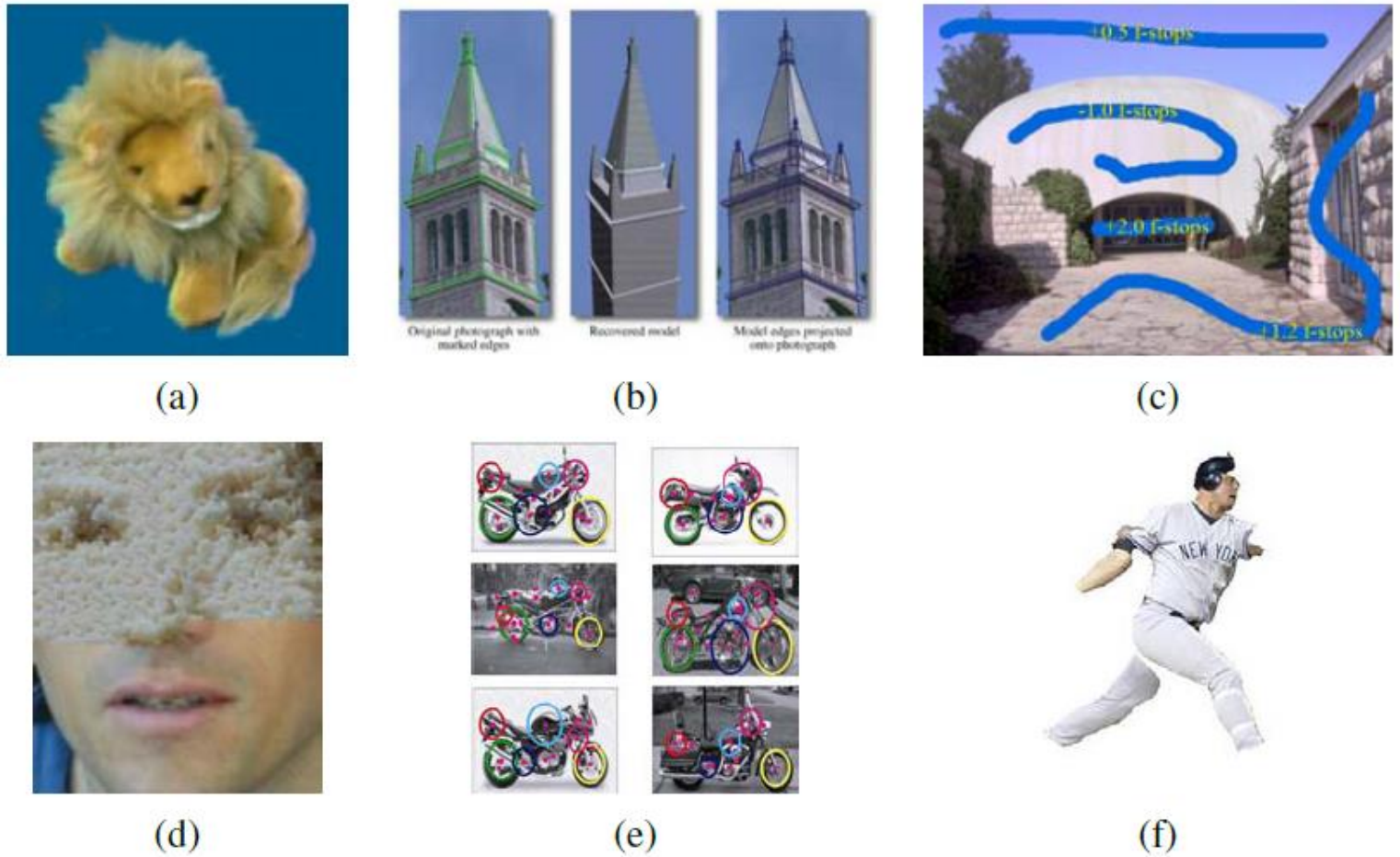
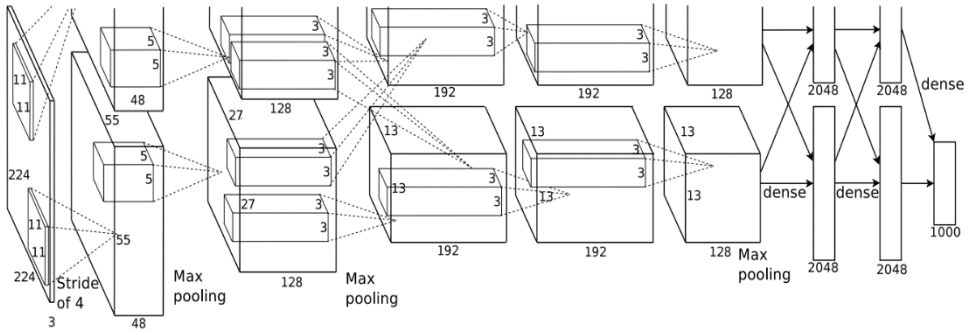
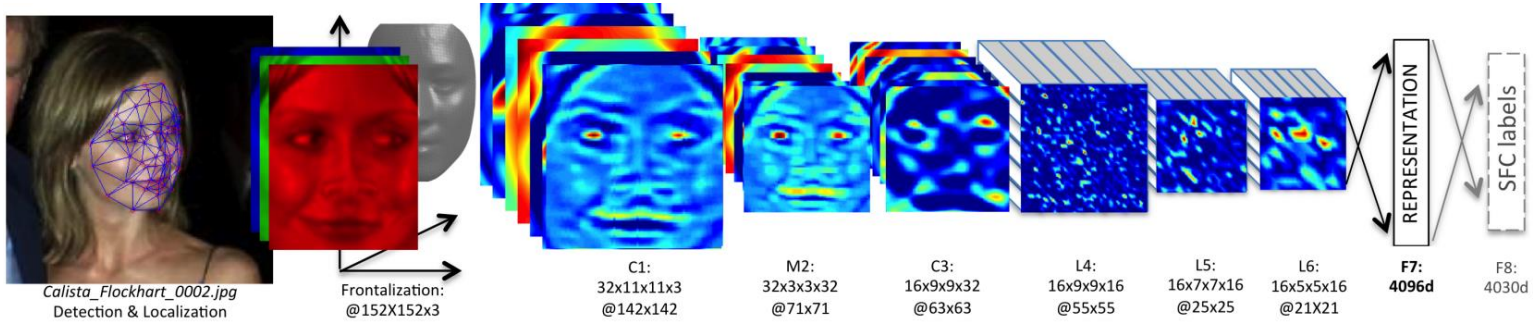


Image credit: Rick Szeliski

2010's: resurgence of deep learning

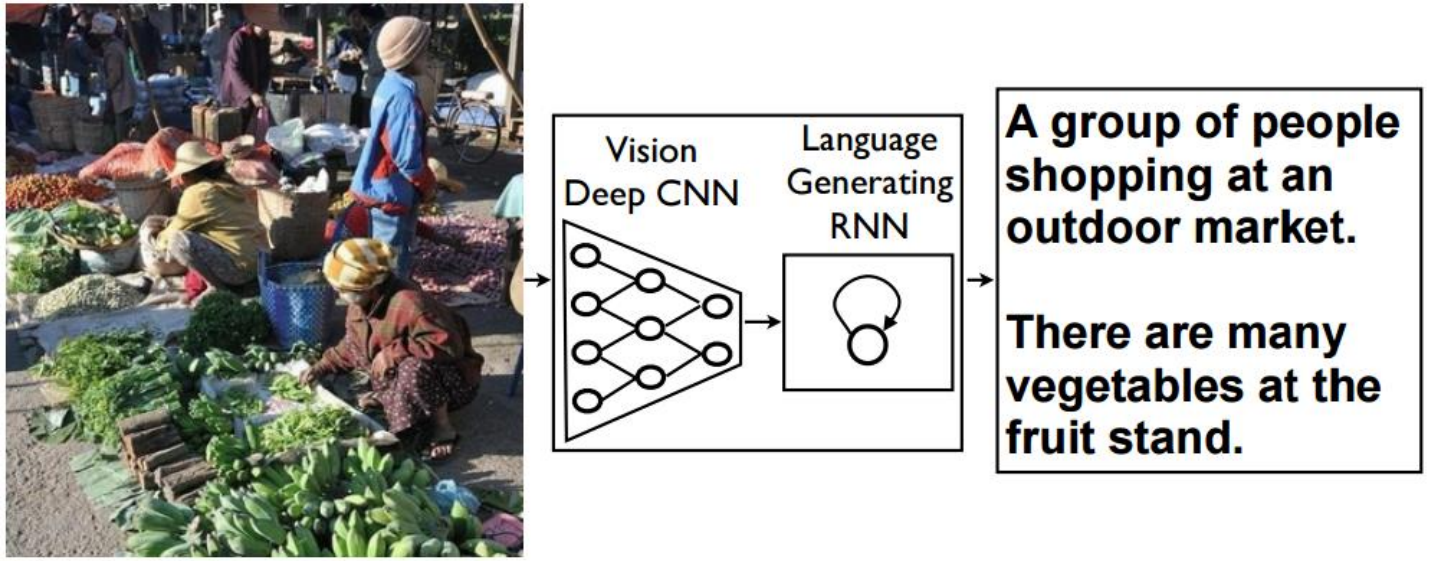


[AlexNet NIPS 2012]



[DeepFace CVPR 2014]

[DeepPose CVPR 2014]



[Show, Attend and Tell ICML 2015]

2020's: autonomous vehicles

2030's: robot uprising?

Examples of Computer Vision Applications

- How is computer vision used today?

Face detection

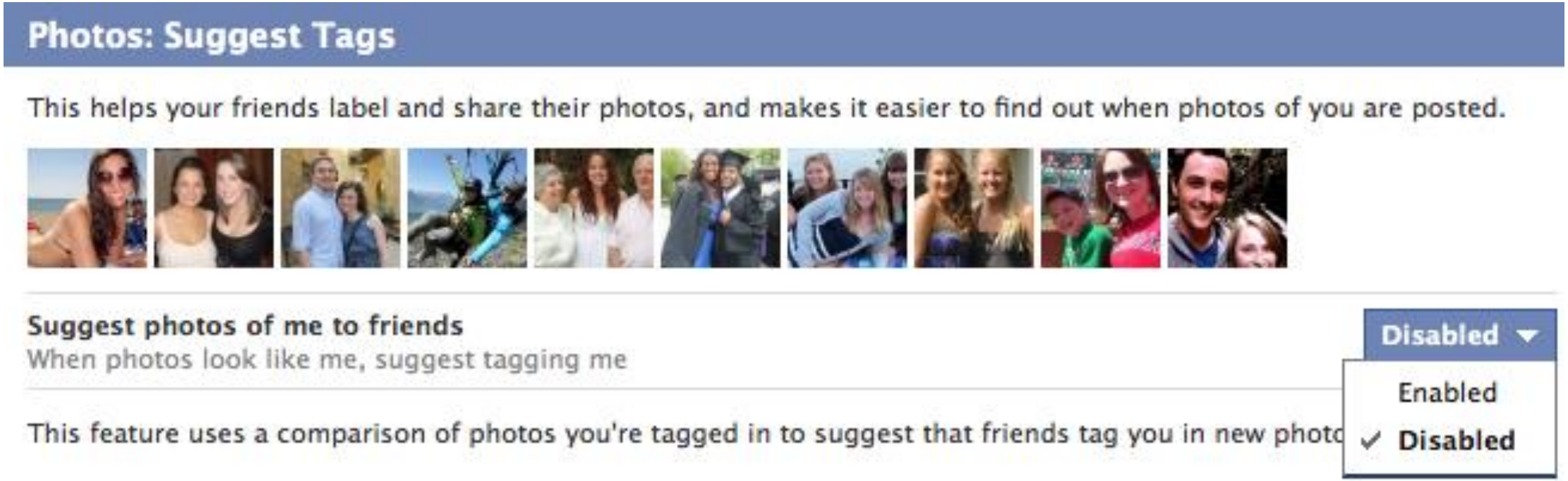
- Most digital cameras and smart phones detect faces (and more)
  - Canon, Sony, Fuji, …
- For smart focus, exposure compensation, and cropping

Slide credit: Steve Seitz

Demo

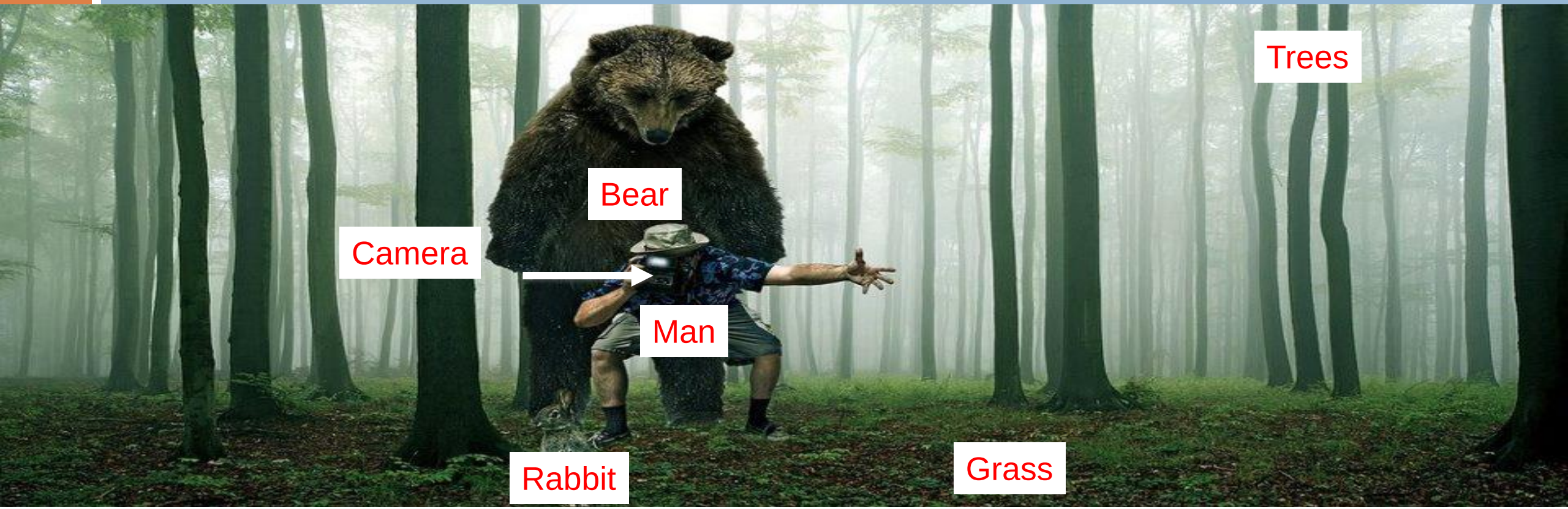
- **Face Detection**

Face recognition



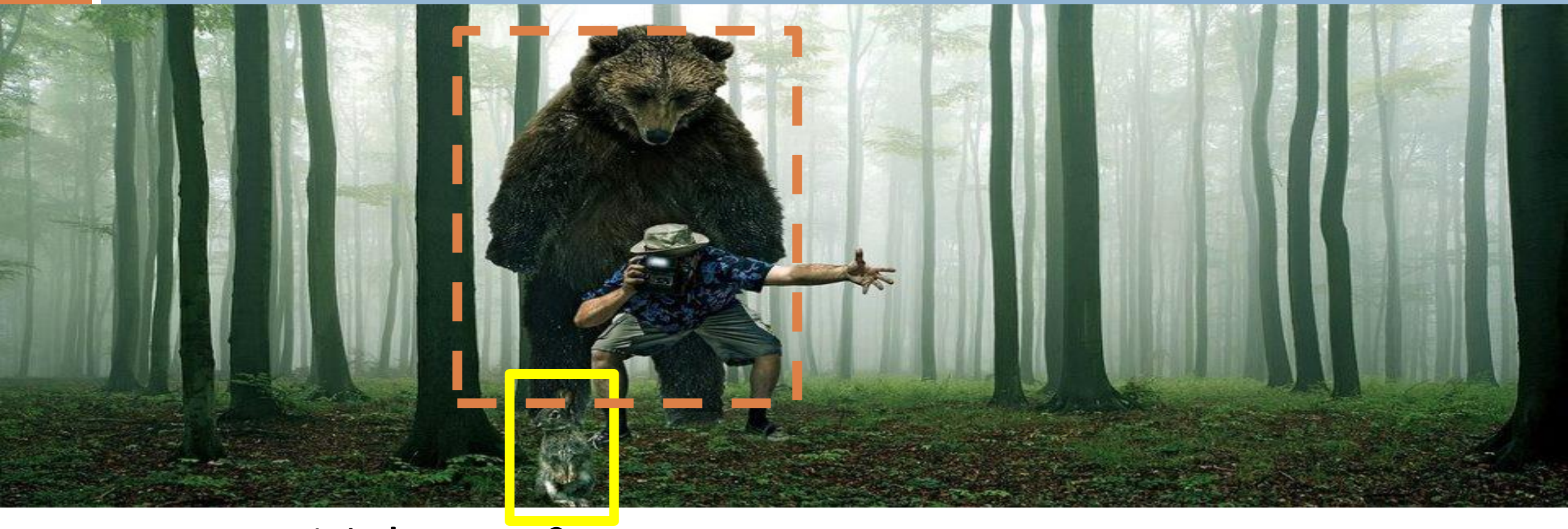
Facebook face auto-tagging

What do you see in this image?



Forest

Describe, predict, or interact with the object based on visual cues



Is it **dangerous**?

Is it **alive**?

How **fast** does it run?

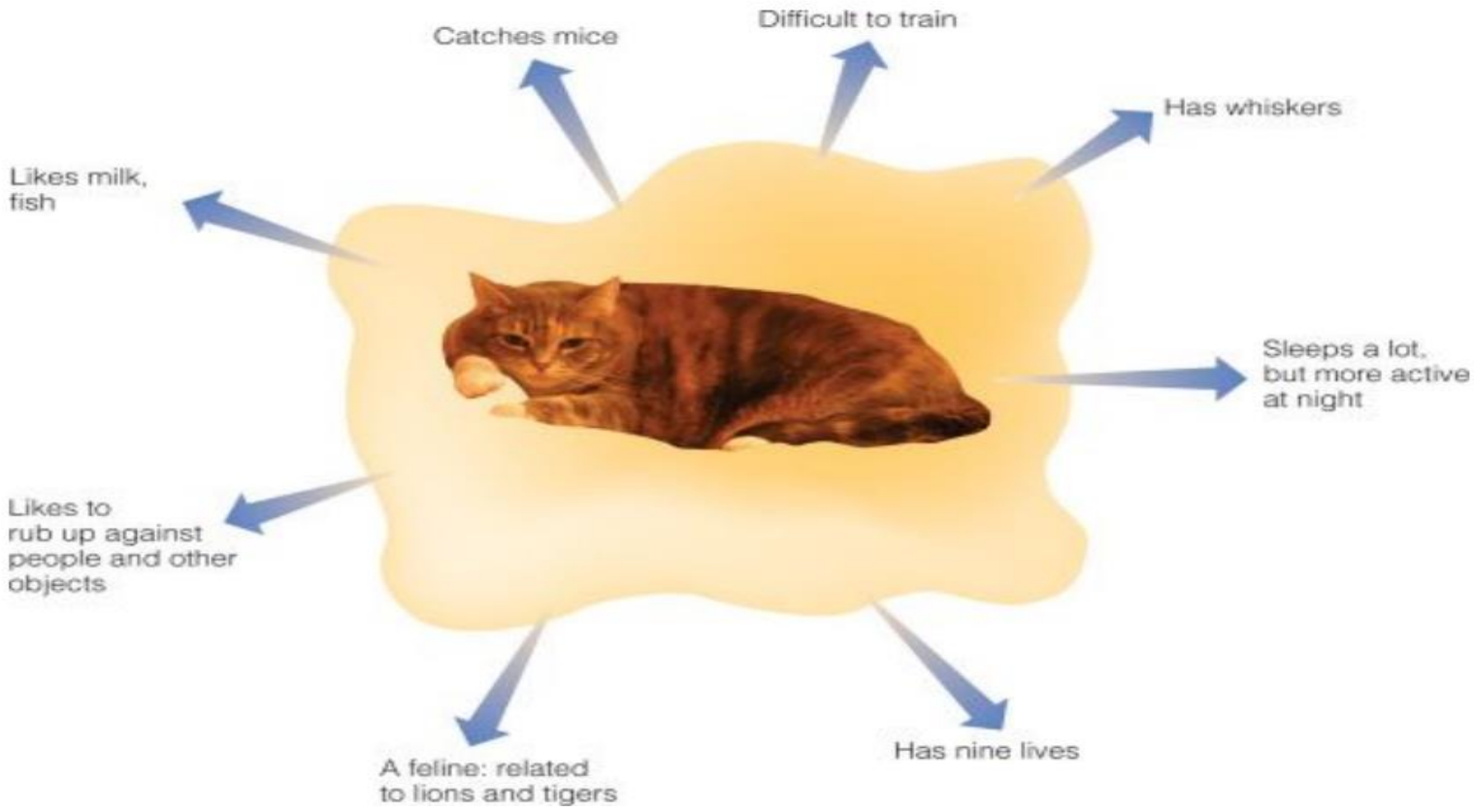
Is it **soft**?

Can I **poke with it**?

Does it have a **tail**?

Why do we care about categories?

- From an object's category, we can make predictions about its behavior in the future, beyond of what is immediately perceived.
- Pointers to knowledge
  - Help to understand individual cases not previously encountered
- Communication



Theory of categorization

How do we determine if something is a member of a particular category?

- Definitional approach
- Prototype approach
- Exemplar approach

Definitional approach: classical view of categories

- Plato & Aristotle
  - Categories are defined by a list of properties shared by all elements in a category
  - Category membership is binary
  - Every member in the category is equal

Aristotle by Francesco Hayez

The Categories (Aristotle)

Slide Credit: A. A. Efros

Prototype or sum of exemplars ?

Prototype Model

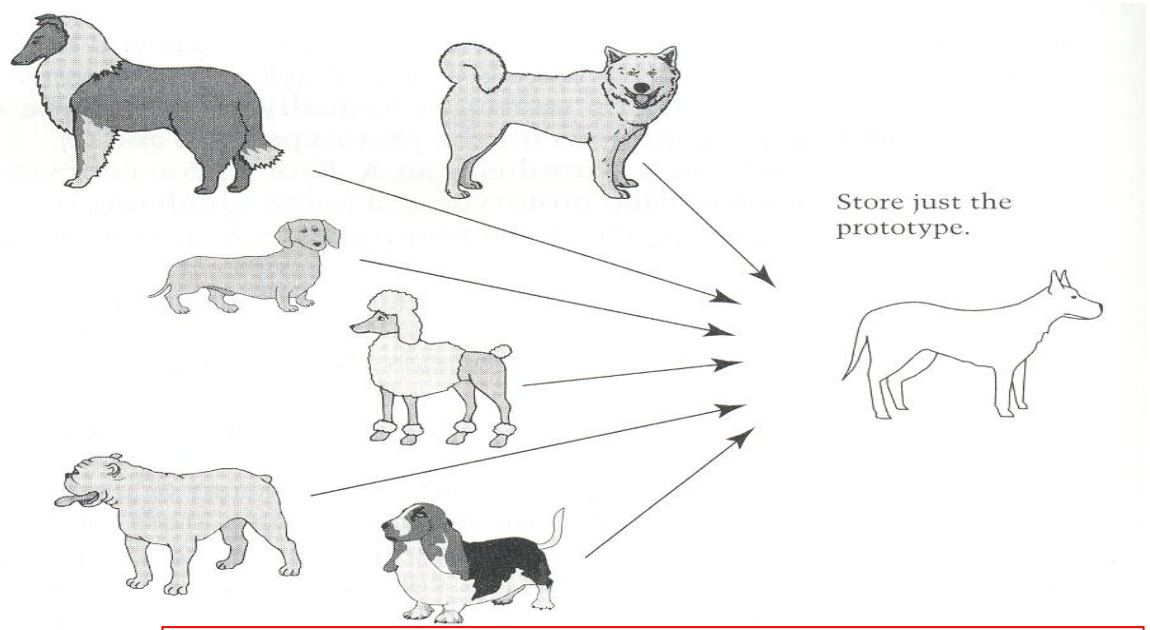


Figure 7.3. *Schematic of the prototype model. Although many exemplars are seen, only the prototype is stored. The prototype is updated continually to incorporate more experience with new exemplars.*

Category judgments are made
by comparing a new exemplar
to the prototype.

