



# Introduction to Computer Vision



Top row, left to right:  
[Image by Roger H Goun](#) is licensed under [CC BY 2.0](#)  
[Image is CC0 1.0](#) public domain  
[Image is CC0 1.0](#) public domain  
[Image is CC0 1.0](#) public domain

Middle row, left to right:  
[Image by BGPHP Conference](#) is licensed under [CC BY 2.0](#); changes made  
[Image is CC0 1.0](#) public domain  
[Image by NASA](#) is licensed under [CC BY 2.0](#)  
[Image is CC0 1.0](#) public domain

Bottom row, left to right:  
[Image is CC0 1.0](#) public domain  
[Image by Derek Keats](#) is licensed under [CC BY 2.0](#); changes made  
[Image is public domain](#)  
[Image is licensed under CC-BY 2.0](#); changes made



# What is Computer Vision?

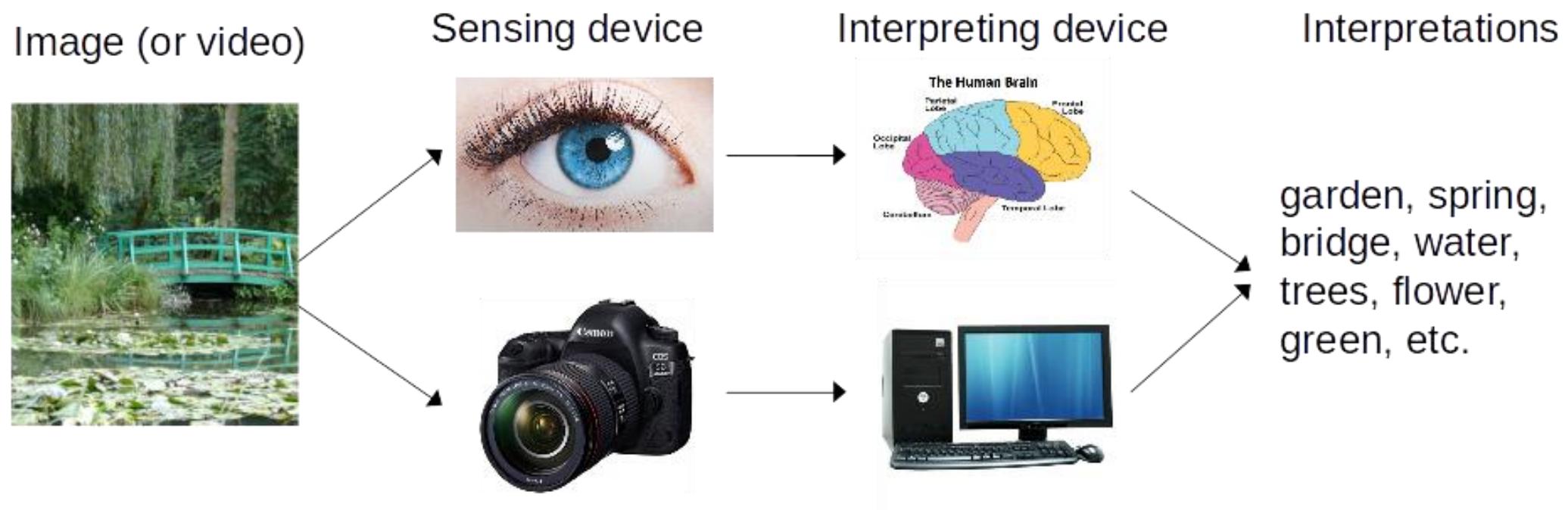
“What does it mean, to see? The plain man's answer (and Aristotle's, too). would be, to know what is where by looking.”

– David Marr, Vision (1982)

“What does it mean, to see? The plain man's answer (and Aristotle's, too). would be, to know **what** is **where** by looking.”

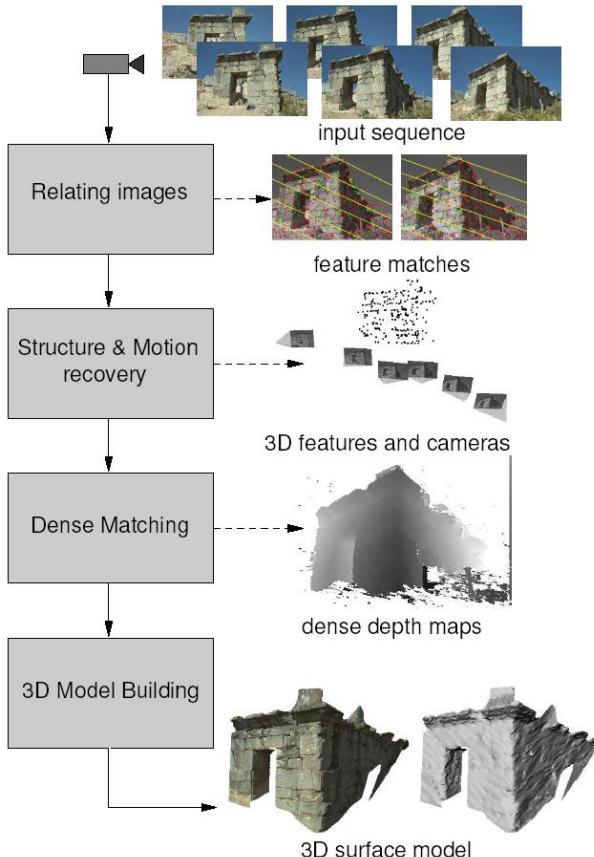
– David Marr, Vision (1982)

# What is (computer) vision?

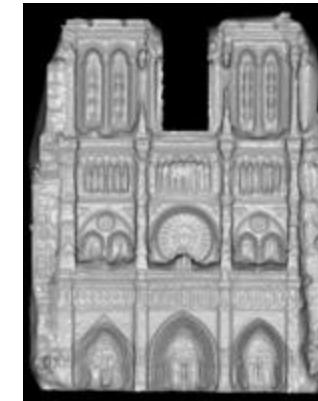


Measurement  Perception

# Measurement

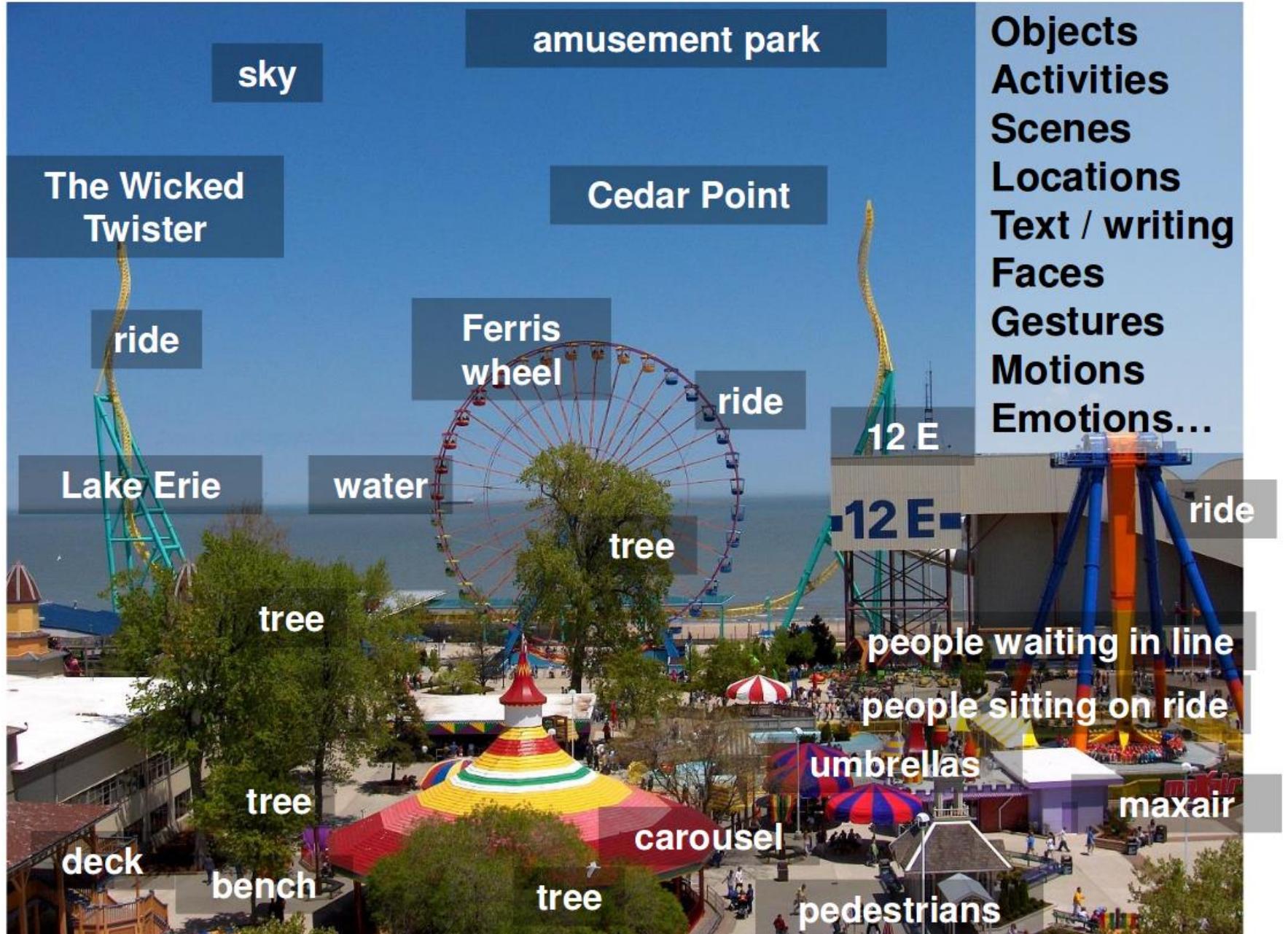


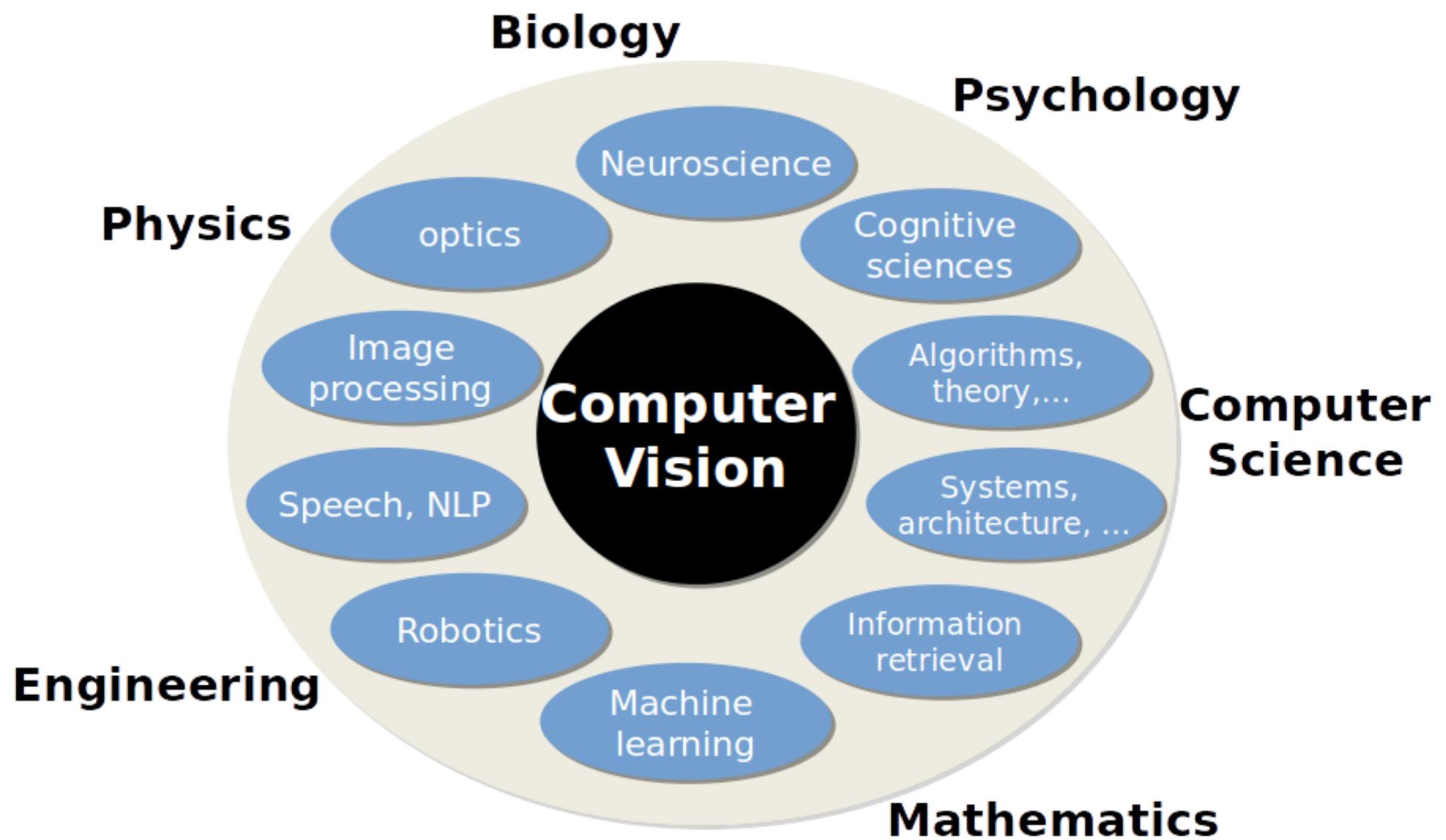
Pollefeys et al.



Goesele et al.

