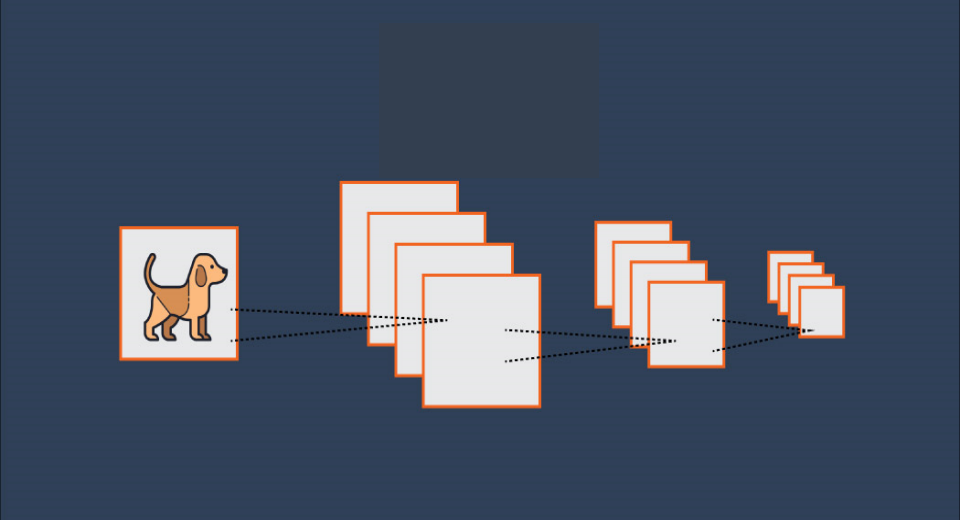
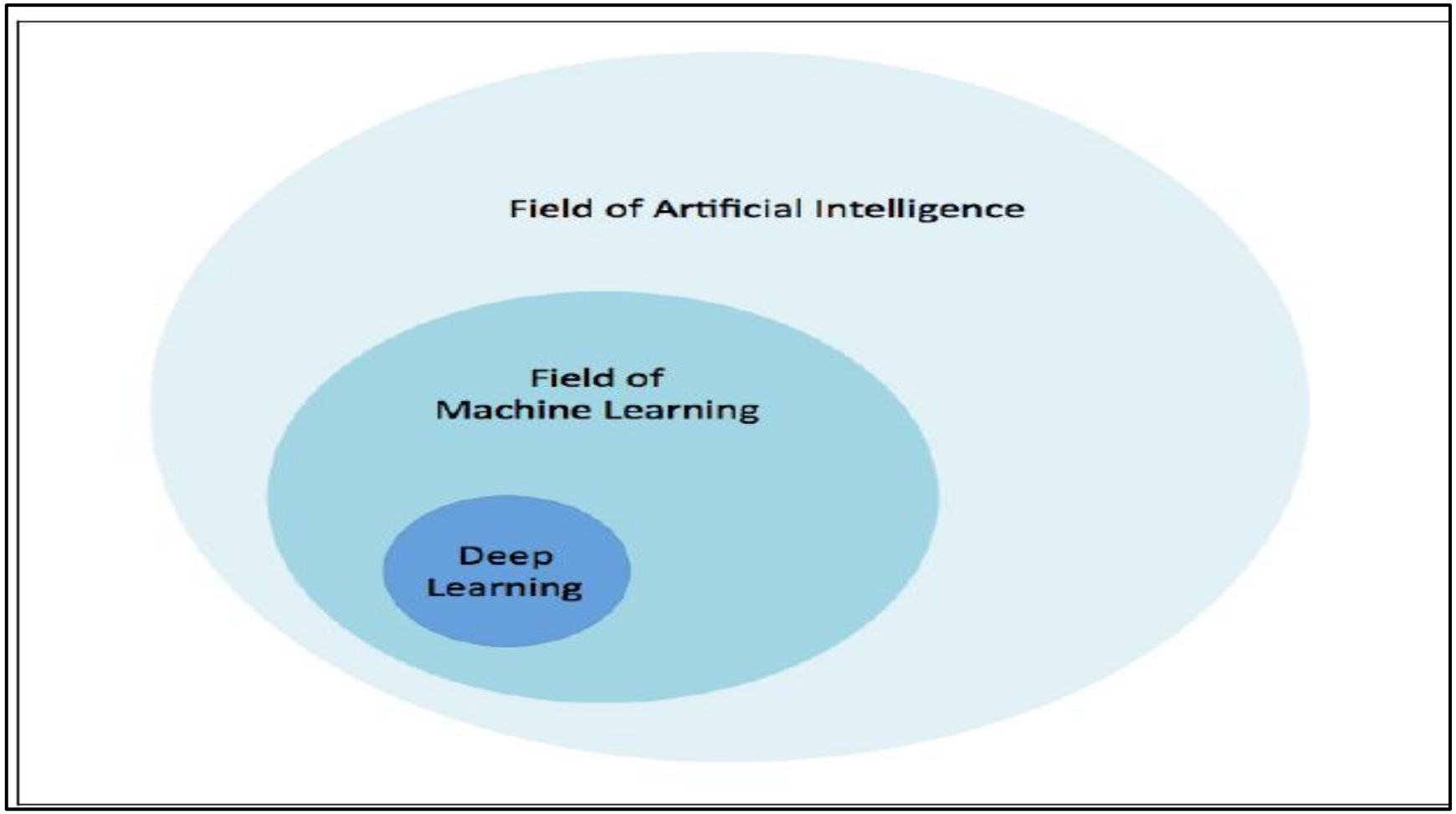


# Convolutional Neural Networks

Dr. V. Uma
Assistant Professor
Dept. of Computer Science
Pondicherry University

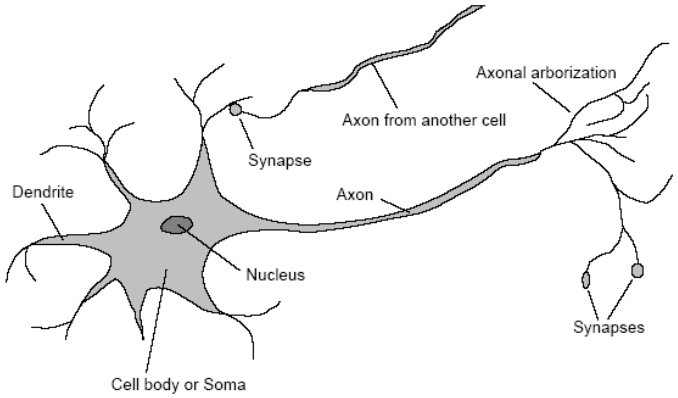

Source: Internet

# Relationship between AI and DL



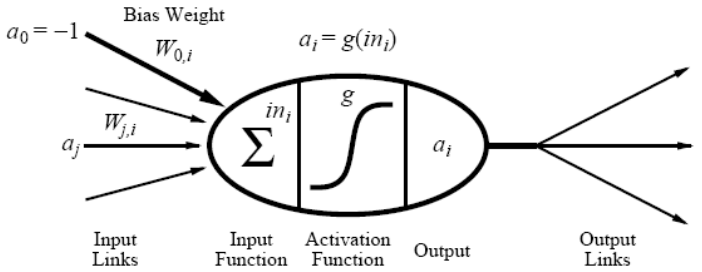
# (Artificial) Neural Networks

- Motivation: human brain
  - massively parallel (*$10^{11}$ neurons, ~20 types*)
  - small computational units with simple low-bandwidth communication (*$10^{14}$ synapses, 1-10ms cycle time*)



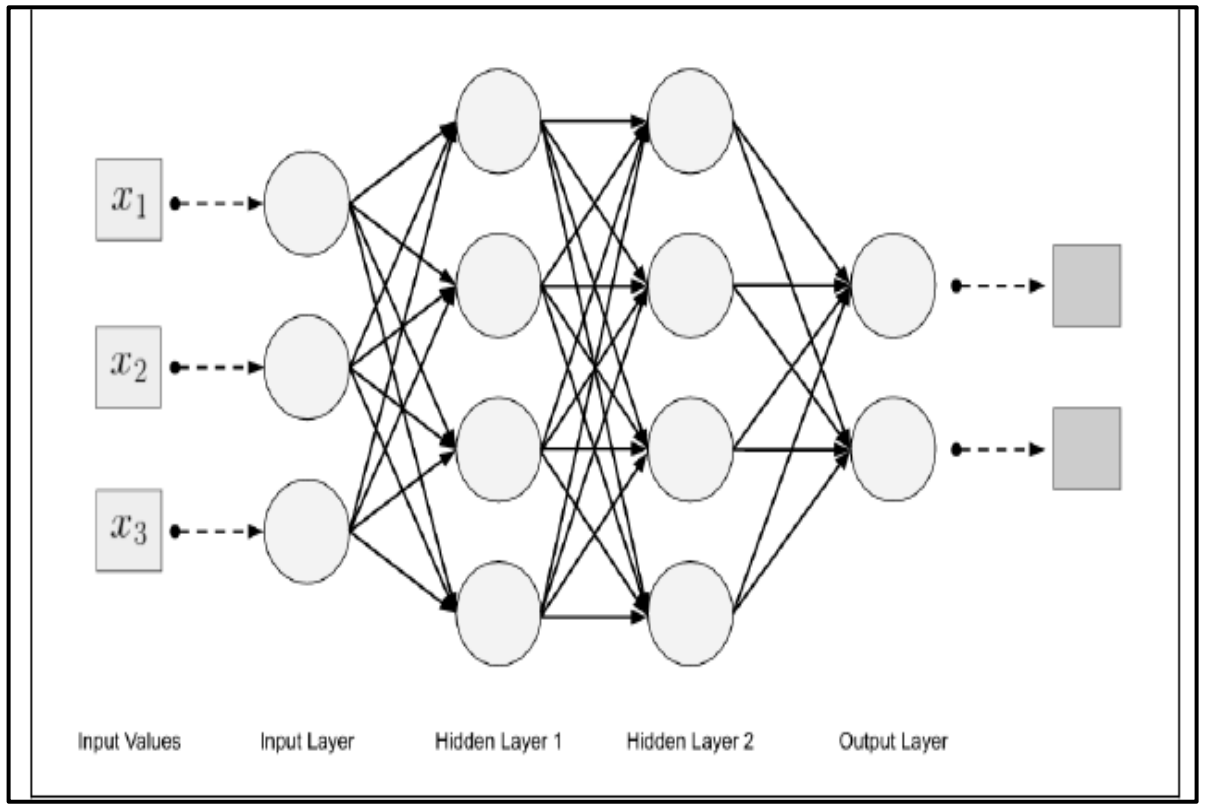

- Realization: neural network
  - *units* (≈ neurons) connected by *directed weighted links*
  - *activation function* from inputs to output

$$a_i \leftarrow g(in_i) = g\left(\sum_j W_{j,i} a_j\right)$$



# Neural network

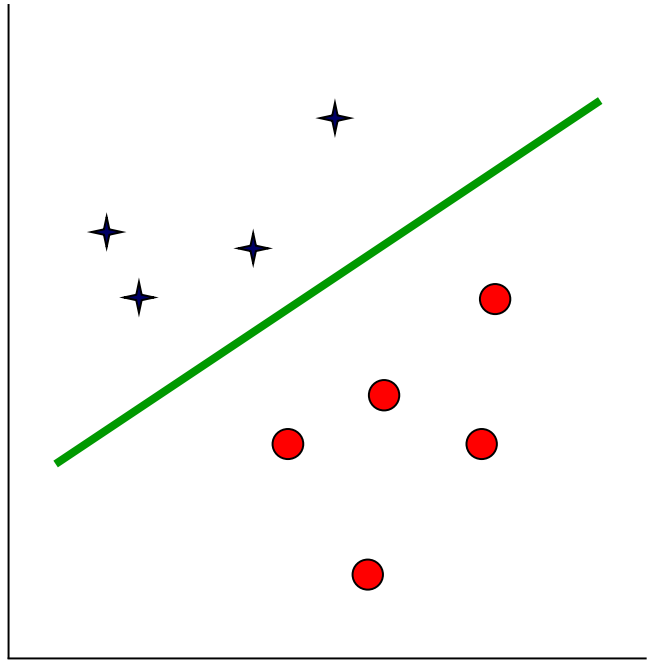
Skin rash
Fever
Cold Cough
Head ache
Vomiting