Exemplars Model

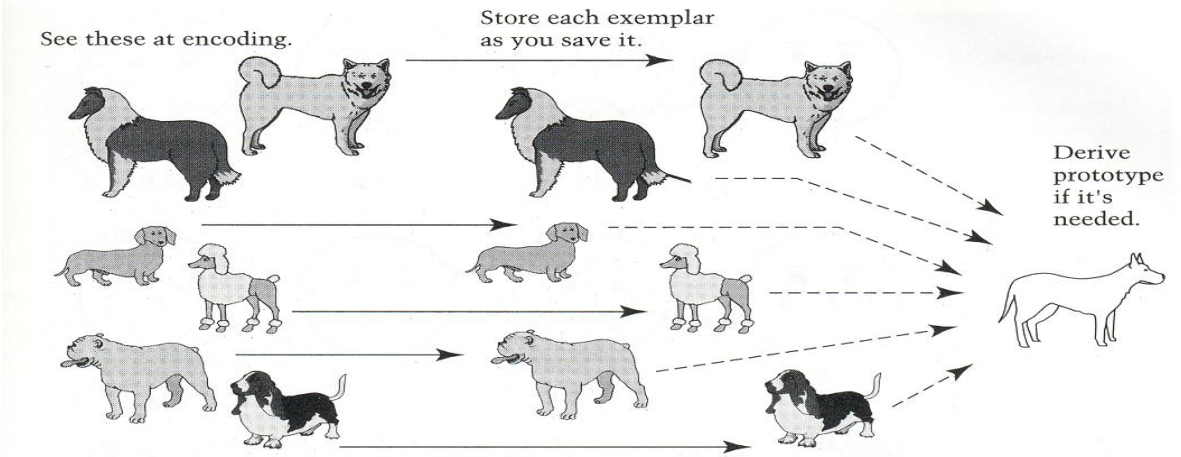


Figure 7.4. *Schematic of the exemplar model. As each exemplar is seen, it is encoded into memory. A prototype is abstracted only when it is needed, for example, when a new exemplar must be categorized.*

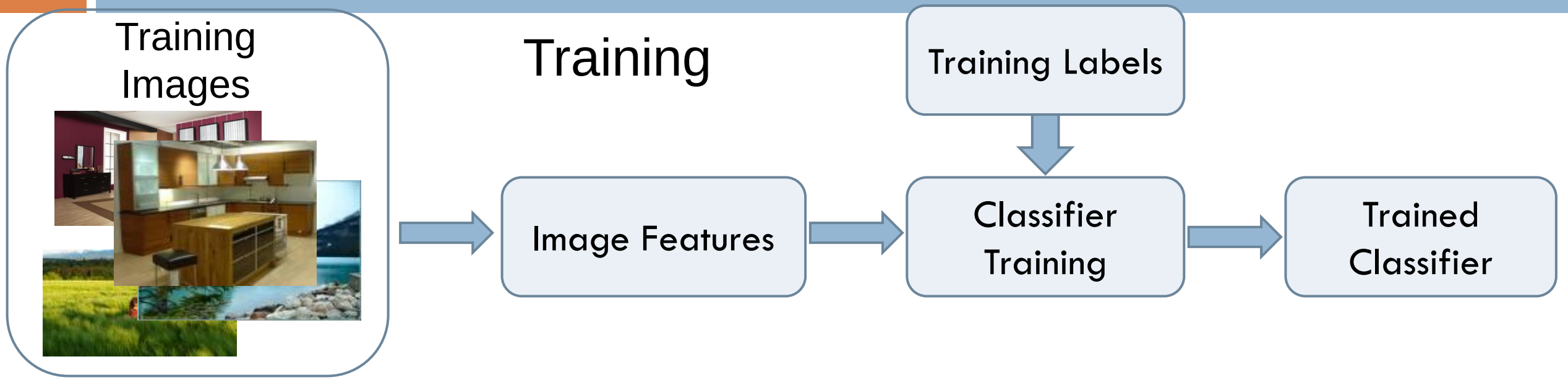
Category judgments are made
by comparing a new exemplar
to all the old exemplars of a category
or to the exemplar that is the most
appropriate

Slide Credit: Torralba

Image categorization

- Cat vs Dog

Training phase



Testing phase

Training Images

Training

Training Labels

Image Features

Classifier Training

Trained Classifier

Testing

Test Image

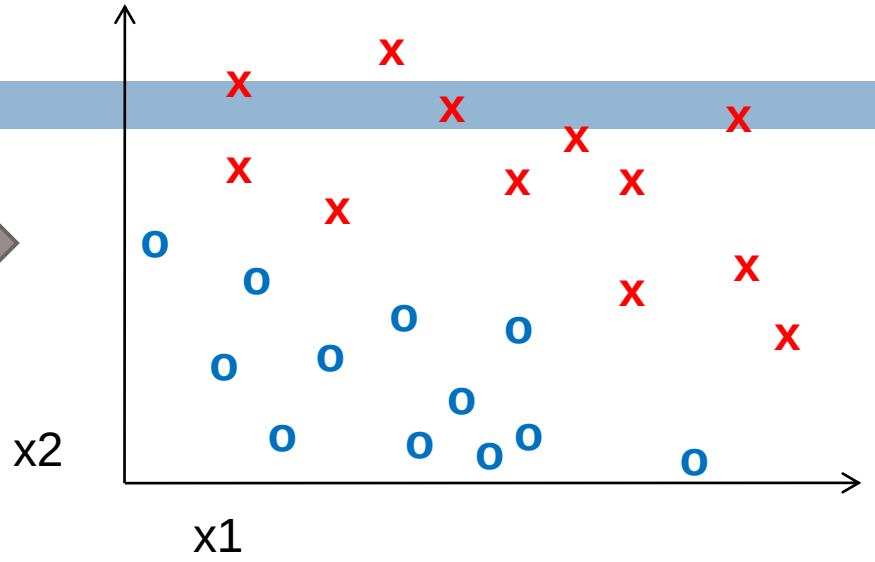
Image Features

Trained Classifier

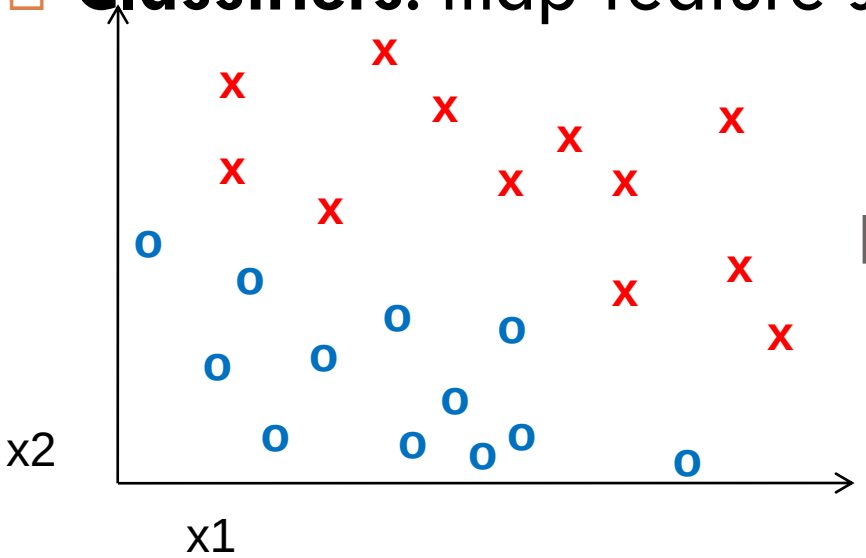
Prediction

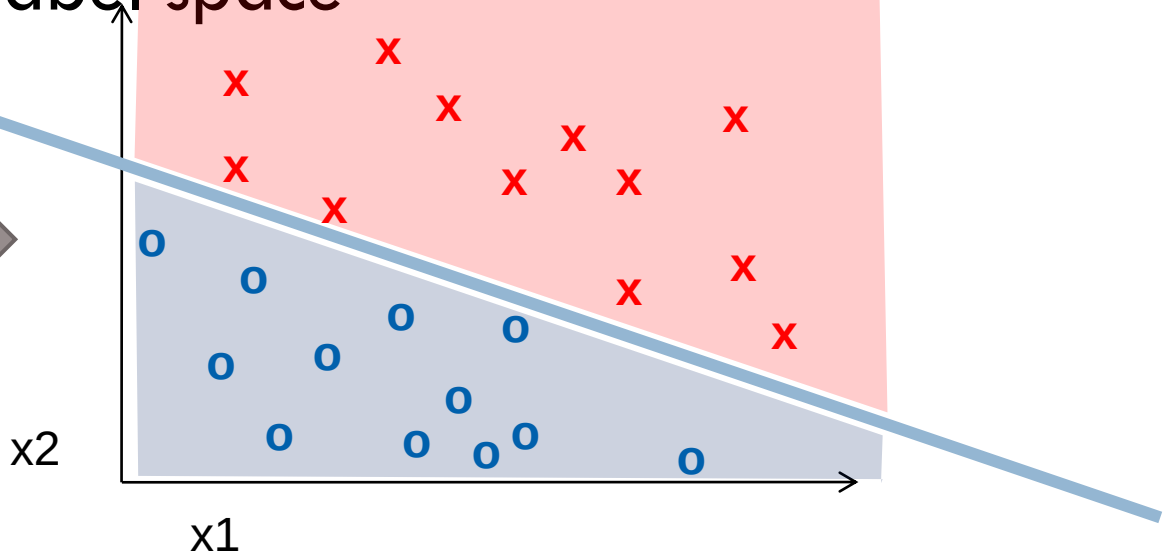
Outdoor

- **Image features:** map images to feature space



- **Classifiers:** map feature space to label space





Different types of classification

- **Exemplar-based**: transfer category labels from examples with most similar features
  - What similarity function? What parameters?
- **Linear classifier**: confidence in positive label is a weighted sum of features
  - What are the weights?
- **Non-linear classifier**: predictions based on more complex function of features
  - What form does the classifier take? Parameters?
- **Generative classifier**: assign to the label that best explains the features (makes features most likely)
  - What is the probability function and its parameters?

Note: You can always fully design the classifier by hand, but usually this is too difficult. Typical solution: learn from training examples.

Testing phase

Training Images

Training

Training Labels

Image Features

Classifier Training

Trained Classifier

Testing

Test Image

Image Features

Trained Classifier

Prediction

Outdoor

Q: What are good features for…

- recognizing a beach?






Q: What are good features for…

- recognizing cloth fabric?






Q: What are good features for…

- recognizing a mug?






What are the right features?

Depend on what you want to know!

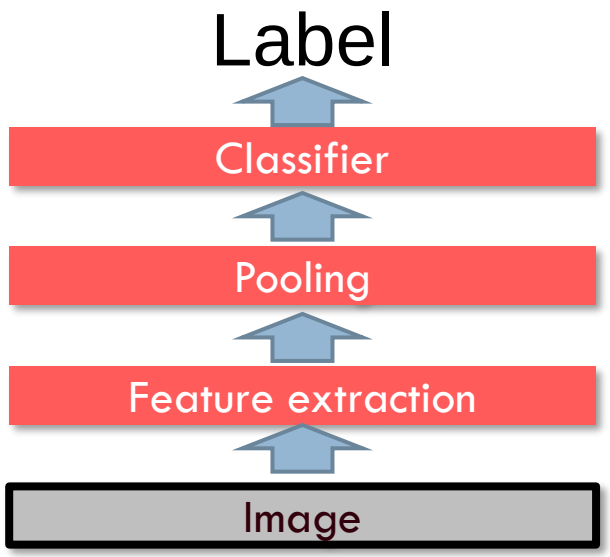
- Object: shape
  - Local shape info, shading, shadows, texture
- Scene : geometric layout
  - linear perspective, gradients, line segments
- Material properties: albedo, feel, hardness
  - Color, texture
- Action: motion
  - Optical flow, tracked points

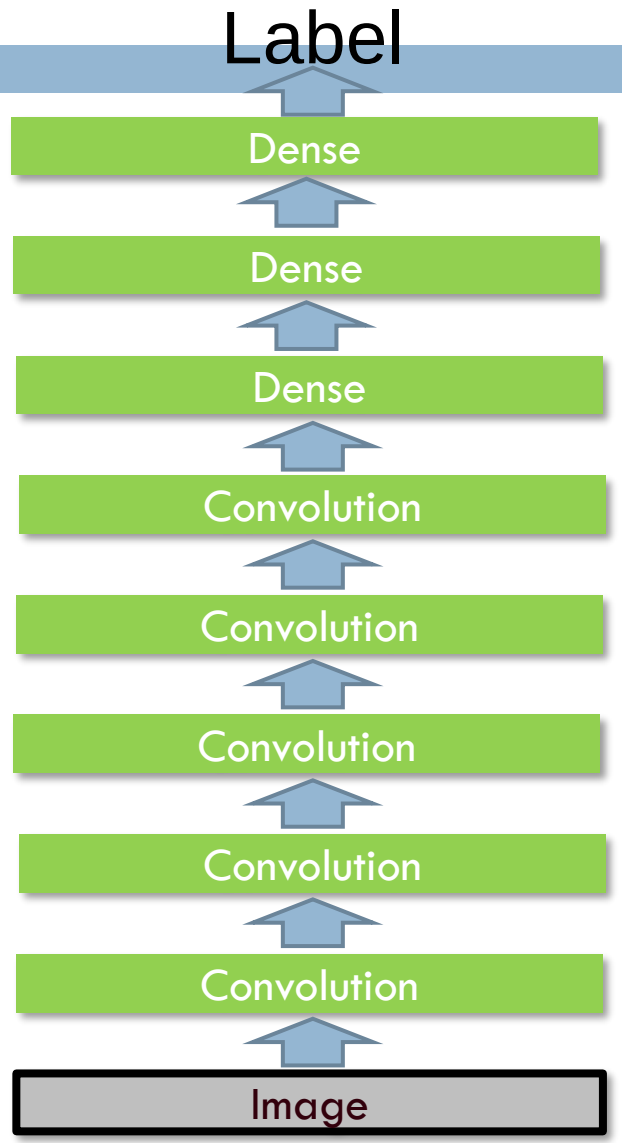
General principles of representation

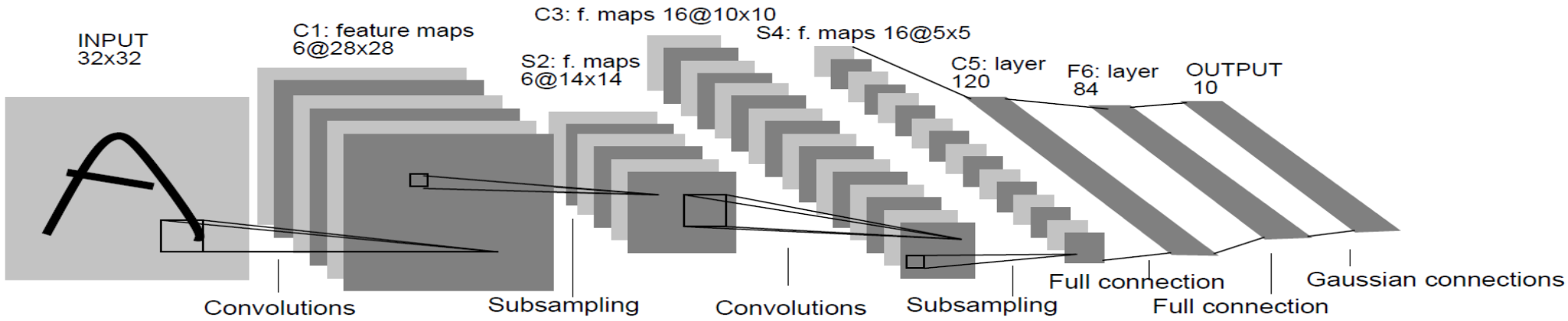
- **Coverage**
  - Ensure that all relevant info is captured
- **Concision**
  - Minimize number of features without sacrificing coverage
- **Directness**
  - Ideal features are independently useful for prediction

Shallow vs. deep learning

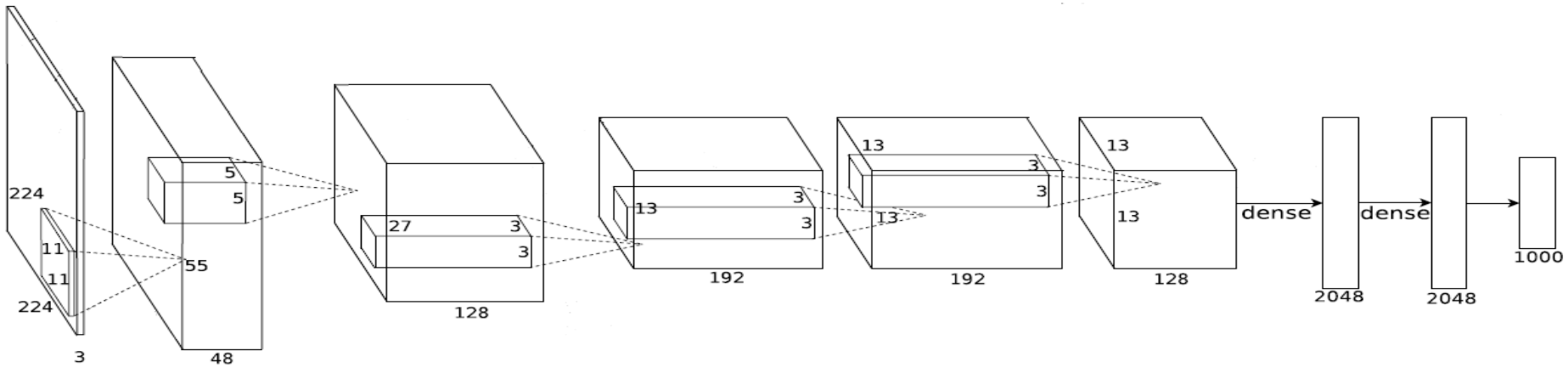
- Engineered vs. learned features





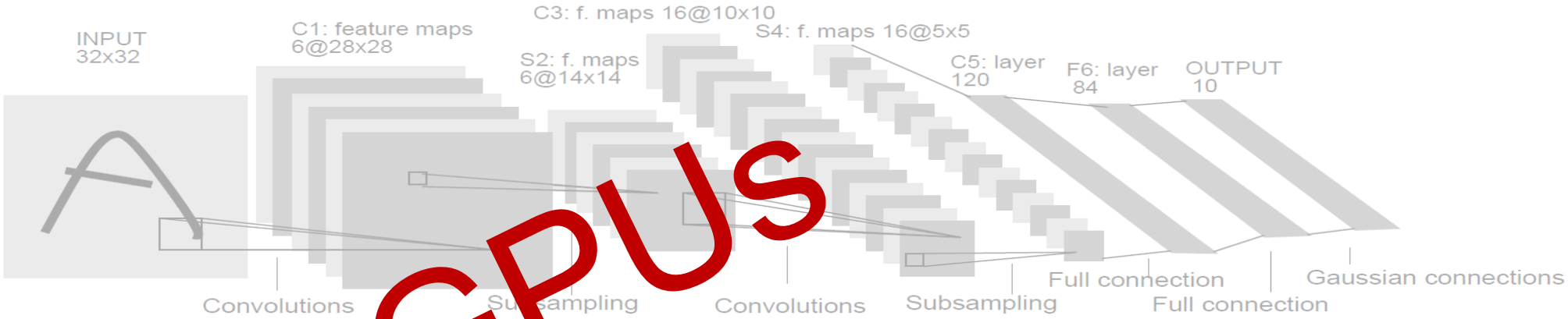


Gradient-Based Learning Applied to Document Recognition, LeCun, Bottou, Bengio and Haffner, Proc. of the IEEE, **1998**



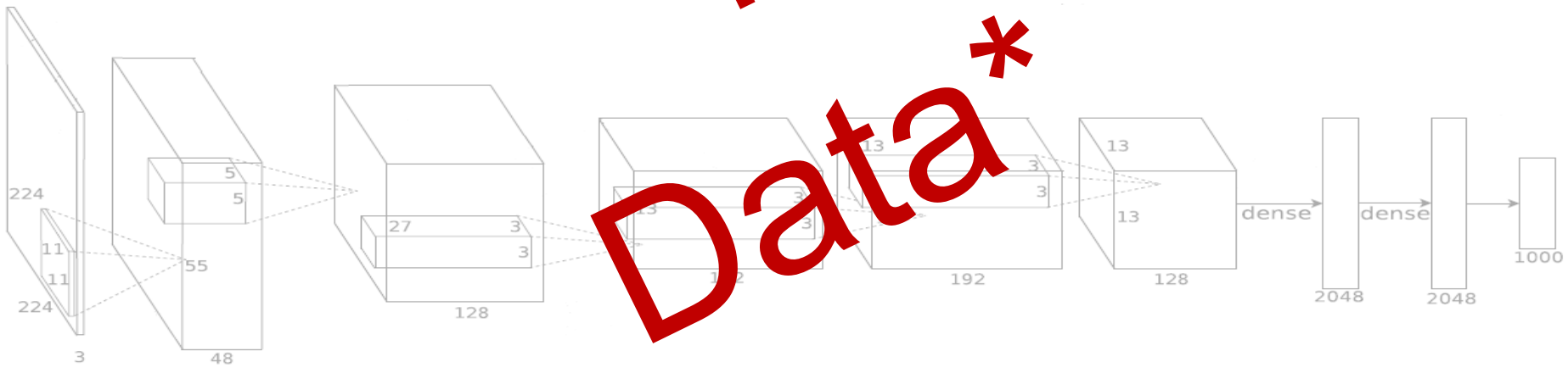
Imagenet Classification with Deep Convolutional Neural Networks, Krizhevsky, Sutskever, and Hinton, NIPS **2012**

Slide Credit: L. Zitnick



Gradient-Based Learning Applied to Document Recognition, LeCun, Bottou, Bengio and Haffner, Proc. of the IEEE, **1998**