# Perception







# Computer Vision Tasks

- Image Classification
- Image Segmentation
- Edge detection
- Image denoising
- Optical flow estimation
- Tracking
- Reconstruction
- Semantic Segmentation
- Object Localization
- Object Detection
- Super-resolution
- Image Captioning
- Visual Question Answering
- Activity Recognition

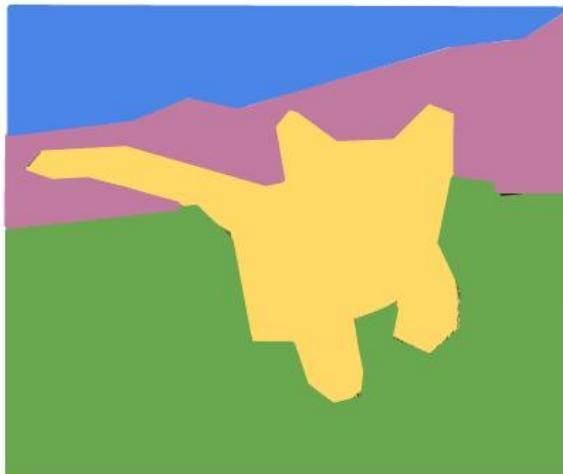
## Classification



CAT

No spatial extent

## Semantic Segmentation



GRASS, CAT,  
TREE, SKY

No objects, just pixels

## Object Detection



DOG, DOG, CAT

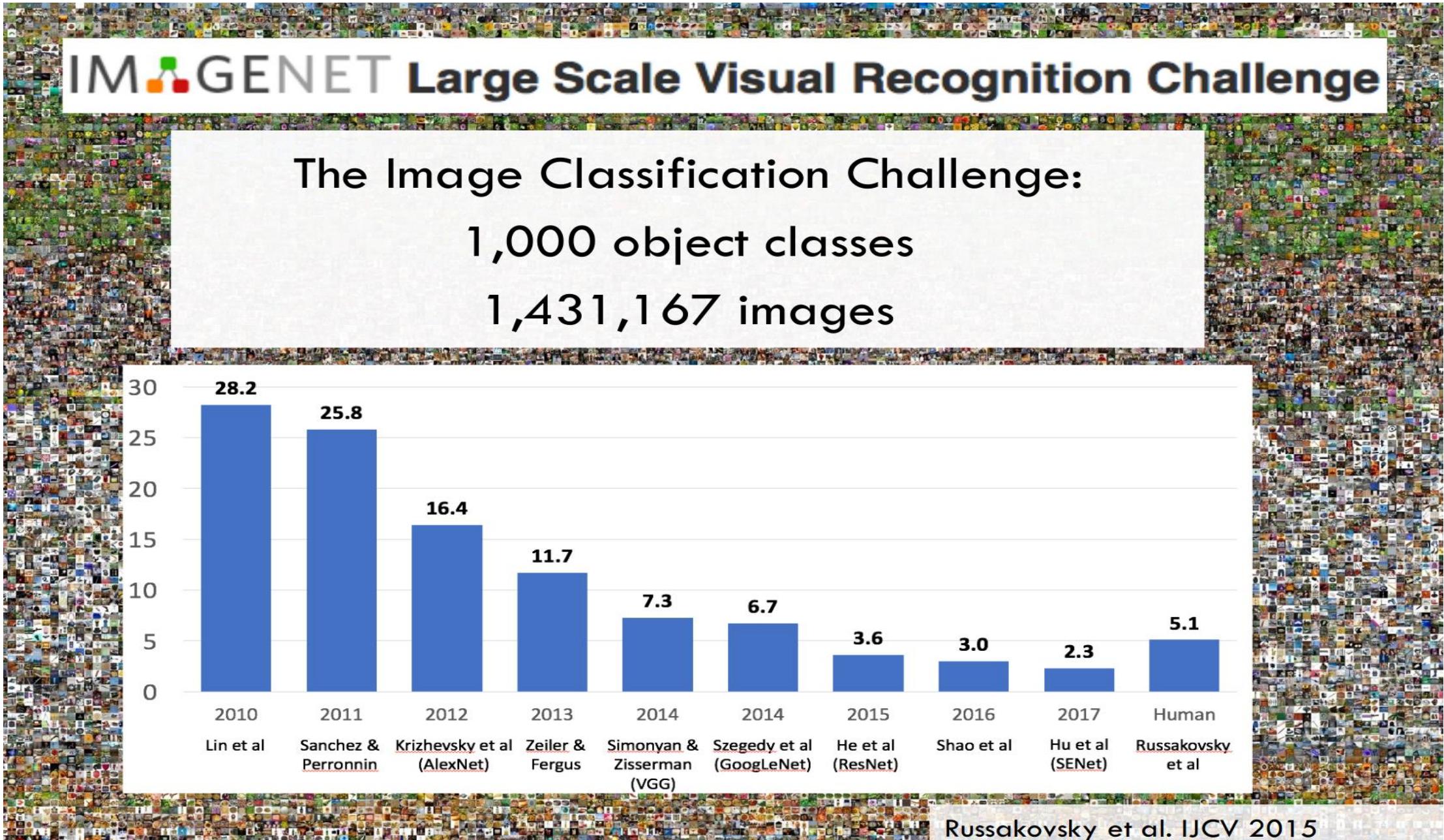
Multiple Object

## Instance Segmentation



DOG, DOG, CAT

[This image is CC0 public domain](#)



Who is wearing glasses?

man



woman



Is the umbrella upside down?

yes



no



Where is the child sitting?

fridge



arms



How many children are in the bed?

2



1



# Image Captioning



"man in black shirt is playing guitar."



"construction worker in orange safety vest is working on road."



"two young girls are playing with lego toy."



**Sadegh's:**

1. a hot dog bun filled with mustard and ketchup.
2. a hot dog bun filled with fries and pickles.
3. a hotdog with mustard, ketchup and onion on a bun.
4. a plate of food with hot dog bun filled with cheese and vegetables.
5. a hotdog with mustard, ketchup, mustard, and onions on a bun.
6. a white plate topped with a sandwich and a salad.
7. a plate of food with a hot dog, tomatoes and sausage.
8. a plate with a sandwich and french fries on a table.
9. a plate with a sandwich and french fries.

**Caption from mode:** a hot dog bun filled with fries and pickles.

**Microsoft:** a hot dog on a bun.



**Sadegh's:**

1. a person holding a rainbow colored umbrella over their head .
2. a woman holding an umbrella in the rain .
3. a woman holding a colorful umbrella over her head .
4. a woman holding an umbrella in the rain .
5. a woman with a red umbrella is standing on the beach .
6. a woman in a red shirt is holding a blue umbrella .
7. a person holding an umbrella while walking down a street .
8. a woman holding an umbrella while walking on a sidewalk .
9. a little girl with a purple umbrella standing in the rain .

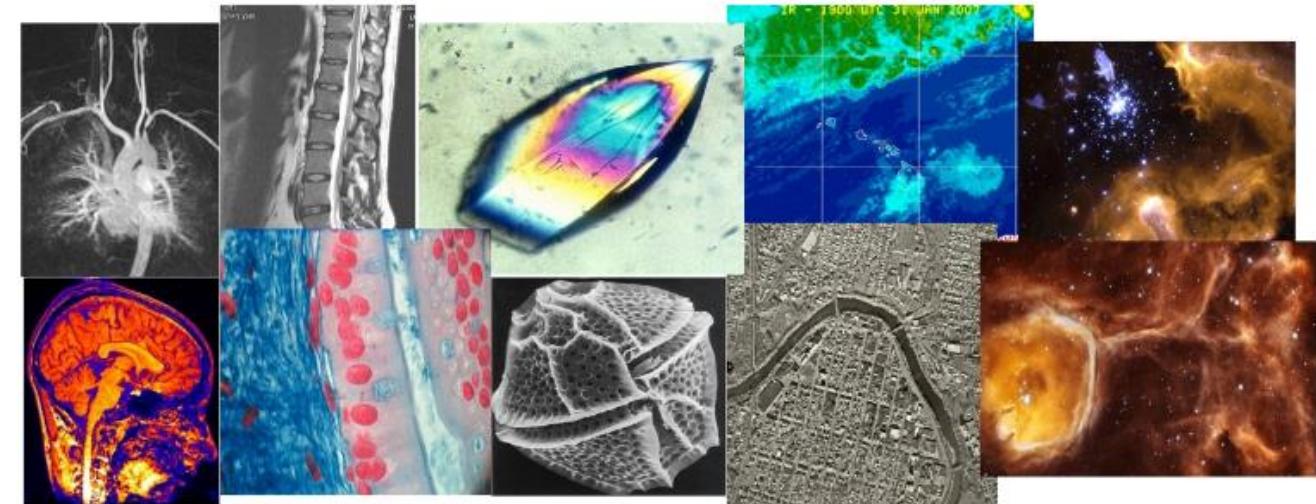
**Caption from mode:** a young woman holding an umbrella standing on a sidewalk .

**Microsoft:** a person holding a blue umbrella



# Why study computer vision?

# Images and video are everywhere!



Surveillance and security

Medical and scientific images

# Computer Vision is Useful and Interesting

- Image Search
- Face detection
- Autonomous driving
- Robotics vision
- 3D city modelling
- Security
- Sports
- Image generation
- Style transfer
- Image-to-image translation
- Image Colorization
- Image Completion
- DeepFakes
- Medical image processing

## Image Search



JPG X

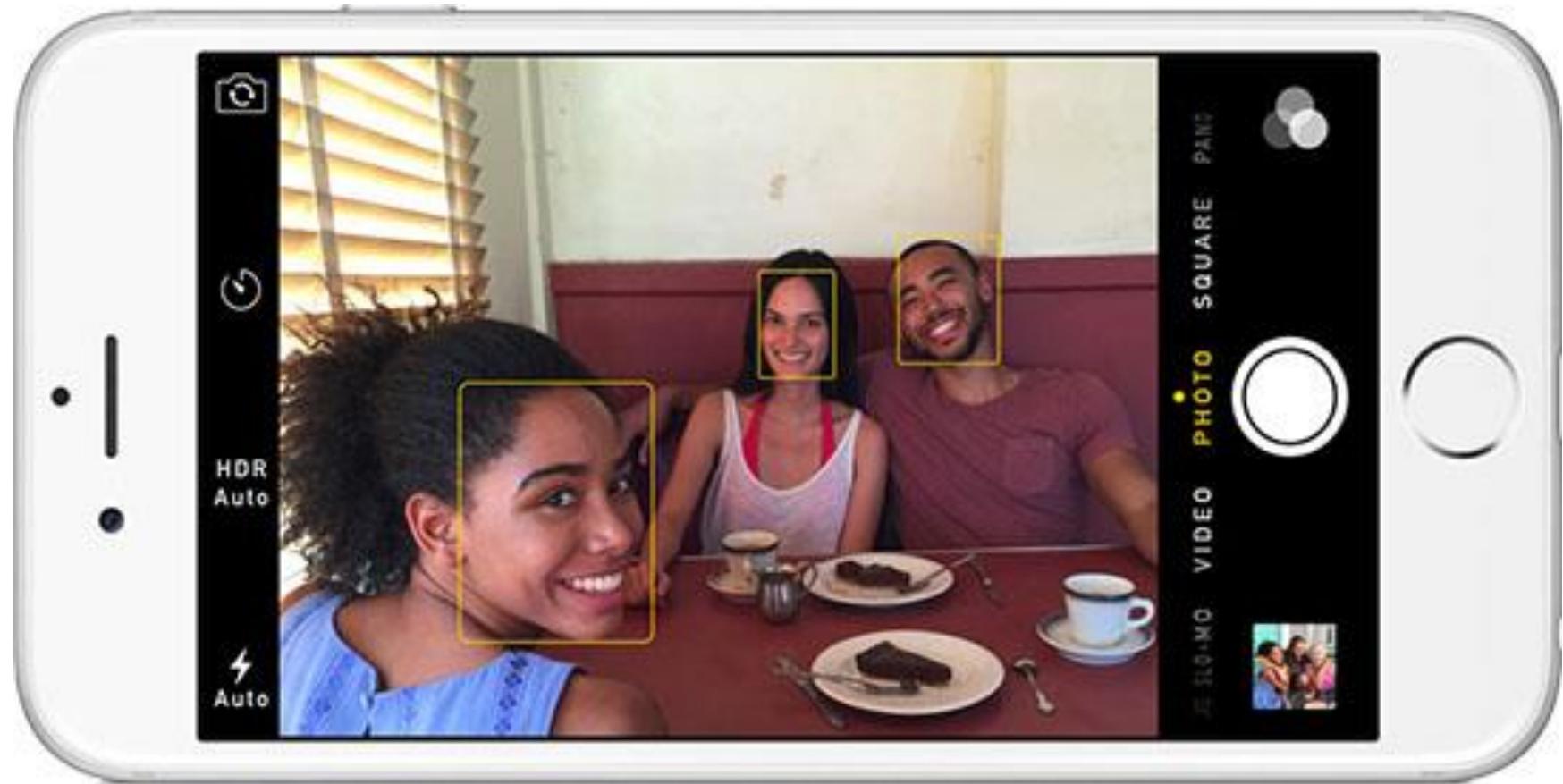
All  Images  Maps  Shopping  More  Settings  Tools