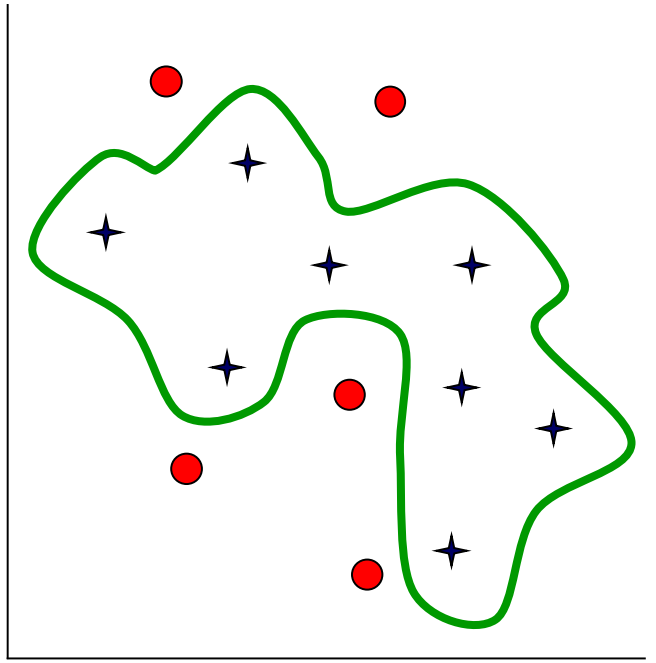

DENGUE

MALARIA

# Neural Network Learning: Decision Boundary



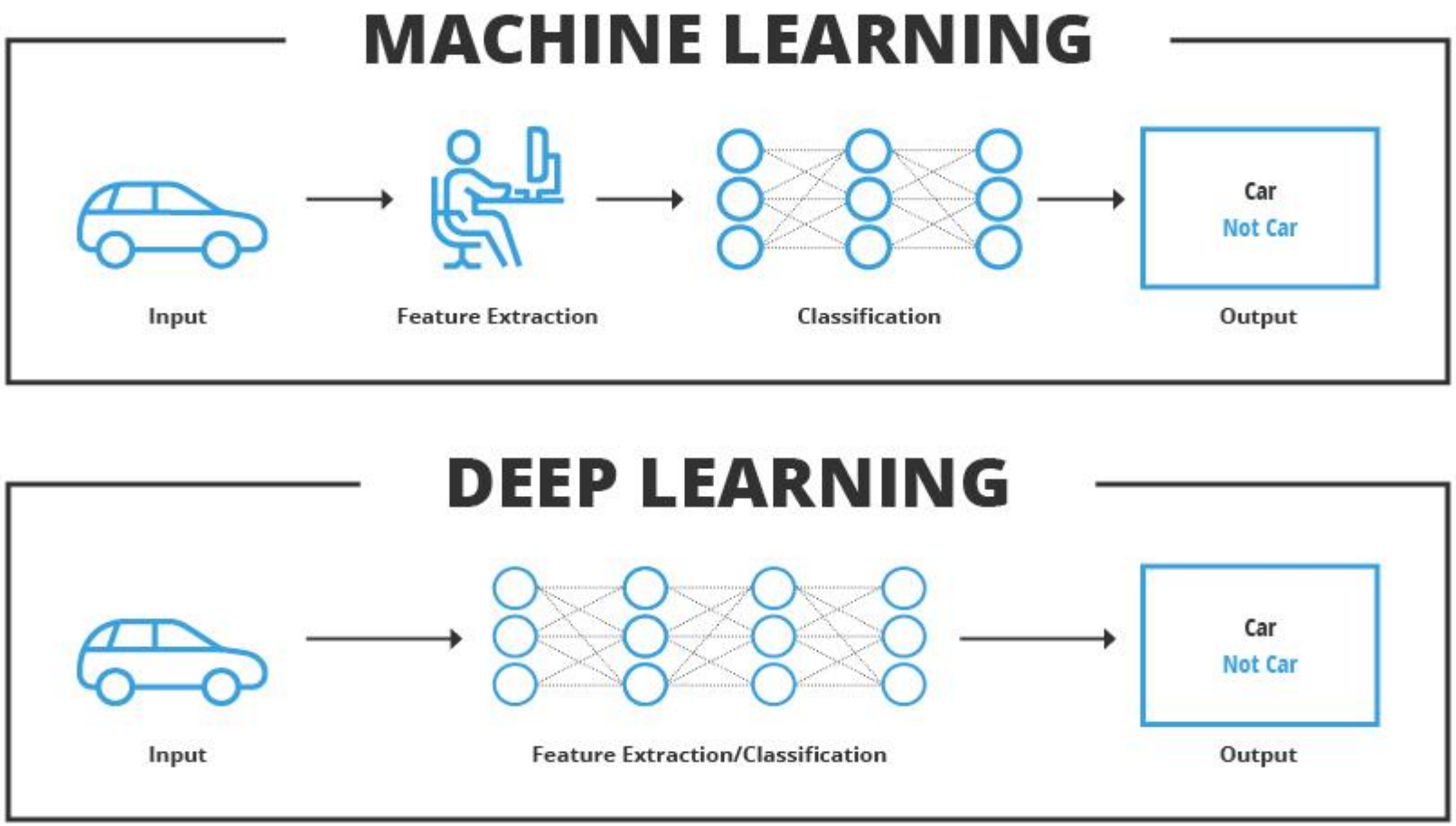
*single-layer perceptron*

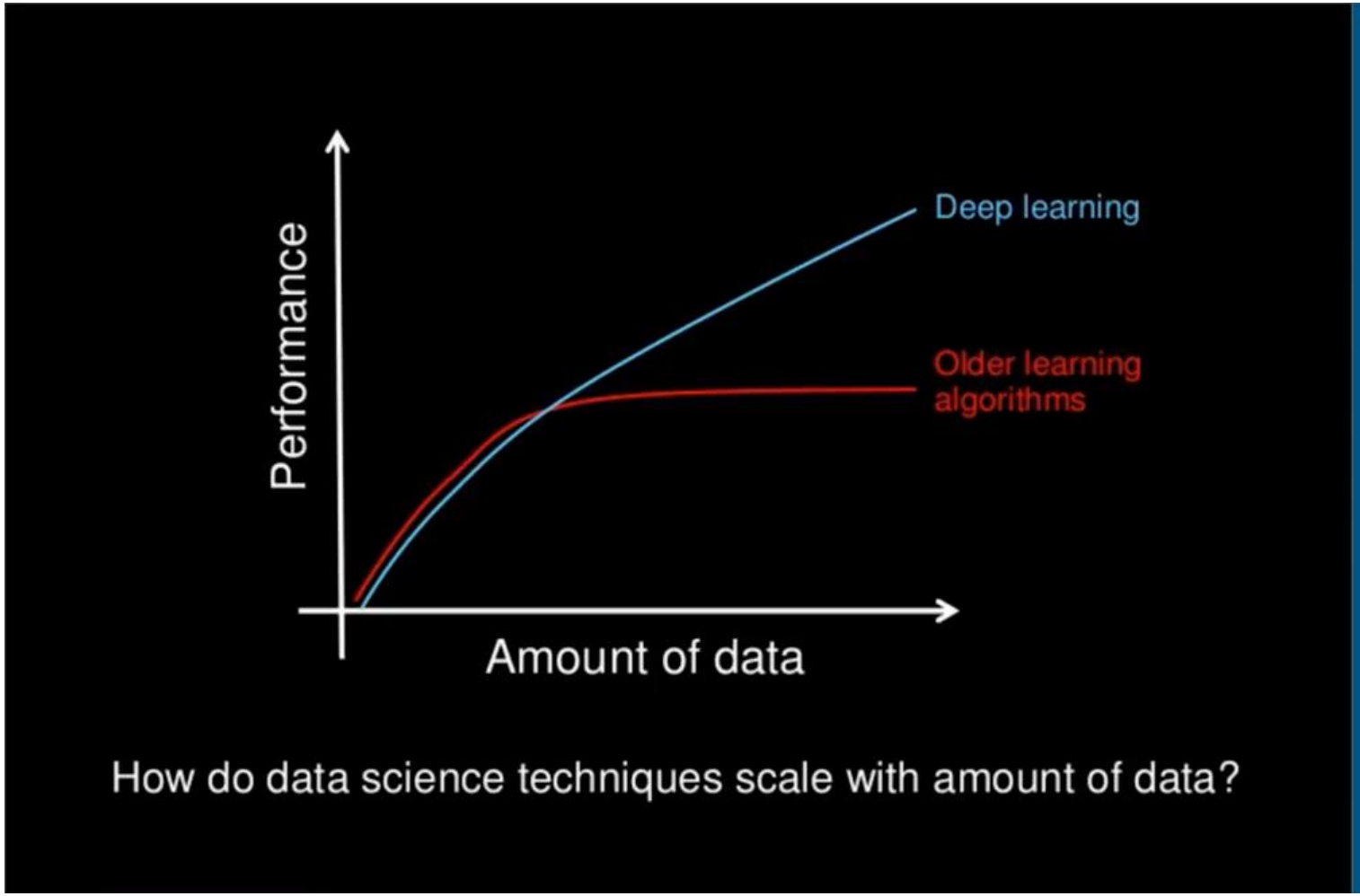


*multi-layer network*

"Deep learning is a particular kind of machine learning that is inspired by the functionality of our brain cells called neurons which led to the concept of artificial neural network"

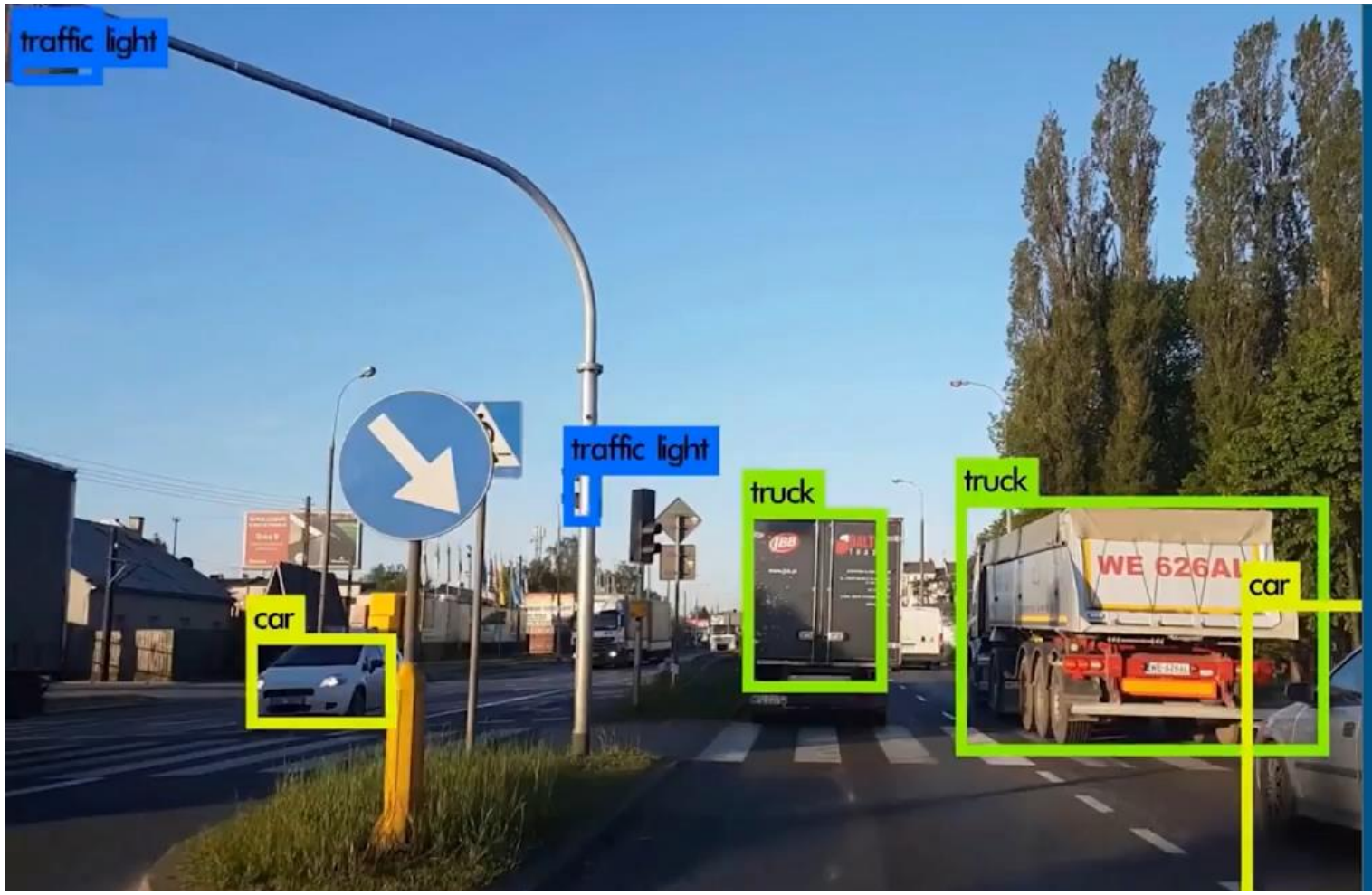
# Deep Learning IS Machine Learning





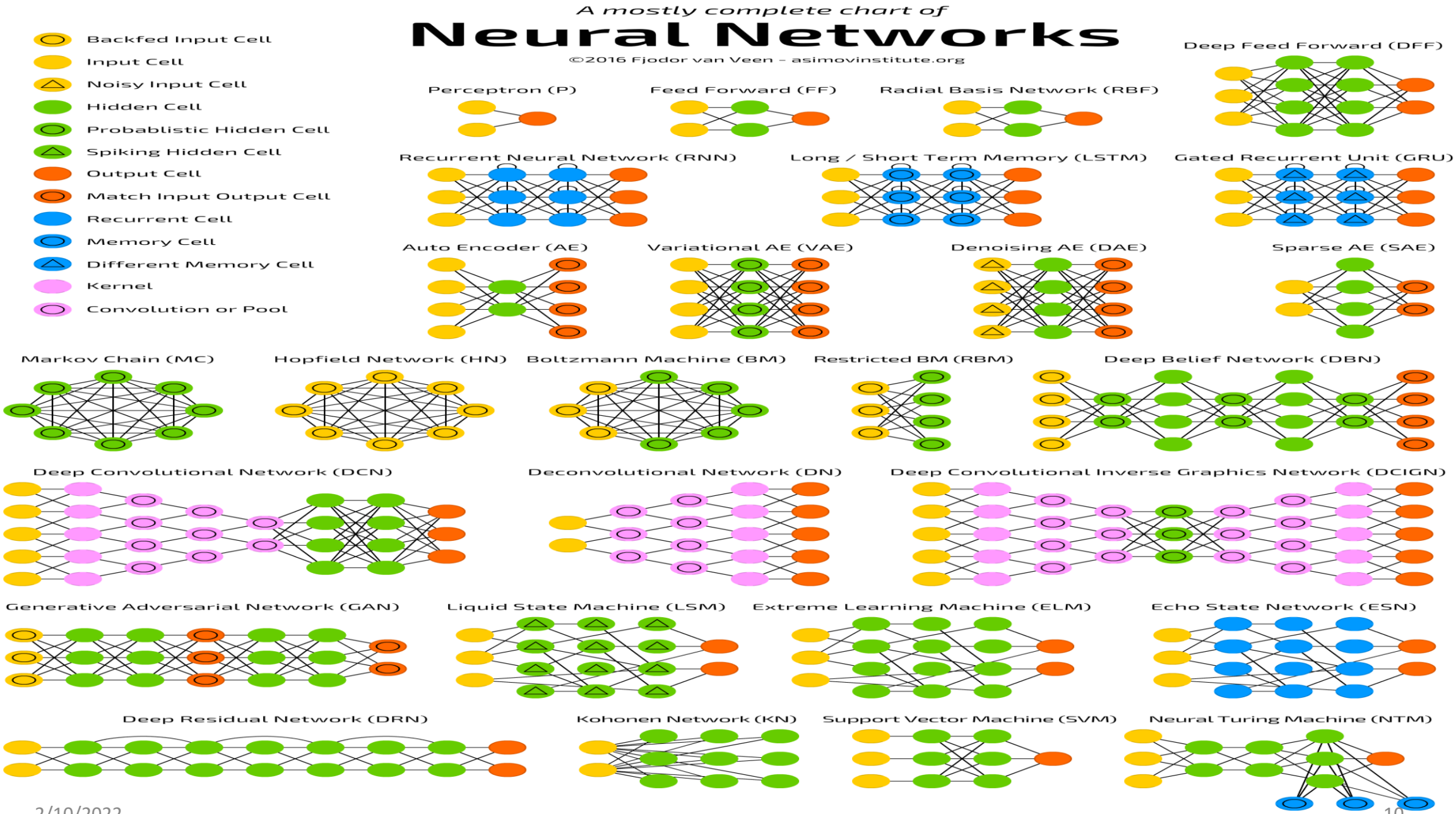

Machine Learning vs Deep Learning

# Data Dependency




Machine Learning
vs Deep Learning

# Problem Solving Approach


A mostly complete chart of
Neural Networks
©2016 Fjodor van Veen - asimovinstitute.org
Backfed Input Cell
Input Cell
Noisy Input Cell
Hidden Cell
Probablistic Hidden Cell
Spiking Hidden Cell
Output Cell
Match Input Output Cell
Recurrent Cell
Memory Cell
Different Memory Cell
Kernel
Convolution or Pool
Perceptron (P)
Feed Forward (FF)
Radial Basis Network (RBF)
Deep Feed Forward (DFF)
Recurrent Neural Network (RNN)
Long / Short Term Memory (LSTM)
Gated Recurrent Unit (GRU)
Auto Encoder (AE)
Variational AE (VAE)
Denoising AE (DAE)
Sparse AE (SAE)
Markov Chain (MC)
Hopfield Network (HN)
Boltzmann Machine (BM)
Restricted BM (RBM)
Deep Belief Network (DBN)
Deep Convolutional Network (DCN)
Deconvolutional Network (DN)
Deep Convolutional Inverse Graphics Network (DCIGN)
Generative Adversarial Network (GAN)
Liquid State Machine (LSM)
Extreme Learning Machine (ELM)
Echo State Network (ESN)
Deep Residual Network (DRN)
Kohonen Network (KN)
Support Vector Machine (SVM)
Neural Turing Machine (NTM)