GPUs + Data*



Imagenet Classifica
Sutskever, and Hinto

* Rectified activations and dropout

Slide Credit: L. Zitnick

CONVOLUTIONAL NEURAL NETWORKS

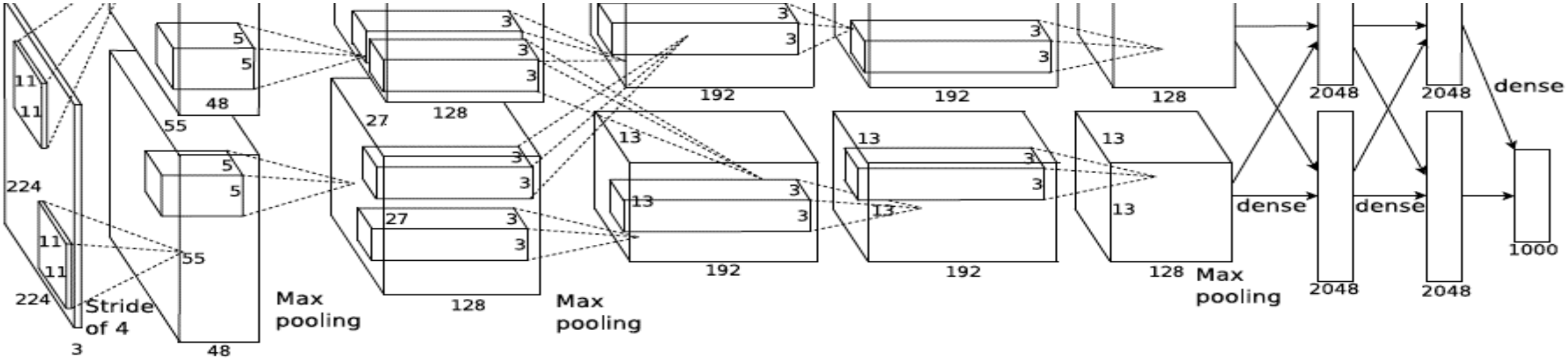


Image Categorization: Training phase

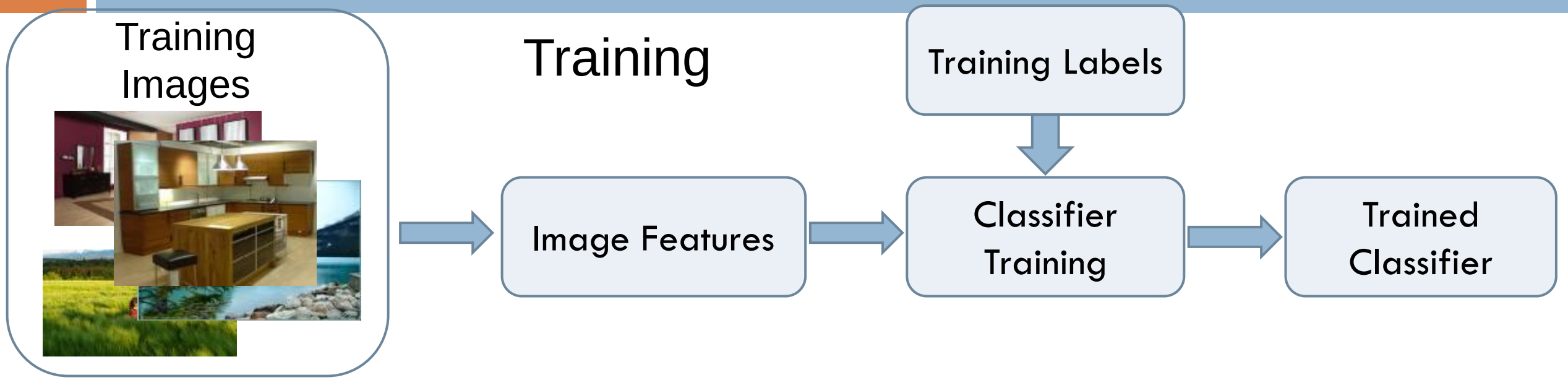


Image Categorization: Testing phase

Training

Training Images

Training Labels

Image Features

Classifier Training

Trained Classifier

Testing

Test Image

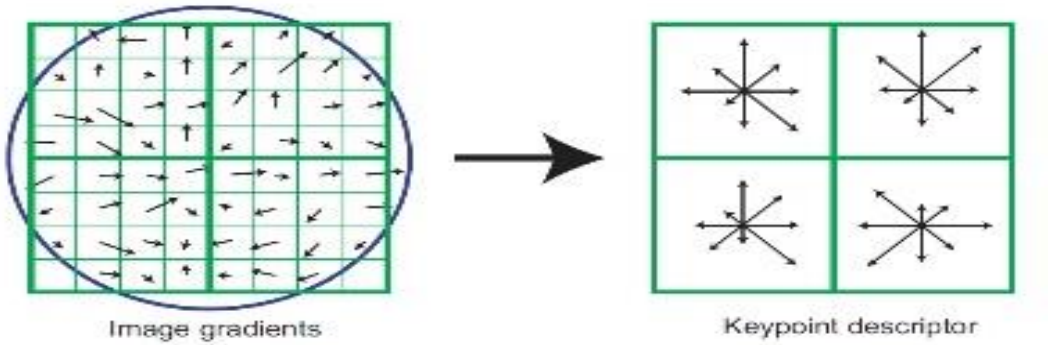
Image Features

Trained Classifier

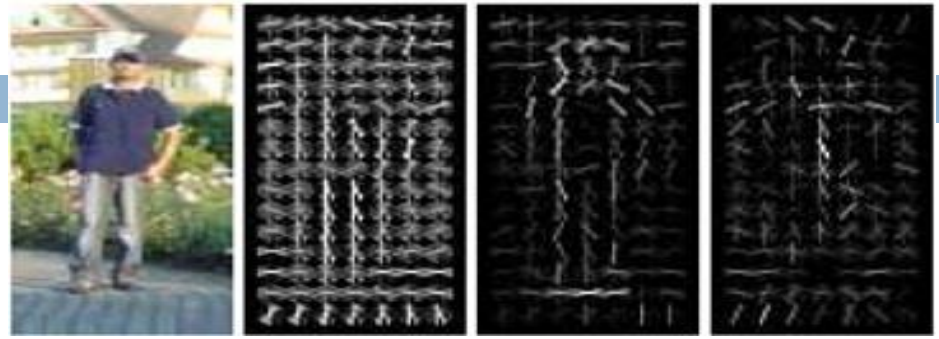
Prediction

Outdoor

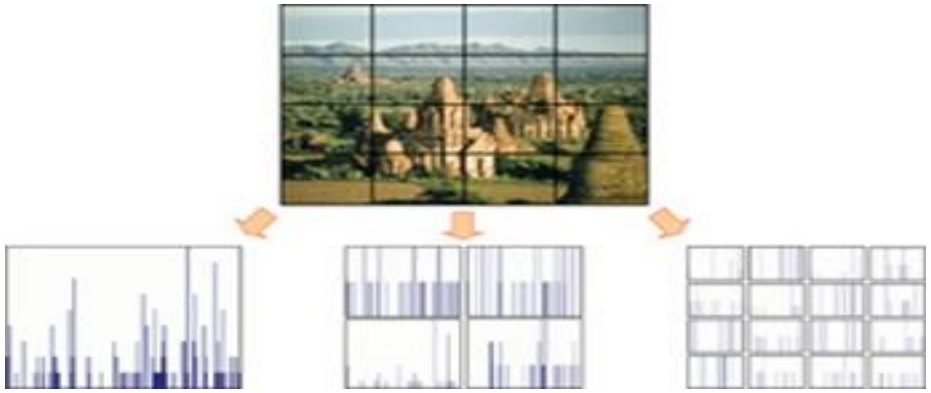
Features are the Keys



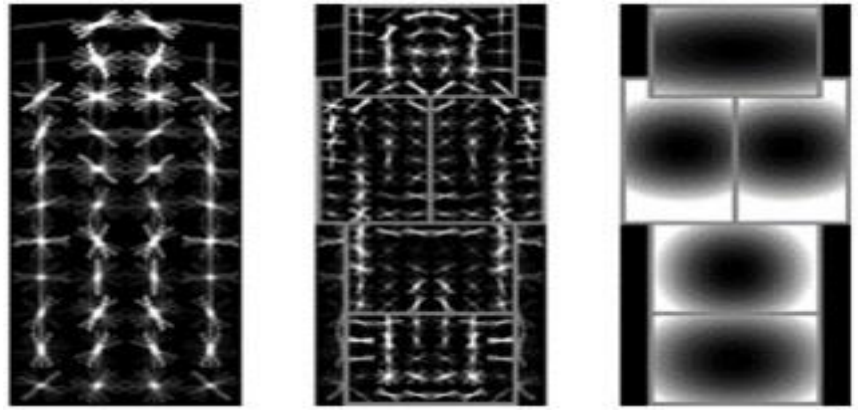
SIFT [Loewe IJCV 04]



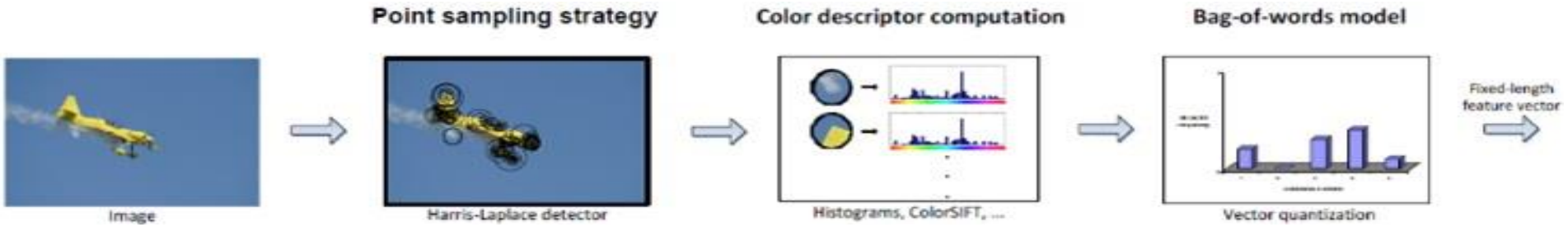
HOG [Dalal and Triggs CVPR 05]



SPM [Lazebnik et al. CVPR 06]



DPM [Felzenszwalb et al. PAMI 10]

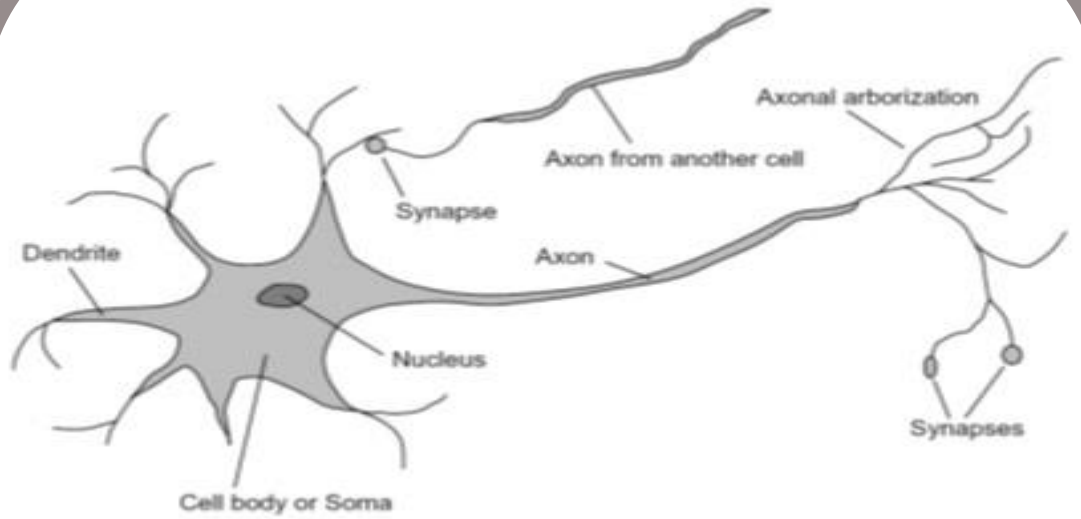


Color Descriptor [Van De Sande et al. PAMI 10]

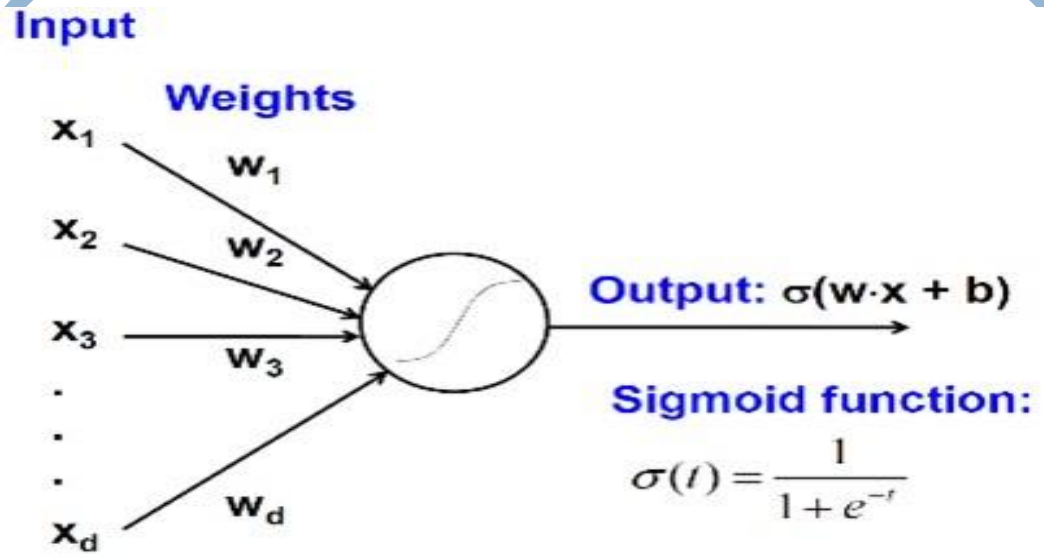
Learning a Hierarchy of Feature Extractors

- Each layer of hierarchy extracts features from output of previous layer
- All the way from pixels → classifier
- Layers have the (nearly) same structure

Biological neuron and Perceptrons



A biological neuron



An artificial neuron (Perceptron)
- a linear classifier

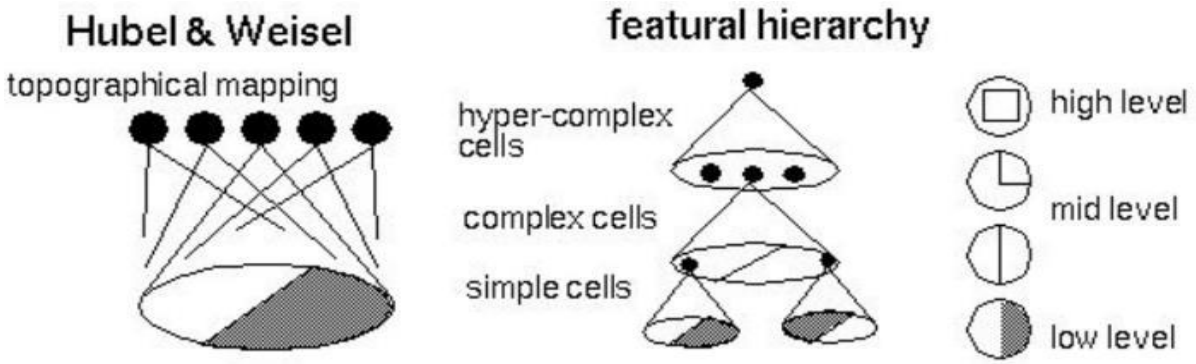
Simple, Complex and Hypercomplex cells

David H. Hubel and Torsten Wiesel

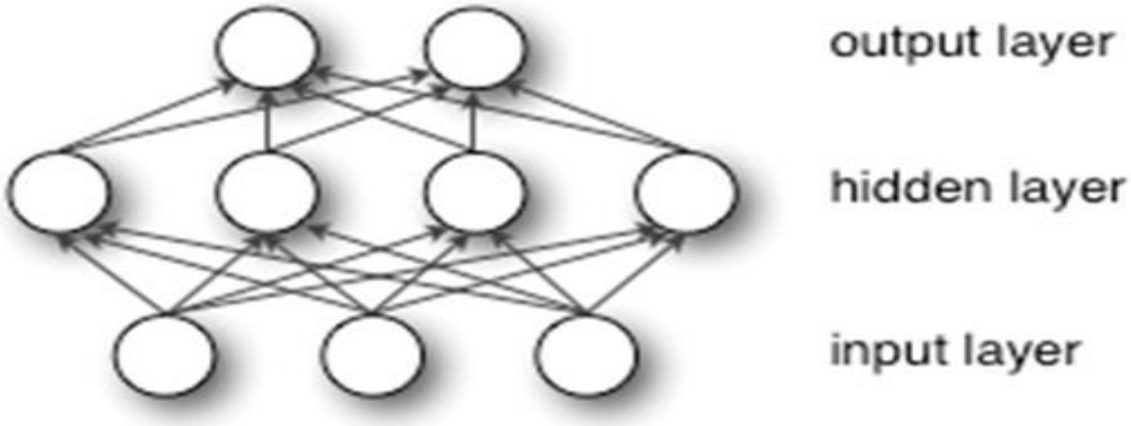
Suggested a **hierarchy** of **feature detectors** in the visual cortex, with higher level features responding to patterns of activation in lower level cells, and propagating activation upwards to still higher level cells.

David Hubel's Eye, Brain, and Vision

Hubel/Wiesel Architecture and Multi-layer Neural Network



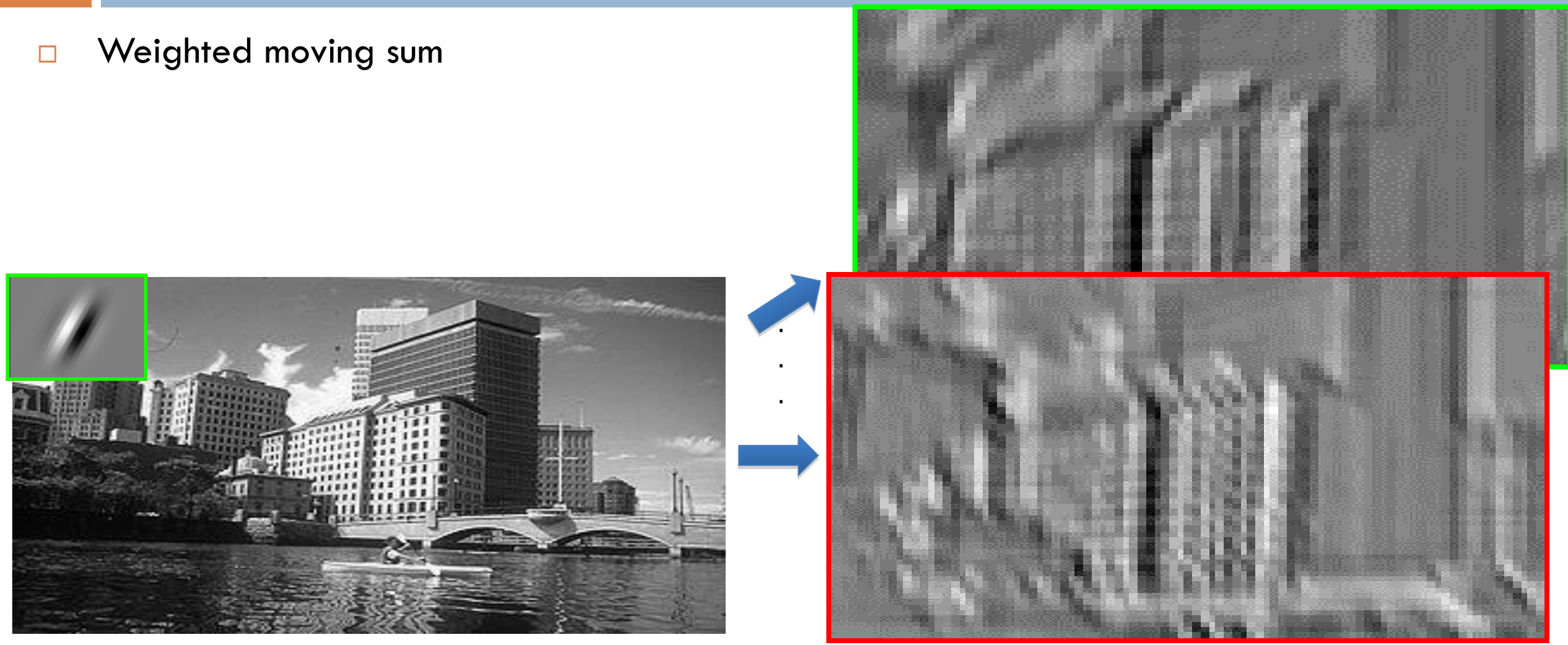
Hubel and Weisel's architecture



Multi-layer Neural Network
- A *non-linear* classifier

What is a Convolution?

- Weighted moving sum



Input

Feature Activation Map

slide credit: S. Lazebnik

Image as an array of numbers

- To bridge the gap between pixels and “meaning”

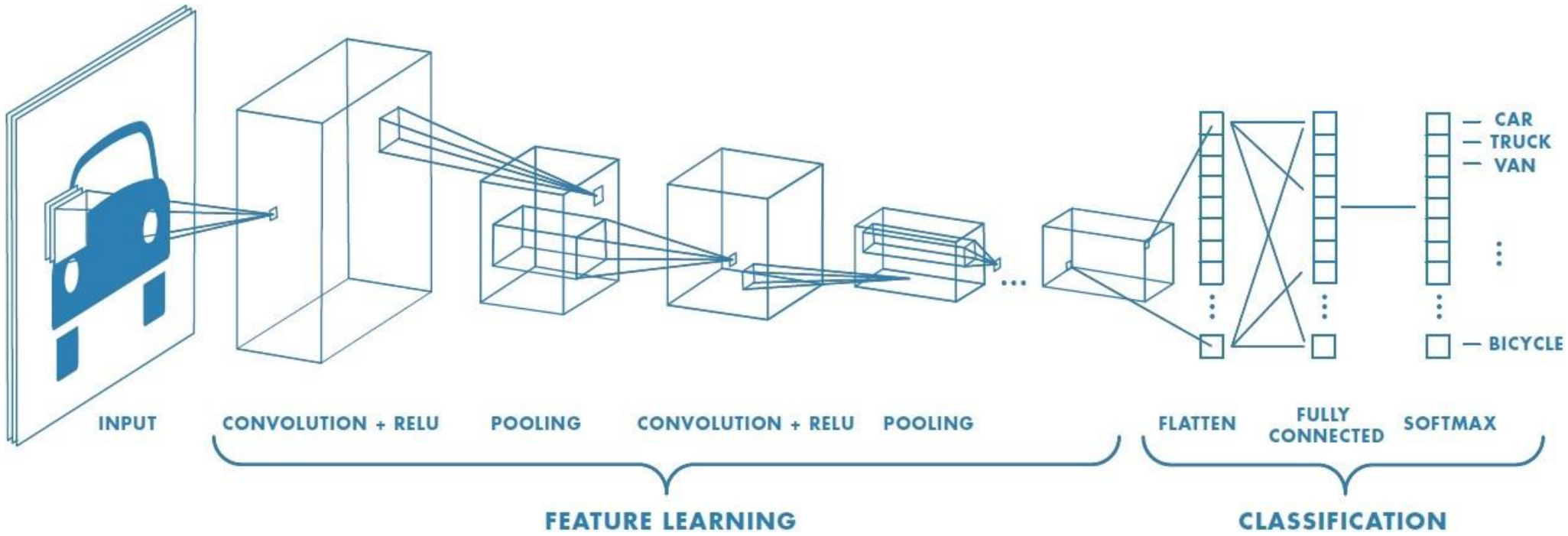
What we see

0	3	2	5	4	7	6	9	8
3	0	1	2	3	4	5	6	7
2	1	0	3	2	5	4	7	6
5	2	3	0	1	2	3	4	5
4	3	2	1	0	3	2	5	4
7	4	5	2	3	0	1	2	3
6	5	4	3	2	1	0	3	2
9	6	7	4	5	2	3	0	1
8	7	6	5	4	3	2	1	0

What a computer sees

Source: S. Narasimhan

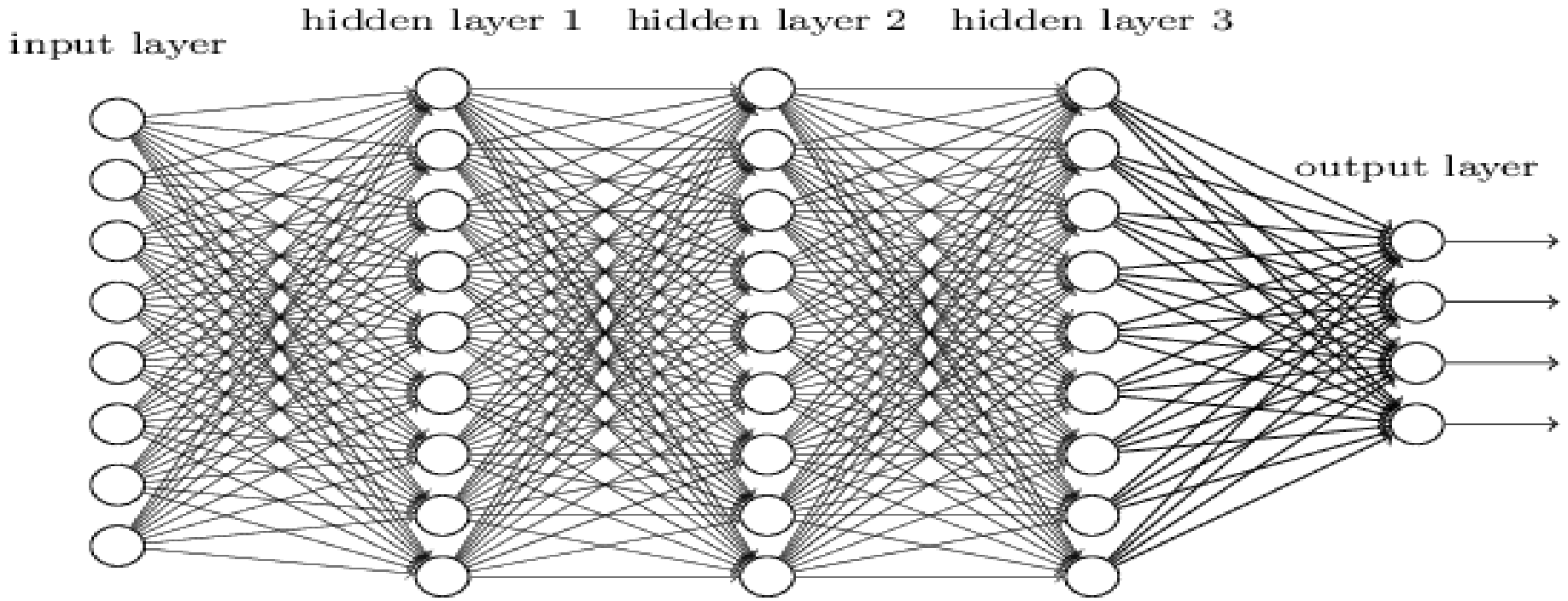
CNN architecture



Ref: [medium] https://medium.com/@RaghavPrabhu/understanding-of-convolutional-neural-network-cnn-deep-learning-99760835f148] for main CNN image.

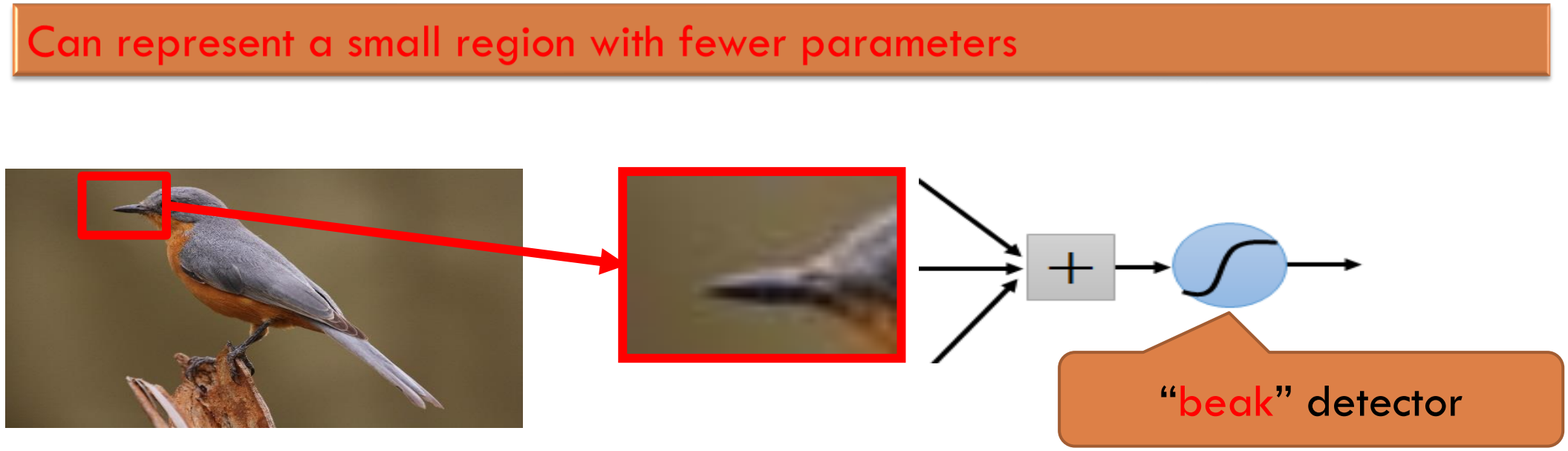
CNN

- We know it is good to learn a small model.
- From this fully connected model, do we really need all the edges?
- Can some of these be shared?