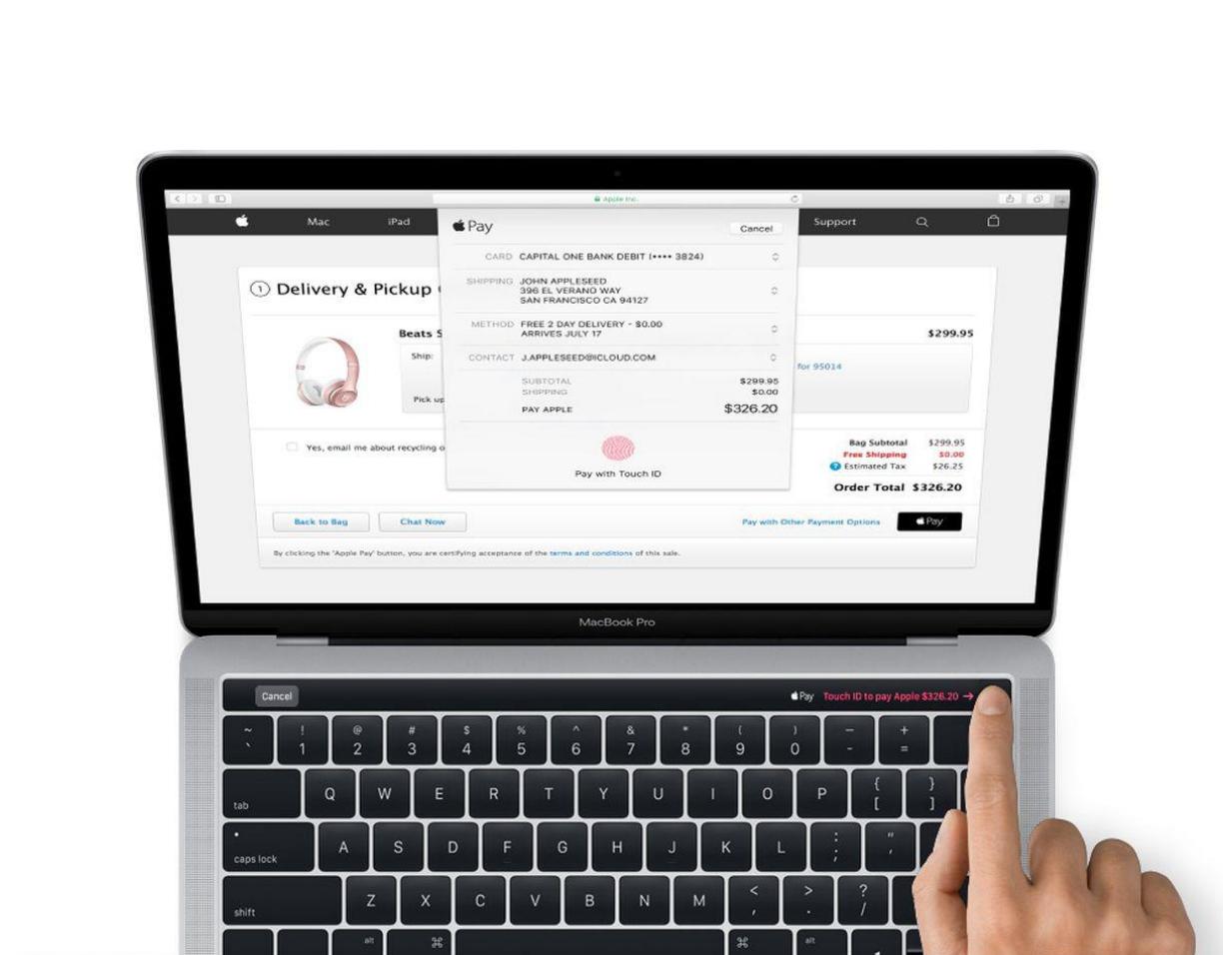
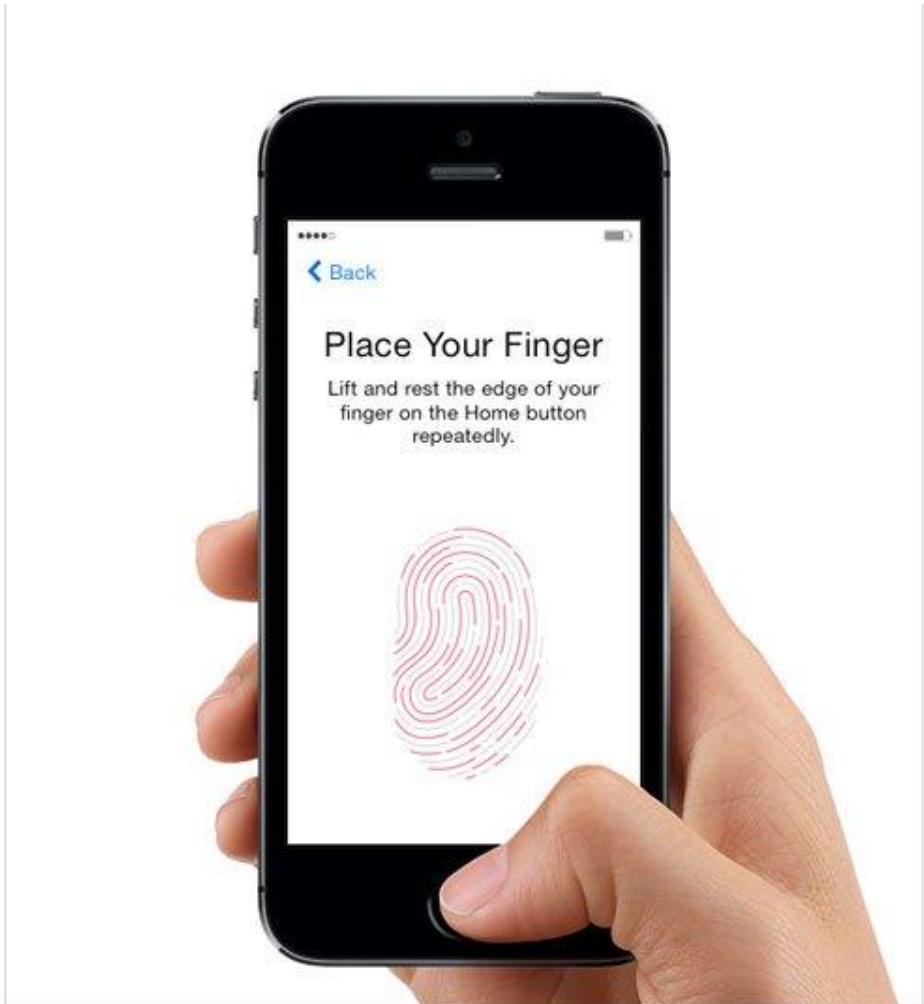
About 25,270,000,000 results (1.30 seconds)

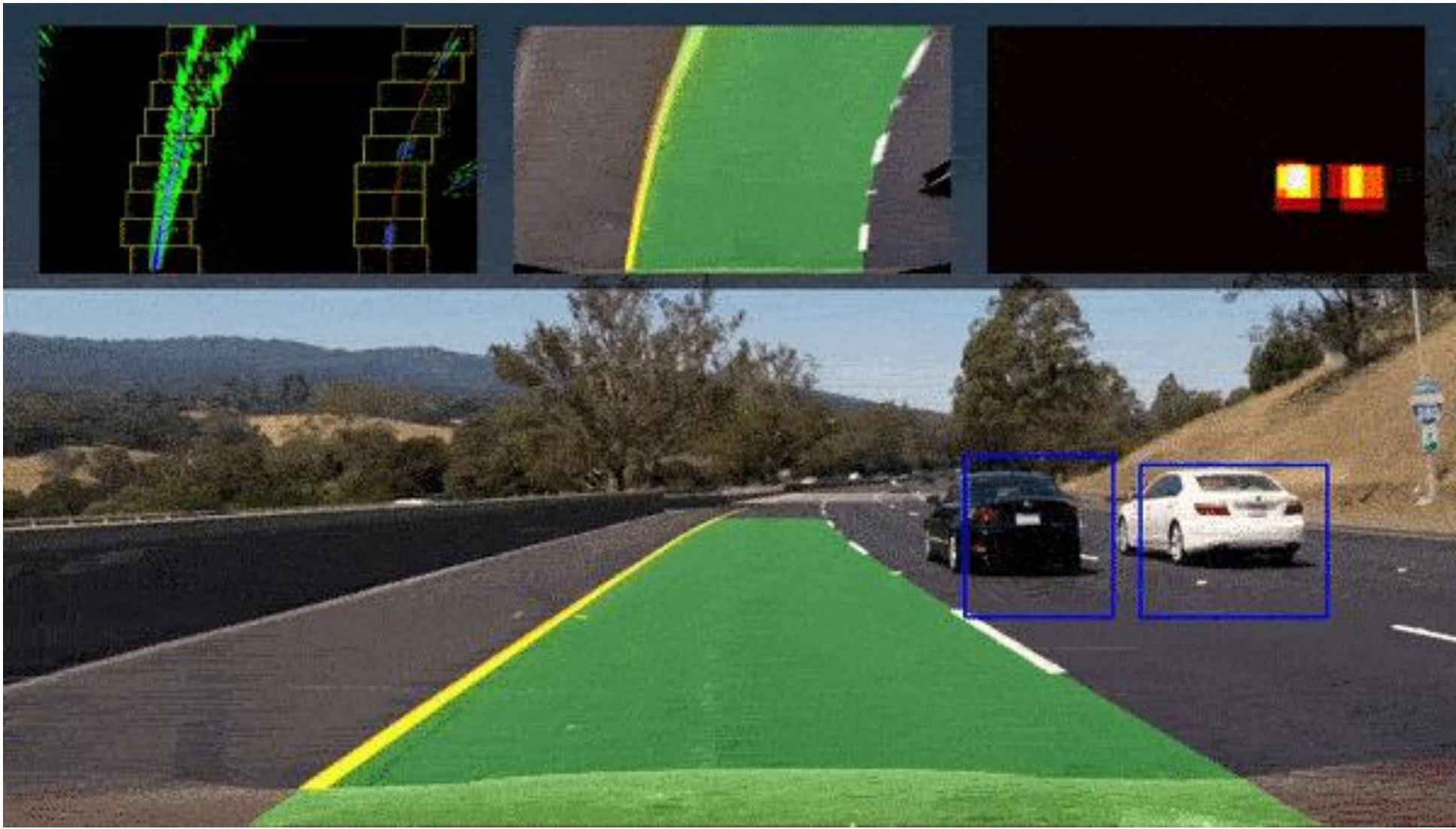
 Image size:  
375 × 300

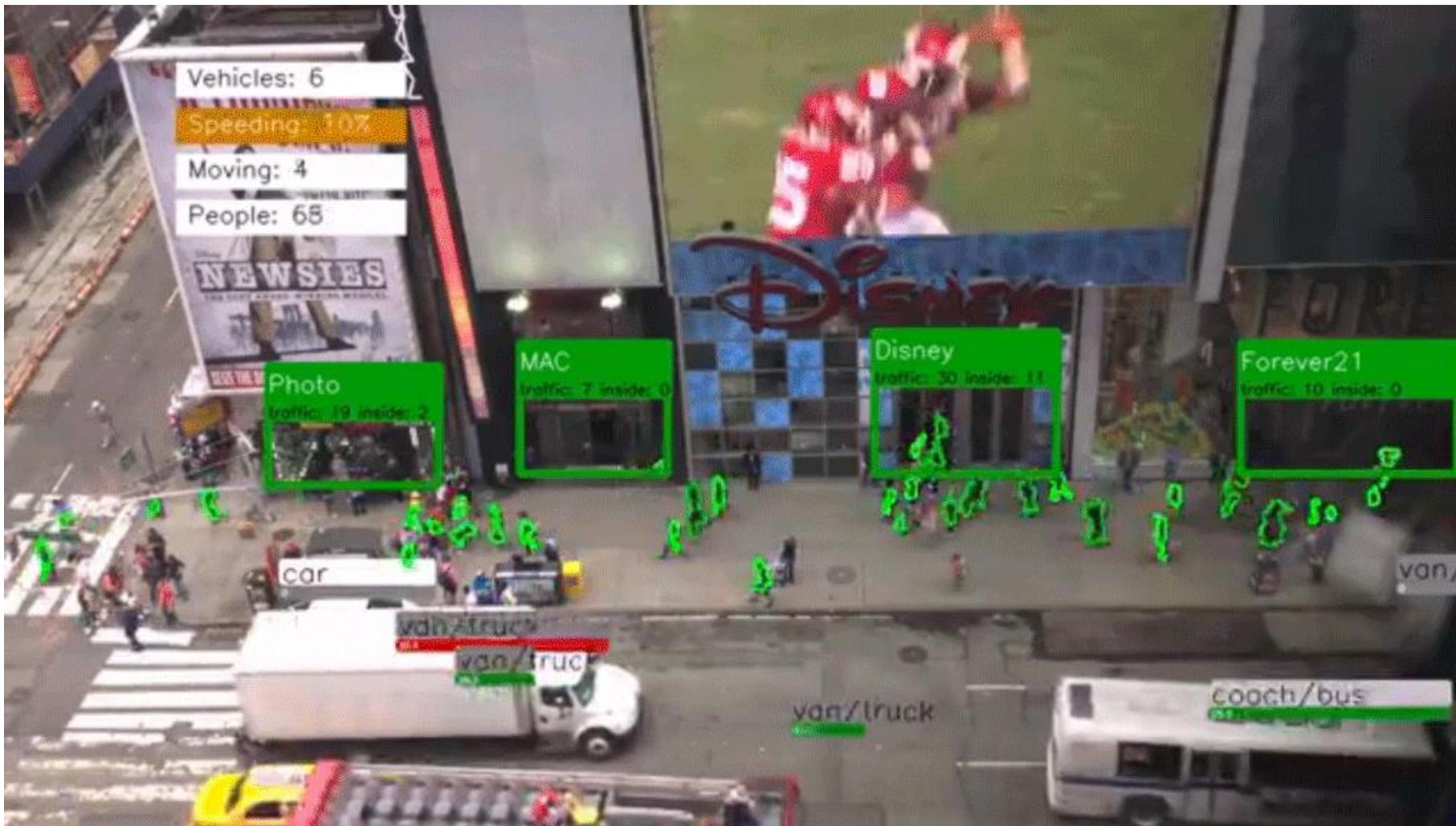
Find other sizes of this image:  
[All sizes](#) - [Small](#)