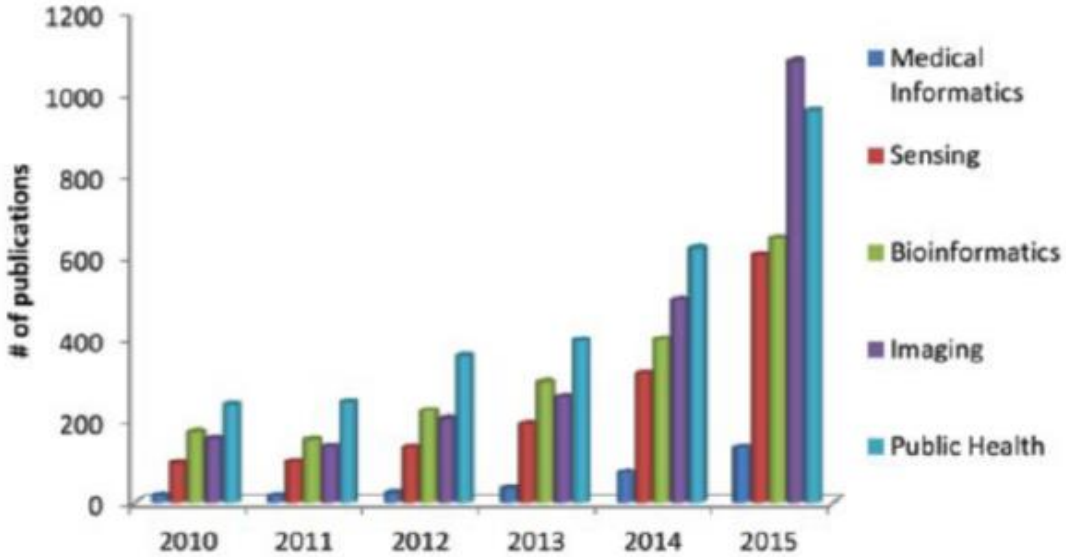
- To generate data (e.g., images, audio, or text), we'd use:
  - — GANs
  - — VAEs
  - — Recurrent Neural Networks
- To model images, we'd likely use:
  - — CNNs
  - — DBNs
- To model sequence data, we'd likely use:
  - — Recurrent Neural Networks/LSTMs

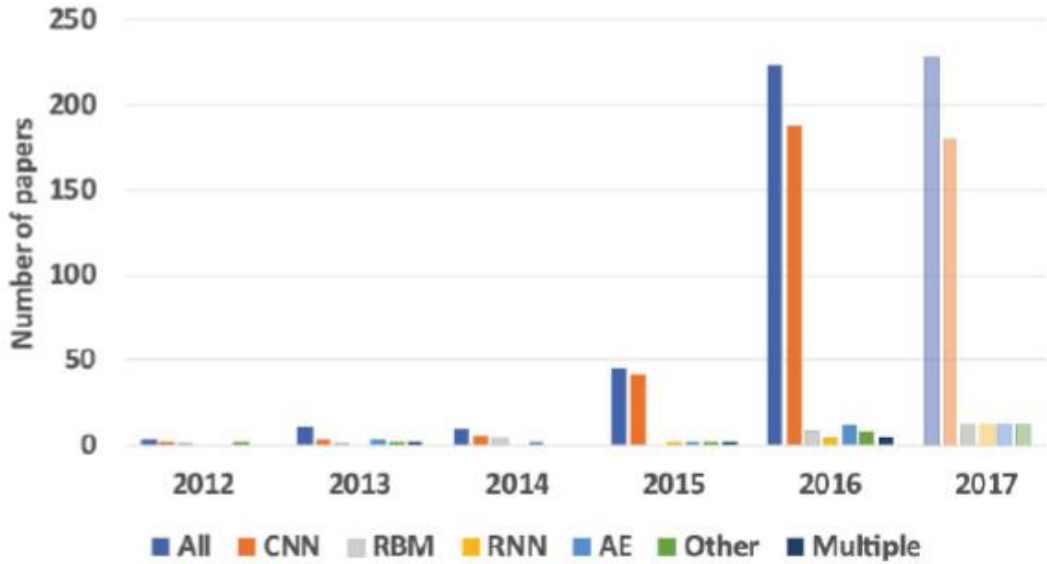
# ImageNet

- The **ImageNet** project is a large visual database designed for use in visual object recognition software research.
- Over ten million URLs of images have been hand-annotated by ImageNet to indicate what objects are pictured; in at least one million of the images, bounding boxes are also provided.
- Since 2010, the ImageNet project runs an annual software contest, the ImageNet Large Scale Visual Recognition Challenge (ILSVRC), where software programs compete to correctly classify and detect objects and scenes.

# Why deep learning for medical imaging

Deep learning is providing exciting solutions for medical image analysis problems




Litjens et al, A survey on Deep learning in medical image analysis, June 2017

https://github.com/terryum/awesome-deep-learning-papers?fbclid=IwAR1pOmIzkbv9yStuzVQS6DHzgKywRp4Ca-jEt8dQmzz17HsWxgh3ToQWCZk

# CNN

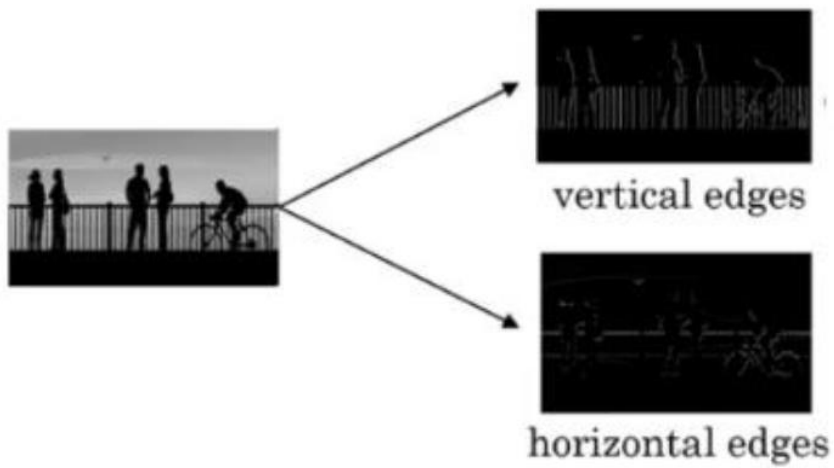
What We See

```
08 02 22 97 38 15 00 40 00 75 04 05 07 78 52 12 50 77 91 08
49 49 99 40 17 81 18 57 60 87 17 40 98 43 69 48 04 56 62 00
81 49 31 73 55 79 14 29 93 71 40 67 53 88 30 03 49 13 36 65
52 70 95 23 04 60 11 42 69 24 68 56 01 32 56 71 37 02 36 91
22 31 16 71 51 67 63 89 41 92 36 54 22 40 40 28 66 33 13 80
24 47 32 60 99 03 45 02 44 75 33 53 78 36 84 20 35 17 12 50
32 98 81 28 64 23 67 10 26 38 40 67 59 54 70 66 18 38 64 70
67 26 20 68 02 62 12 20 95 63 94 39 63 08 40 91 66 49 94 21
24 55 58 05 66 73 99 26 97 17 78 78 96 83 14 88 34 89 63 72
21 36 23 09 75 00 76 44 20 45 35 14 00 61 33 97 34 31 33 95
78 17 53 28 22 75 31 67 15 94 03 80 04 62 16 14 09 53 56 92
16 39 05 42 96 35 31 47 55 58 88 24 00 17 54 24 36 29 85 57
86 56 00 48 35 71 89 07 05 44 44 37 44 60 21 58 51 54 17 58
19 80 81 68 05 94 47 69 28 73 92 13 86 52 17 77 04 89 55 40
04 52 08 83 97 35 99 16 07 97 57 32 16 26 26 79 33 27 98 66
88 36 68 87 57 62 20 72 03 46 33 67 46 55 12 32 63 93 53 69
04 42 16 73 38 25 39 11 24 94 72 18 08 46 29 32 40 62 76 36
20 69 36 41 72 30 23 88 34 62 99 69 82 67 59 85 74 04 36 16
20 73 35 29 78 31 90 01 74 31 49 71 48 86 81 16 23 57 05 54
01 70 54 71 83 51 54 69 16 92 33 48 61 43 52 01 89 19 67 48
```

What Computers See

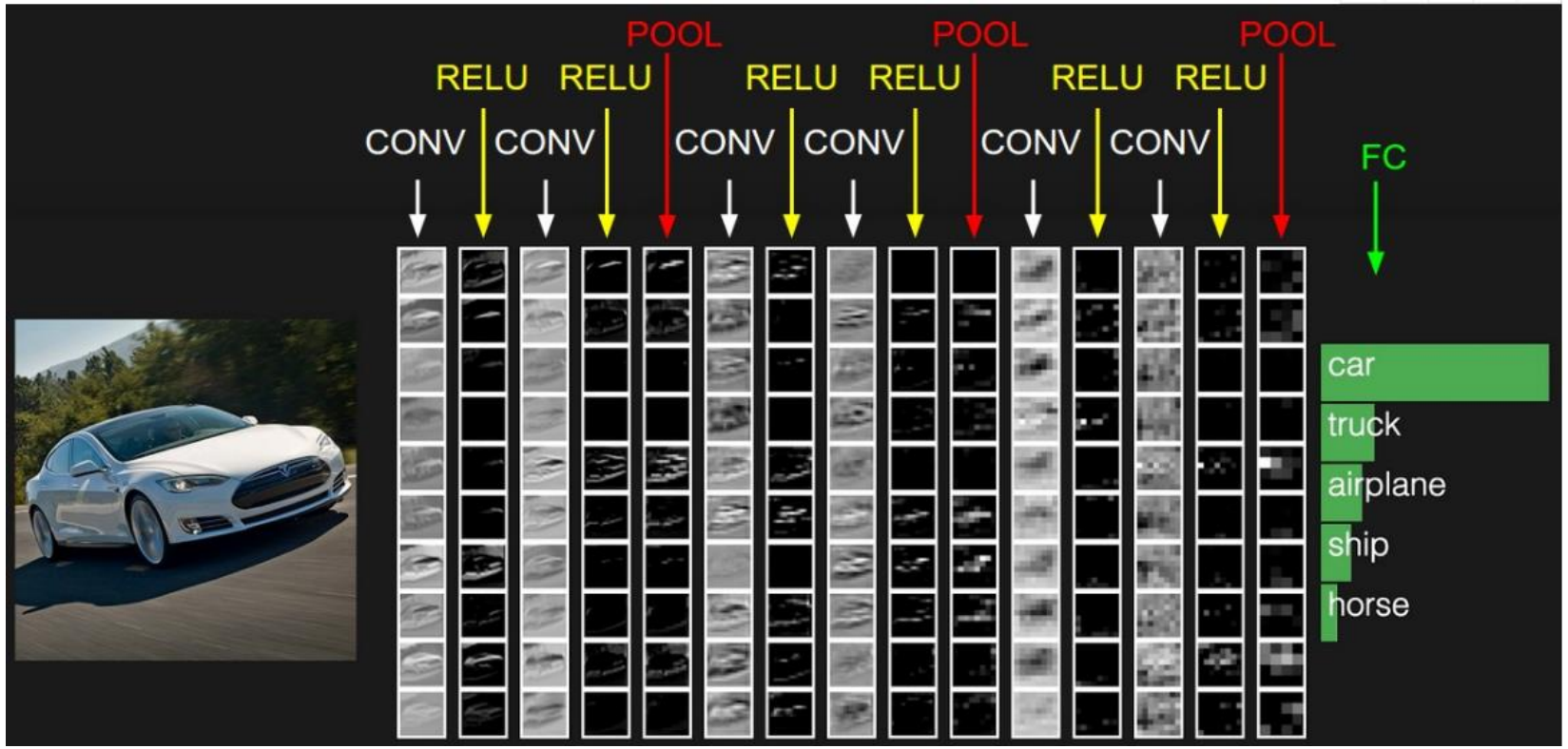
Edge Detection

there are many vertical and horizontal edges in the image. The first thing to do is to detect these edges:



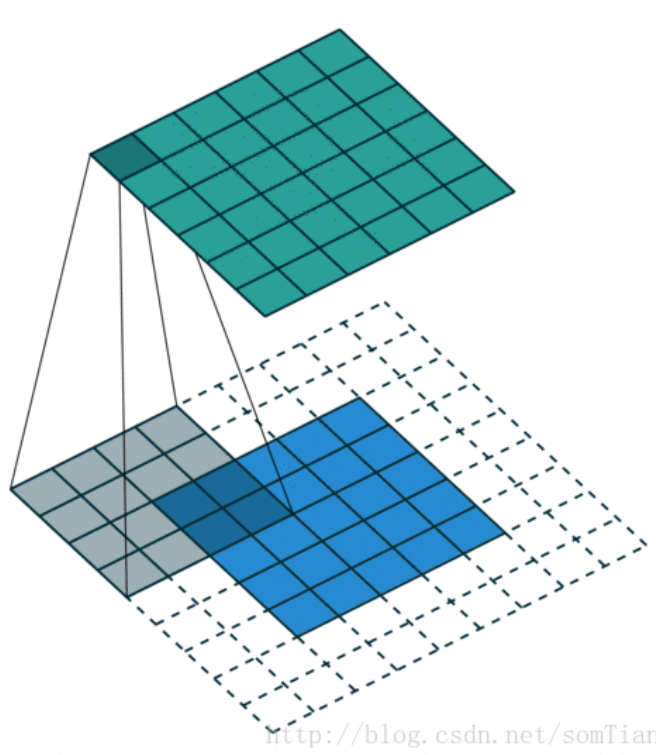

9

# A simple CNN structure



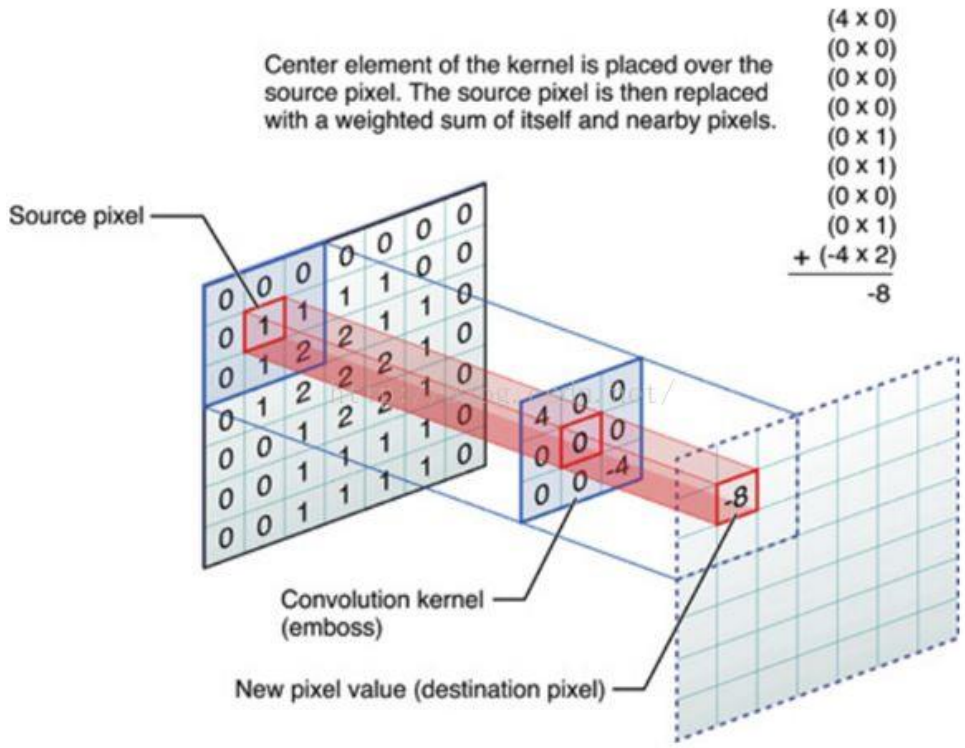

CONV: Convolutional kernel layer
RELU: Activation function
POOL: Dimension reduction layer
FC: Fully connection layer