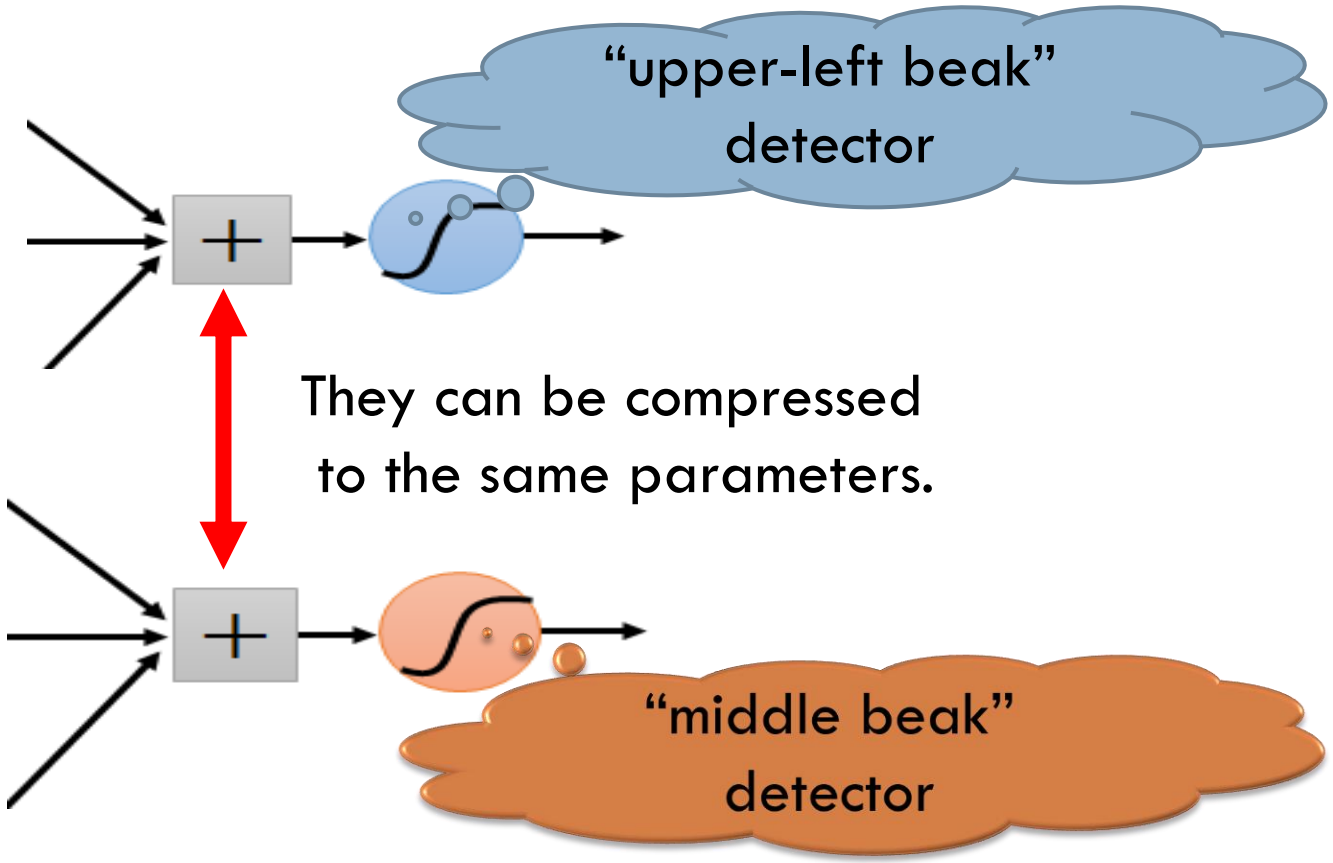
Consider learning an image:

- Some patterns are much smaller than the whole image



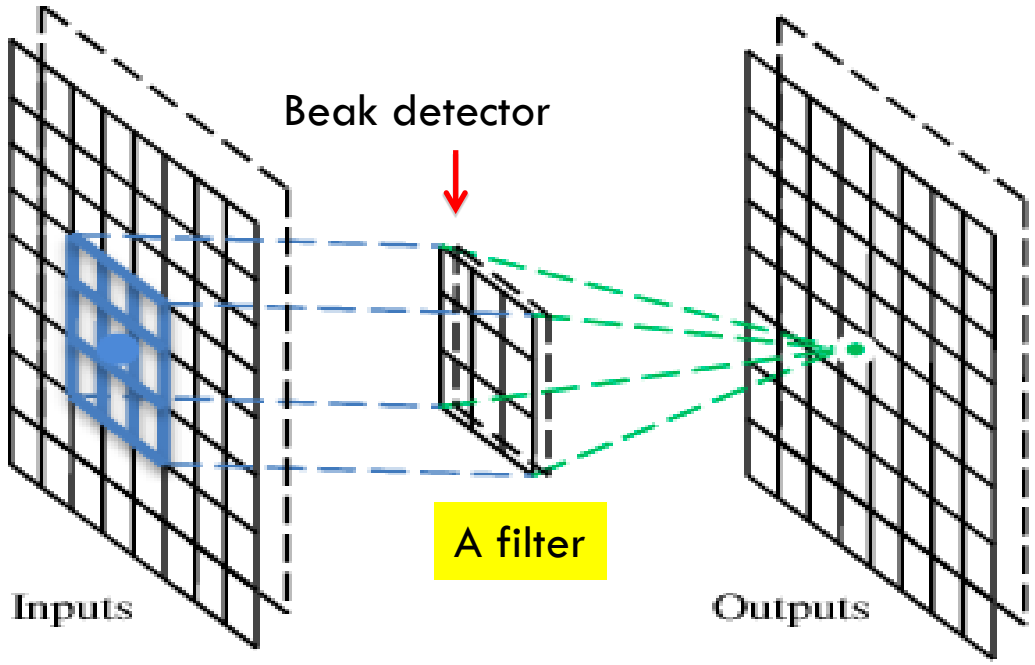
Same pattern appears in different places:They can be compressed!

What about training a lot of such "small" detectors and each detector must "move around".



A convolutional layer

A CNN is a neural network with some convolutional layers (and some other layers). A convolutional layer has a number of filters that does convolutional operation.



Convolution

1	0	0	0	0	1
0	1	0	0	1	0
0	0	1	1	0	0
1	0	0	0	1	0
0	1	0	0	1	0
0	0	1	0	1	0

6 x 6 image

These are the network parameters to be learned.

1	-1	-1
-1	1	-1
-1	-1	1

Filter 1

-1	1	-1
-1	1	-1
-1	1	-1

Filter 2

⋮ ⋮

Each filter detects a small pattern (3 x 3).

Convolution

1	-1	-1
-1	1	-1
-1	-1	1

Filter 1

stride=1

1	0	0	0	0	1
0	1	0	0	1	0
0	0	1	1	0	0
1	0	0	0	1	0
0	1	0	0	1	0
0	0	1	0	1	0

6 x 6 image

Dot product →

3 -1

Convolution

1	-1	-1
-1	1	-1
-1	-1	1

Filter 1

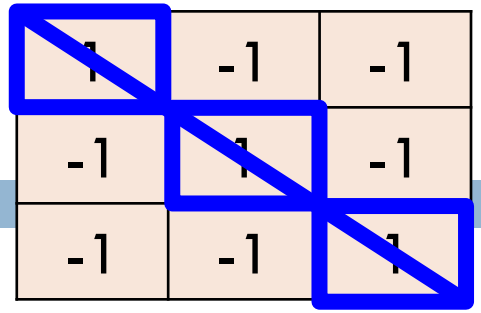
If stride=2

1	0	0	0	0	1
0	1	0	0	1	0
0	0	1	1	0	0
1	0	0	0	1	0
0	1	0	0	1	0
0	0	1	0	1	0

6 x 6 image

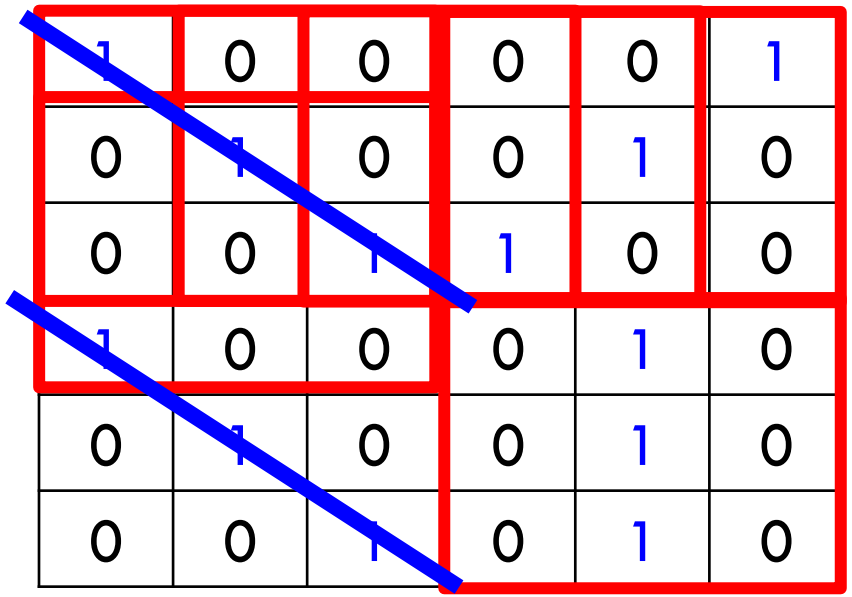
3 -3

Convolution

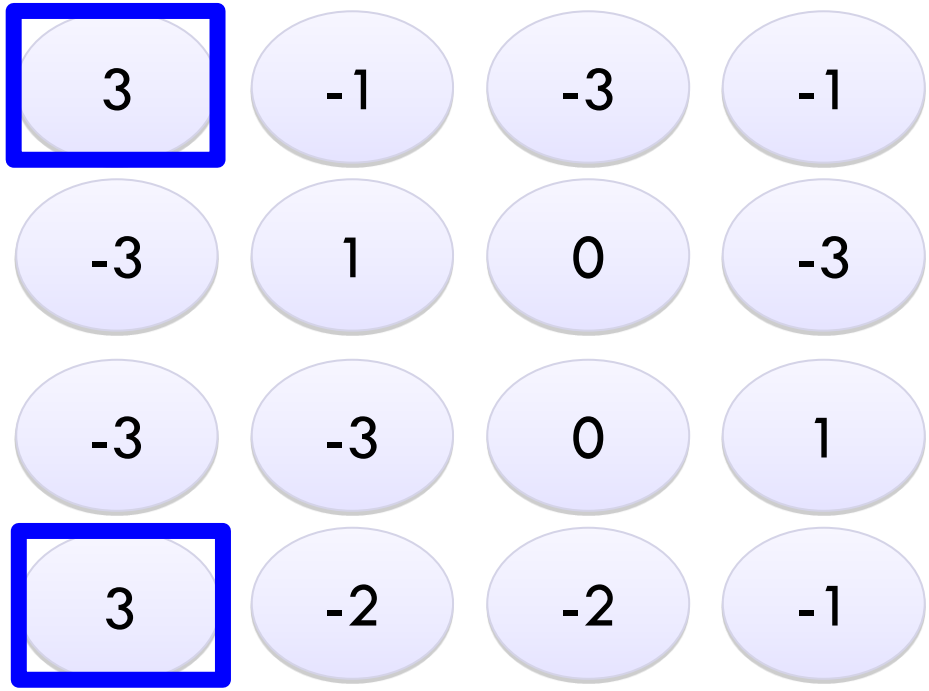


Filter 1

stride=1



6 x 6 image



Convolution

-1	1	-1
-1	1	-1
-1	1	-1

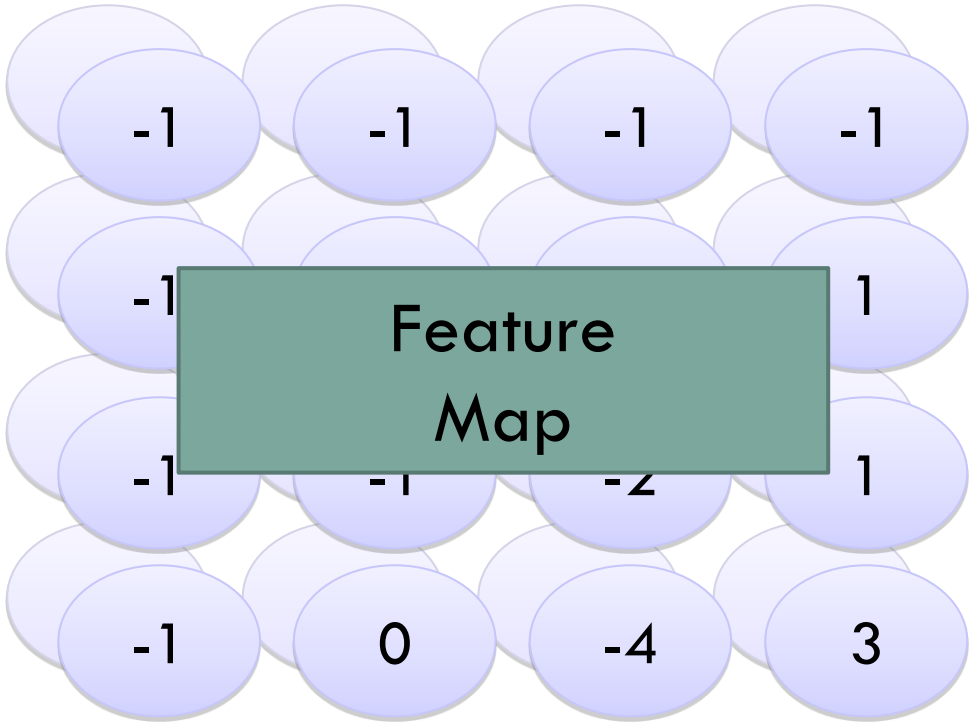
Filter 2

stride=1

1	0	0	0	0	1
0	1	0	0	1	0
0	0	1	1	0	0
1	0	0	0	1	0
0	1	0	0	1	0
0	0	1	0	1	0

6 x 6 image

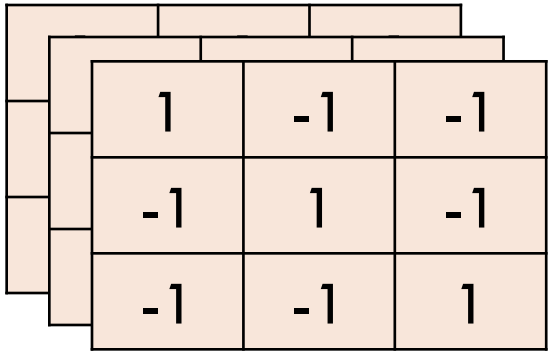
Repeat this for each filter



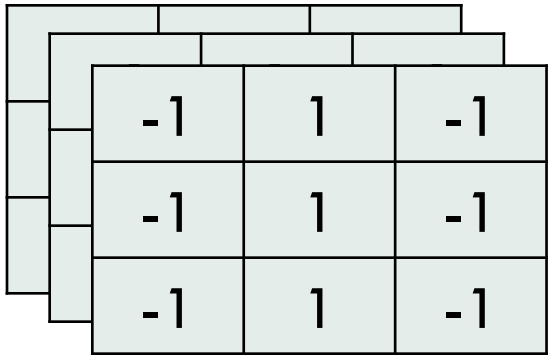
Two 4 x 4 images

Forming 2 x 4 x 4 matrix

Color image: RGB 3 channels

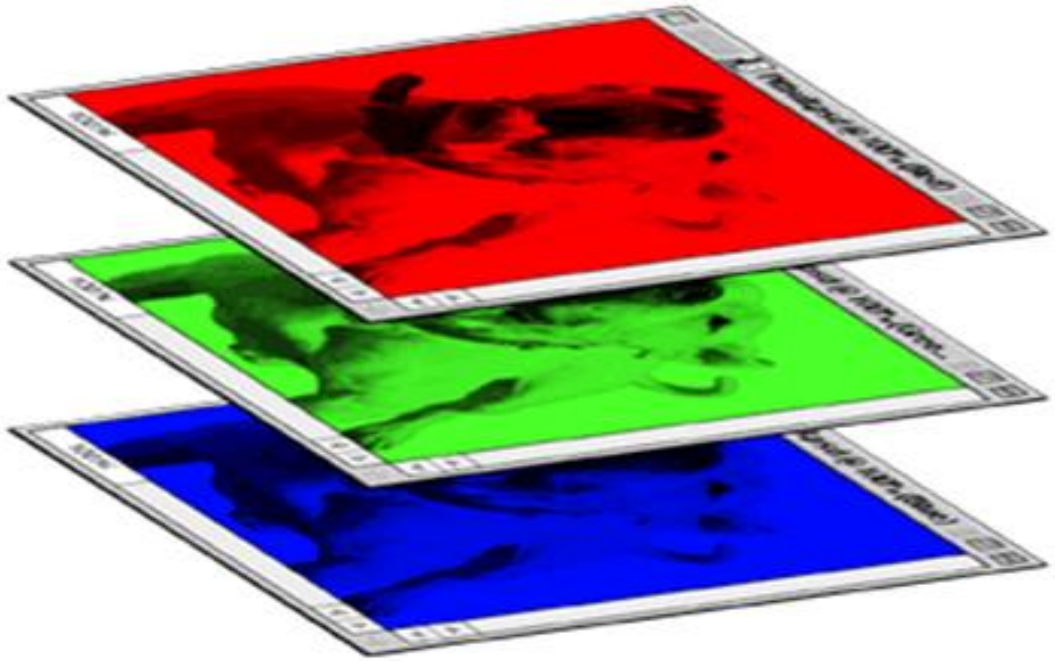


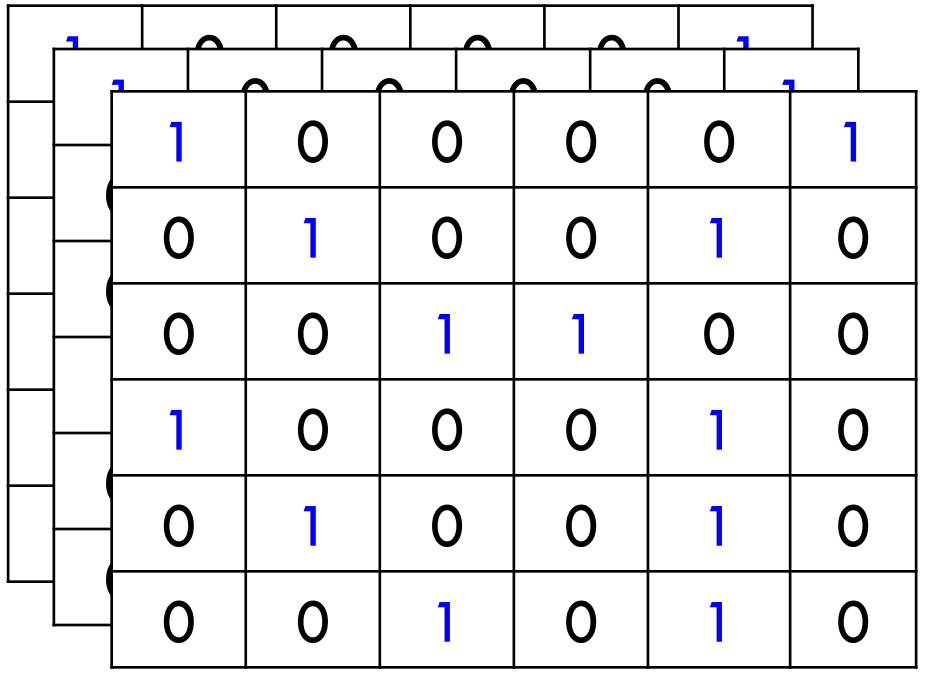
Filter 1



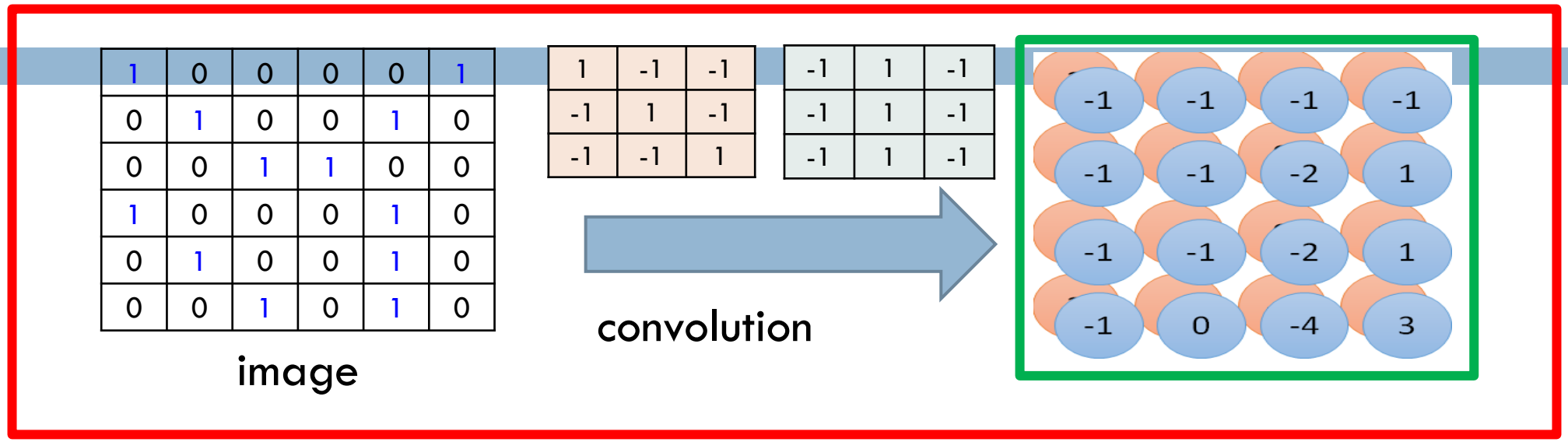
Filter 2

Color image



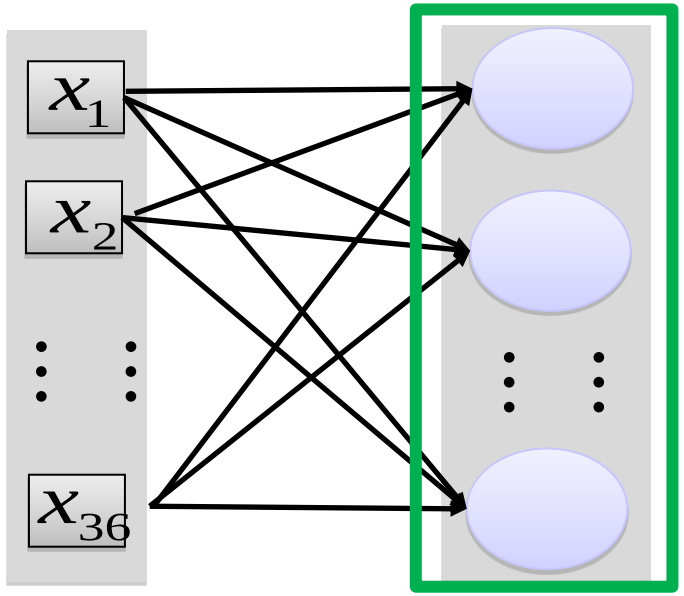


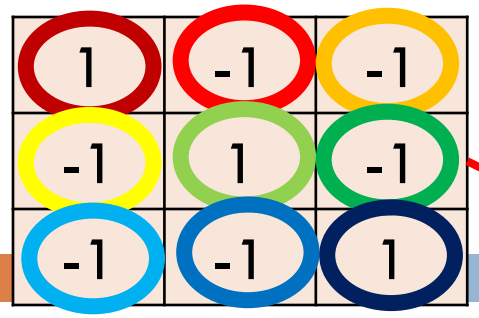
Convolution v.s. Fully Connected



Fully-connected

1	0	0	0	0	1
0	1	0	0	1	0
0	0	1	1	0	0
1	0	0	0	1	0
0	1	0	0	1	0
0	0	1	0	1	0





Filter 1

1	0	0	0	0
0	1	0	0	1
0	0	1	1	0
1	0	0	0	1
0	1	0	0	1
0	0	1	0	1

6 x 6 image

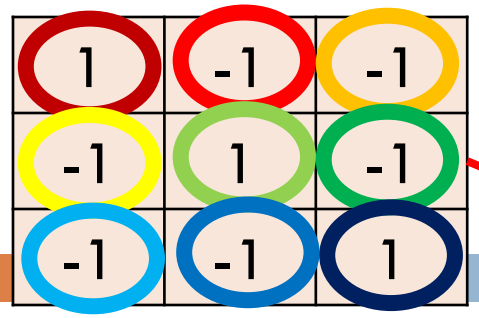
3	-1	-3	-1
-3	1	0	-3
-3	-3	0	1
3	-2	-2	-1

1 1
2 0
3 0
4: 0
⋮
0
8 1
9 0
10: 0
⋮
13 0
14 0
15 1
16 1
⋮

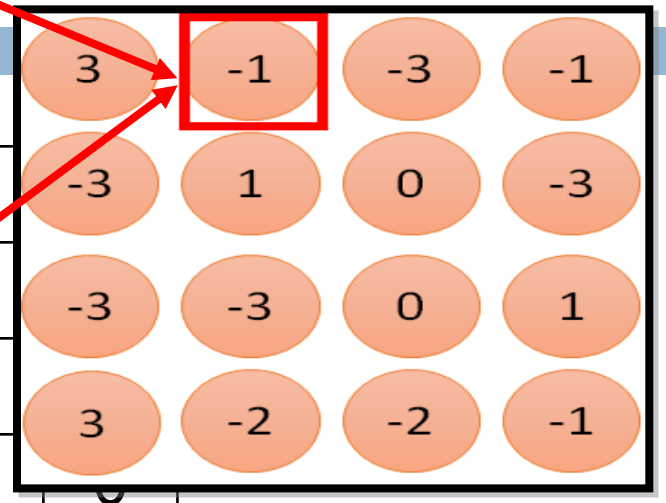
3

fewer parameters!