Possible related search: **boston terrier dog breeds**









# 3D Maps



Google Maps

# Sports

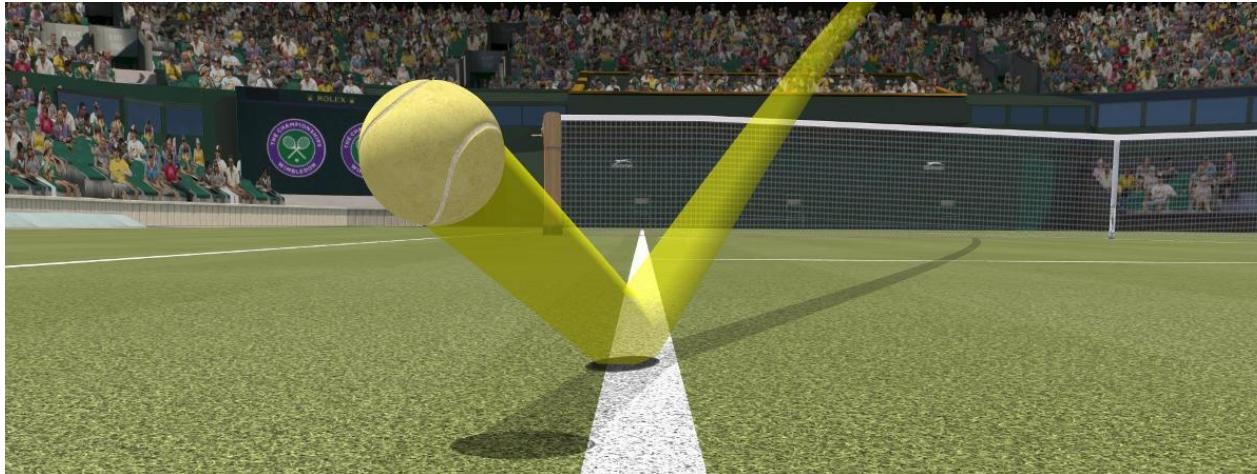
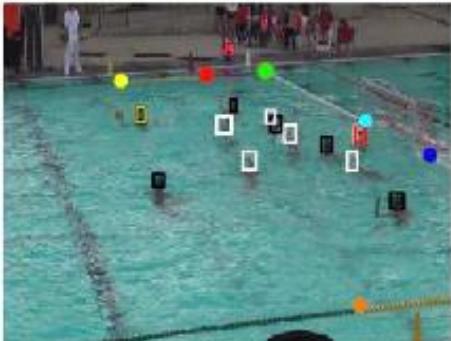
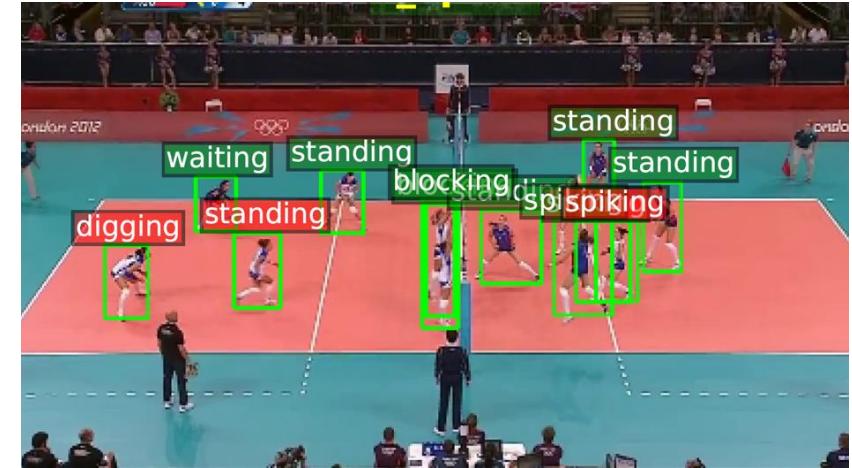
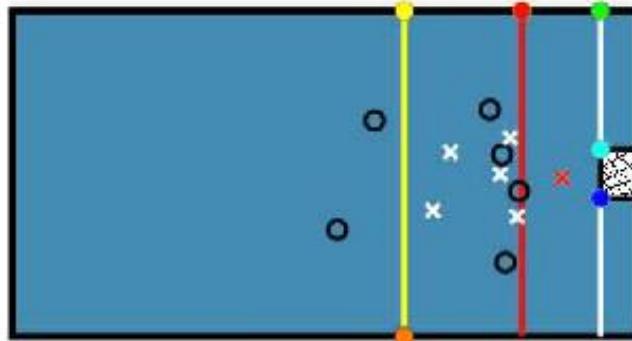


image view



overhead view

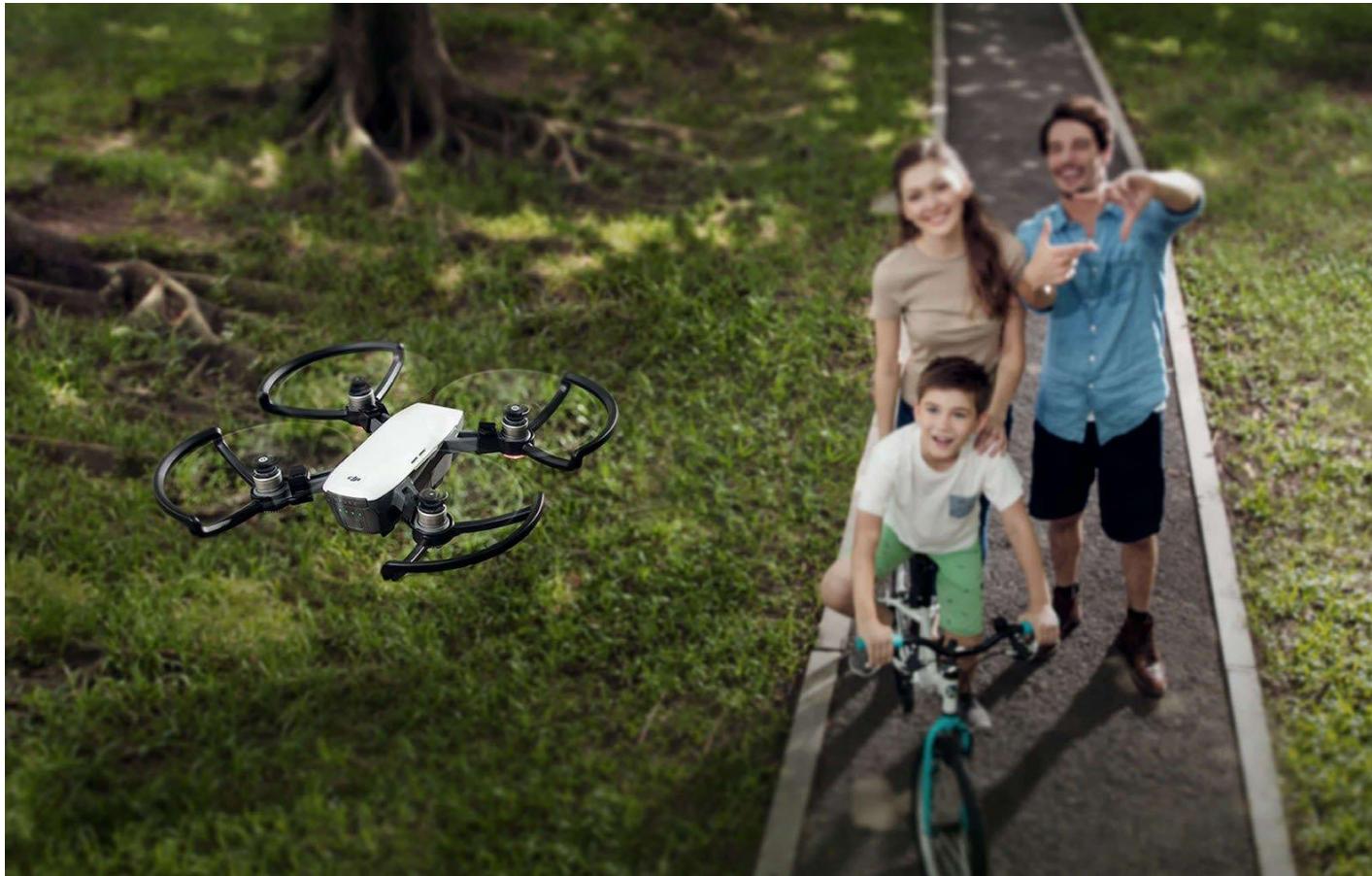


1. Hawk-Eye
2. Goelline technology
3. What will Happen Next? Forecasting Player Moves in Sports Videos
4. <https://indatalabs.com/blog/how-does-computer-vision-work>



# Gaming

**KINECT  
SPORTS**



# Gesture Recognition

# Special Effects

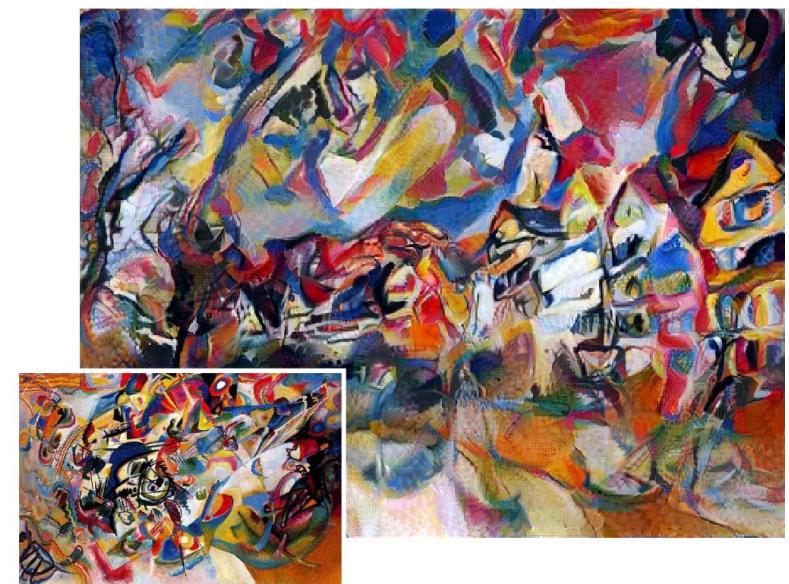


## Real vs Fake





# Image Generation



# Deepfake





Norwegian Bride (est late 1890s)









Outside border images, clockwise, starting from top left:

[Image by Pop Culture Geek](#) is licensed under CC BY 2.0; changes made

[Image by the US Government](#) is in the public domain

[Image by the US Government](#) is in the public domain

[Image by Glogger](#) is licensed under CC BY-SA 3.0; changes made

[Image by Sylenus](#) is licensed under CC BY 3.0; changes made

[Image by US Government](#) is in the public domain

Inside four images, clockwise, starting from top left:

[Image](#) is CCO 1.0 public domain

[Image by Tucania](#) is licensed under CC BY-SA 3.0; changes made

[Image by Intuitive Surgical, Inc.](#) is licensed under CC BY-SA 3.0; changes made

[Image by Oyundari Zorigtbaatar](#) is licensed under CC BY-SA 4.0

## Diamond Sponsors



## Platinum Sponsors



## Gold Sponsors



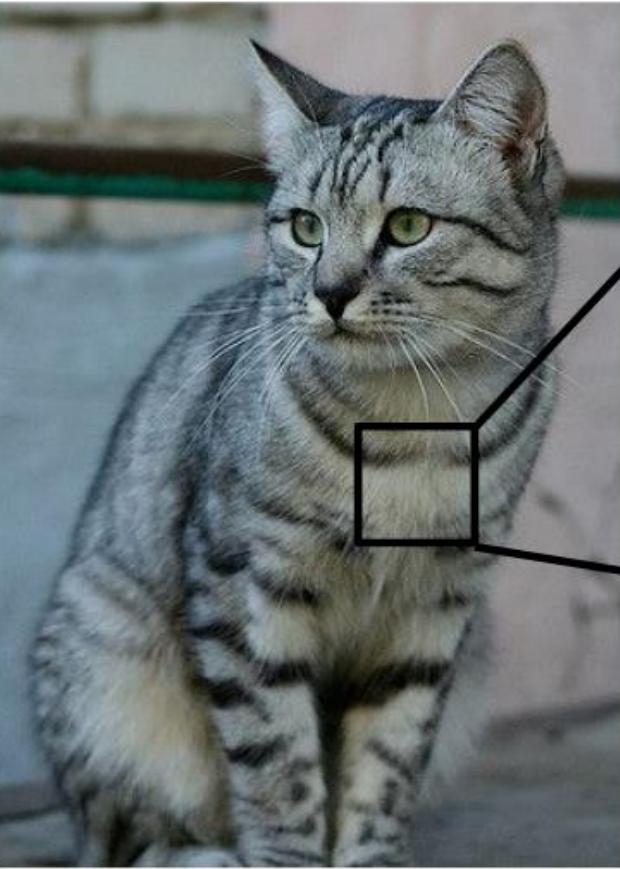
## Silver Sponsors



ZILLION GROUP ZO

Computer  
Vision is  
Challenging





What we see

```
[[105 112 108 111 104 99 106 99 96 103 112 119 104 97 93 87]
 [91 98 102 106 104 79 98 103 99 105 123 136 110 105 94 85]
 [76 85 90 105 128 105 87 96 95 99 115 112 106 103 99 85]
 [99 81 81 93 120 131 127 100 95 98 102 99 96 93 101 94]
 [106 91 61 64 69 91 88 85 101 107 109 98 75 84 96 95]
 [114 108 85 55 55 69 64 54 64 87 112 129 98 74 84 91]
 [133 137 147 103 65 81 80 65 52 54 74 84 102 93 85 82]
 [128 137 144 140 109 95 86 70 62 65 63 63 60 73 86 101]
 [125 133 148 137 119 121 117 94 65 79 80 65 54 64 72 98]
 [127 125 131 147 133 127 126 131 111 96 89 75 61 64 72 84]
 [115 114 109 123 150 148 131 118 113 109 100 92 74 65 72 78]
 [89 93 90 97 188 147 131 118 113 114 113 109 106 95 77 80]
 [63 77 86 81 77 79 102 123 117 115 117 125 125 130 115 87]
 [62 65 82 89 78 71 80 101 124 126 119 101 107 114 131 119]
 [63 65 75 88 89 71 62 81 120 138 135 105 81 98 110 118]
 [87 65 71 87 106 95 69 45 76 130 126 107 92 94 105 112]
 [118 97 82 86 117 123 116 66 41 51 95 93 89 95 102 107]
 [164 146 112 80 82 120 124 104 76 48 45 66 88 101 102 109]
 [157 170 157 120 93 86 114 132 112 97 69 55 70 82 99 94]
 [130 128 134 161 139 100 109 118 121 134 114 87 65 53 69 86]
 [128 112 96 117 150 144 120 115 104 107 102 93 87 81 72 79]
 [123 107 96 86 83 112 153 149 122 109 104 75 80 107 112 99]
 [122 121 102 80 82 86 94 117 145 148 153 102 58 78 92 107]
 [122 164 148 103 71 56 78 83 93 103 119 139 102 61 69 84]]]
```