# Convolutional kernel




**This is a gif image**

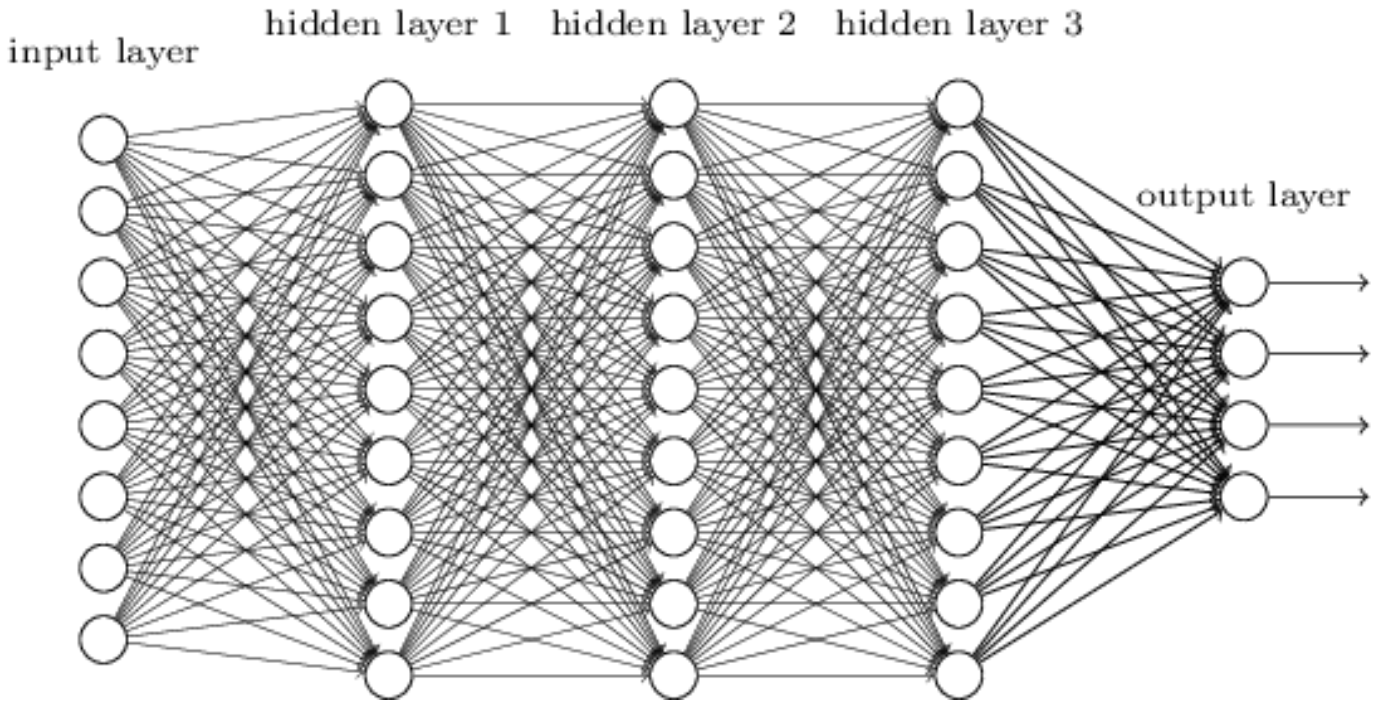
# Convolutional kernel




Padding on the input volume with zeros in such way that the conv layer does not alter the spatial dimensions of the input

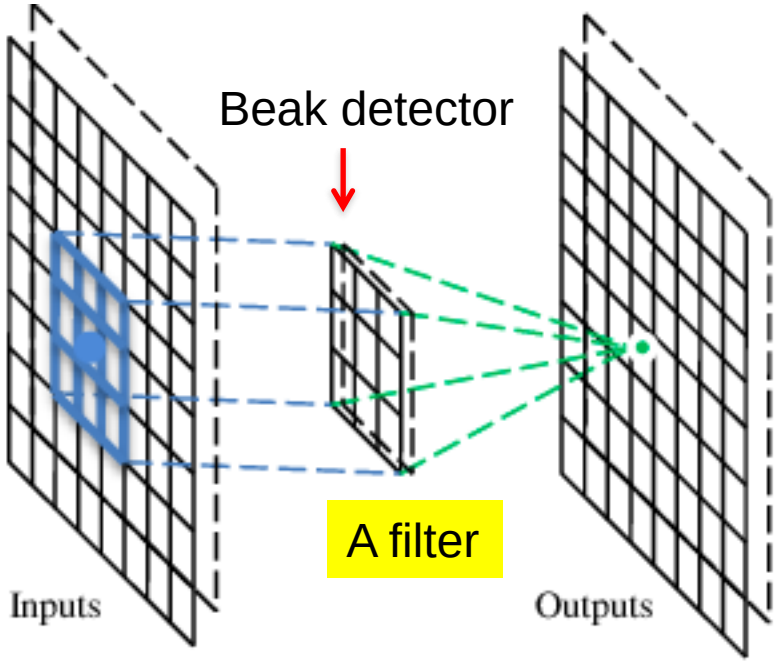
# Smaller Network: CNN

- 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?



# 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

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

| | | |
|---|---|---|
| -1 | 1 | -1 |
| -1 | 1 | -1 |
| -1 | 1 | -1 |

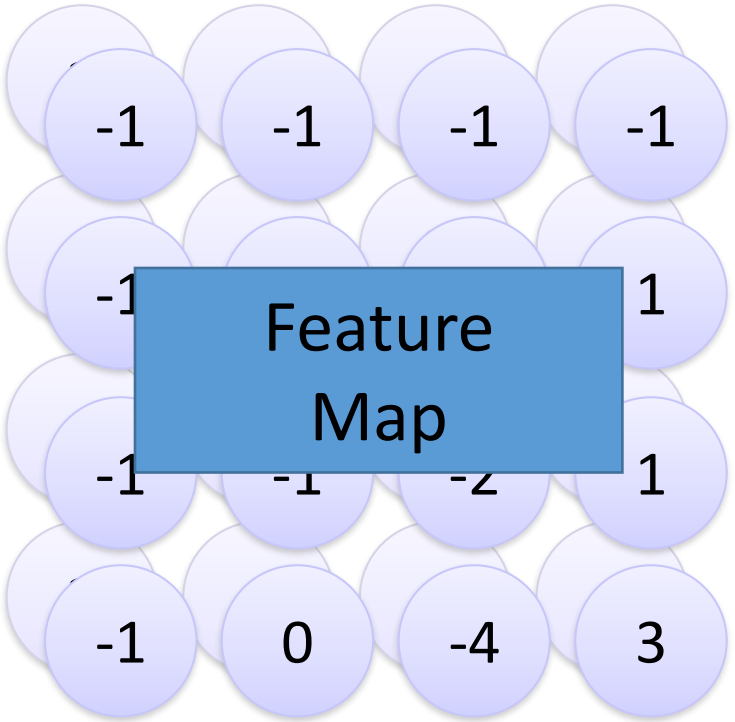
Filter 2

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 |

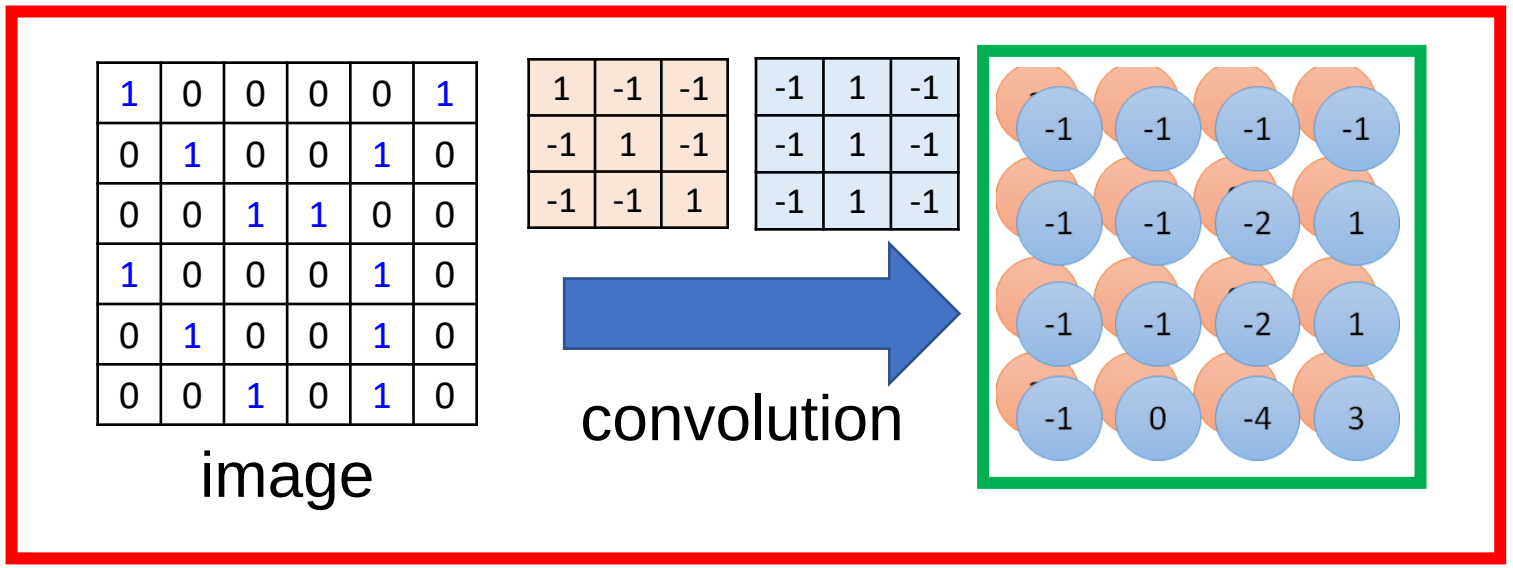
6 x 6 image

Repeat this for each filter



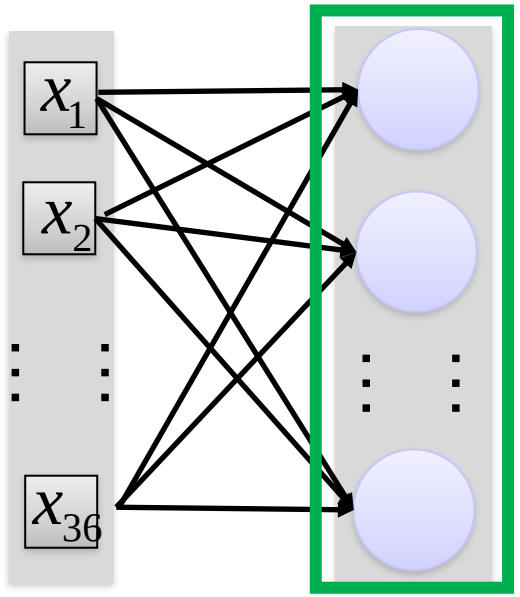

Two 4 x 4 images
Forming 2 x 4 x 4 matrix

# 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 |



Convolution of an image with different filters can perform operations such as edge detection, blur and sharpen by applying filters.

| Operation | Filter | Convolved Image |
|---|---|---|
| Identity | $\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$ |  |
| Edge detection | $\begin{bmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix}$ |  |
| | $\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$ |  |
| | $\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$ |  |
| Sharpen | $\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$ |  |
| Box blur (normalized) | $\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$ |  |
| Gaussian blur (approximation) | $\frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$ |  |

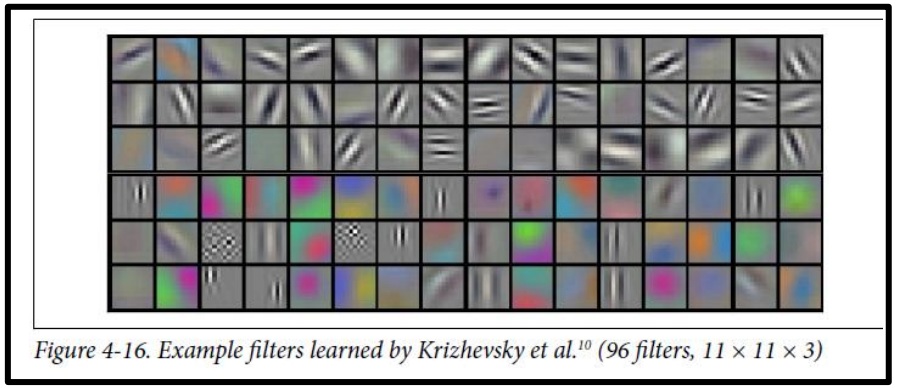


*Figure 4-16. Example filters learned by Krizhevsky et al.[10] (96 filters, 11 × 11 × 3)*

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

→

| | | | |
|---|---|---|---|
| -5 | -4 | 0 | 8 |
| -10 | -2 | 2 | 3 |
| 0 | -2 | -4 | -7 |
| -3 | -2 | -3 | -16 |

| | | | | | |
|---|---|---|---|---|---|
| 10 | 10 | 10 | 0 | 0 | 0 |
| 10 | 10 | 10 | 0 | 0 | 0 |
| 10 | 10 | 10 | 0 | 0 | 0 |
| 10 | 10 | 10 | 0 | 0 | 0 |
| 10 | 10 | 10 | 0 | 0 | 0 |
| 10 | 10 | 10 | 0 | 0 | 0 |

*

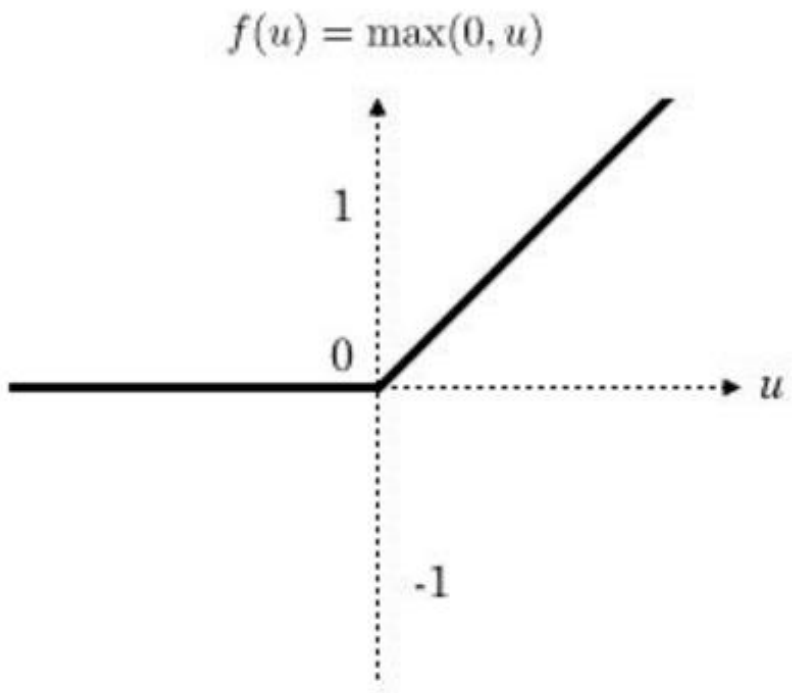
| | | |
|---|---|---|
| 1 | 0 | -1 |
| 1 | 0 | -1 |
| 1 | 0 | -1 |