Only connect to 9 inputs, not fully connected



Filter 1

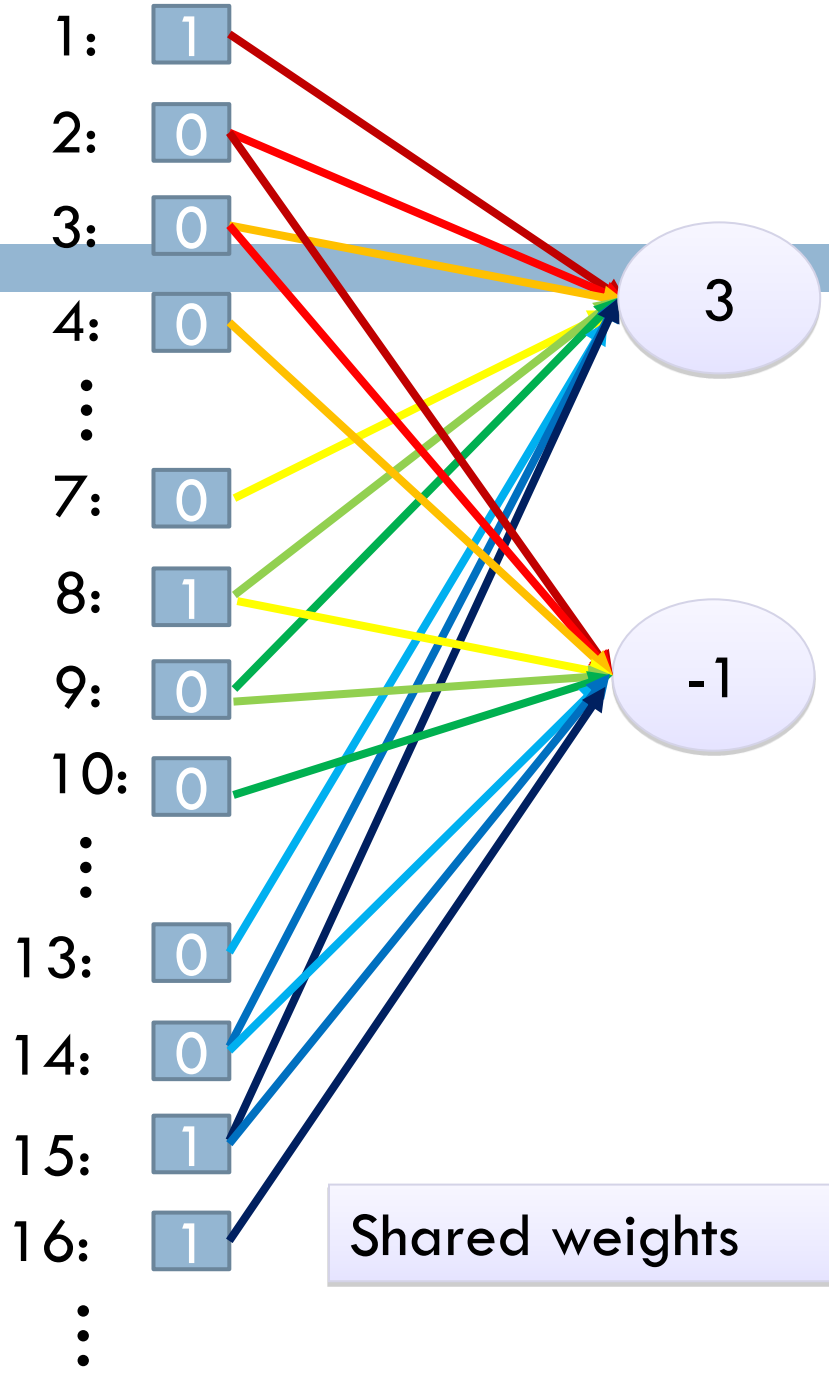


1	0	0	0	0	
0	1	0	0	1	
0	0	1	1	0	
1	0	0	0	1	
0	1	0	0	1	0
0	0	1	0	1	0

6 x 6 image

Fewer parameters

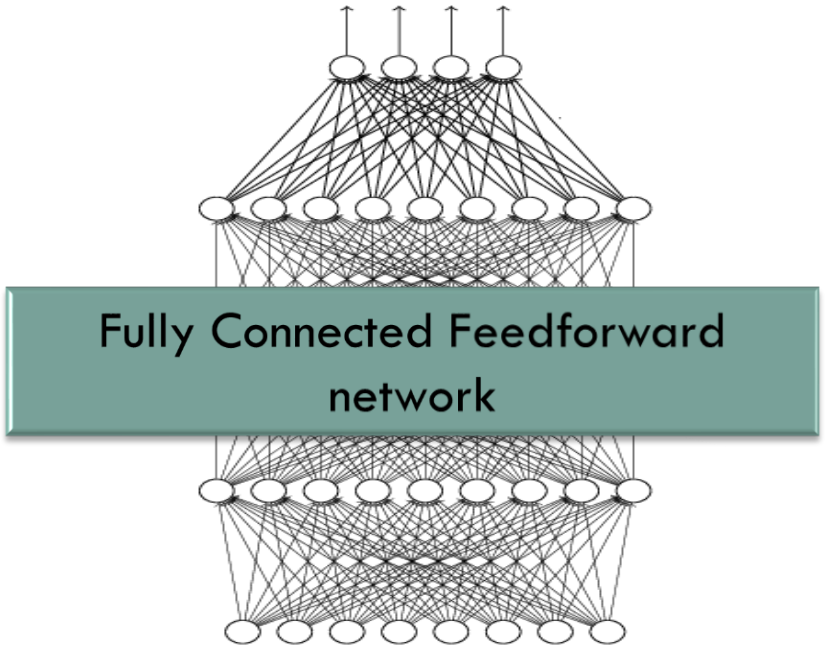
Even fewer parameters



Shared weights

The whole CNN

cat dog ……



Convolution

Max Pooling

Convolution

Max Pooling

Can repeat many times

Flattened

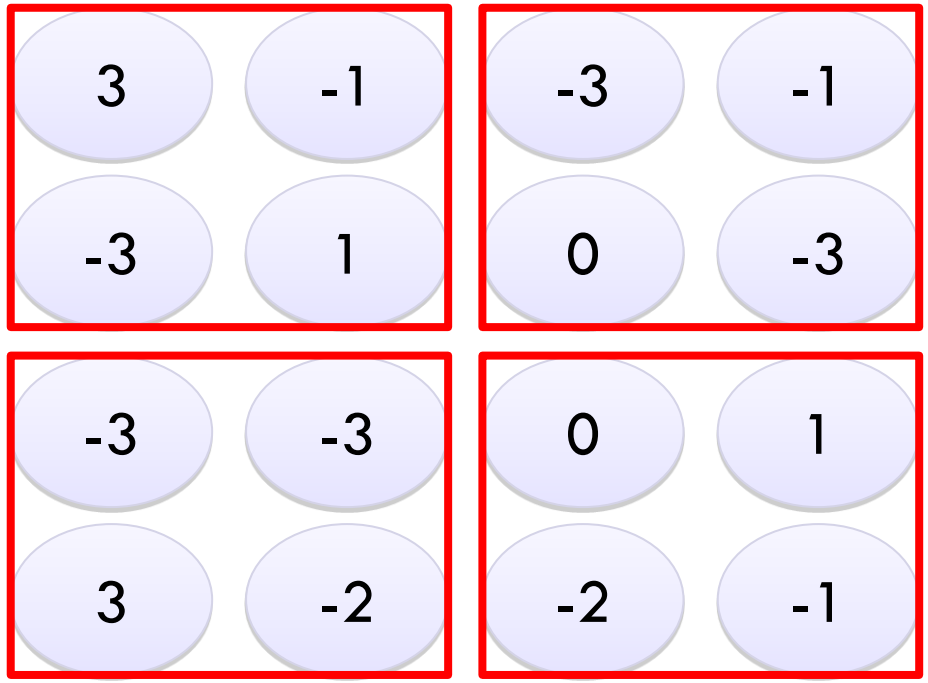
Max Pooling

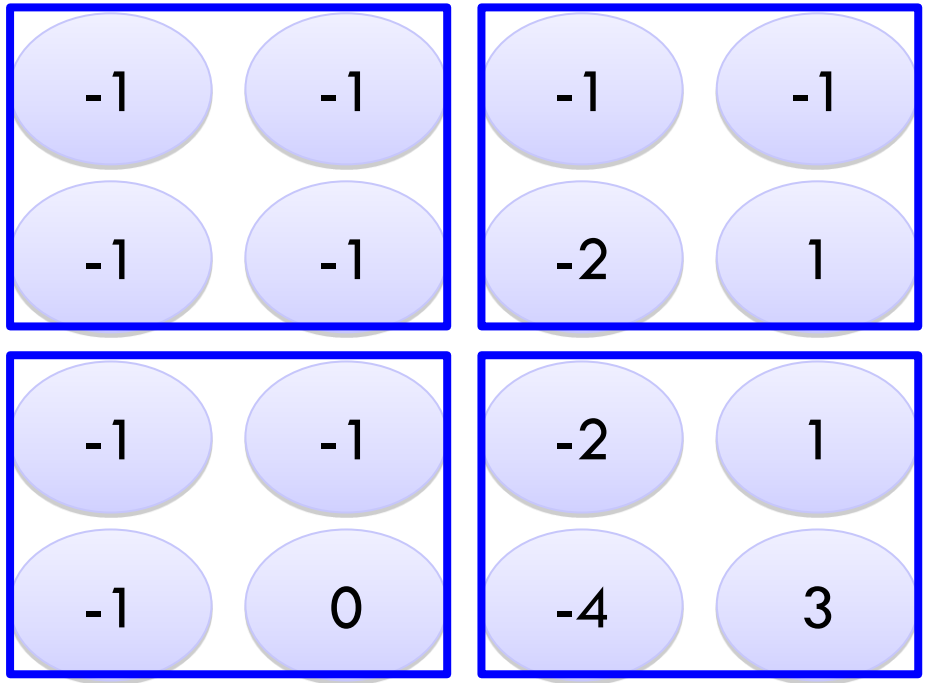
1	-1	-1
-1	1	-1
-1	-1	1

Filter 1

-1	1	-1
-1	1	-1
-1	1	-1

Filter 2





Why Pooling

- Subsampling pixels will not change the object

bird

Subsampling

bird

We can subsample the pixels to make image smaller

→ fewer parameters to characterize the image

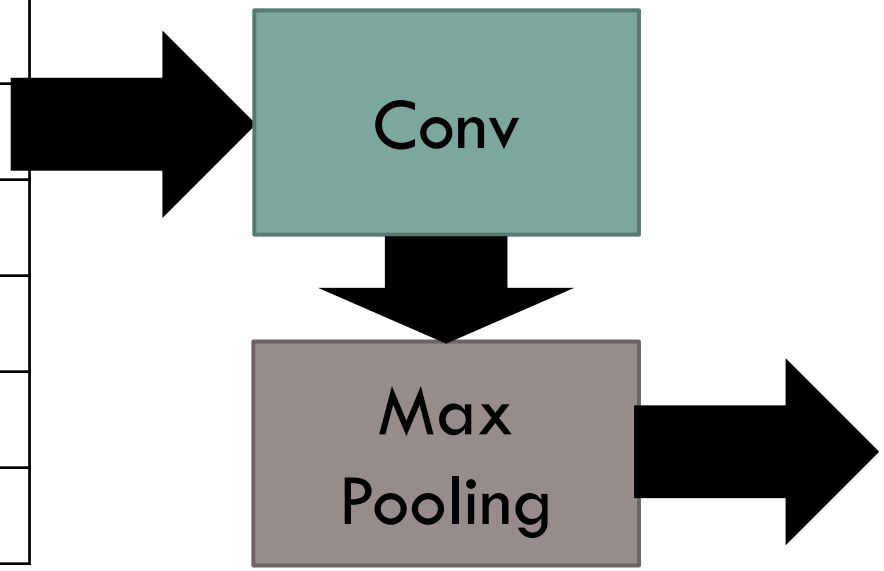
A CNN compresses a fully connected network in two ways:

- Reducing number of connections
- Shared weights on the edges
- Max pooling further reduces the complexity

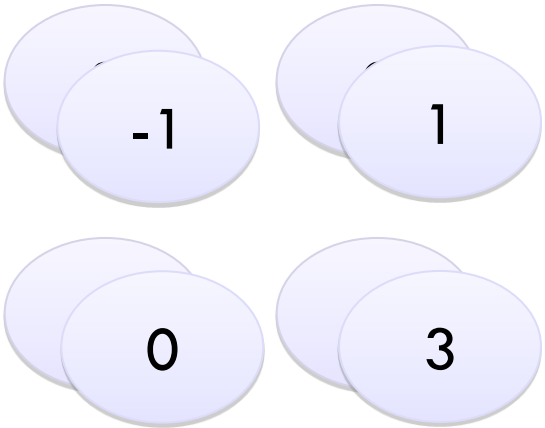
Max Pooling

1	0	0	0	0	1
0	1	0	0	1	0
0	0	1	1	0	0
1	0	0	0	1	0
0	1	0	0	1	0
0	0	1	0	1	0

6 x 6 image



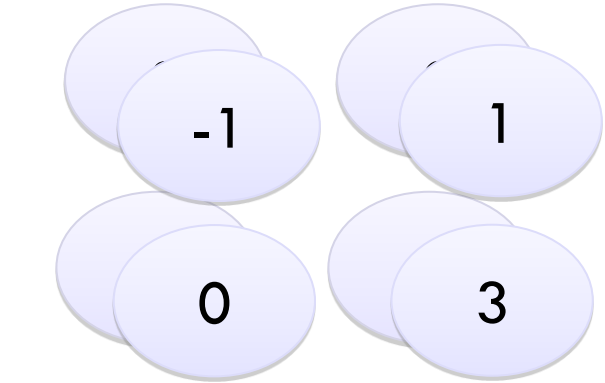
New image but smaller

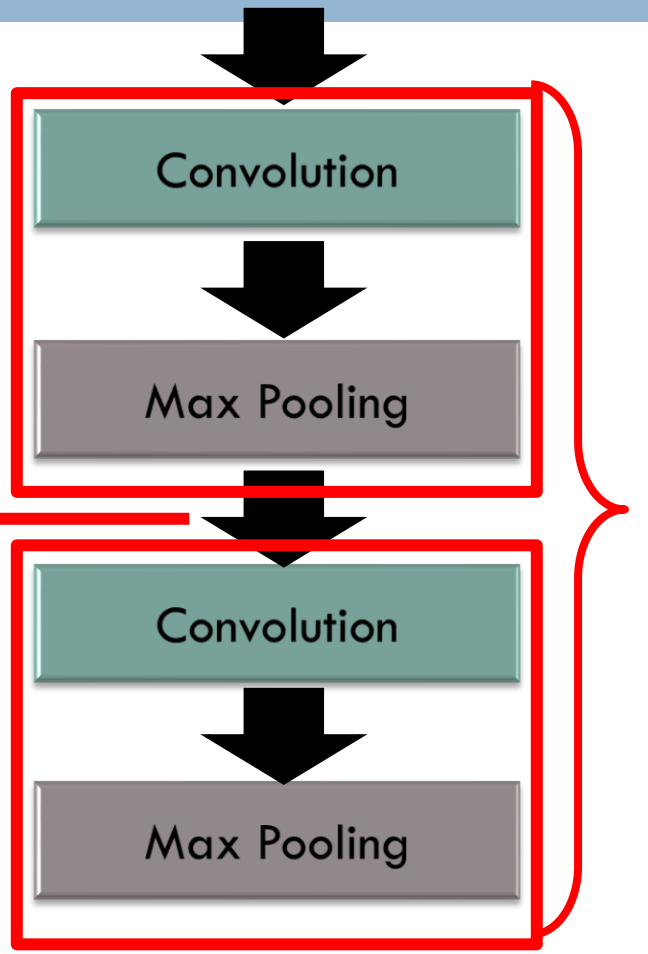


2 x 2 image

Each filter is a channel

The whole CNN



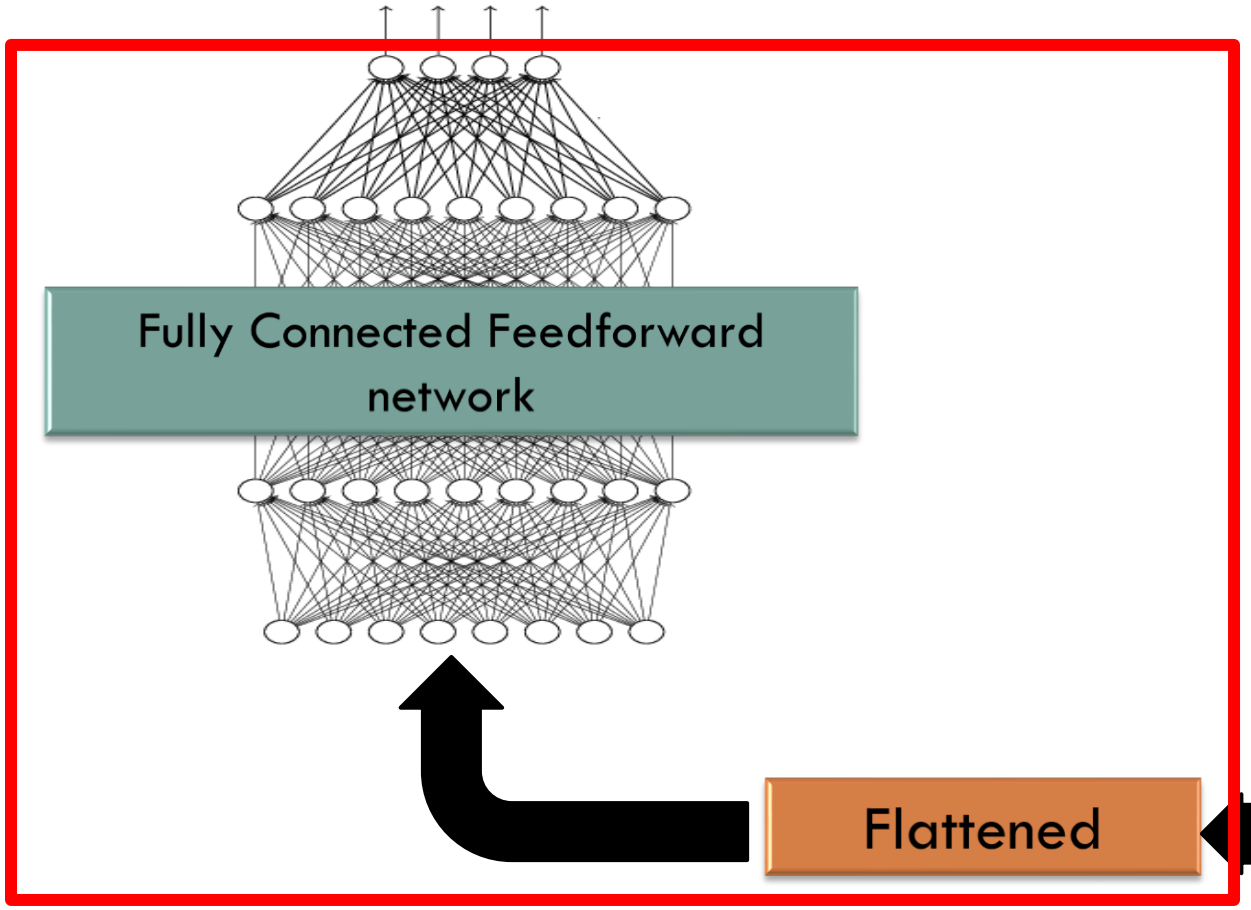


Smaller than the original image

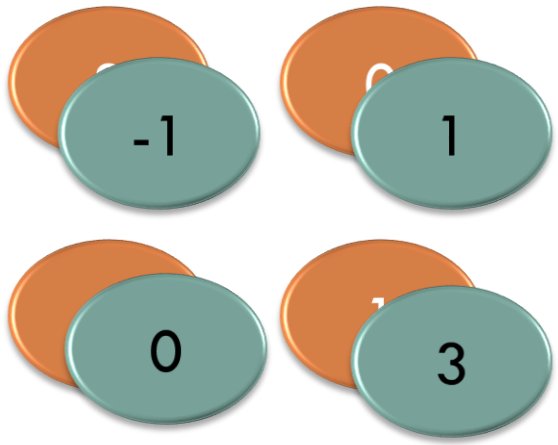
The number of channels is the number of filters

The whole CNN

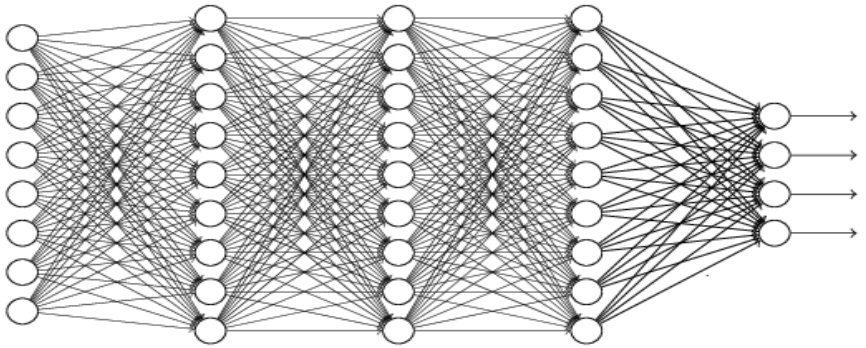
cat dog ……



Flattening



Flattened

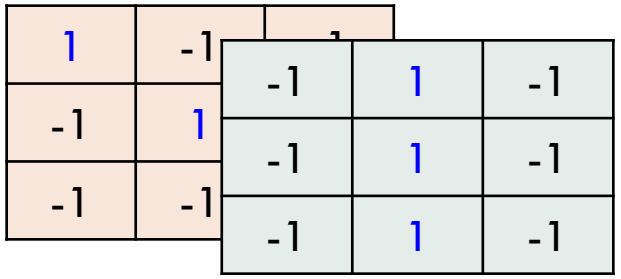


Fully Connected Feedforward network

CNN in Keras

Only modified the ***network structure*** and ***input format (vector -> 3-D tensor)***

```
model2.add( Convolution2D( 25,3,3,
                input_shape=(28,28,1)) )
```



……

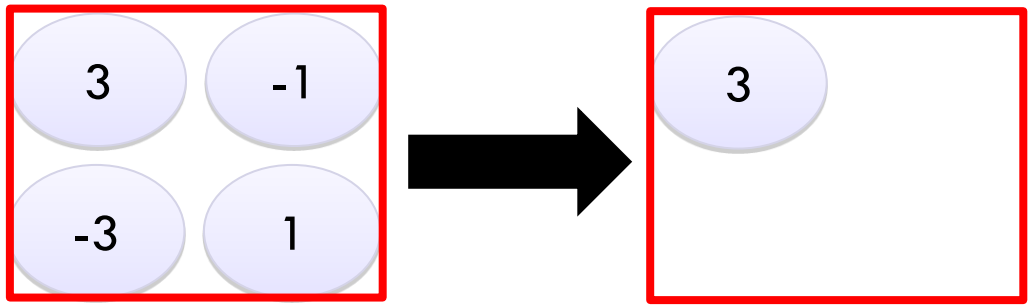
There are 25 3x3 filters.