What a computer sees

# Challenges: Background Clutter



[This image is CC0 1.0 public domain](#)



[This image is CC0 1.0 public domain](#)

# Challenges: Illumination



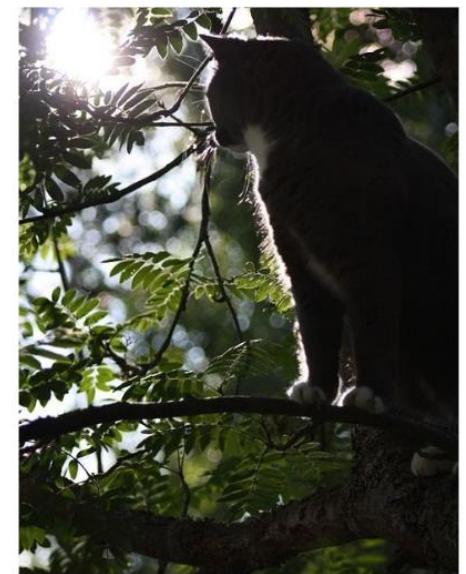
[This image is CC0 1.0 public domain](#)



[This image is CC0 1.0 public domain](#)



[This image is CC0 1.0 public domain](#)



[This image is CC0 1.0 public domain](#)

# Challenges: Deformation



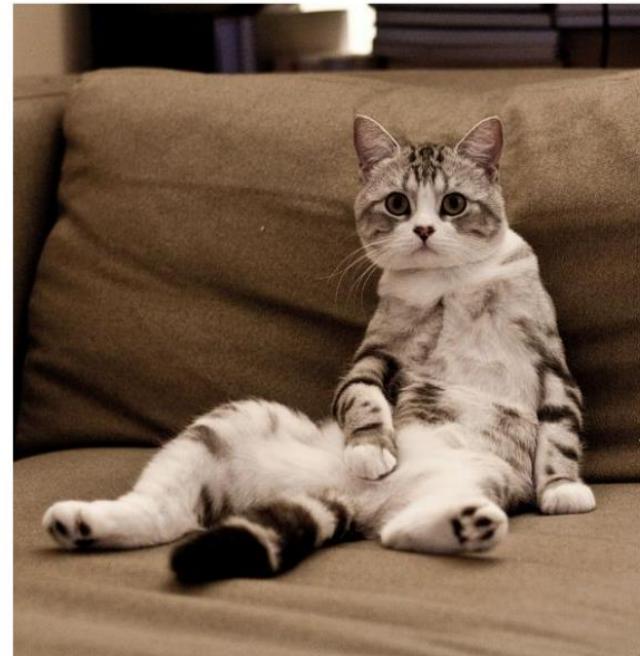
[This image by Umberto Salvagnin](#)  
is licensed under [CC-BY 2.0](#)



[This image by Umberto Salvagnin](#)  
is licensed under [CC-BY 2.0](#)



[This image by sare bear is](#)  
licensed under [CC-BY 2.0](#)



[This image by Tom Thai is](#)  
licensed under [CC-BY 2.0](#)

# Challenges: Occlusion



[This image is CC0 1.0 public domain](#)



[This image is CC0 1.0 public domain](#)



[This image by \*\*jonsson\*\* is licensed under CC-BY 2.0](#)

# Challenges: Intraclass variation



This image is [CC0 1.0 public domain](#)

# dog



# mug



# hat





How do we solve it?

# Deep Learning

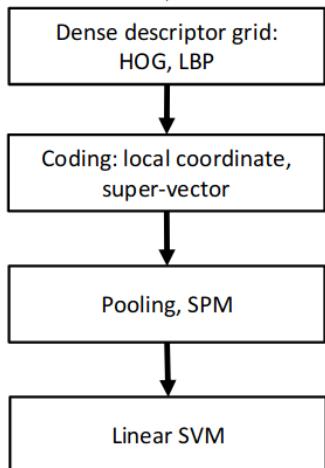


**Deep Convolutional Neural Networks**  
have become an important tool for Computer Vision.

# IMAGENET Large Scale Visual Recognition Challenge

## Year 2010

NEC-UIUC

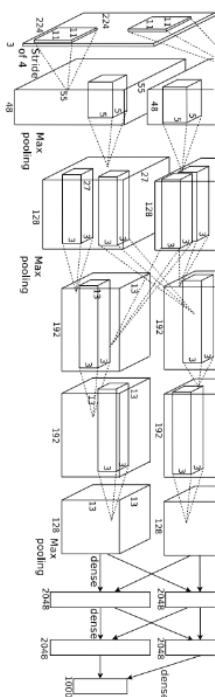


[Lin CVPR 2011]

Lion image by Swissfrog is licensed under CC BY 3.0

## Year 2012

SuperVision



[Krizhevsky NIPS 2012]

Figure copyright Alex Krizhevsky, Ilya Sutskever, and Geoffrey Hinton, 2012. Reproduced with permission.

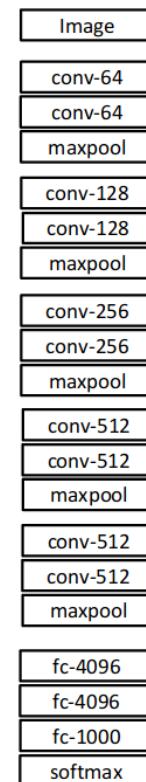
## Year 2014

GoogLeNet

- Pooling
- Convolution
- Softmax
- Other



VGG

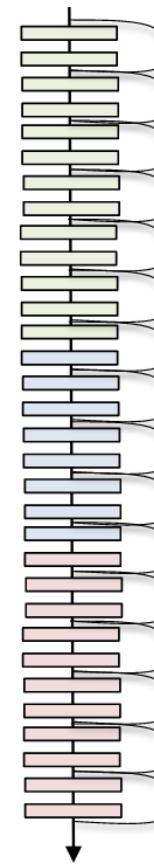


[Szegedy arxiv 2014]

[Simonyan arxiv 2014]

## Year 2015

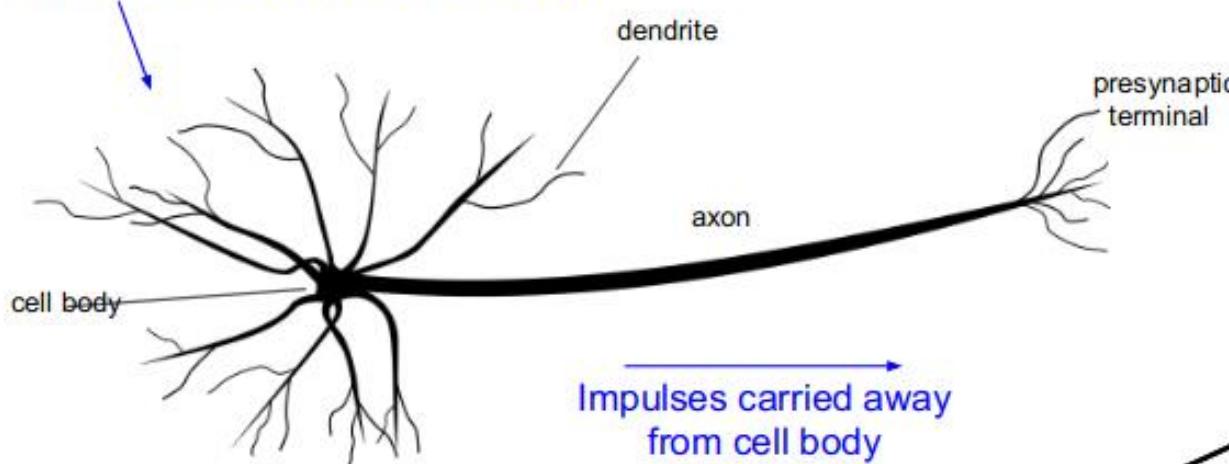
MSRA



[He ICCV 2015]