=

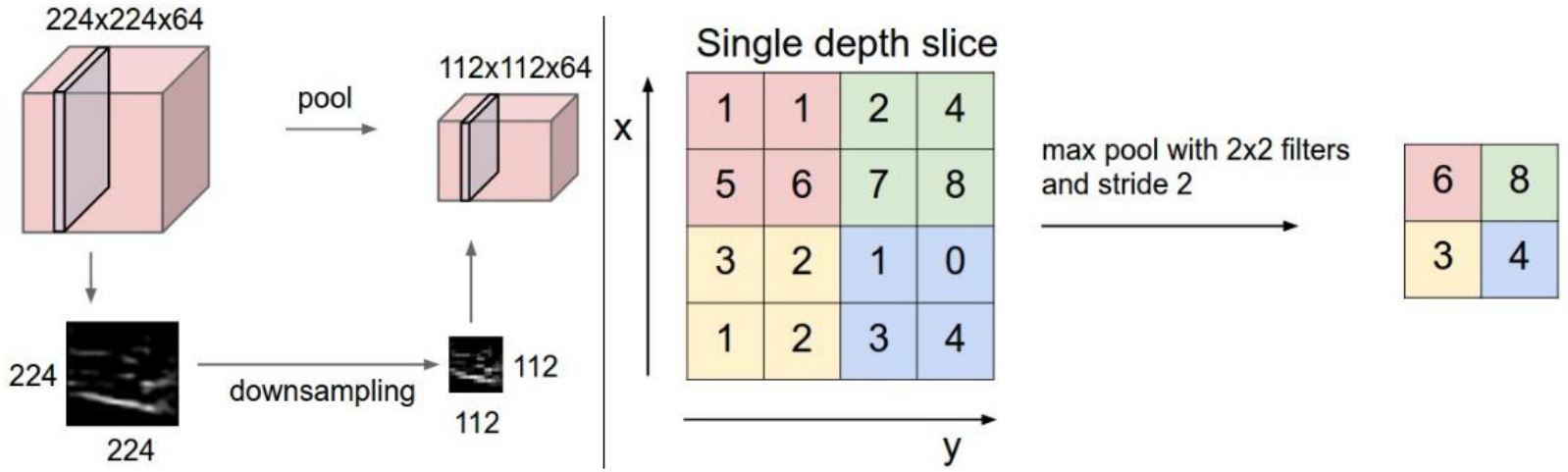
| | | | |
|---|---|---|---|
| 0 | 30 | 30 | 0 |
| 0 | 30 | 30 | 0 |
| 0 | 30 | 30 | 0 |
| 0 | 30 | 30 | 0 |

# Rectified linear unit， ReLU

rectified linear function， $f(x) = max(0, x)$



# Pooling layer




Pooling layer downsamples the volume spatially, independently in each depth slice of the input volume. **Left:** In this example, the input volume of size [224x224x64] is pooled with filter size 2, stride 2 into output volume of size [112x112x64]. Notice that the volume depth is preserved. **Right:** The most common downsampling operation is max, giving rise to **max pooling**, here shown with a stride of 2. That is, each max is taken over 4 numbers (little 2x2 square).

# Max Pooling

| | | |
|---|---|---|
| 1 | -1 | -1 |
| -1 | 1 | -1 |
| -1 | -1 | 1 |

Filter 1

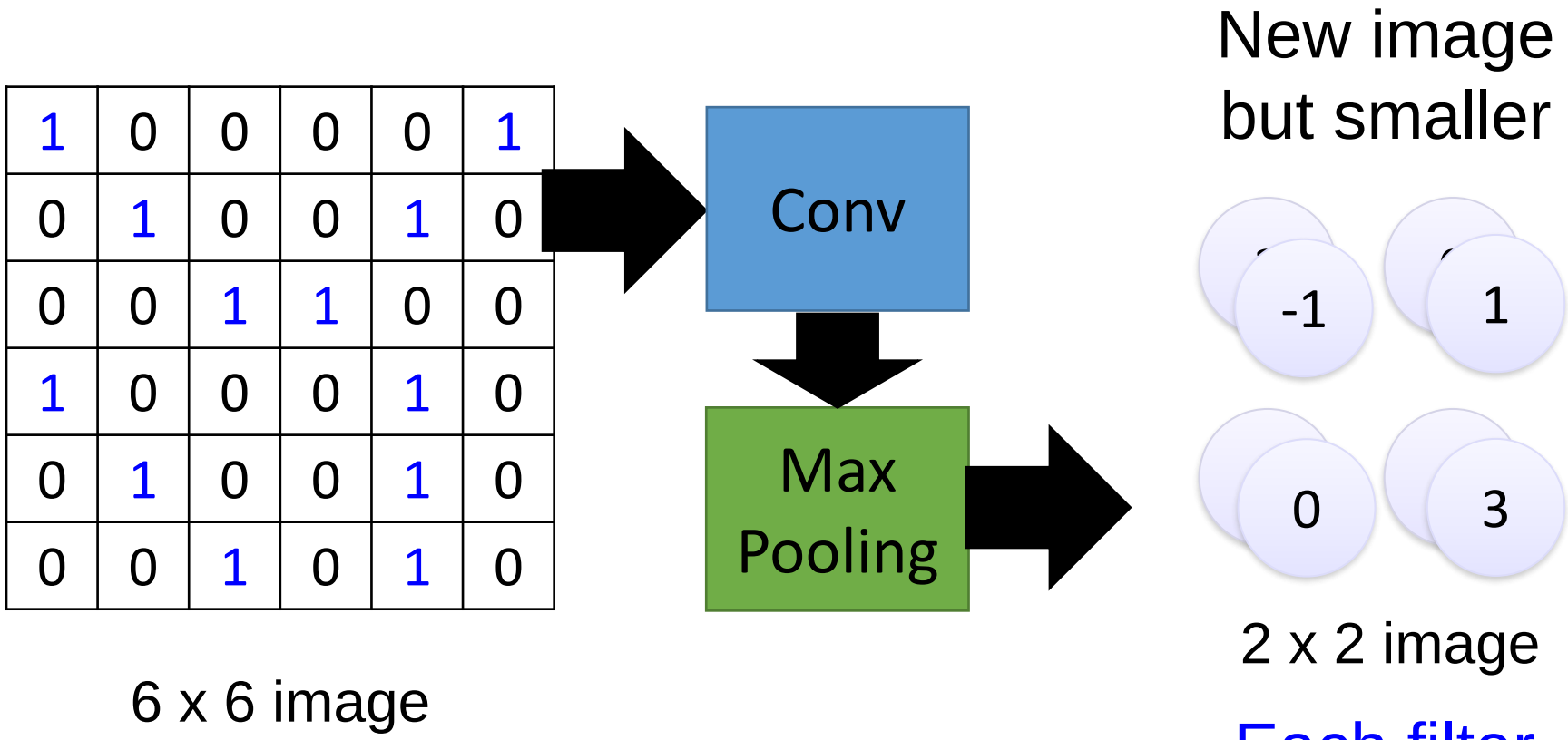
| | | |
|---|---|---|
| -1 | 1 | -1 |
| -1 | 1 | -1 |
| -1 | 1 | -1 |

Filter 2

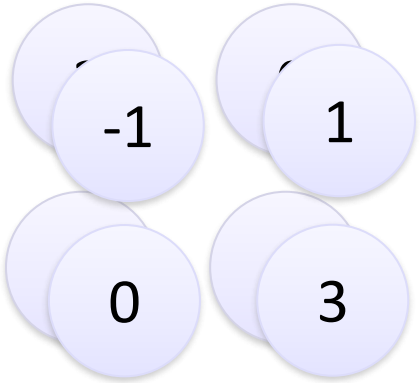
| | | | |
|---|---|---|---|
| 3 | -1 | -3 | -1 |
| -3 | 1 | 0 | -3 |
| -3 | -3 | 0 | 1 |
| 3 | -2 | -2 | -1 |

| | | | |
|---|---|---|---|
| -1 | -1 | -1 | -1 |
| -1 | -1 | -2 | 1 |
| -1 | -1 | -2 | 1 |
| -1 | 0 | -4 | 3 |

# 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
Conv
Max Pooling
New image but smaller
-1
1
0
3
6 x 6 image
2 x 2 image
Each filter is a channel

# The whole CNN




A new image

Convolution

Max Pooling

Convolution

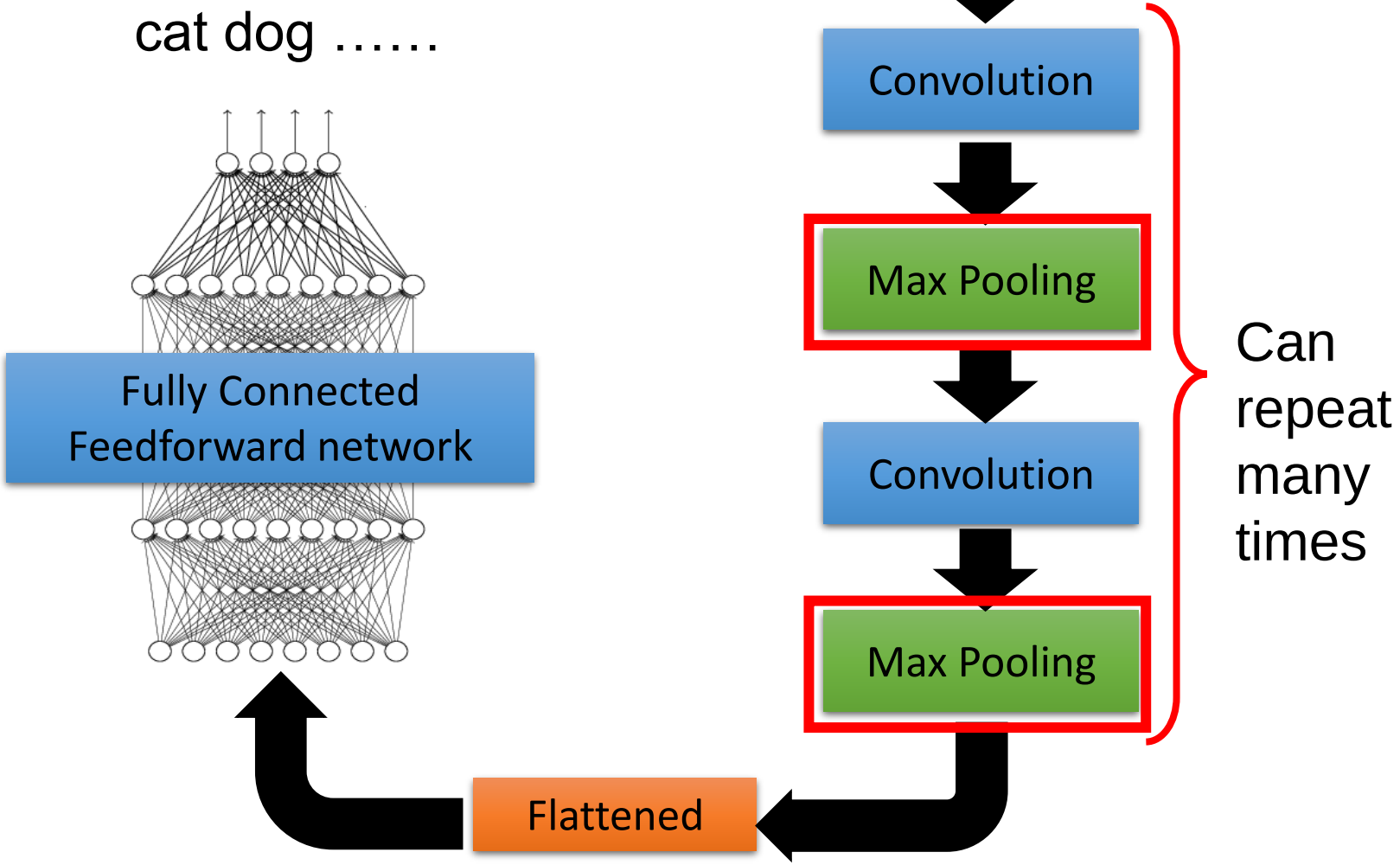
Max Pooling

Can repeat many times

Smaller than the original image

The number of channels is the number of filters

# The whole CNN



# The whole CNN

cat dog ……

Convolution

Max Pooling

A new image

Convolution

Max Pooling

A new image

Flattened

Fully Connected Feedforward network

# Applications

Convolutional neural network (ConvNets or CNNs) is one of the main categories to do images recognition, images classifications. Objects detections, recognition faces etc., are some of the areas where CNNs are widely used.

# AlphaGo

19 x 19 matrix

Black: 1

white: -1

none: 0

Neural Network

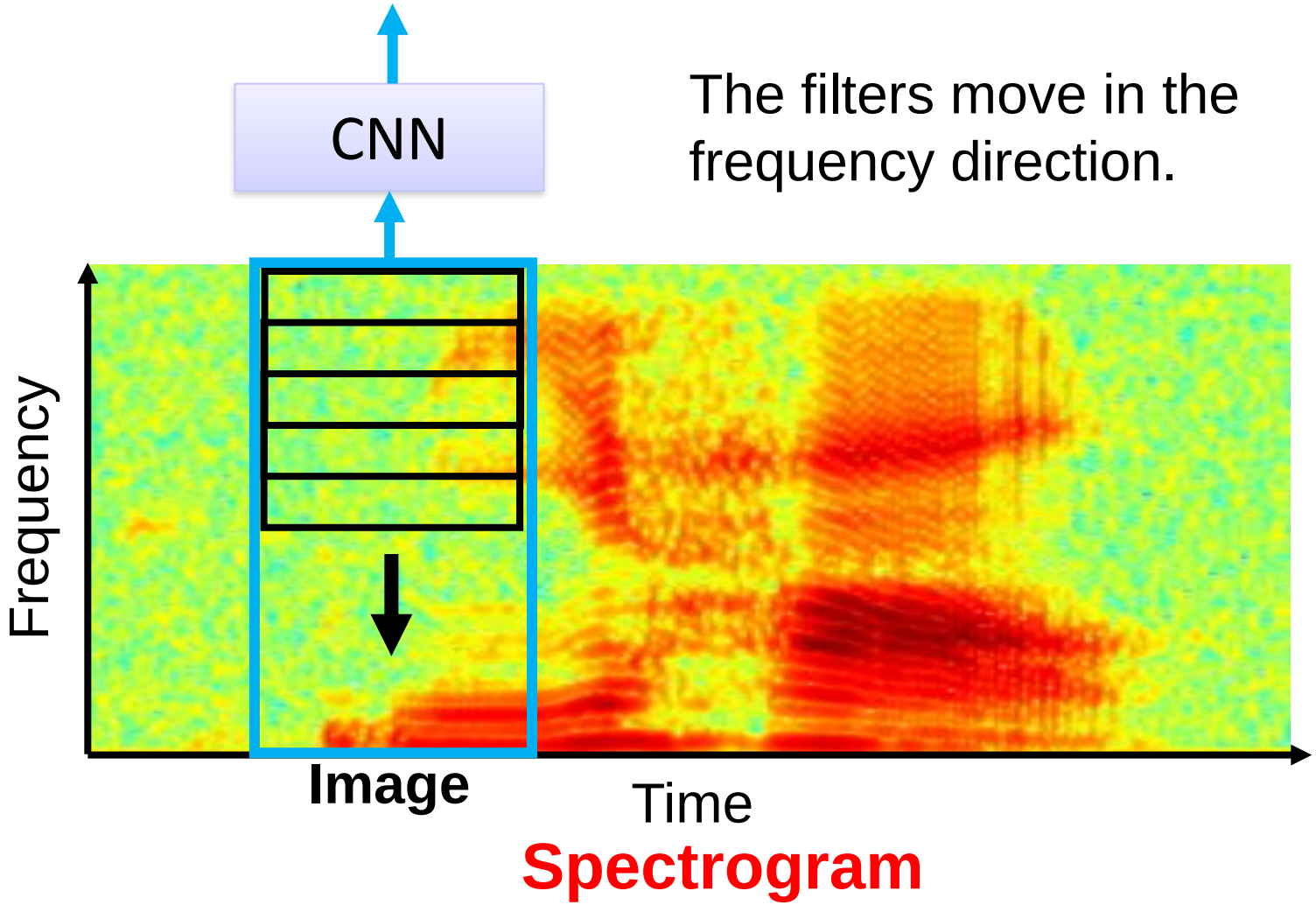
Next move

(19 x 19 positions)