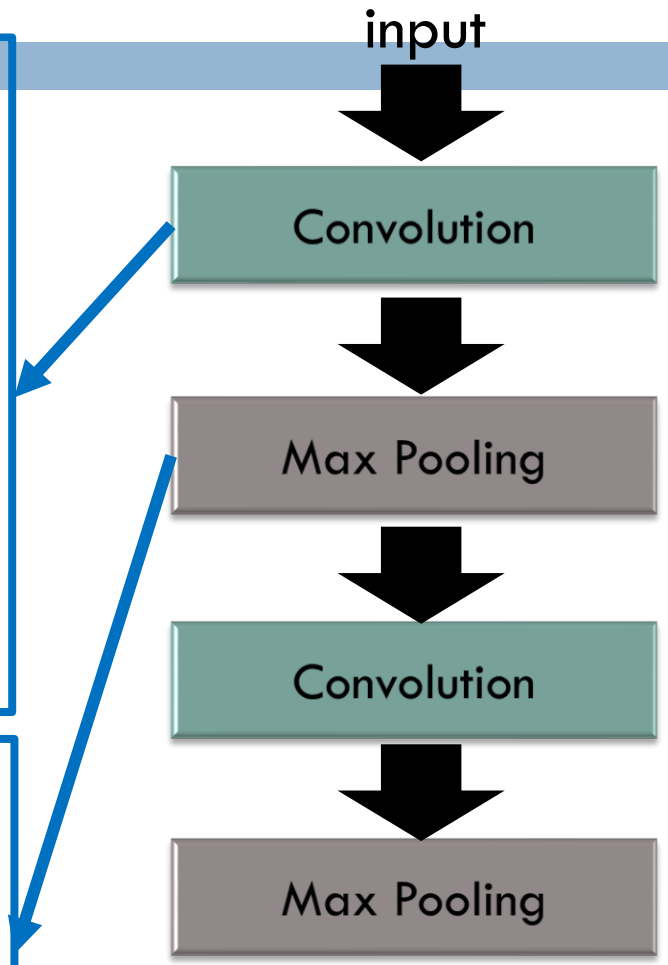
Input_shape = (28 , 28 , 1)

28 x 28 pixels

1: black/white, 3: RGB

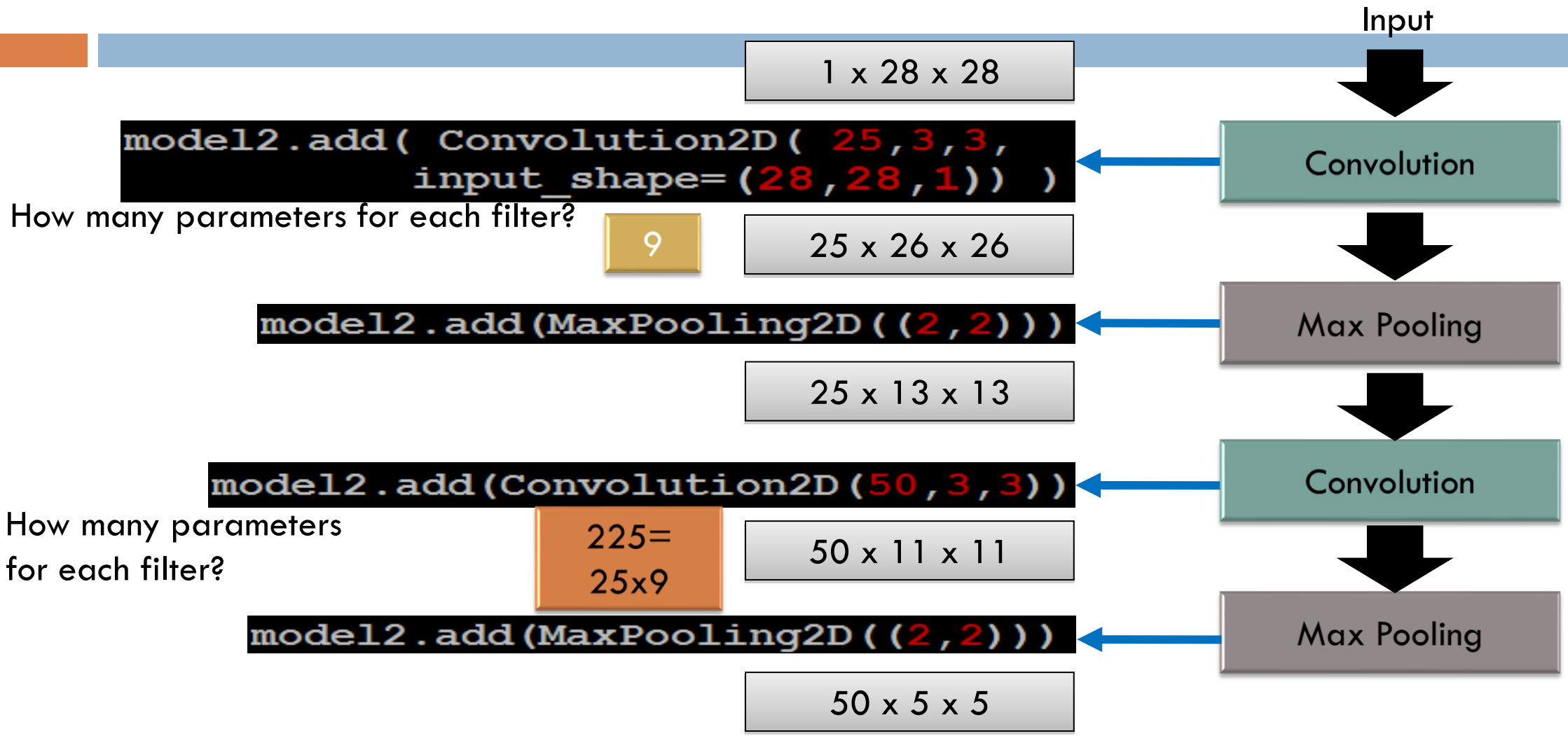
```
model2.add(MaxPooling2D((2,2)))
```





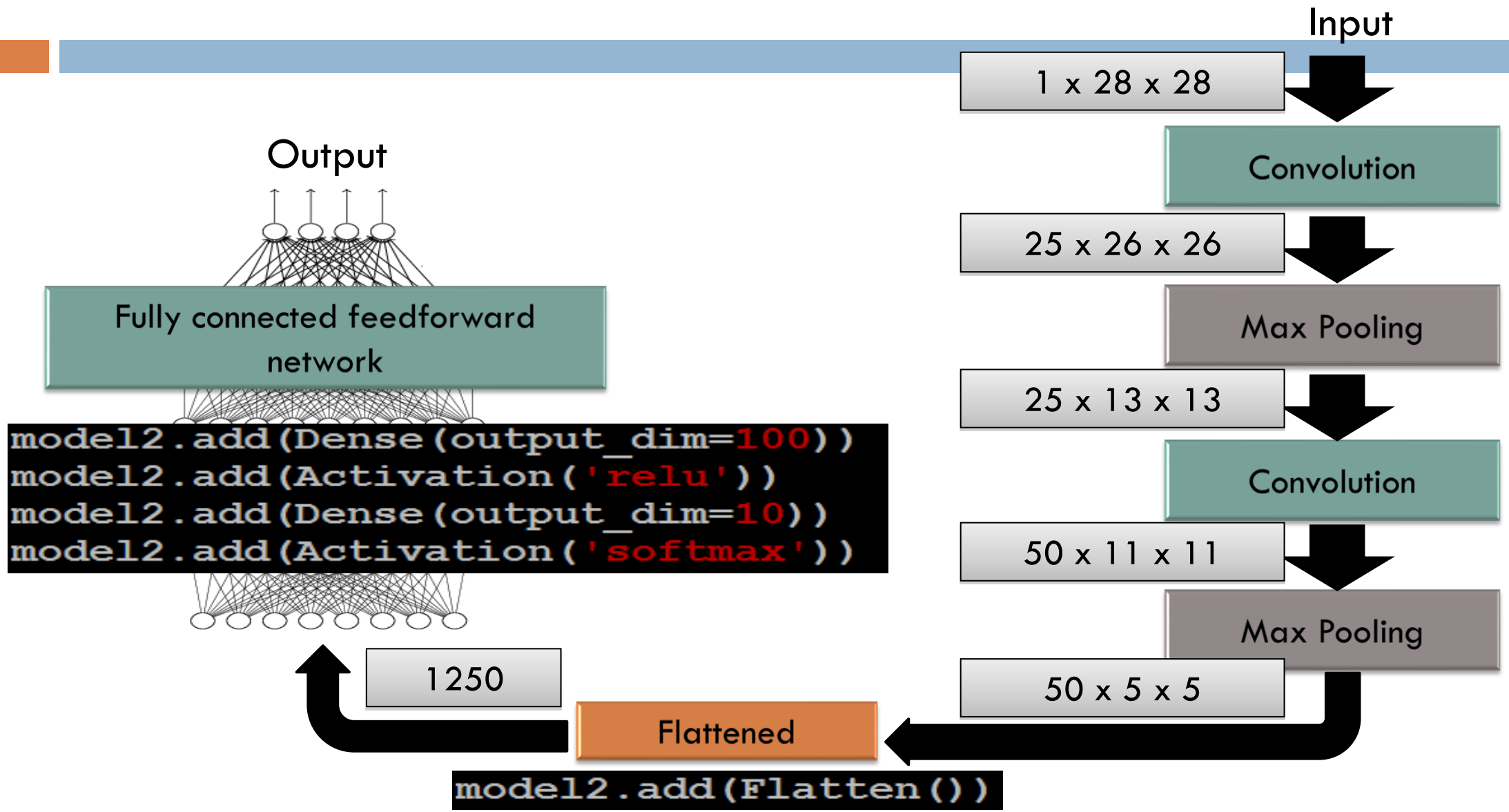
CNN in Keras

Only modified the ***network structure*** and ***input format (vector -> 3-D array)***

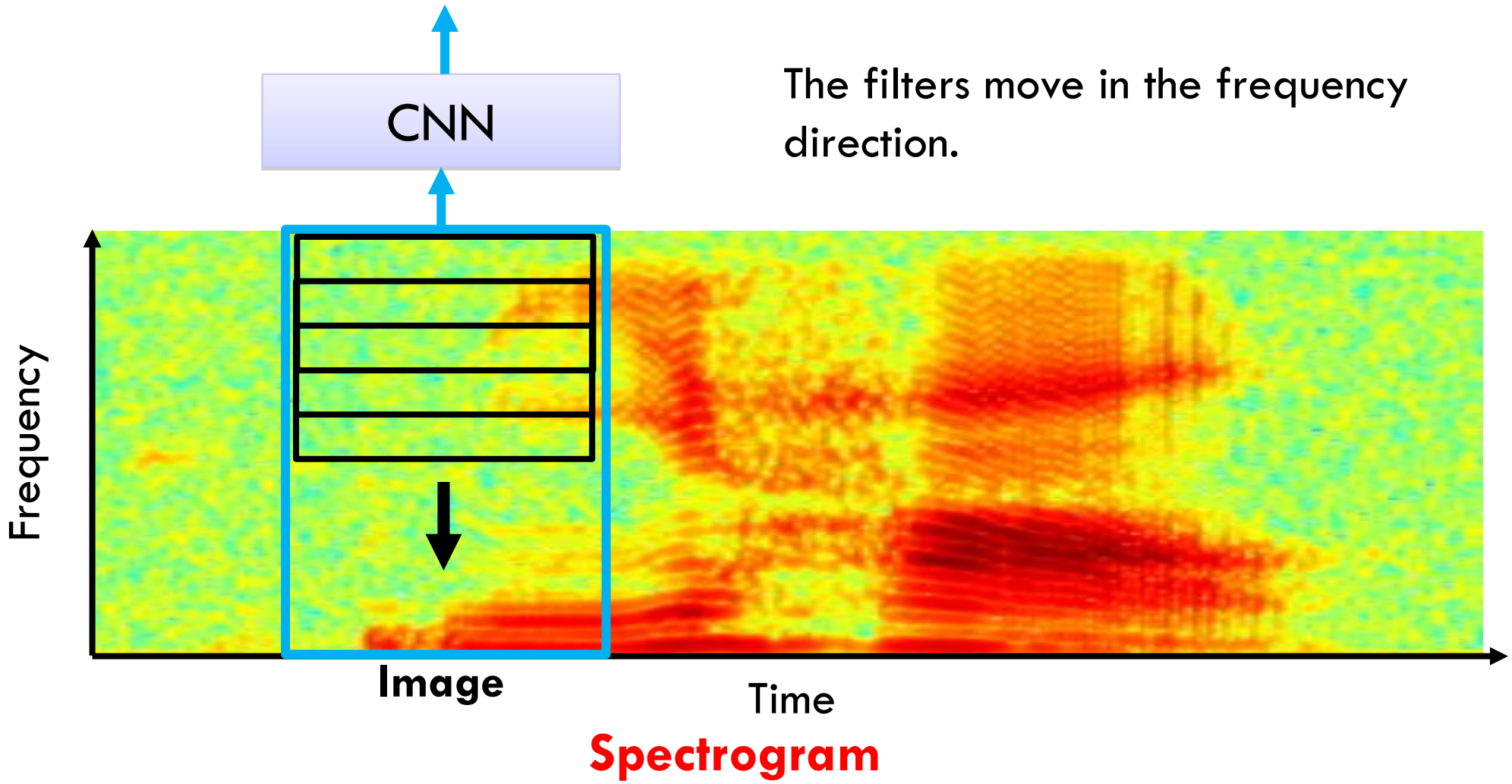


CNN in Keras

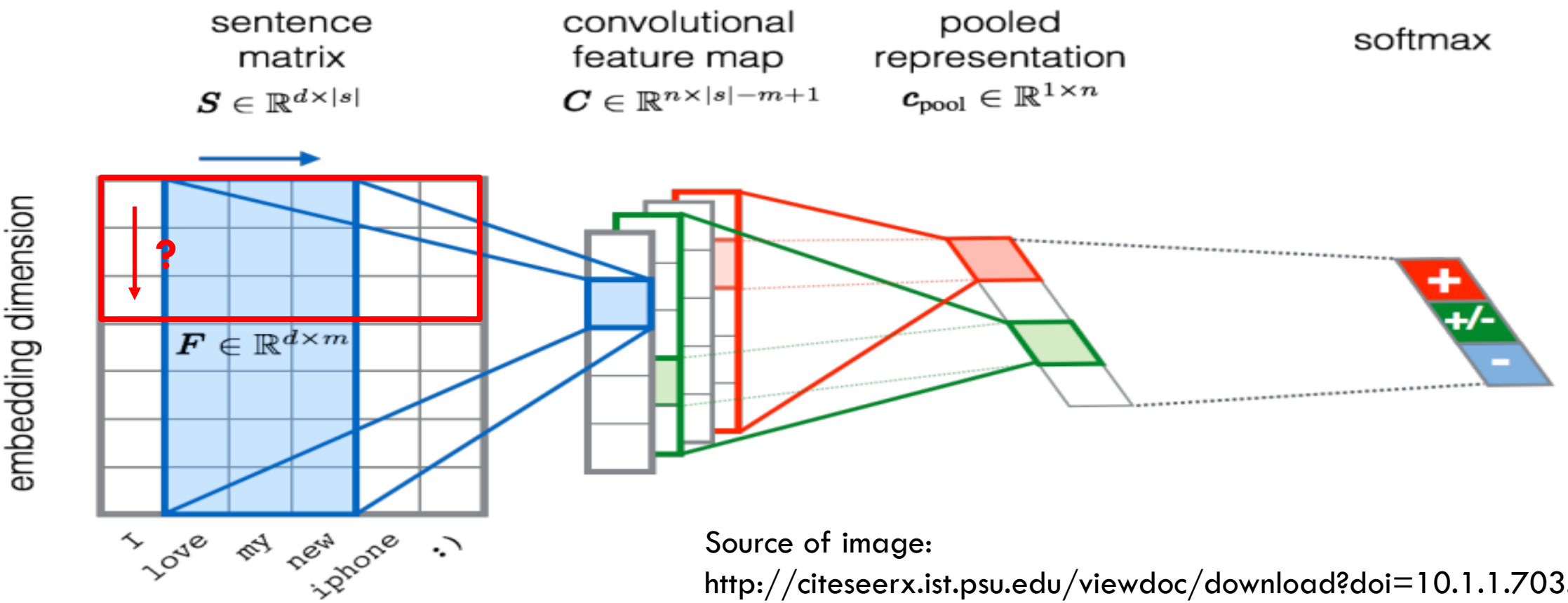
Only modified the ***network structure*** and ***input format (vector -> 3-D array)***



CNN in speech recognition

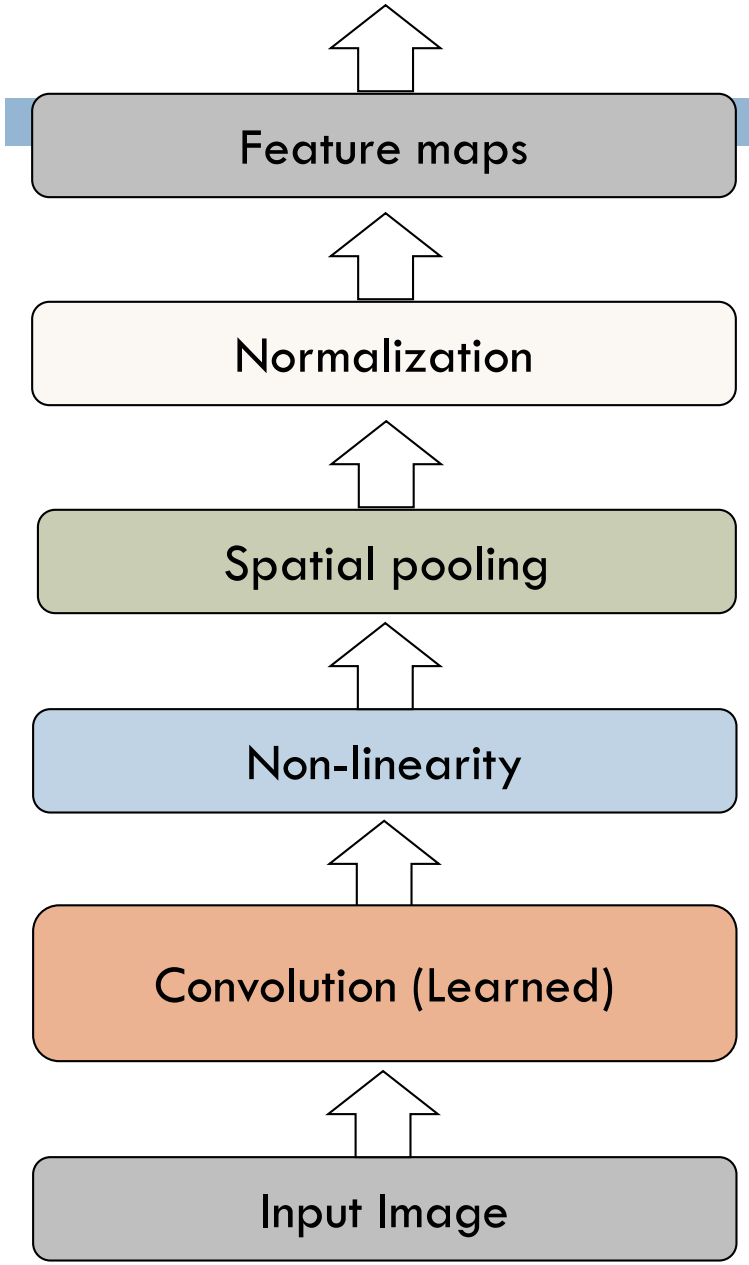


CNN in text classification



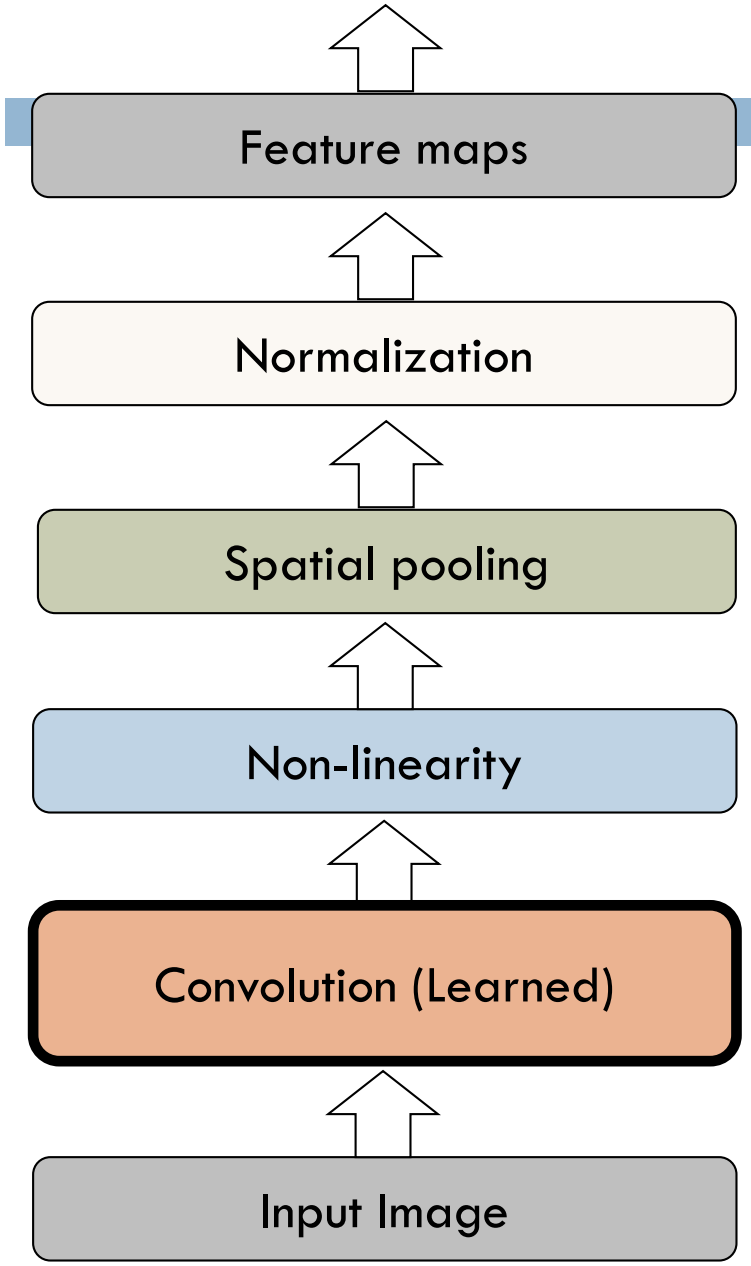
Source of image:
http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.703.6858&rep=rep1&type=pdf