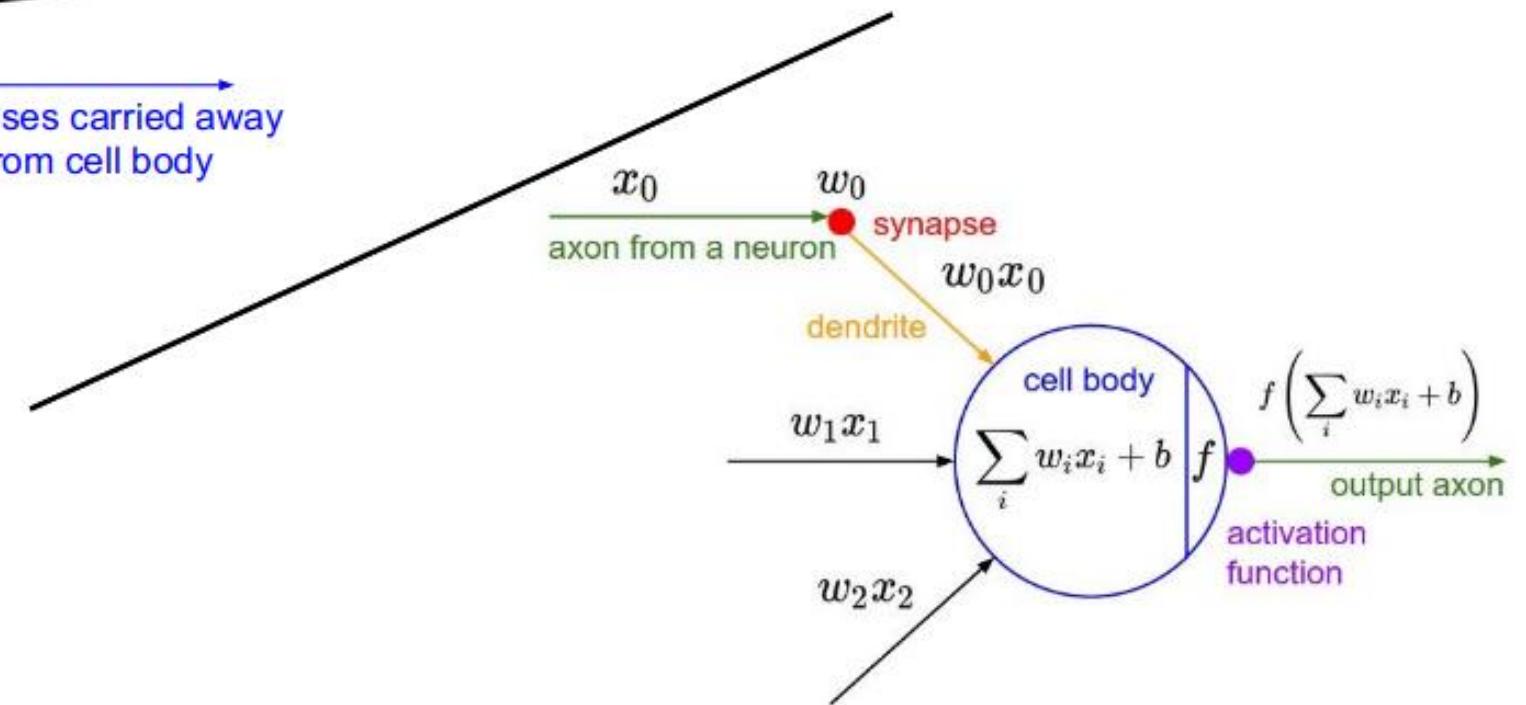
# Neural Networks

Impulses carried toward cell body



This image by Felipe Perucho  
is licensed under CC-BY 3.0

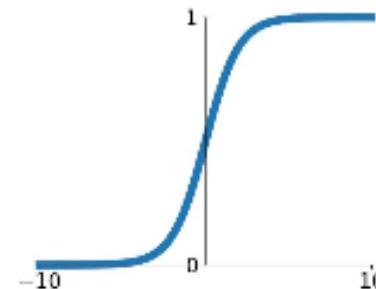
Impulses carried away  
from cell body



# Activation Functions: Non-linear Computations

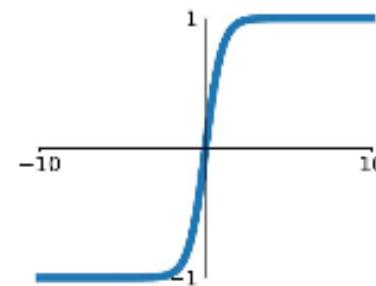
## Sigmoid

$$\sigma(x) = \frac{1}{1+e^{-x}}$$



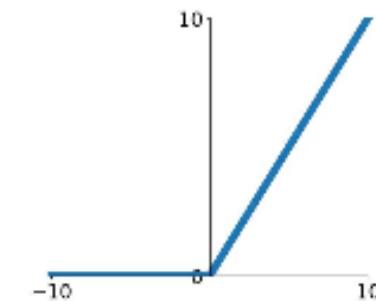
## tanh

$$\tanh(x)$$



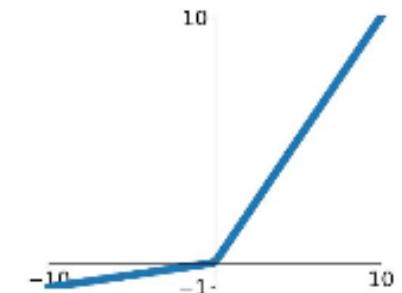
## ReLU

$$\max(0, x)$$



## Leaky ReLU

$$\max(0.1x, x)$$

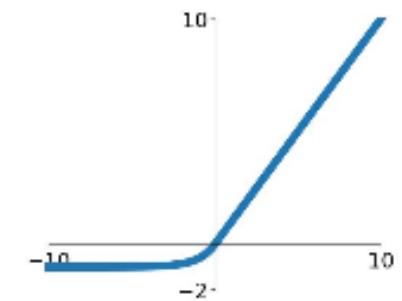


## Maxout

$$\max(w_1^T x + b_1, w_2^T x + b_2)$$

## ELU

$$\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$$



# Neural Networks

- Linear score function

$$f = Wx$$

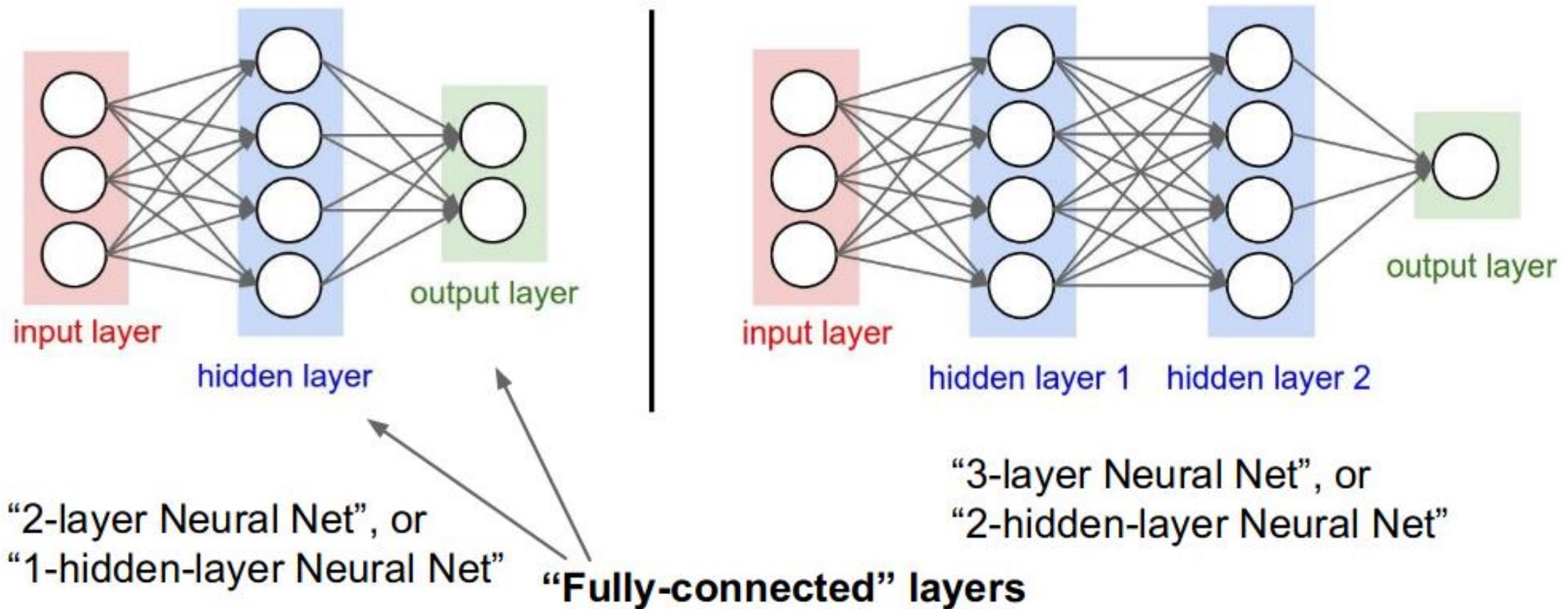
- 2-layer Neural Network

$$f = W_2 \max(0, W_1 x)$$

- 3-layer Neural Network

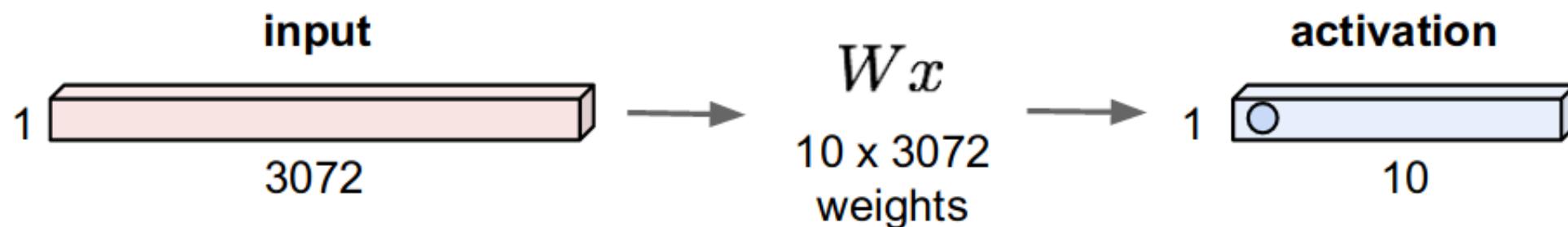
$$f = W_3 \max(0, W_2 \max(0, W_1 x))$$