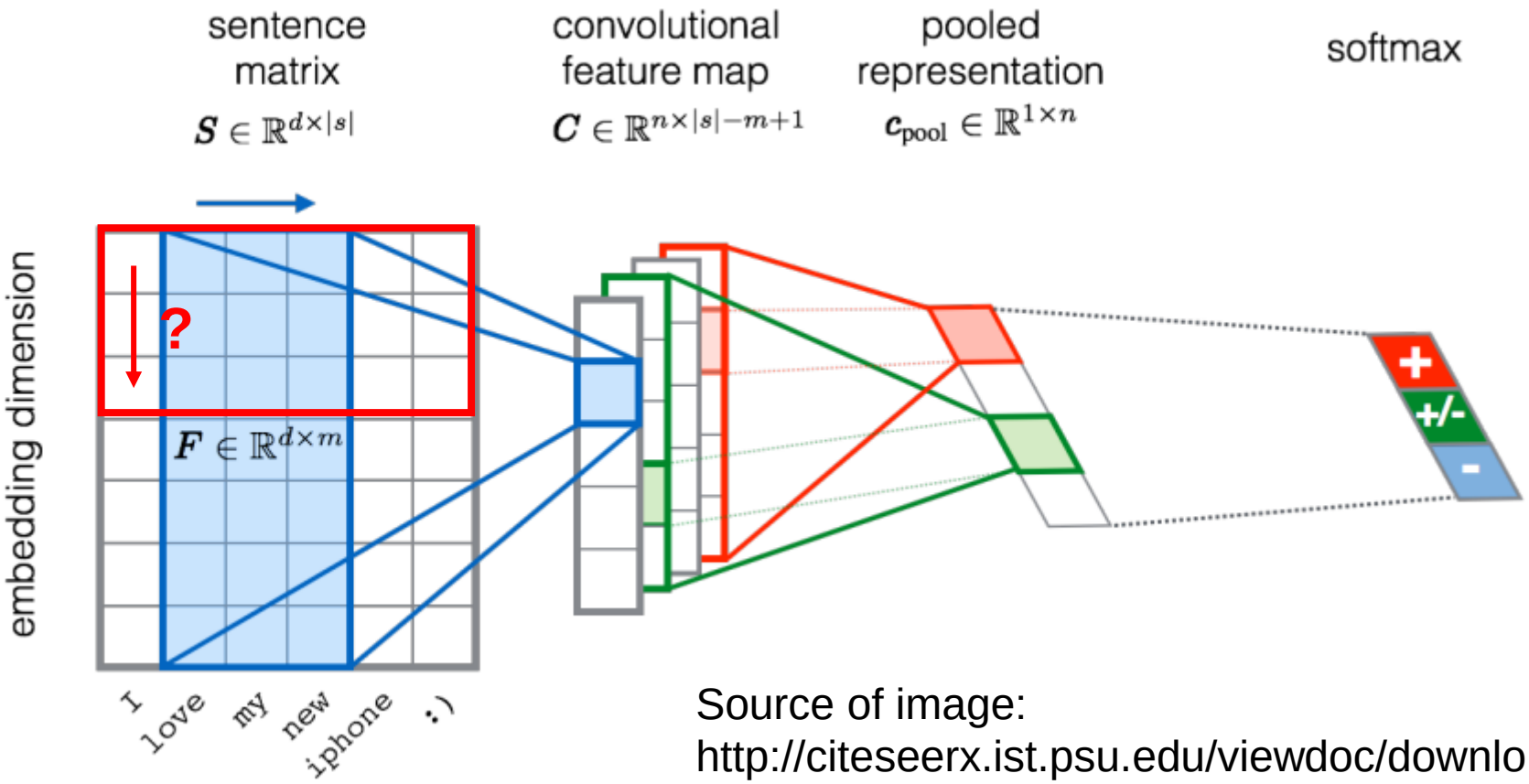
Fully-connected feedforward network can be used

But CNN performs much better

# CNN in speech recognition



# CNN in text classification

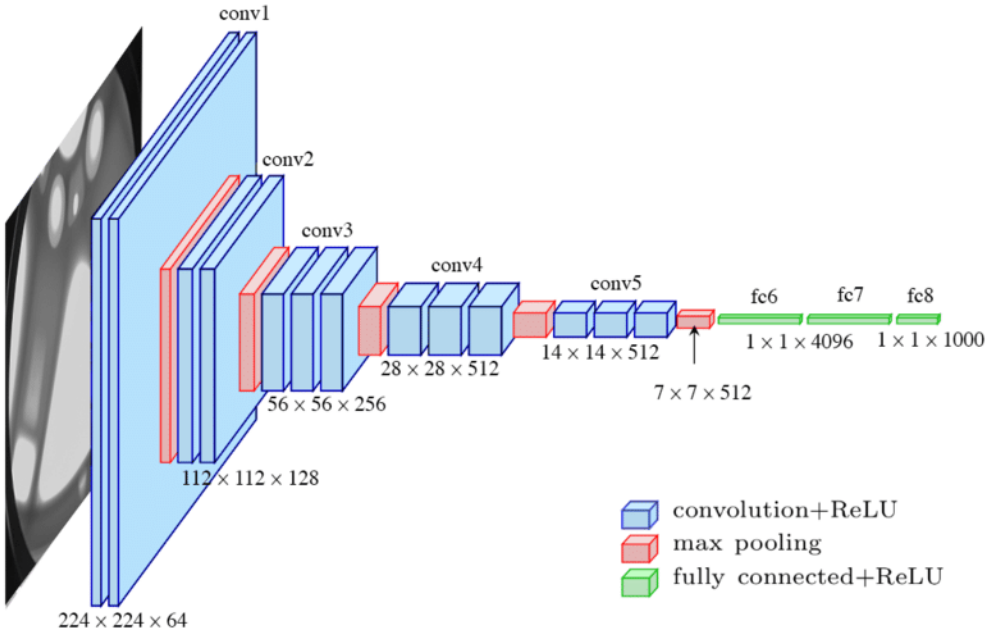



Source of image: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.703.6858&rep=rep1&type=pdf

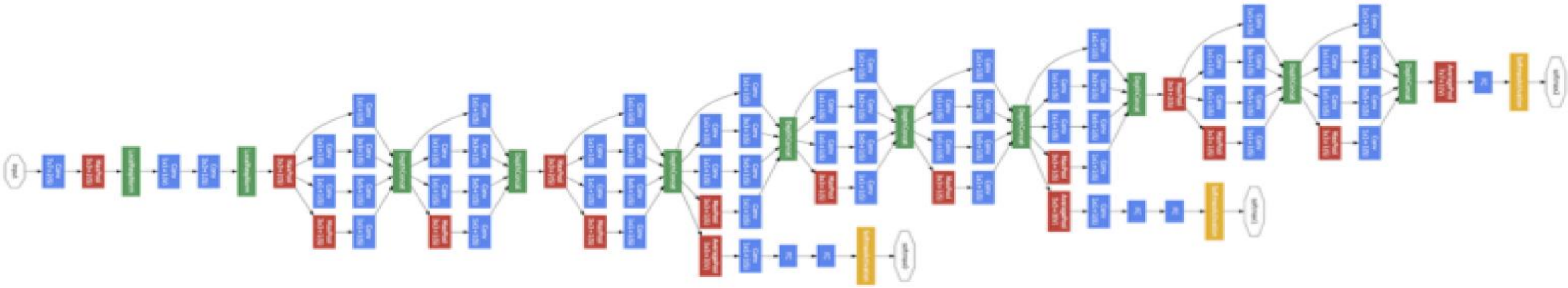
- LeNet[18]
  - One of the earliest successful architectures of CNNs
  - Developed by Yann Lecun
  - Originally used to read digits in images
- AlexNet[19]
  - Helped popularize CNNs in computer vision
  - Developed by Alex Krizhevsky, Ilya Sutskever, and Geoff Hinton
  - Won the ILSVRC 2012
- ZF Net[20]
  - Won the ILSVRC 2013
  - Developed by Matthew Zeiler and Rob Fergus
  - Introduced the visualization concept of the Deconvolutional Network
- GoogLeNet[21]
  - Won the ILSVRC 2014
  - Developed by Christian Szegedy and his team at Google
  - Codenamed "Inception," one variation has 22 layers
- VGGNet[22]
  - Runner-Up in the ILSVRC 2014
  - Developed by Karen Simonyan and Andrew Zisserman
  - Showed that depth of network was a critical factor in good performance
- ResNet[23]
  - Trained on very deep networks (up to 1,200 layers)
  - Won first in the ILSVRC 2015 classification task

# VGGNet

- The runner-up in ILSVRC 2014 was the network from Karen Simonyan and Andrew Zisserman that became known as the VGGNet.
- Their final best network contains 16 CONV/FC layers and, appealingly, features an extremely homogeneous architecture that only performs 3x3 convolutions and 2x2 pooling from the beginning to the end.
- A downside of the VGGNet is that it is more expensive to evaluate and uses a lot more memory and parameters (140M).
- Most of these parameters are in the first fully connected layer, and it was since found that these FC layers can be removed with no performance downgrade, significantly reducing the number of necessary parameters.



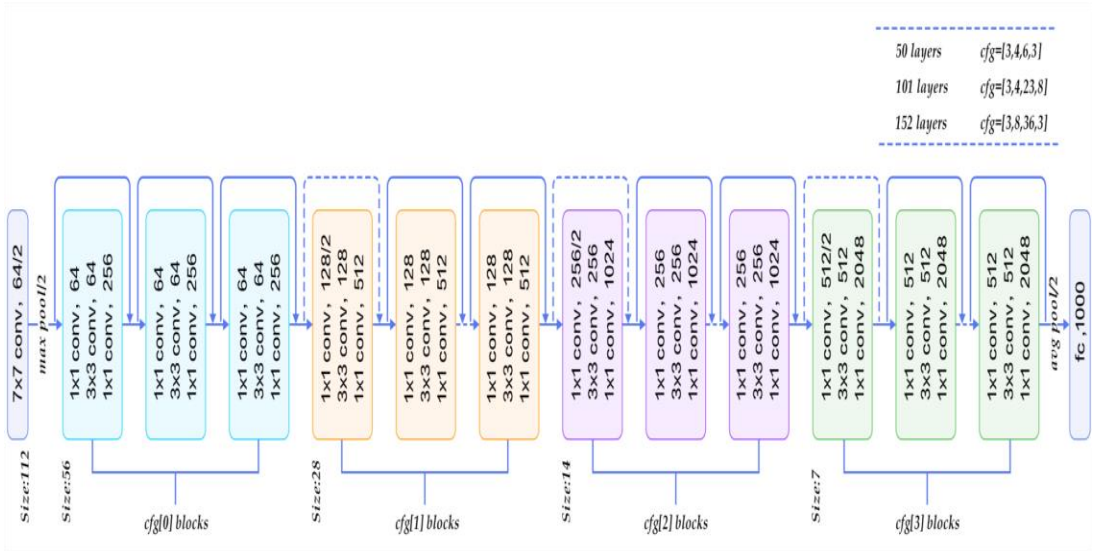
# Inception(GoogLeNet)

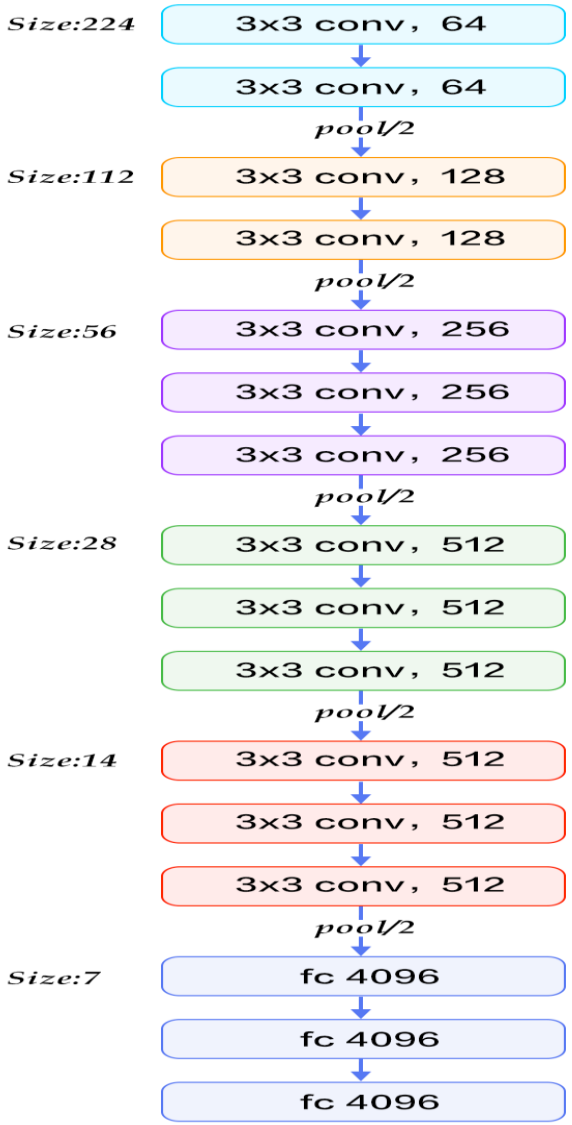


Convolution
Pooling
Softmax
Other

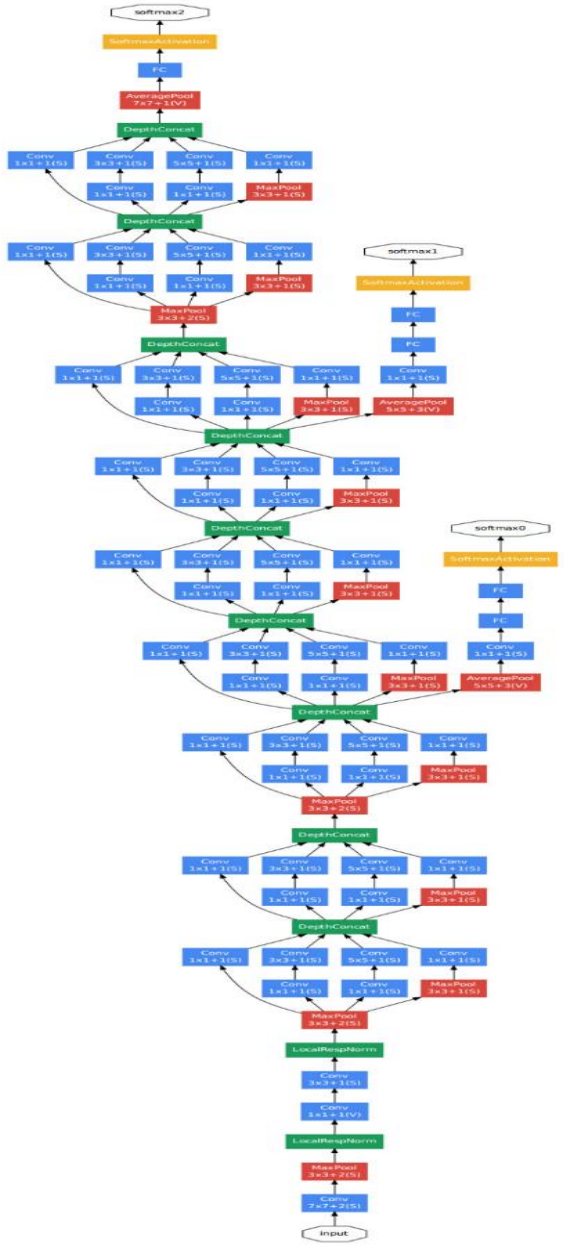
# ResNet

- Residual Network developed by Kaiming He et al. was the winner of ILSVRC 2015.
- It features special *skip connections* and a heavy use of batch normalization.
- The architecture is also missing fully connected layers at the end of the network.