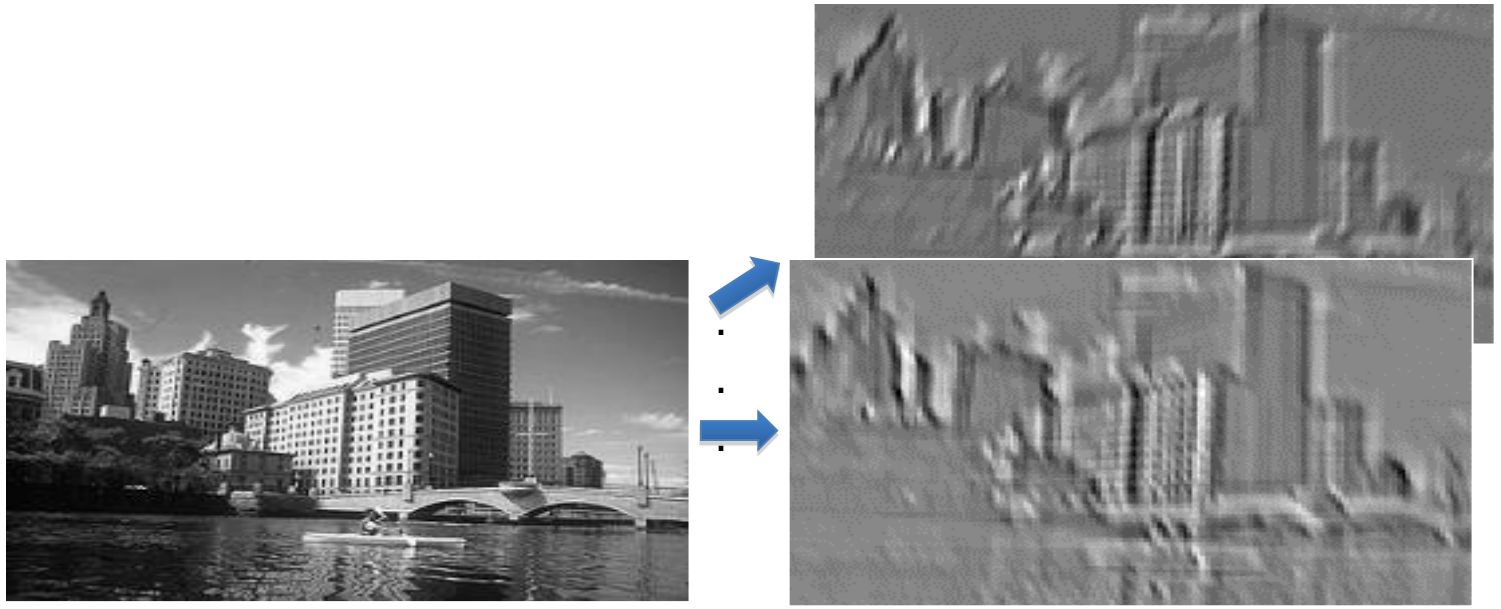
Convolutional Neural Networks



slide credit: S. Lazebnik

Convolutional Neural Networks



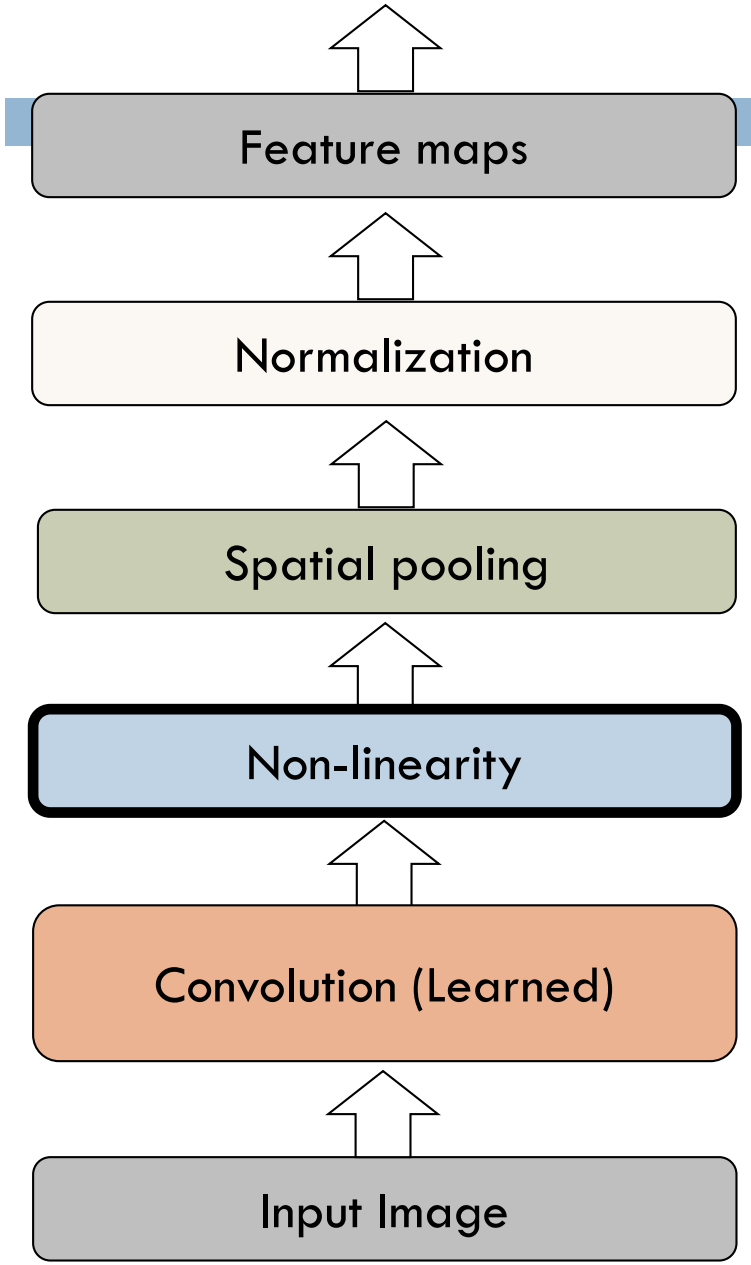


Input

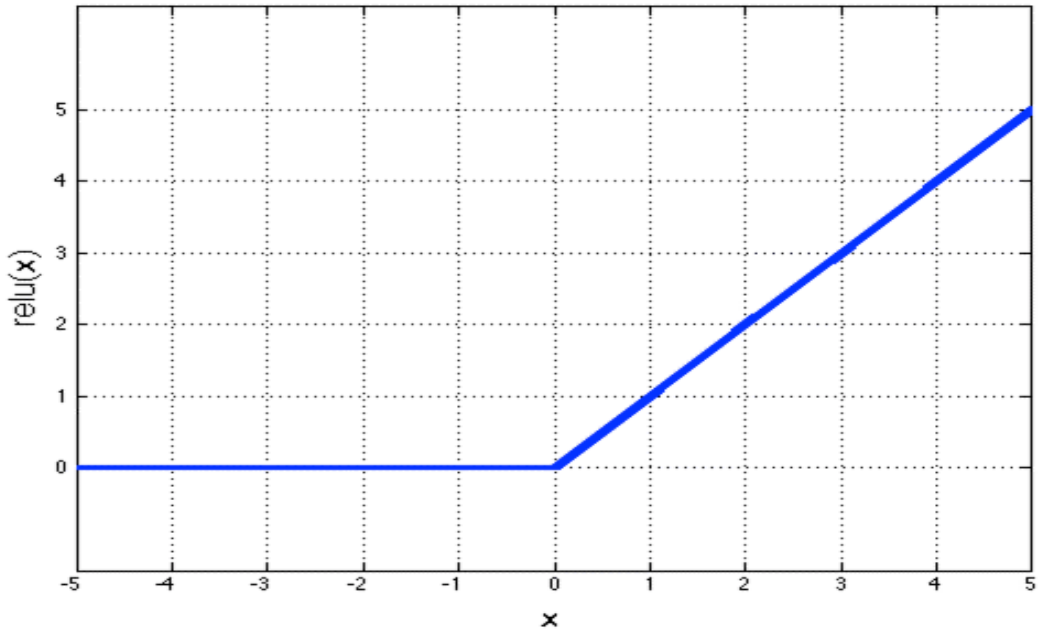
Feature Map

slide credit: S. Lazebnik

Convolutional Neural Networks

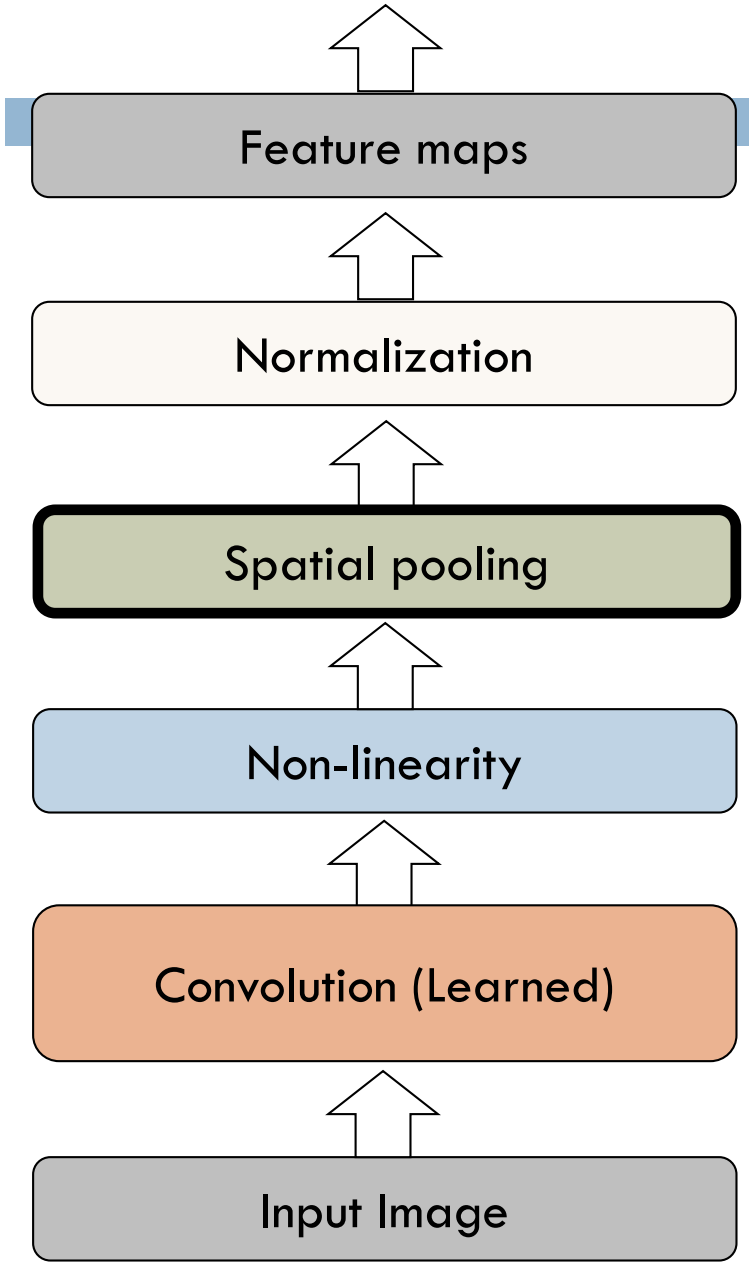


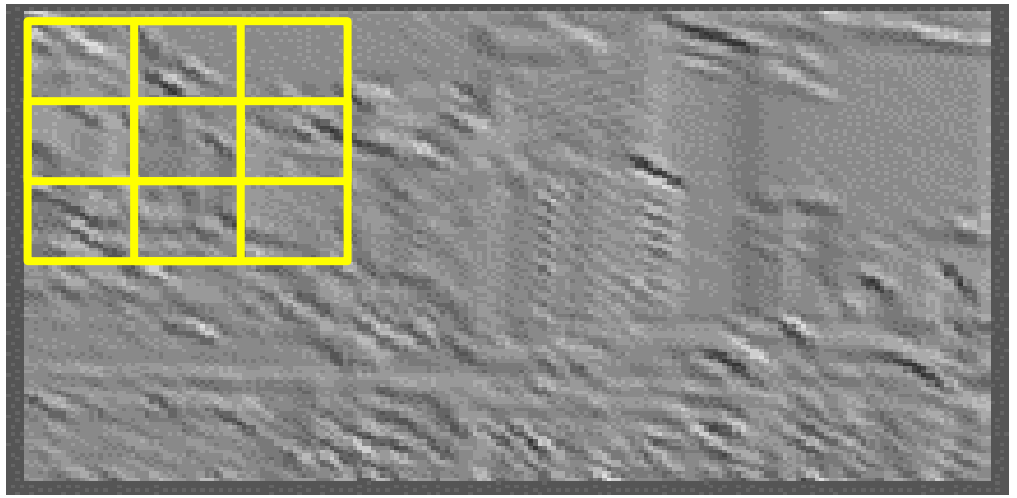
Rectified Linear Unit (ReLU)



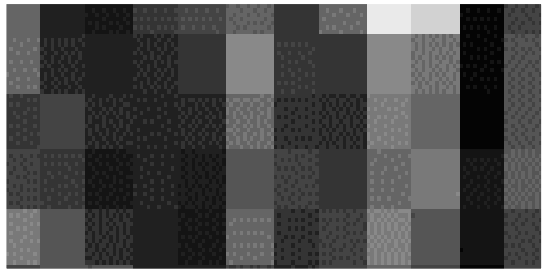
slide credit: S. Lazebnik

Convolutional Neural Networks





Max pooling

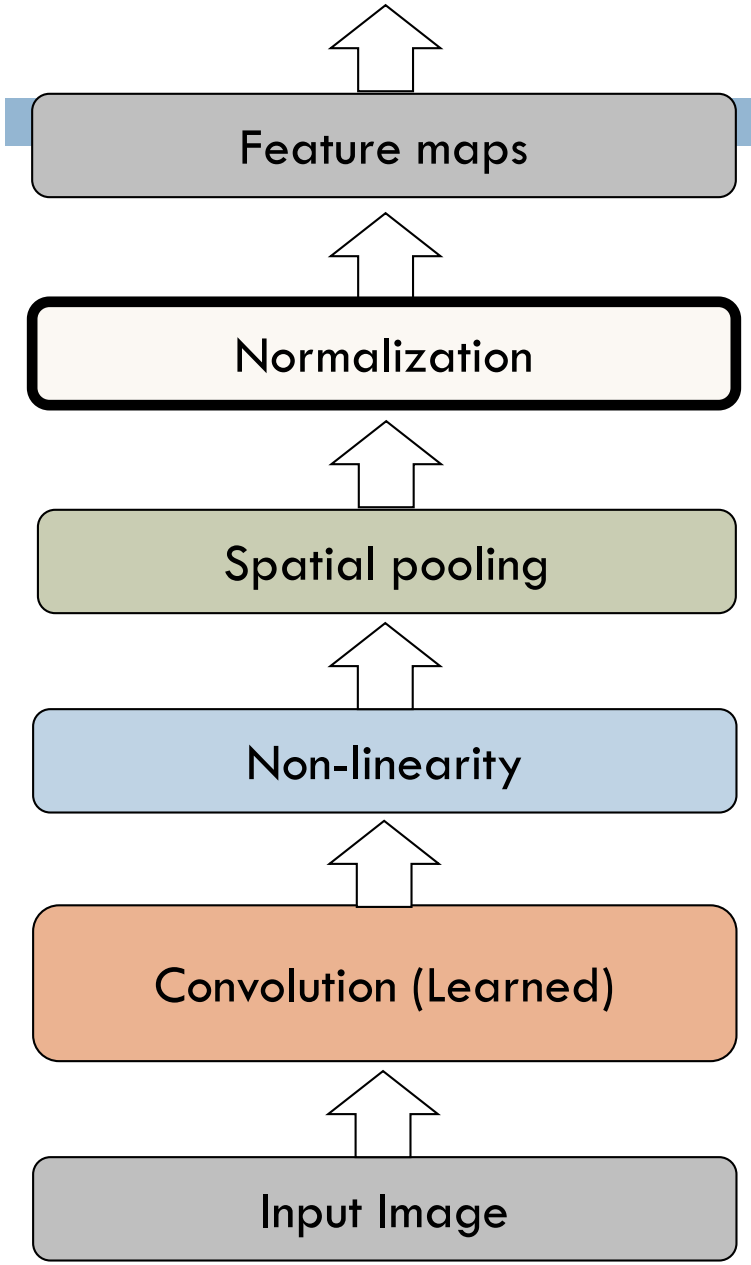


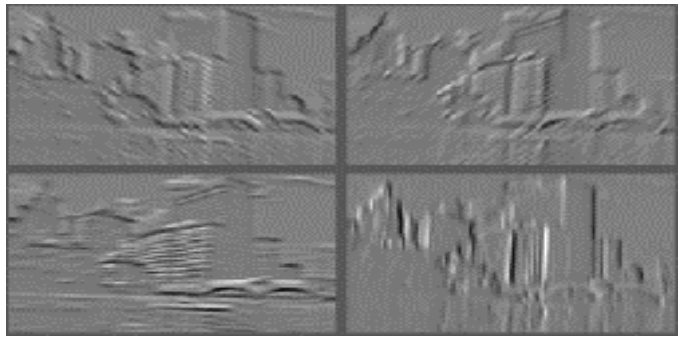
Max-pooling: a non-linear down-sampling

Provide *translation invariance*

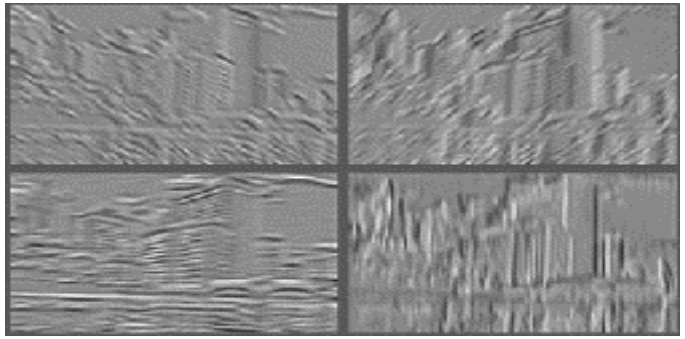
slide credit: S. Lazebnik

Convolutional Neural Networks





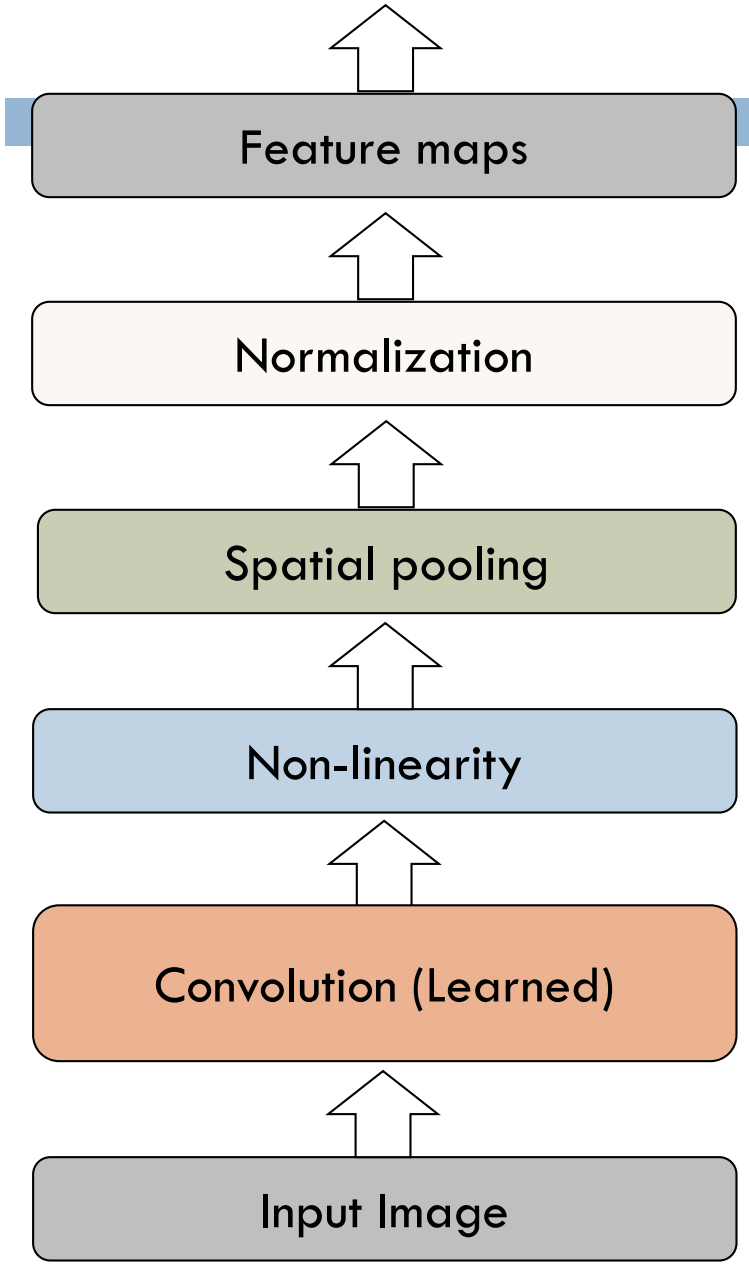
Feature Maps



Feature Maps
After Contrast
Normalization

slide credit: S. Lazebnik

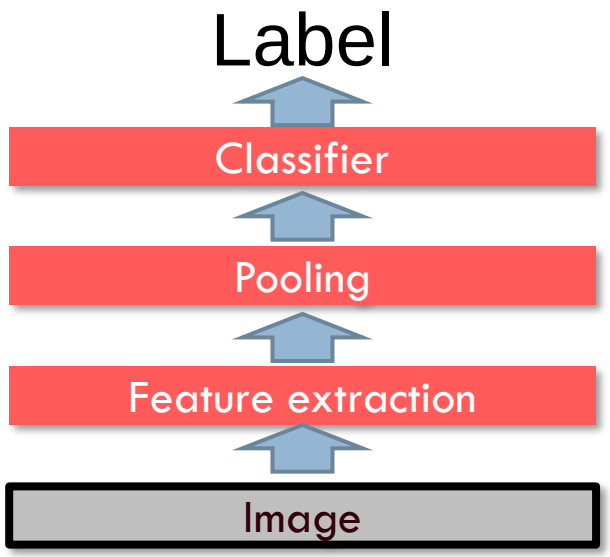
Convolutional Neural Networks

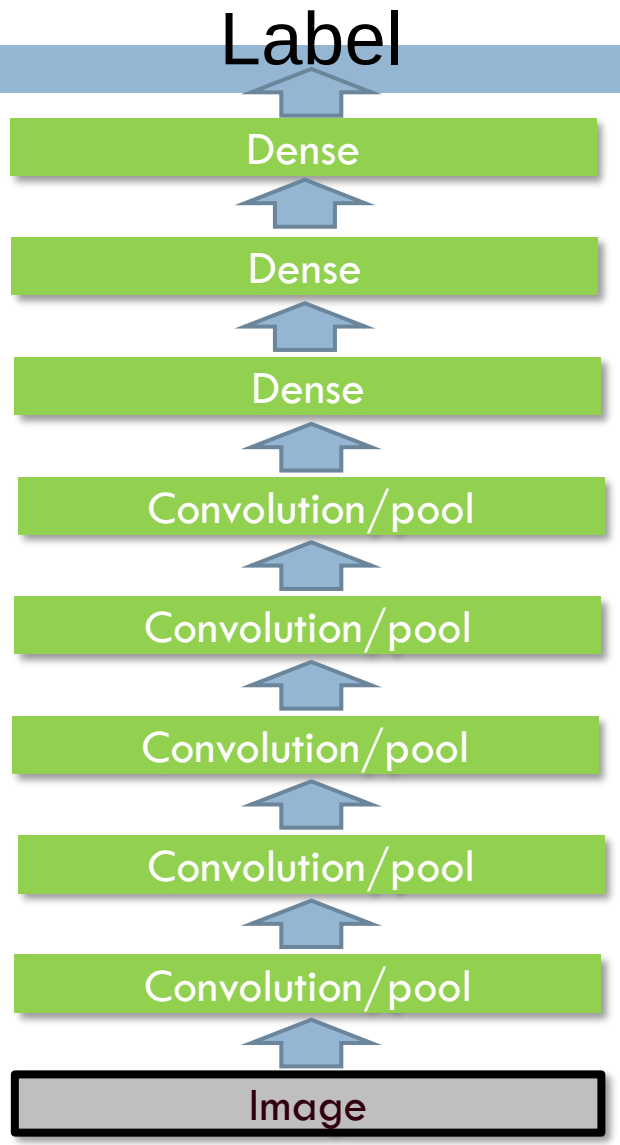


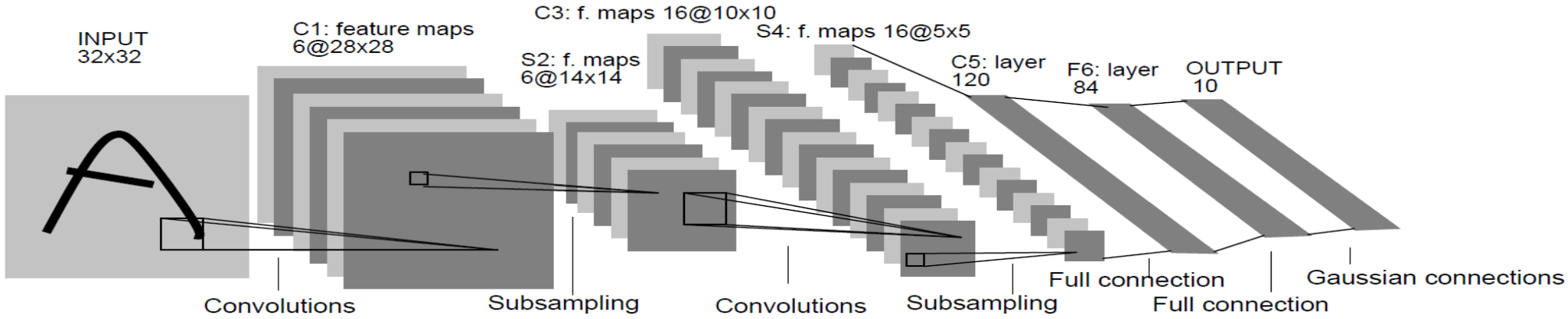
slide credit: S. Lazebnik

Engineered vs. learned features

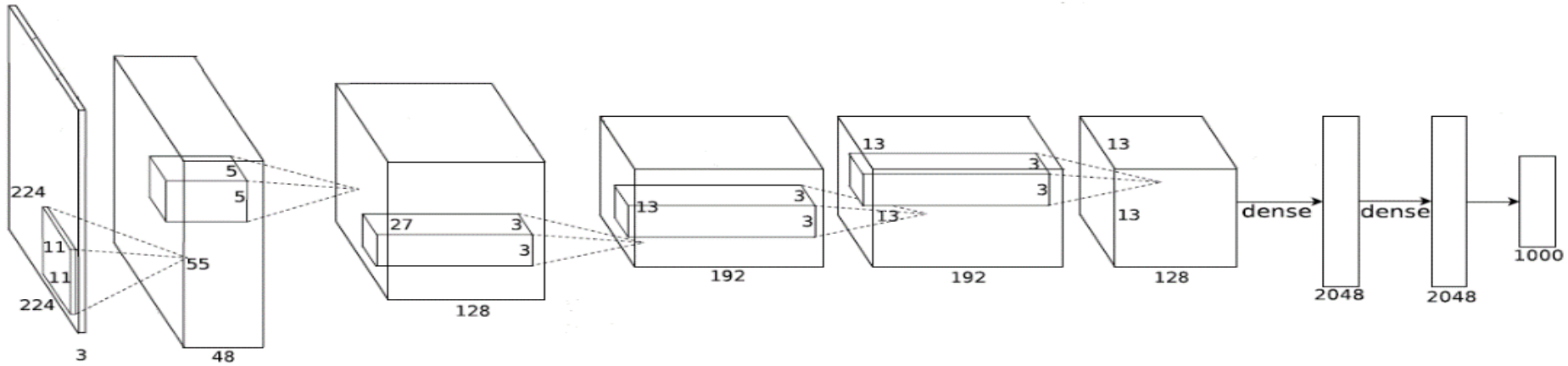
Convolutional filters are trained in a supervised manner by back-propagating classification error