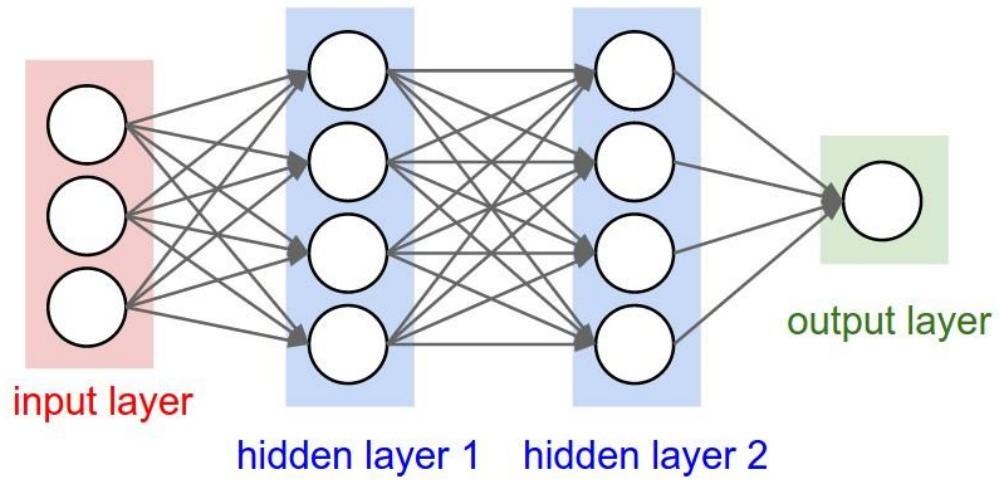
# Neural Networks



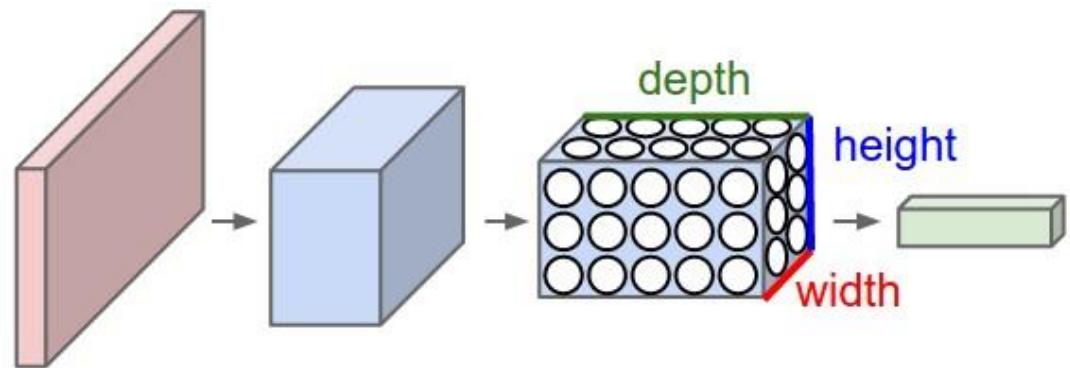
# Fully Connected Layer

32x32x3 image -> stretch to 3072 x 1



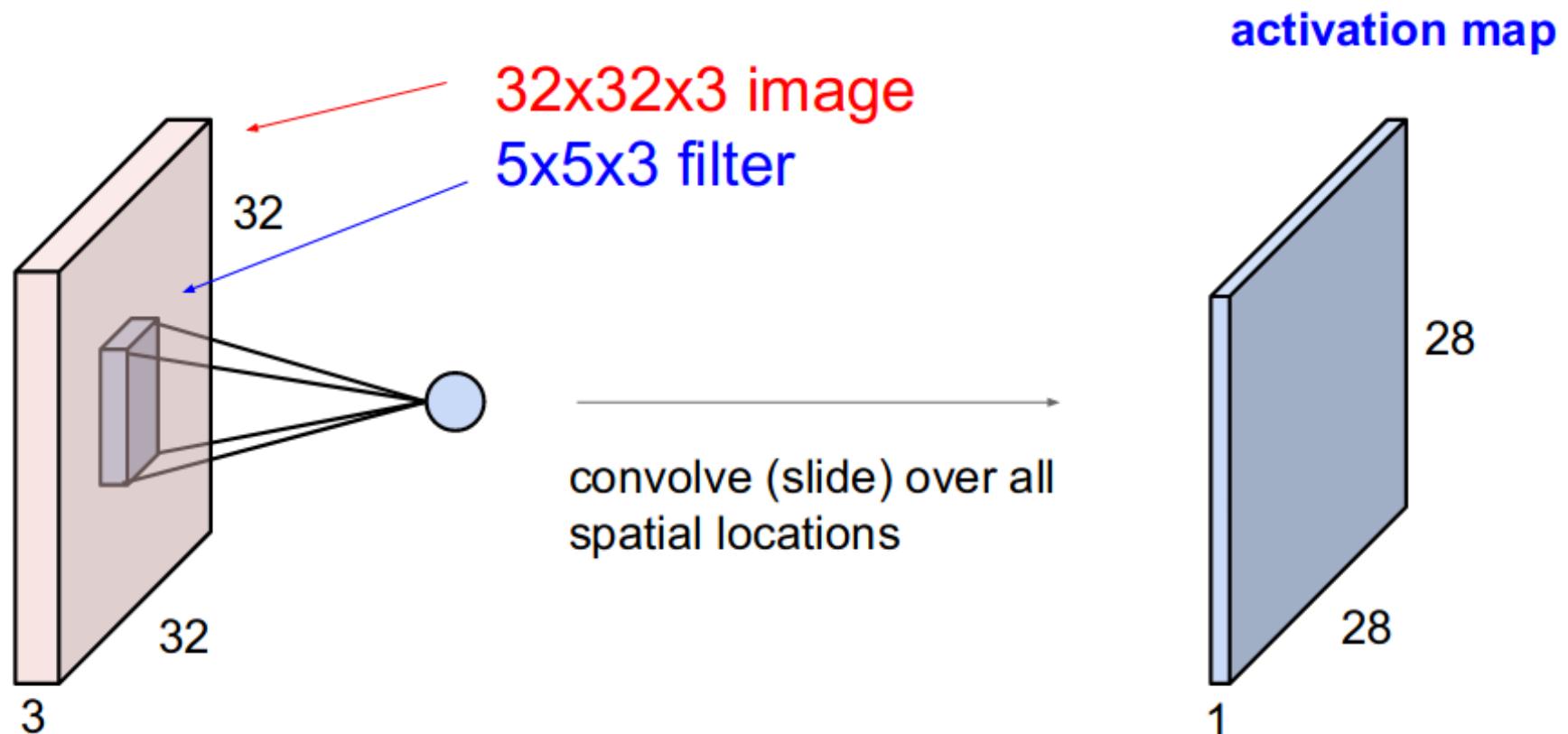


Fully-Connected Neural Network



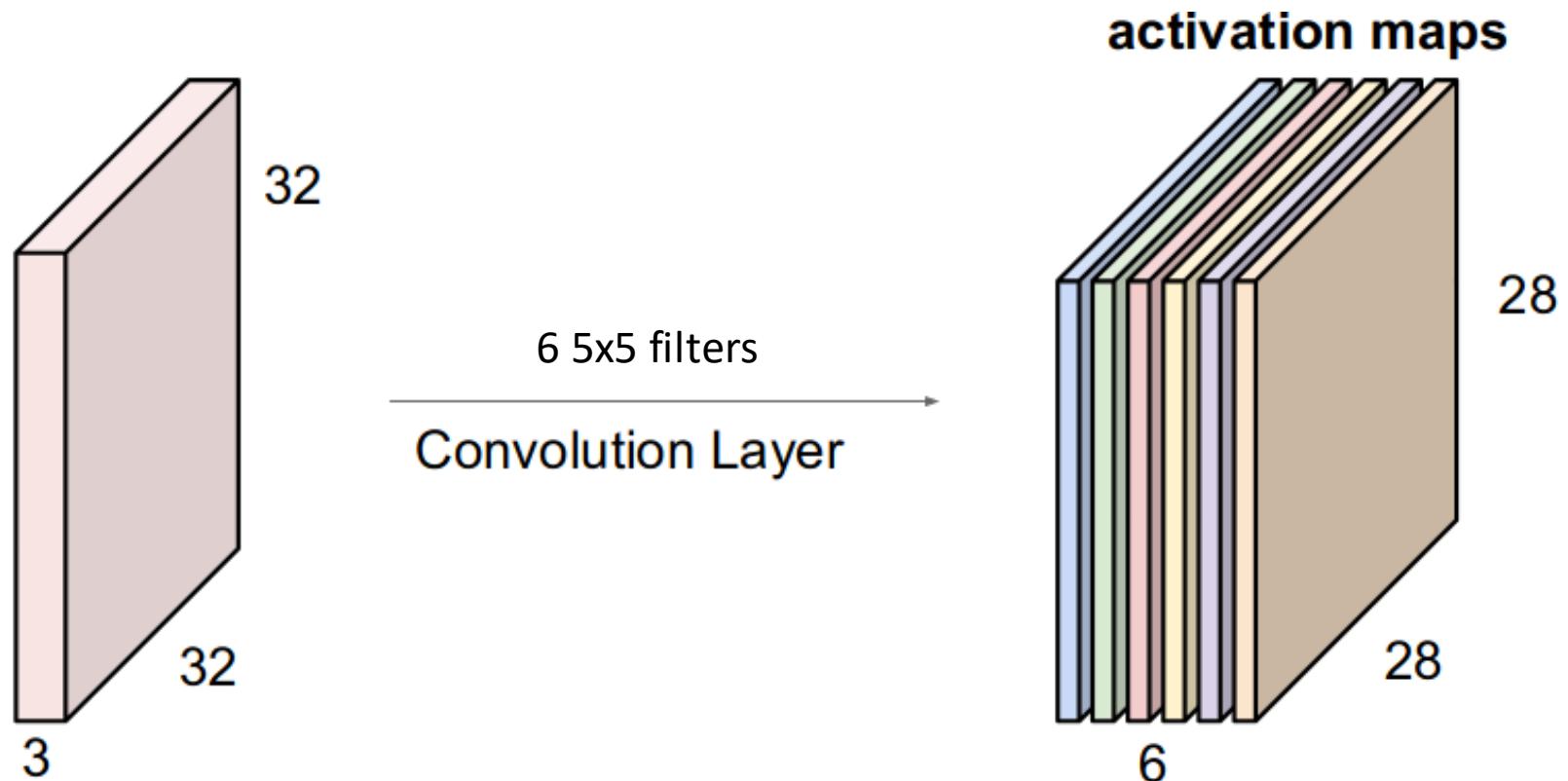
Convolutional Neural Network

# Convolution Layer

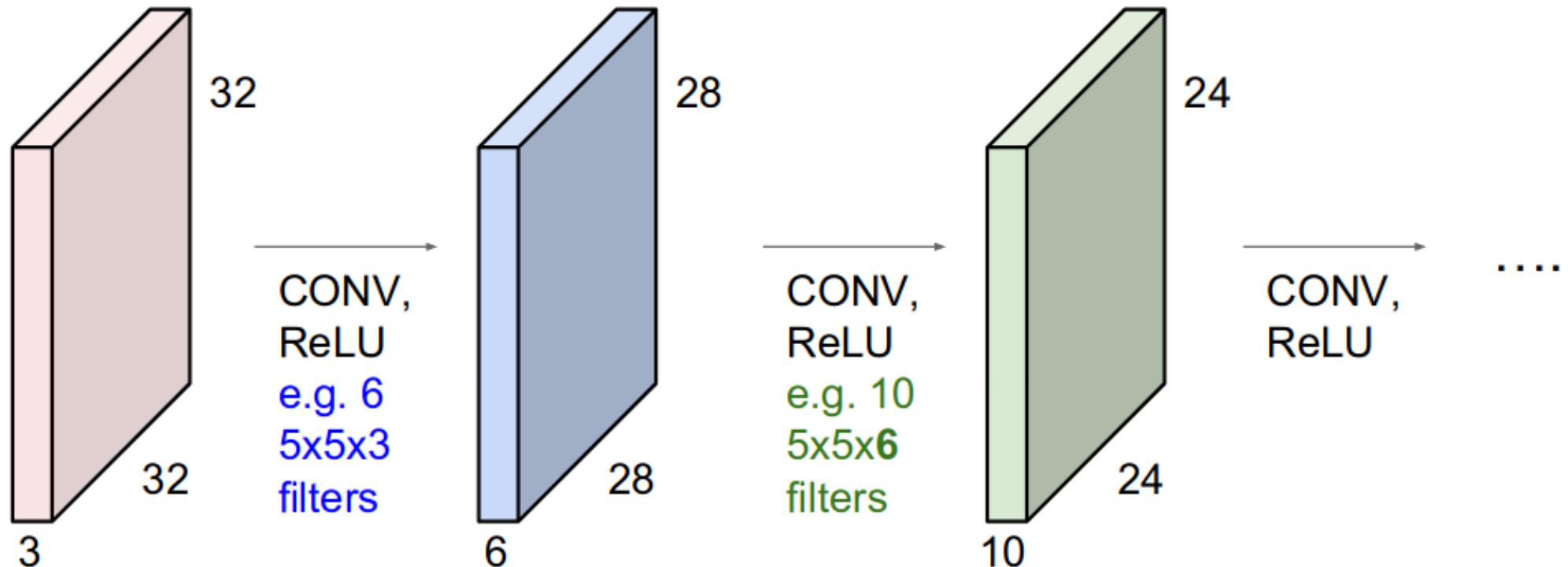


**Convolve** the filter with the image i.e. “slide over the image spatially, computing **dot products**”

# Convolution Layer



# Convolution Layers



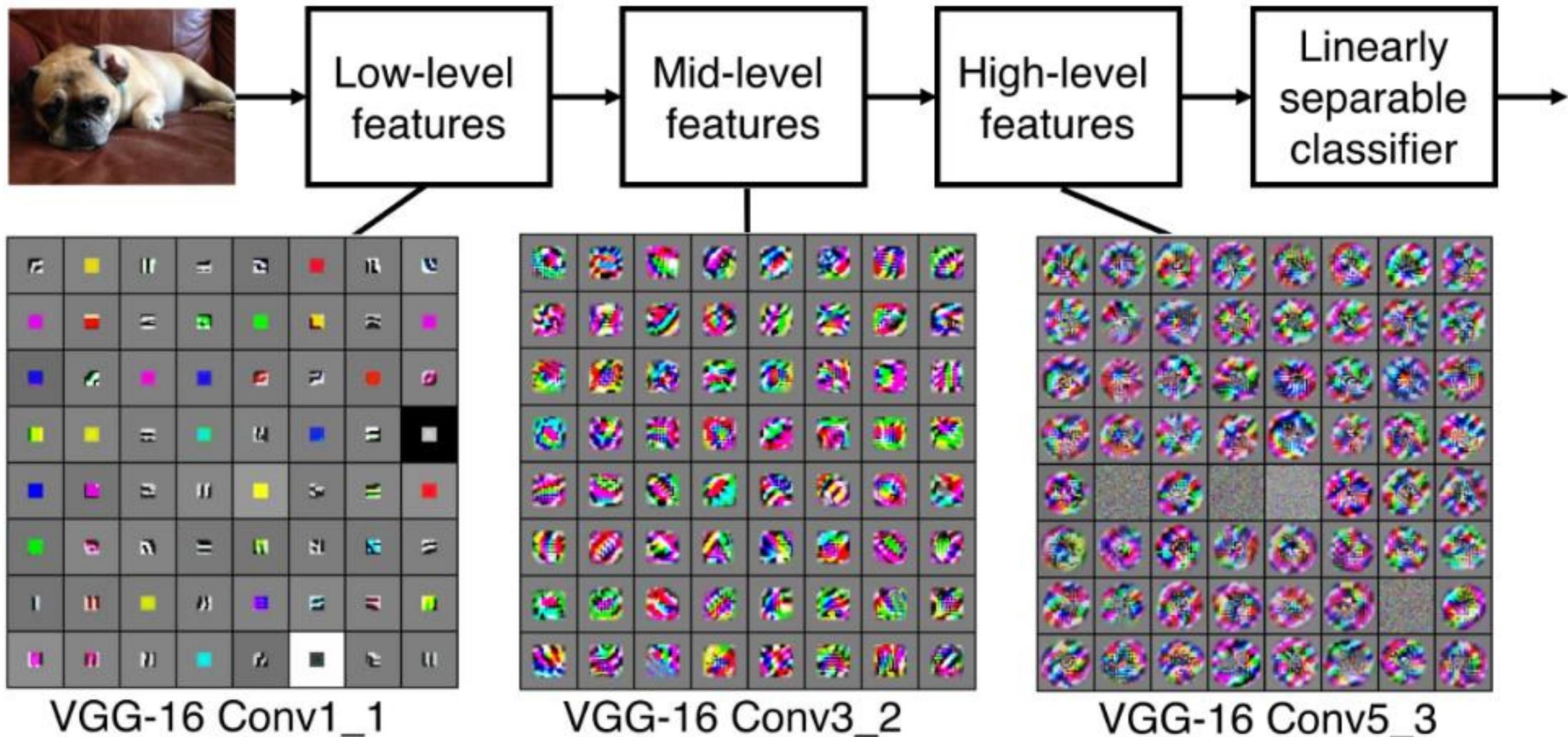
# ReLU

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

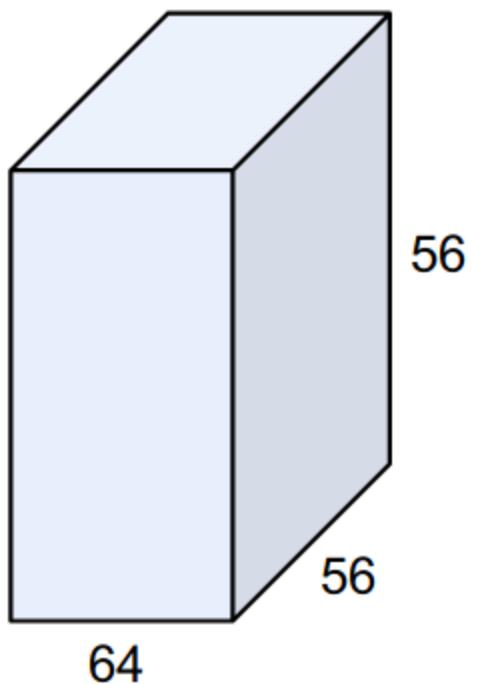
ReLU



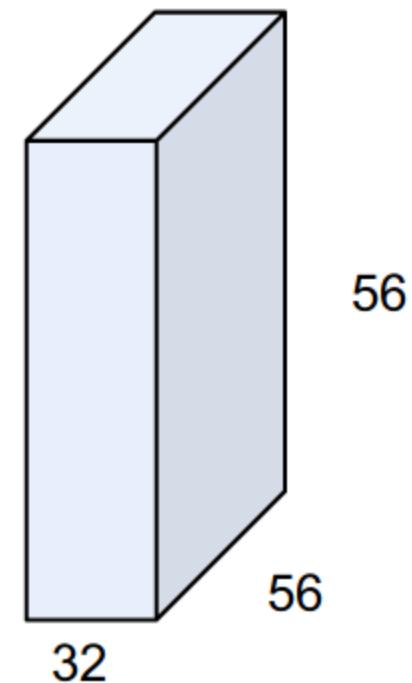
0	5	0	2
0	3	0	6
6	4	0	0
0	8	0	9



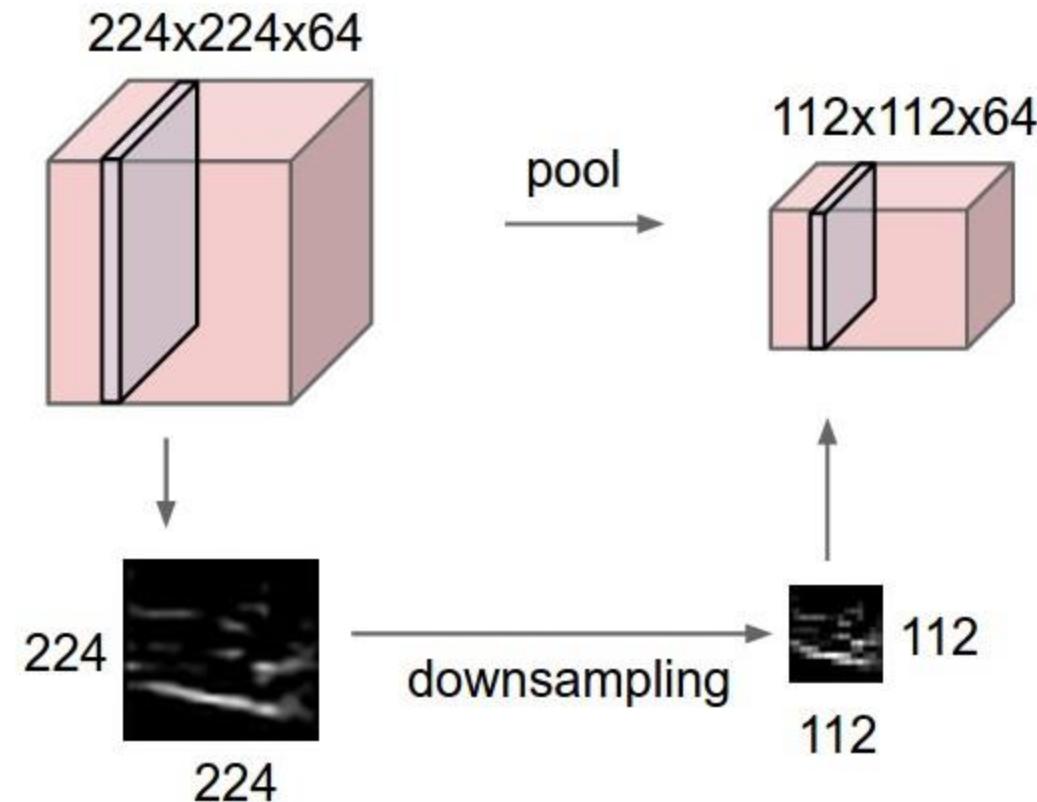
# 1x1 Convolution layer



1x1 CONV  
with 32 filters  
→  
(each filter has size  
1x1x64, and performs a  
64-dimensional dot  
product)