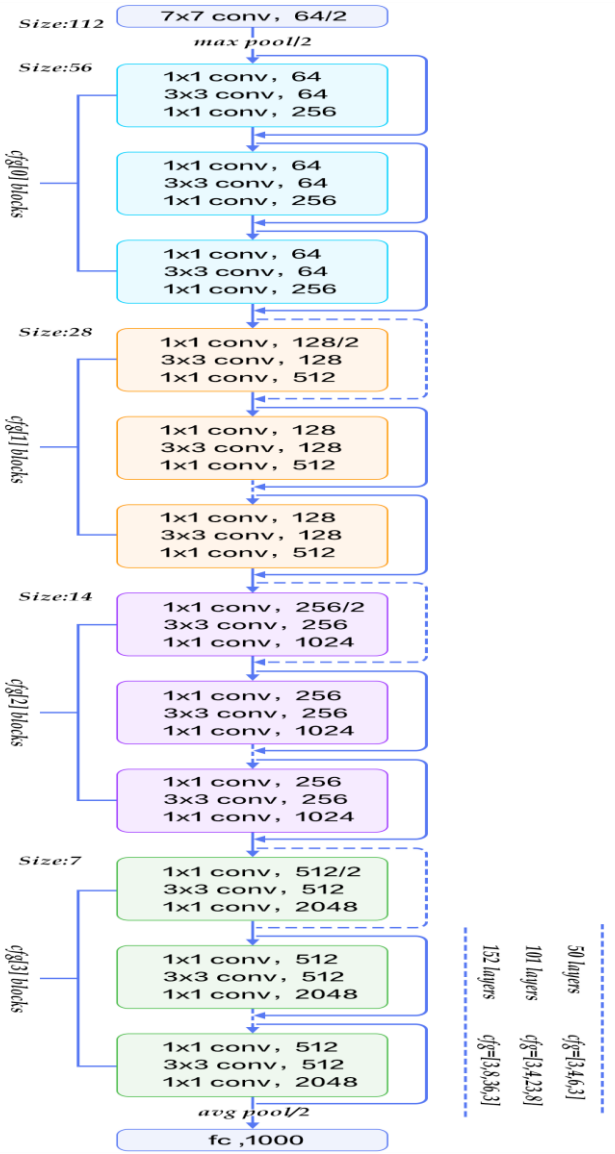





VGG-16



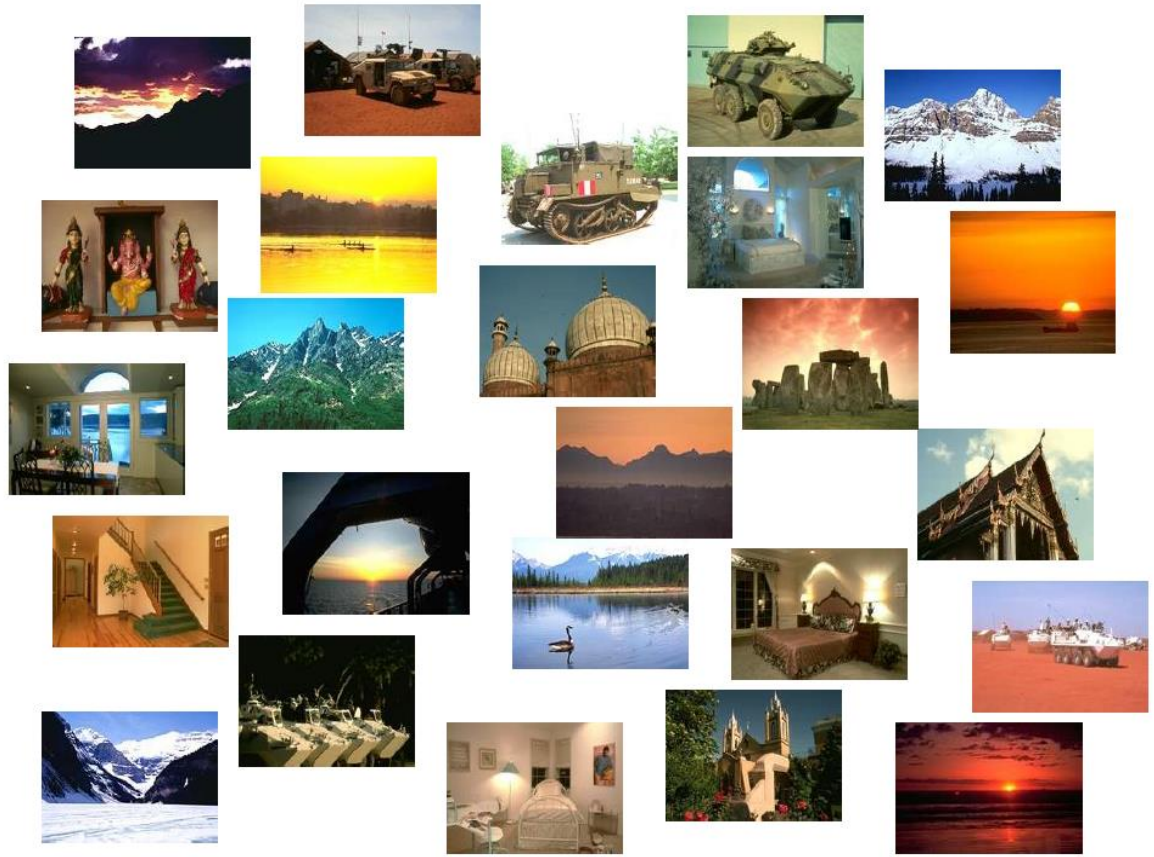

GoogleNet




ResNet

# Scene Image Classification

# Scene Image Clustering



# Scene Image Clustering

Residential Interiors

Mountains

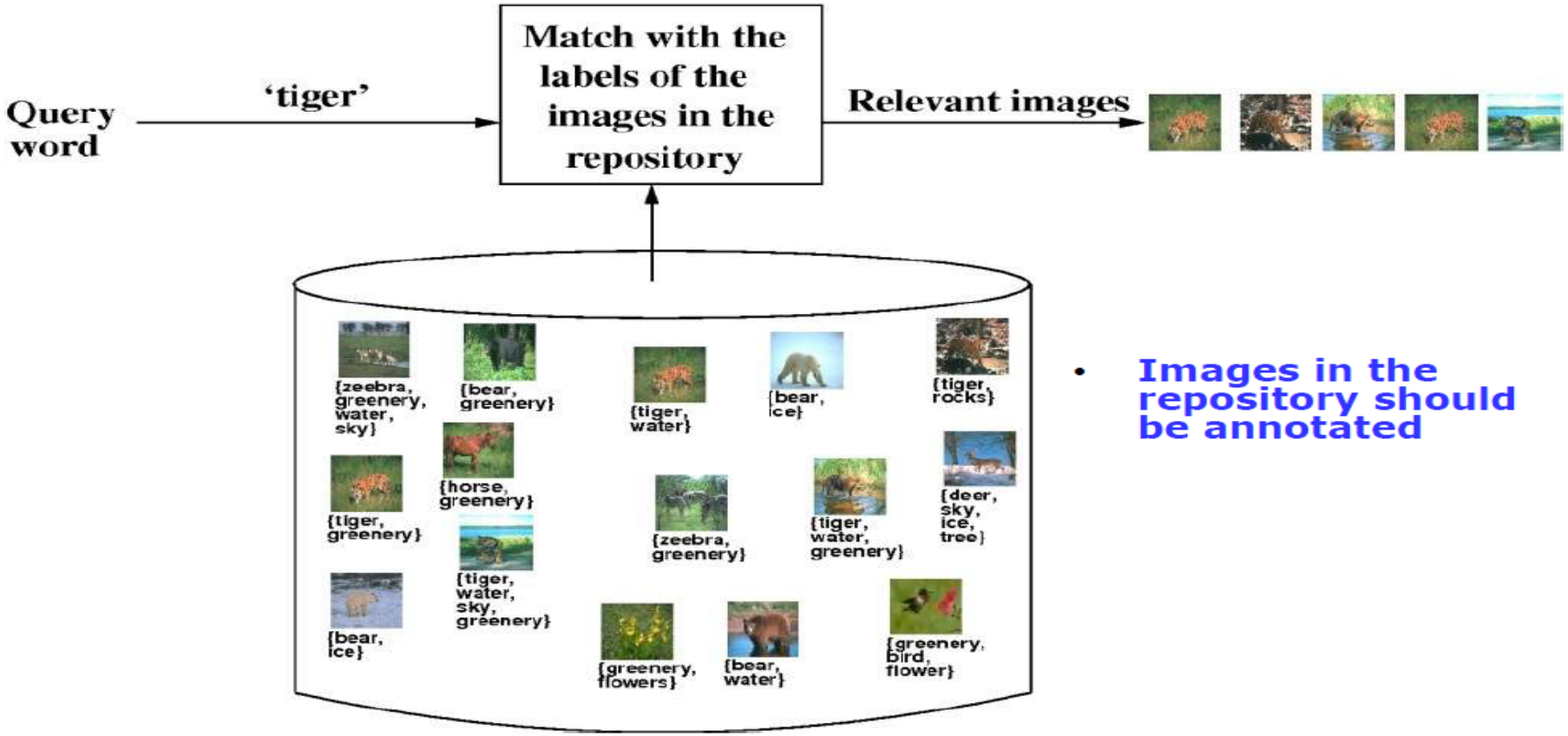
Military Vehicles

Sacred Places

Sunsets & Sunrises

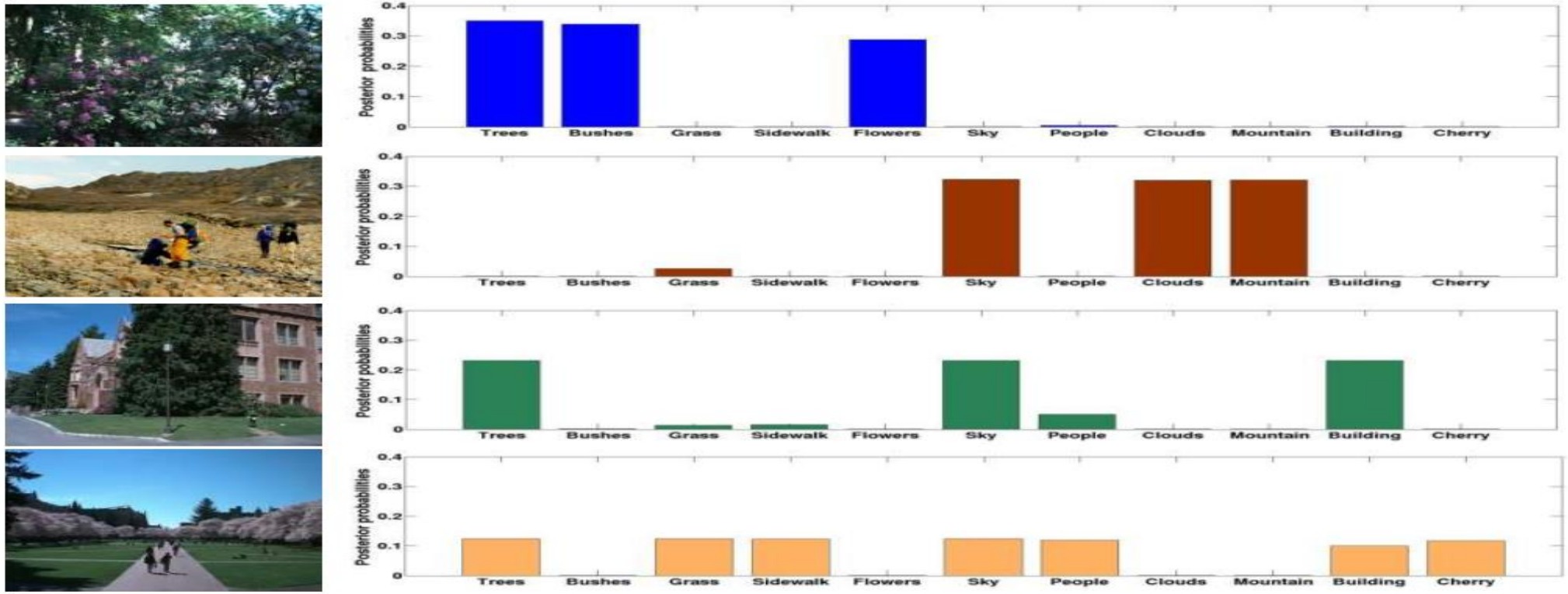
# Content based Image Retrieval

- **Query-by-semantics (QBS) Approach**




- **Images in the repository should be annotated**

# Intermediate Scene Representation

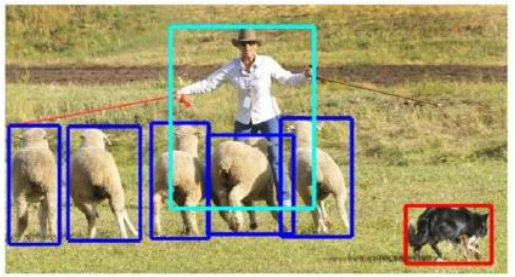


# Object detection

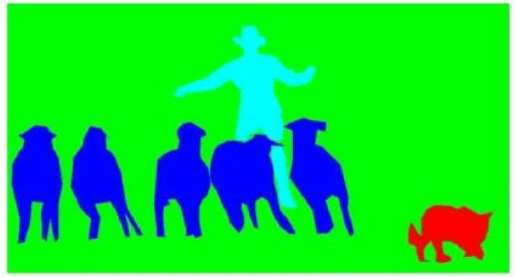
- Object detection is a computer technology related to computer vision and image processing that deals with detecting instances of semantic objects of a certain class (such as humans, buildings, or cars) in digital images and videos (Wikipedia)

# Instance Segmentation Examples


(a) Image classification



(b) Object localization



(c) Semantic segmentation

(d) This work

Microsoft COCO: Common Objects in Context

# Object detection

## Object Detection as Classification

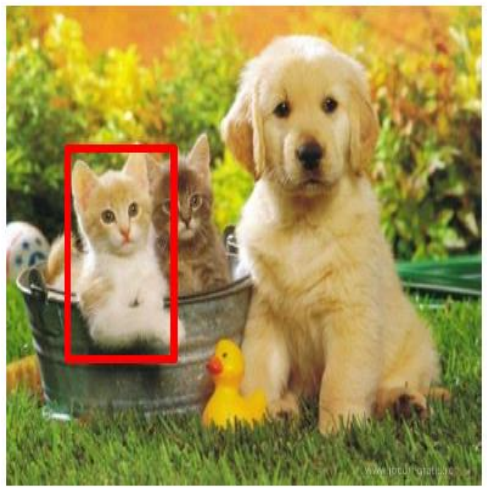
Classes = [cat, dog, duck]