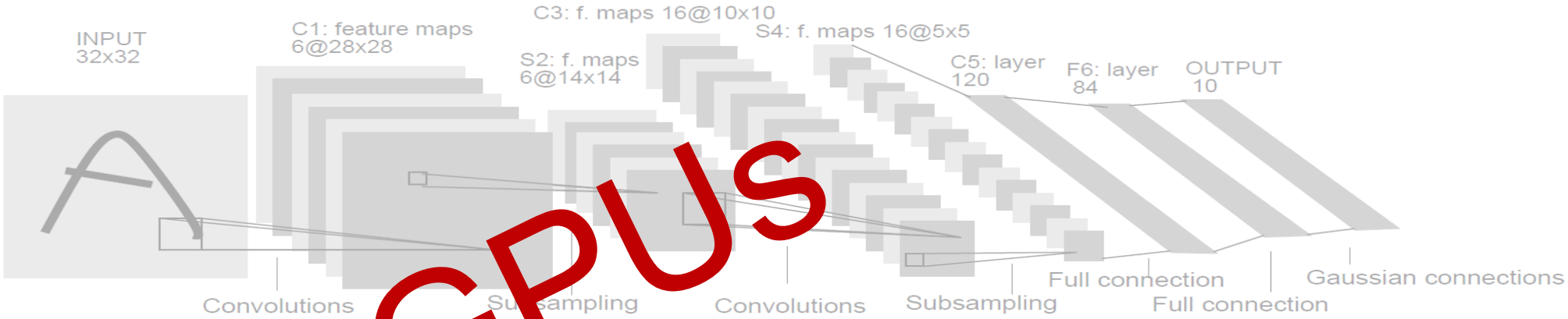


Gradient-Based Learning Applied to Document Recognition, LeCun, Bottou, Bengio and Haffner, Proc. of the IEEE, **1998**



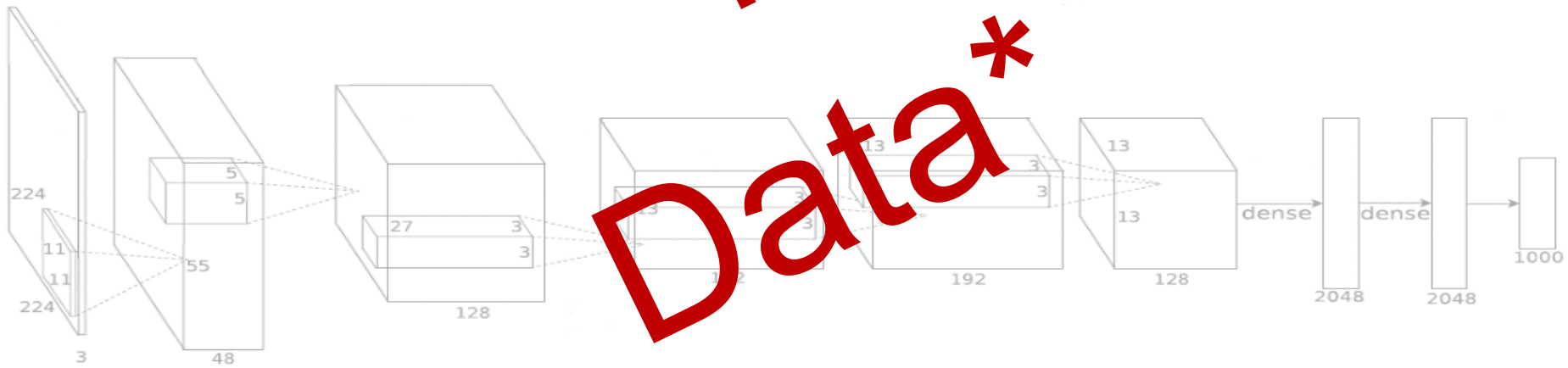
Imagenet Classification with Deep Convolutional Neural Networks, Krizhevsky, Sutskever, and Hinton, NIPS **2012**

Slide Credit: L. Zitnick



Gradient-Based Learning Applied to Document Recognition, LeCun, Bottou, Bengio and Haffner, Proc. of the IEEE, **1998**

GPUs + Data*

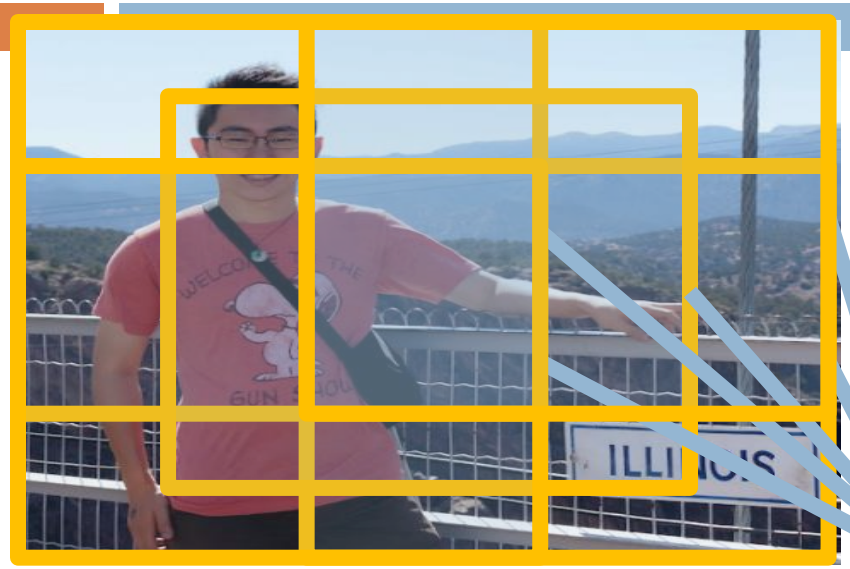


Imagenet Classifica
Sutskever, and Hinto

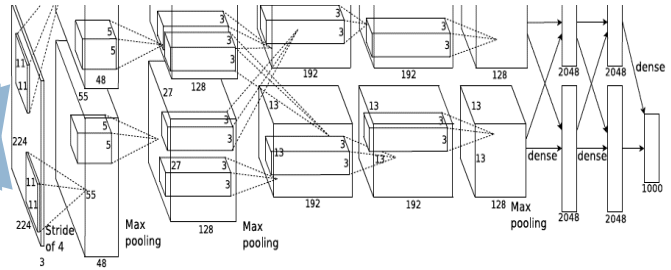
* Rectified activations and dropout

Slide Credit: L. Zitnick

Using CNN for Image Classification

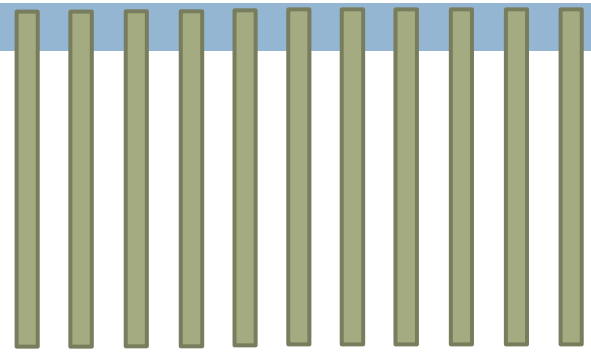


AlexNet



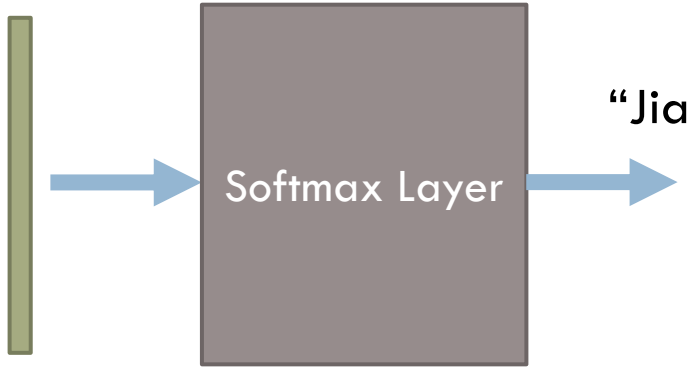
Fixed input size:
224x224x3

Fully connected layer Fc7
d = 4096



Averaging

d = 4096

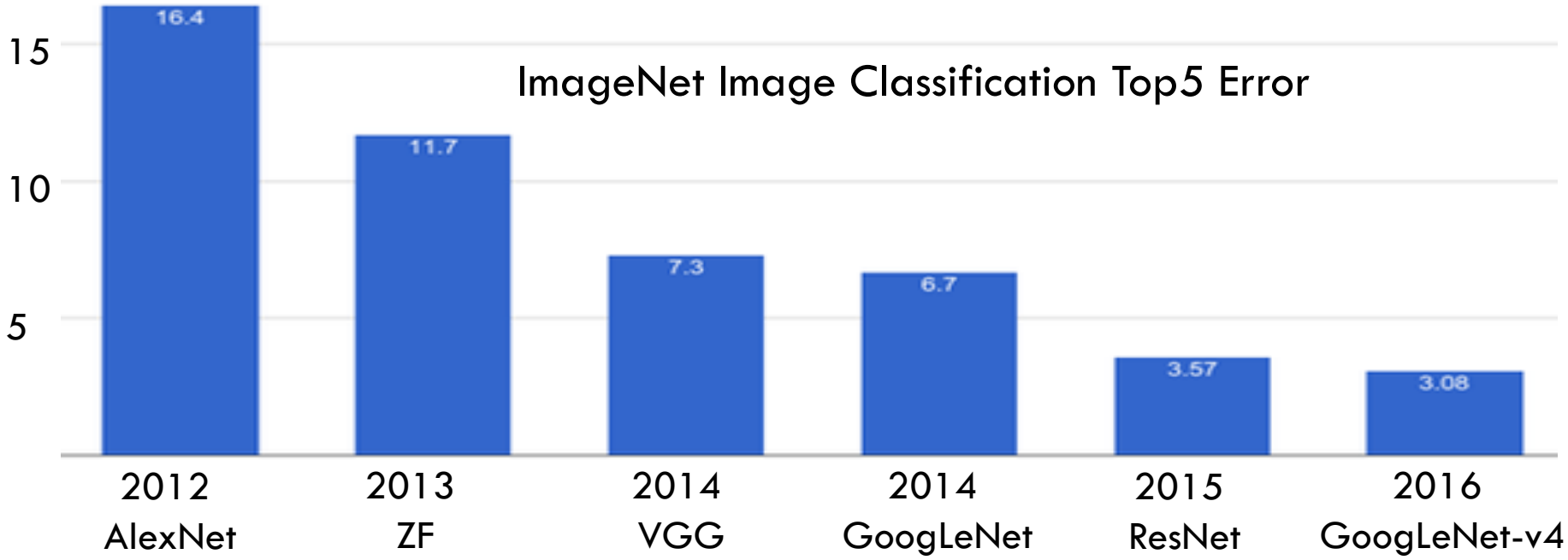


"Jia-Bin"

DEMO-Basic CNN and Classification

- **Classification of images on CIFAR-10 Dataset Code demo**

Progress on ImageNet



VGG-Net

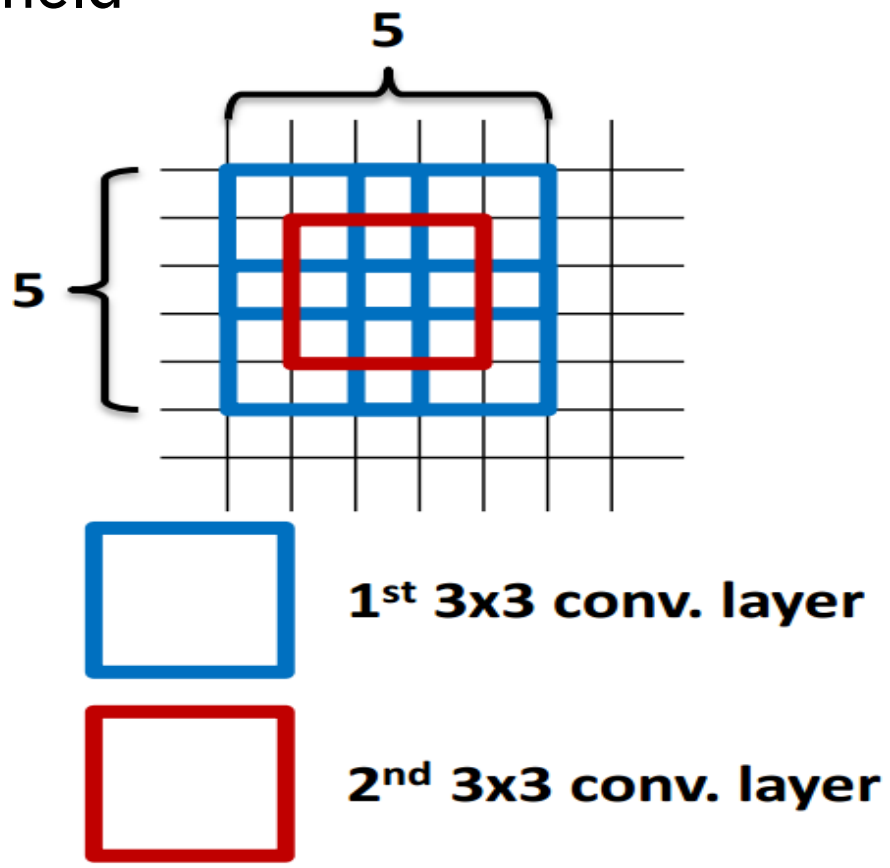
- The deeper, the better

- Key design choices:
  - 3x3 conv. Kernels
    - very small
  - conv. stride 1
    - no loss of information
- Other details:
  - Rectification (ReLU) non-linearity
  - 5 max-pool layers (x2 reduction)
  - no normalization
  - 3 fully-connected (FC) layers

VGG-Net

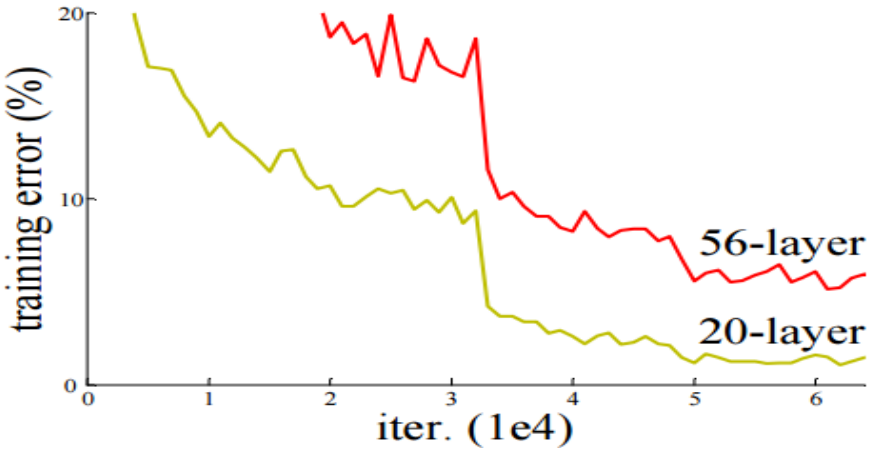
- Why 3x3 layers?
  - Stacked conv. layers have a large receptive field
  - two 3x3 layers – 5x5 receptive field
  - three 3x3 layers – 7x7 receptive field

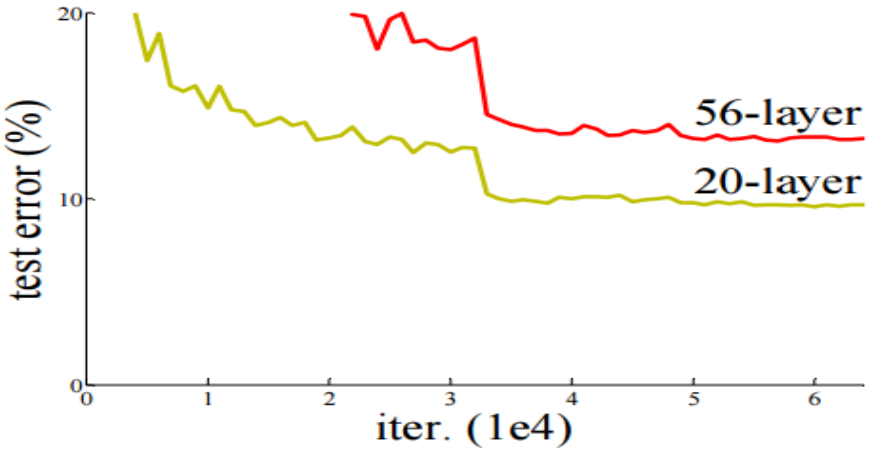
- More non-linearity
  - Less parameters to learn
  - ~140M per net



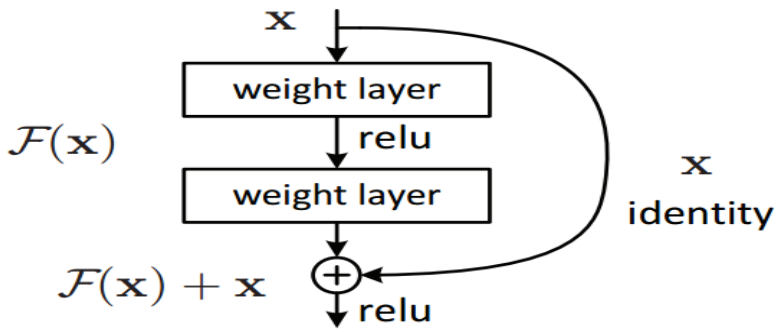
ResNet

- Car





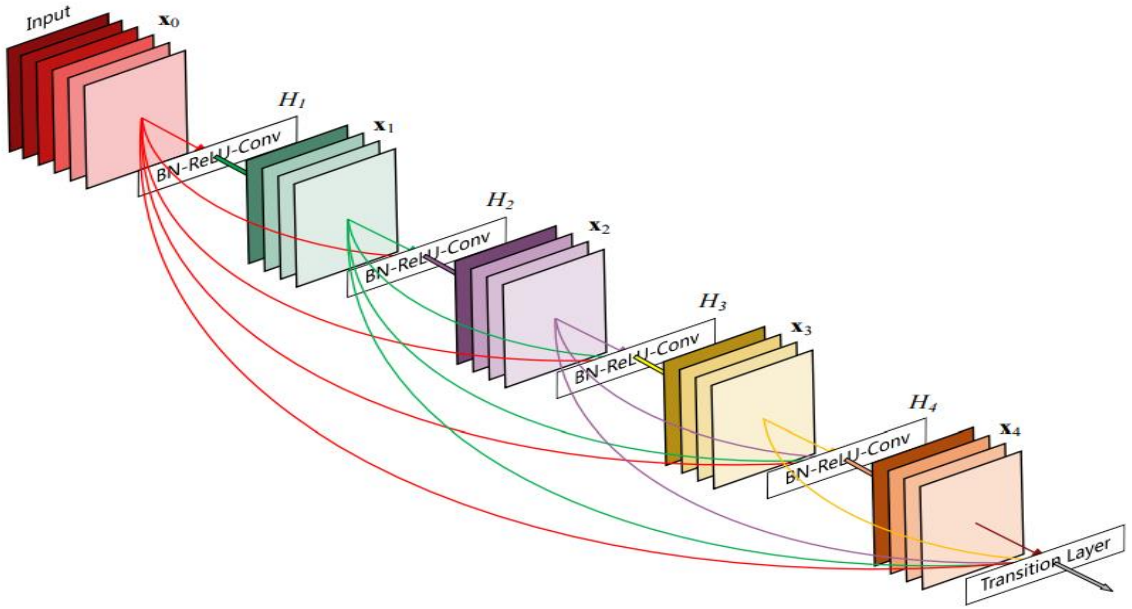
- How can we train very deep network?
  - Residual learning

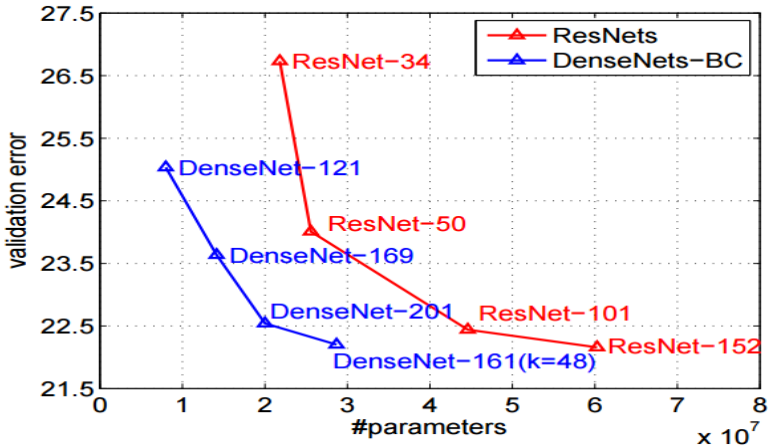


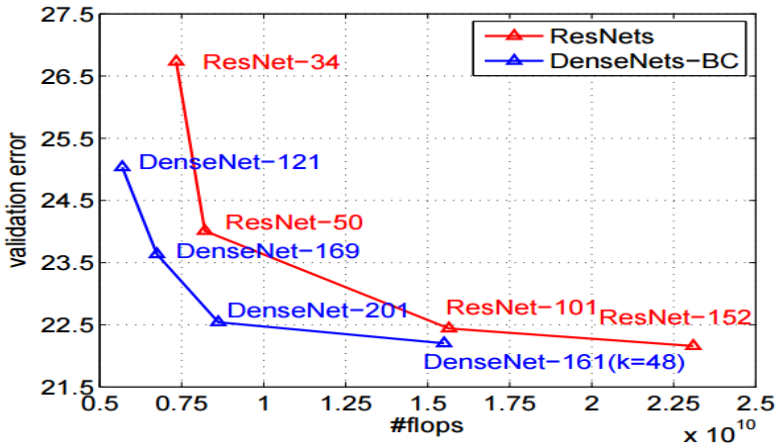
method	top-5 err. (**test**)
VGG [41] (ILSVRC'14)	7.32
GoogLeNet [44] (ILSVRC'14)	6.66
VGG [41] (v5)	6.8
PReLU-net [13]	4.94
BN-inception [16]	4.82
ResNet (ILSVRC'15)	**3.57**

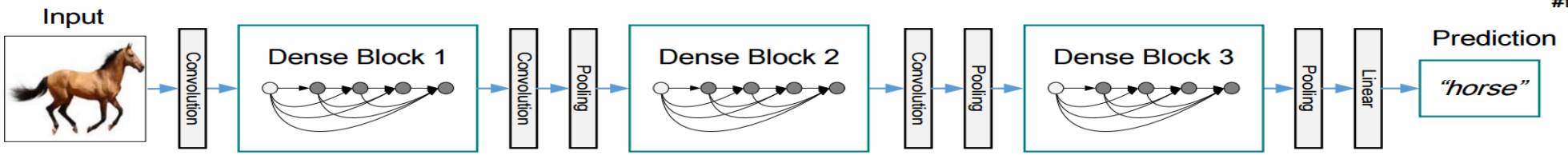
DenseNet

- Shorter connections (like ResNet) help
- Why not just connect them all?









Training Convolutional Neural Networks

- Backpropagation + stochastic gradient descent with momentum
  - Neural Networks: Tricks of the Trade
- Dropout
- Data augmentation
- Batch normalization
- Initialization
  - Transfer learning

Deep learning library

- TensorFlow
  - Research + Production
- PyTorch
  - Research
- Caffe2
  - Production

Resources

- http://deeplearning.net/
  - Hub to many other deep learning resources
- https://github.com/ChristosChristofidis/awesome-deep-learning
  - A resource collection deep learning
- https://github.com/kjw0612/awesome-deep-vision
  - A resource collection deep learning for computer vision
- http://cs231n.stanford.edu/syllabus.html
  - Nice course on CNN for visual recognition


xtranormal