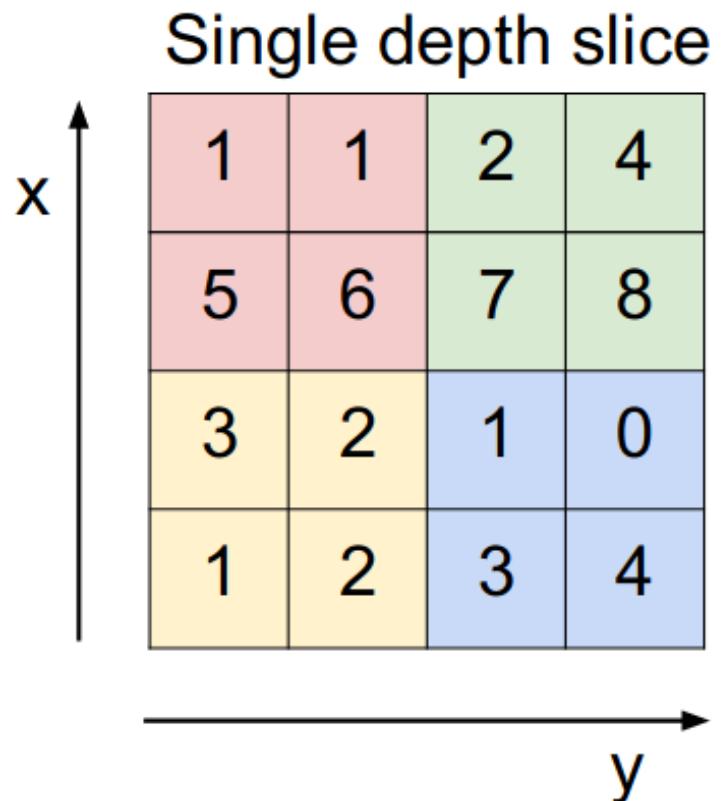


# Pooling layer



- makes the representations smaller and more manageable
- operates over each activation map independently

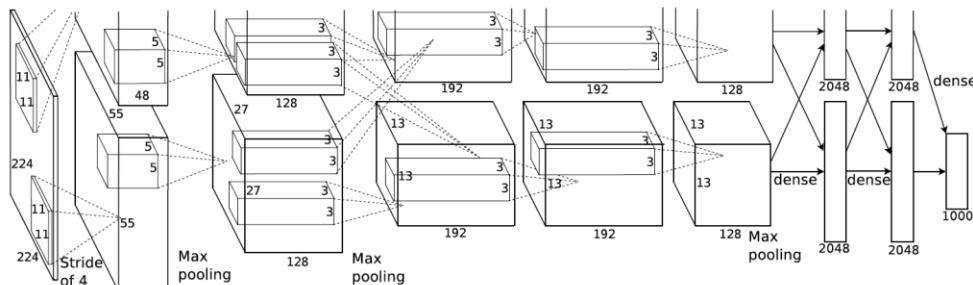
# Max Pooling



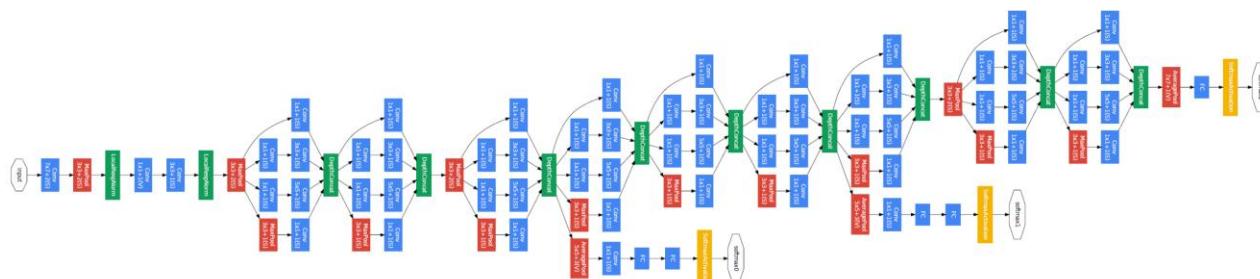
max pool with 2x2 filters  
and stride 2

6	8
3	4

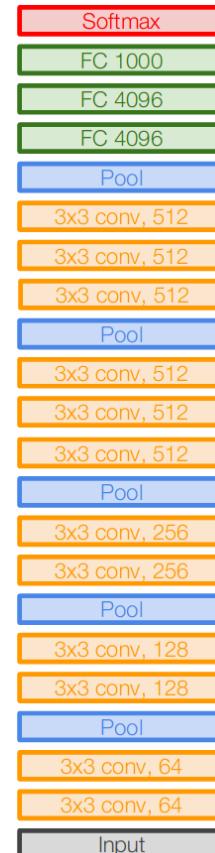
# CNN Architectures



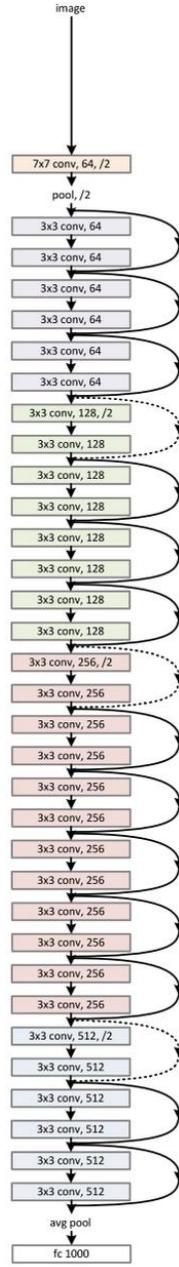
AlexNet



GoogleNet

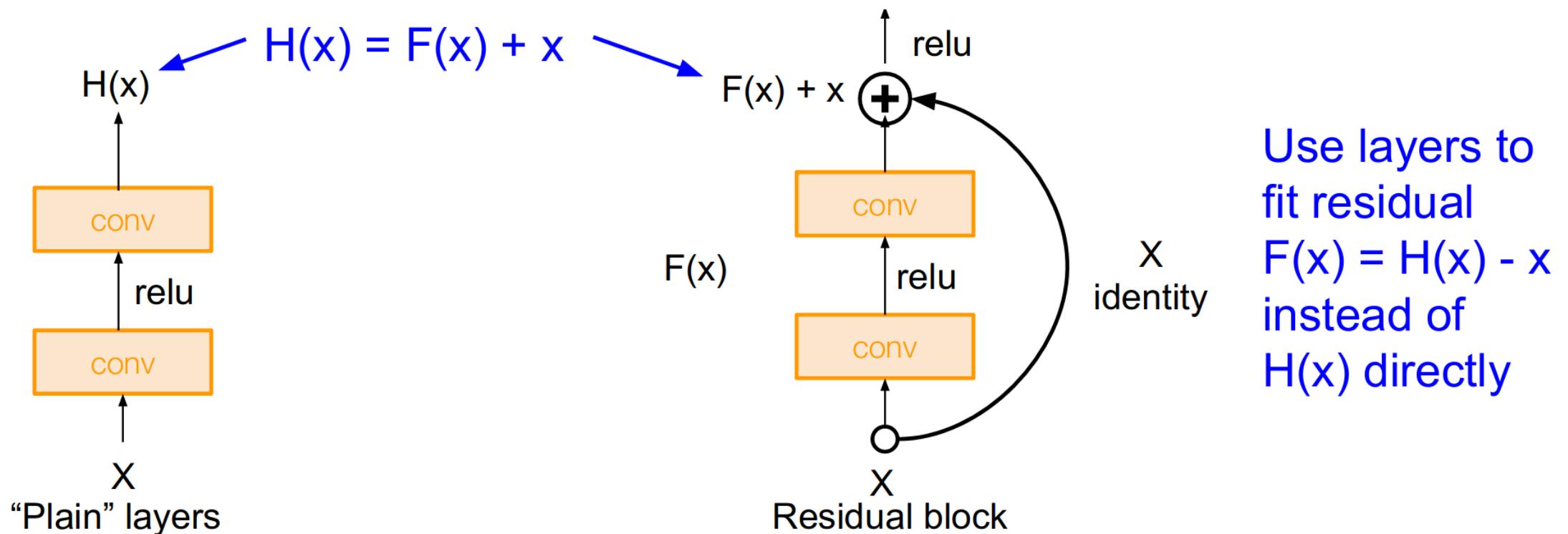


VGGNet



ResNet

# ResNet



# Loss Functions

Suppose: 3 training examples, 3 classes.

With some  $W$  the scores  $f(x, W) = Wx$  are:



cat	<b>3.2</b>	1.3	2.2
car	<b>5.1</b>	<b>4.9</b>	2.5
frog	-1.7	2.0	<b>-3.1</b>

A **loss function** tells how good our current classifier is

Given a dataset of examples

$$\{(x_i, y_i)\}_{i=1}^N$$

Where  $x_i$  is image and  
 $y_i$  is (integer) label

Loss over the dataset is a sum of loss over examples:

$$L = \frac{1}{N} \sum_i L_i(f(x_i, W), y_i)$$

Suppose: 3 training examples, 3 classes.  
With some  $W$  the scores  $f(x, W) = Wx$  are:



cat	<b>3.2</b>	1.3	2.2
car	5.1	<b>4.9</b>	2.5
frog	-1.7	2.0	<b>-3.1</b>

## Multiclass SVM loss:

Given an example  $(x_i, y_i)$  where  $x_i$  is the image and where  $y_i$  is the (integer) label,

and using the shorthand for the scores vector:  $s = f(x_i, W)$

the SVM loss has the form:

$$L_i = \sum_{j \neq y_i} \begin{cases} 0 & \text{if } s_{y_i} \geq s_j + 1 \\ s_j - s_{y_i} + 1 & \text{otherwise} \end{cases}$$
$$= \sum_{j \neq y_i} \max(0, s_j - s_{y_i} + 1)$$

# Softmax Classifier (Multinomial Logistic Regression)



cat	<b>3.2</b>
car	5.1
frog	-1.7

**scores = unnormalized log probabilities of the classes.**

$$P(Y = k | X = x_i) = \frac{e^{s_k}}{\sum_j e^{s_j}} \quad \text{where} \quad s = f(x_i; W)$$

Want to maximize the log likelihood, or (for a loss function) to minimize the negative log likelihood of the correct class:

$$L_i = -\log P(Y = y_i | X = x_i)$$

---

in summary:  $L_i = -\log\left(\frac{e^{s_{y_i}}}{\sum_j e^{s_j}}\right)$



# Stochastic Gradient Descent (SGD)

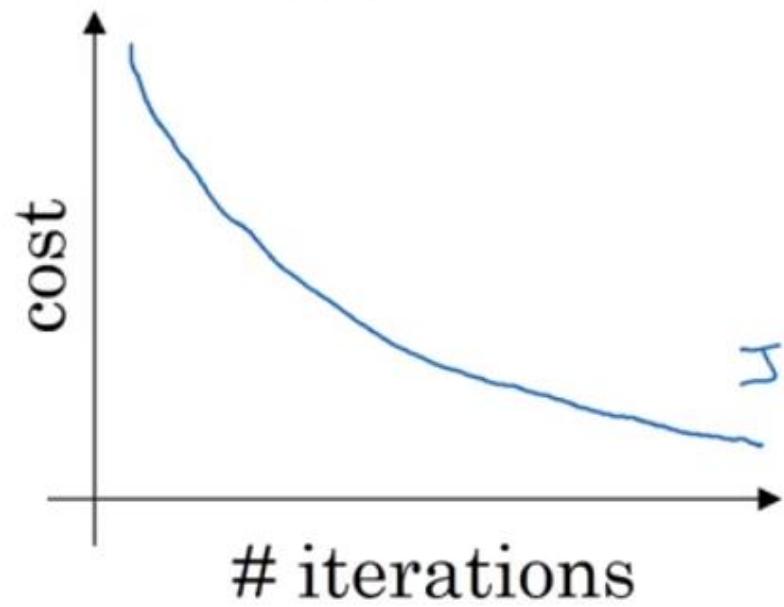
$$L(W) = \frac{1}{N} \sum_{i=1}^N L_i(x_i, y_i, W) + \lambda R(W)$$

$$\nabla_W L(W) = \frac{1}{N} \sum_{i=1}^N \nabla_W L_i(x_i, y_i, W) + \lambda \nabla_W R(W)$$

Full sum expensive  
when N is large!

Approximate sum  
using a **minibatch** of  
examples  
32 / 64 / 128 common

Batch gradient descent



Mini-batch gradient descent