Cat ? NO

Dog ? NO

Duck? NO

# Object Detection as Classification

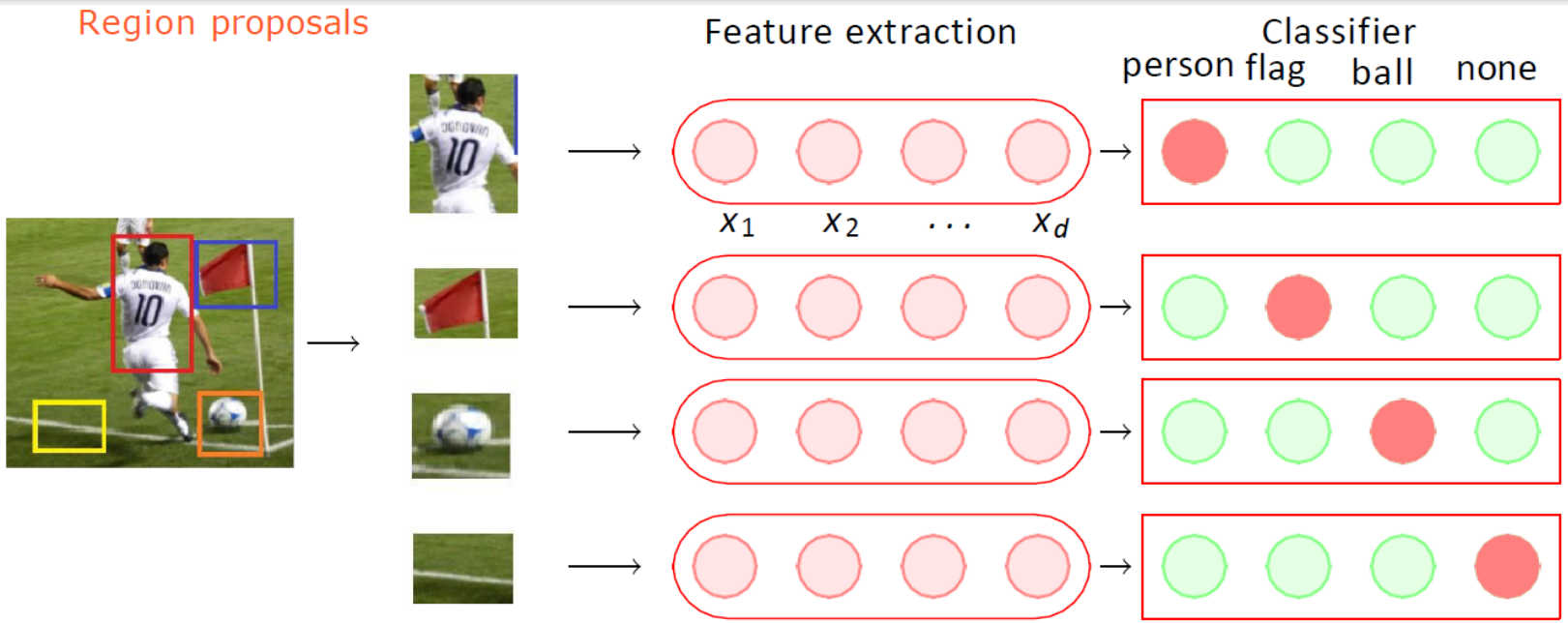


Classes = [cat, dog, duck]

Cat ? YES

Dog ? NO

Duck? NO




- Let us see a typical pipeline for *object detection*
- It starts with a region proposal stage where we identify potential regions which may contain objects
- We could think of these regions as mini-images

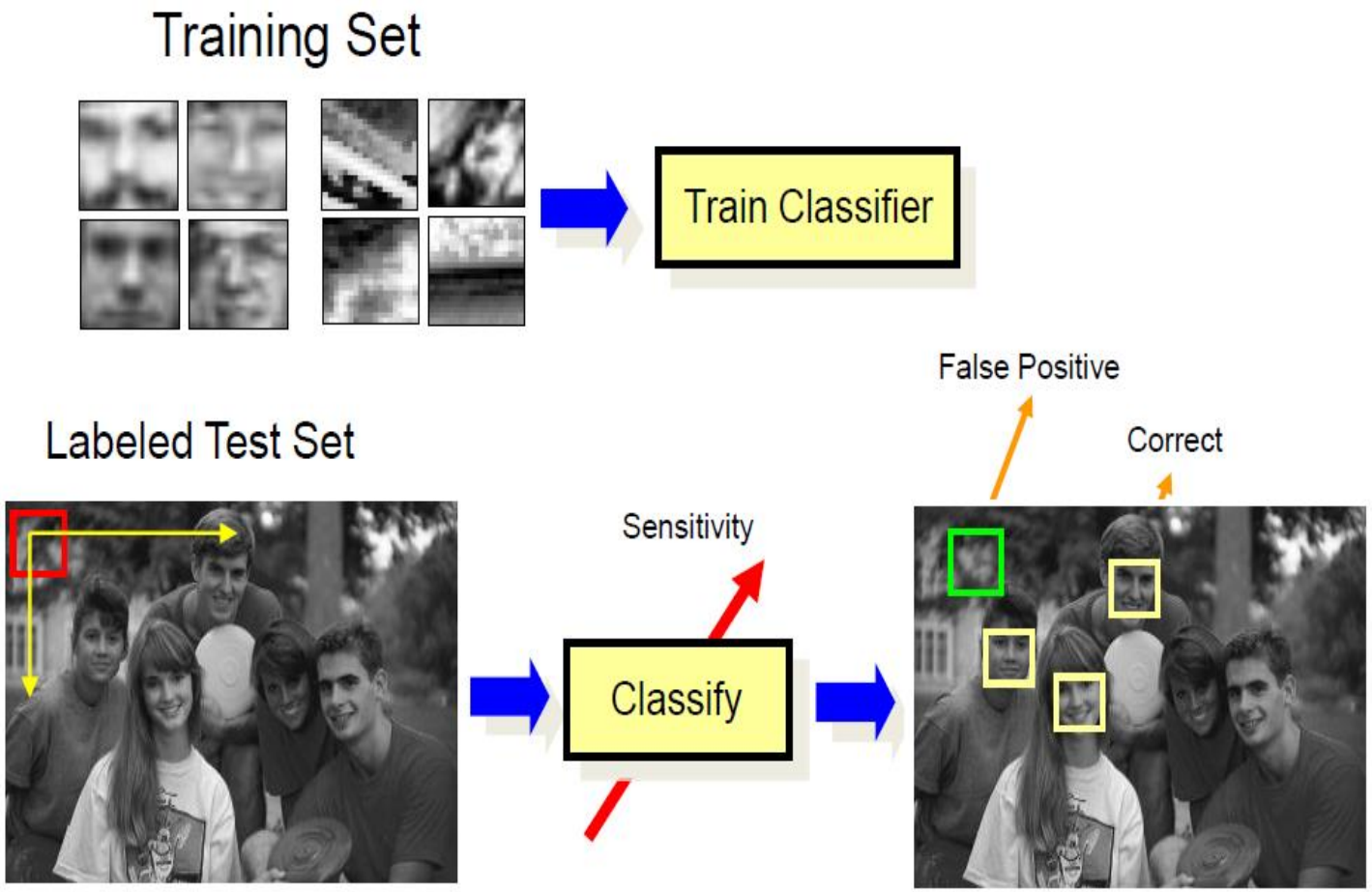
6/47

# General Problems of Recognition

Invariance:
- "External parameters"
  - Pose
  - Illumination
- "Internal parameters"
  - Person identity
  - Facial expression

Applicable to many classes of objects

# Training and Testing

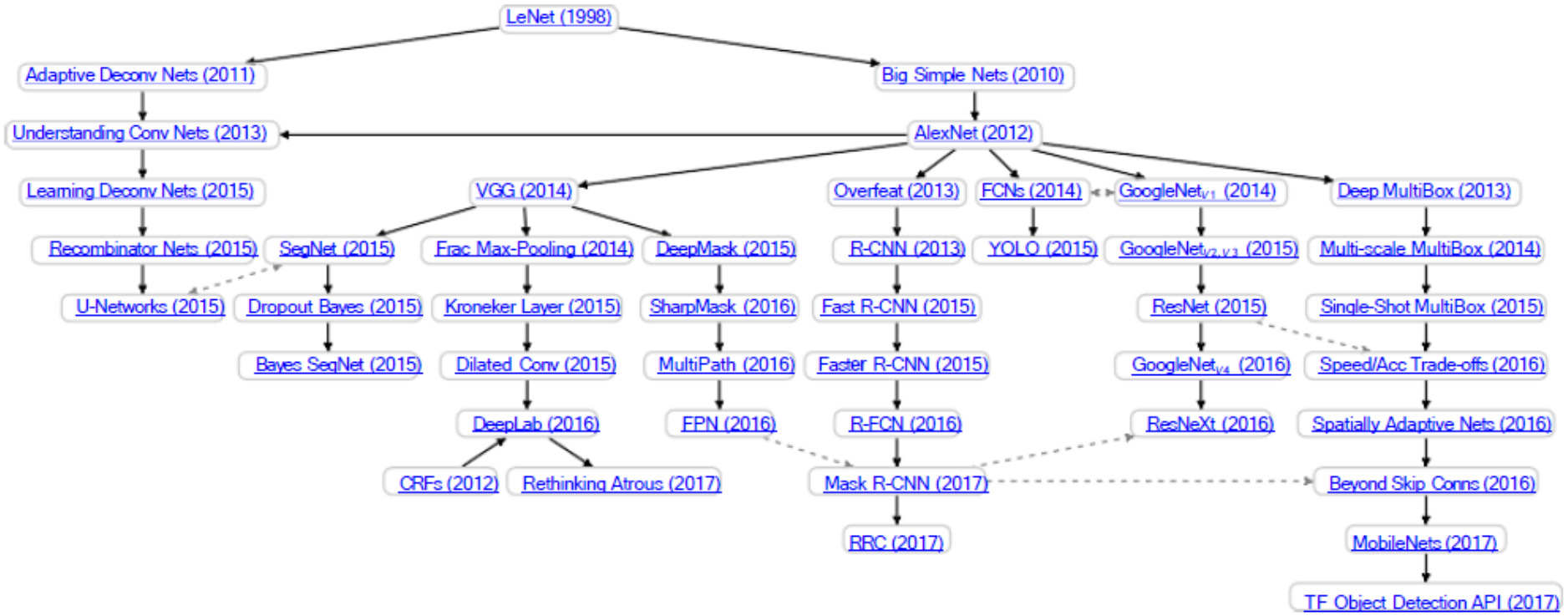


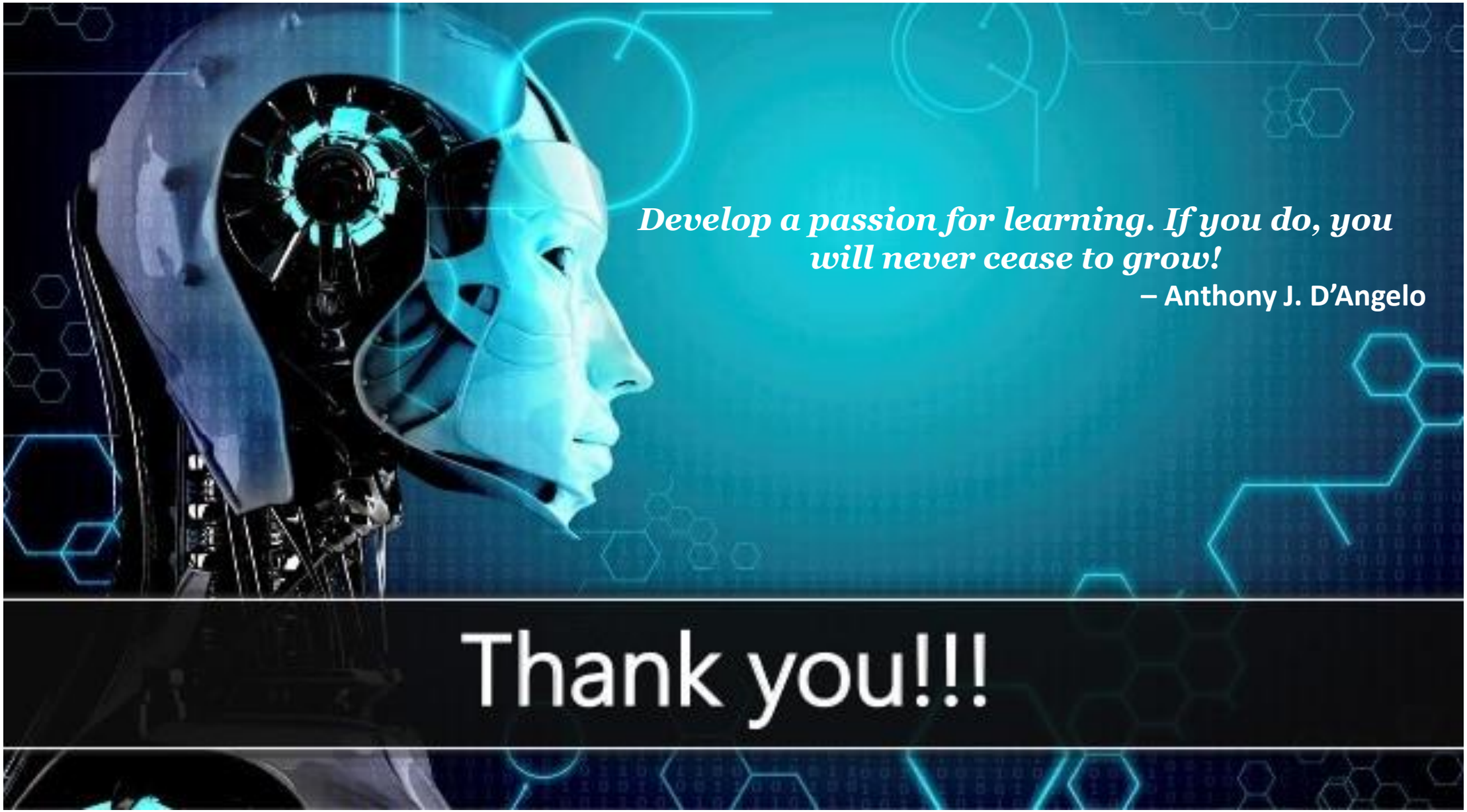
# Datasets

| Name | # Images (trainval) | # Classes | Last updated |
|---|---|---|---|
| ImageNet | 450k | 200 | 2015 |
| COCO | 120K | 80 | 2014 |
| Pascal VOC | 12k | 20 | 2012 |
| Oxford-IIIT Pet | 7K | 37 | 2012 |
| KITTI Vision | 7K | 3 | 2014 |

# Detection and Segmentation Atlas

There are too many outstanding contributions to cover in a single deck. Here is a brief high-level overview intended to provide broader context and help guide additional exploration.




*Develop a passion for learning. If you do, you will never cease to grow!*
– Anthony J. D'Angelo
Thank you!!!