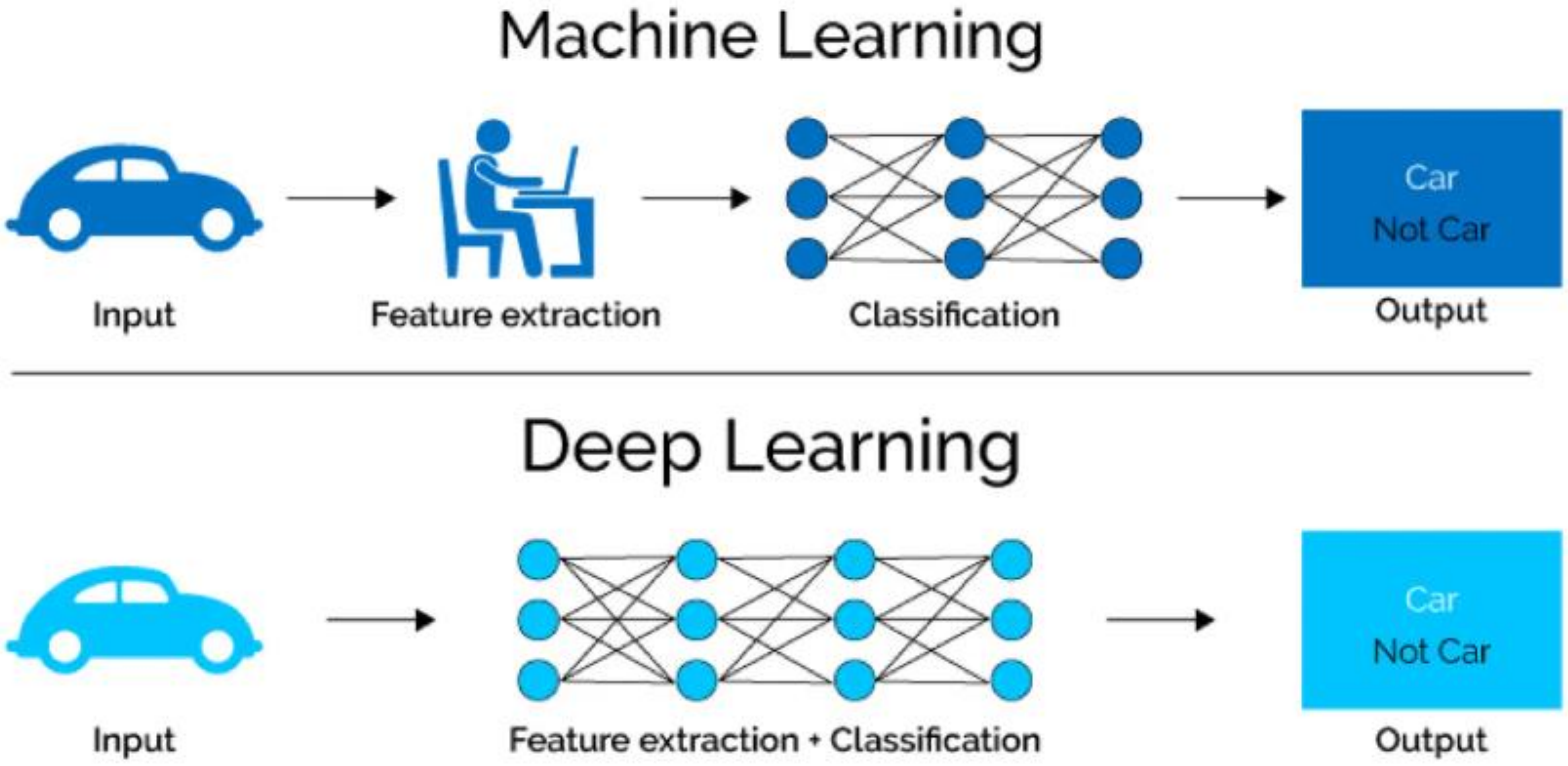
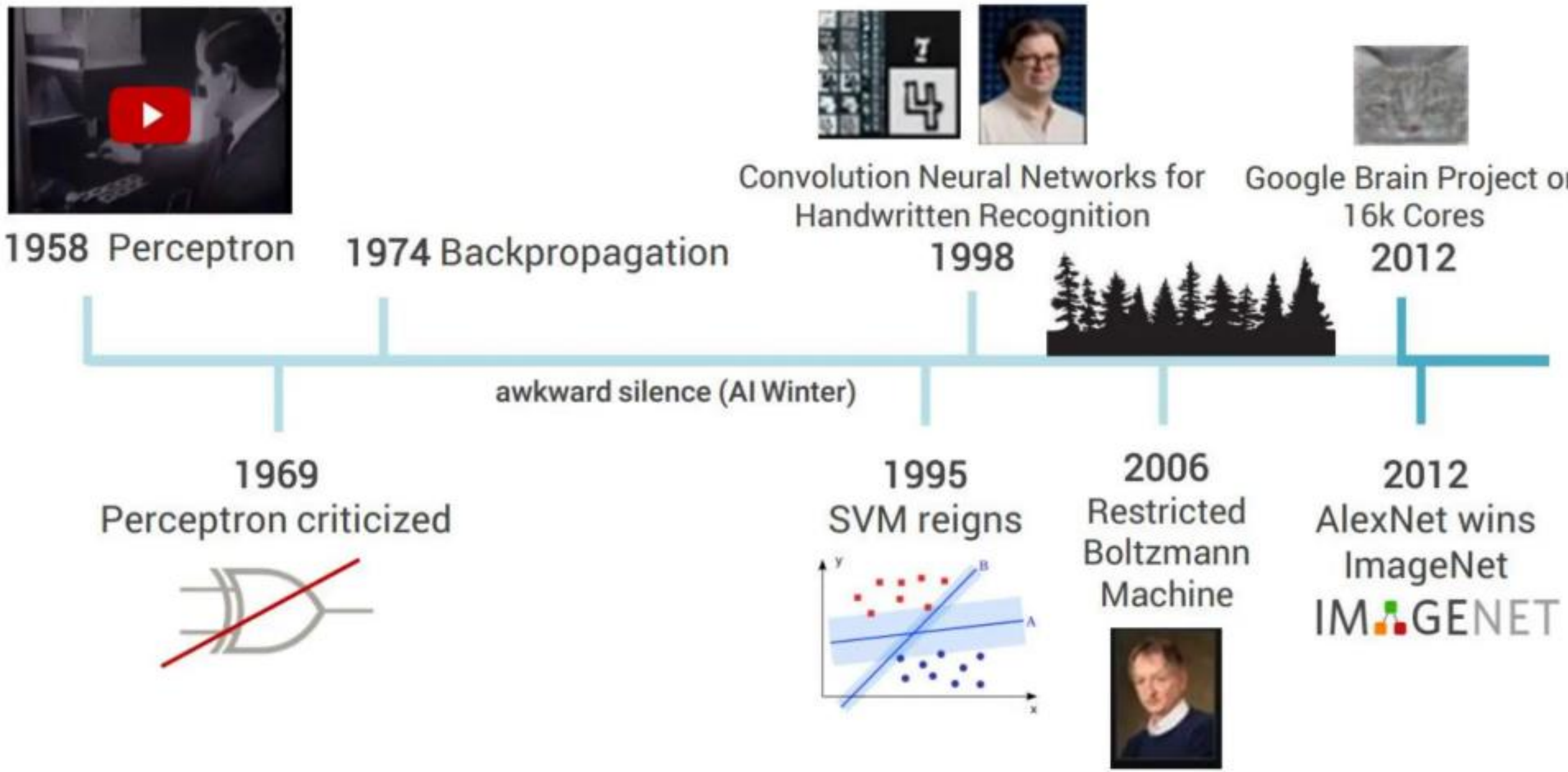
# Deep vs Machine Learning



# Why we needs Deep Learning?

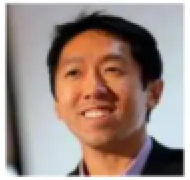
- **SuperIntelligent** Devices
- Best Solution for
    - image recognition
    - speech recognition
    - natural language processing
    - Big Data


1958 Perceptron
1969
Perceptron criticized
1974 Backpropagation
awkward silence (AI Winter)
1995
SVM reigns
Convolution Neural Networks for
Handwritten Recognition
1998
2006
Restricted
Boltzmann
Machine
Google Brain Project on
16k Cores
2012
2012
AlexNet wins
ImageNet
IMAGENET

**Geoffrey Hinton:** University of Toronto & Google

**Yann LeCun:** New York University & Facebook

**Andrew Ng:** Stanford & Baidu



## Deep Learning Requirements

- ❑Large data set with good quality
- ❑Measurable and describable goals
- ❑Enough computing power
- ❑Neural Network (Brain of Human)

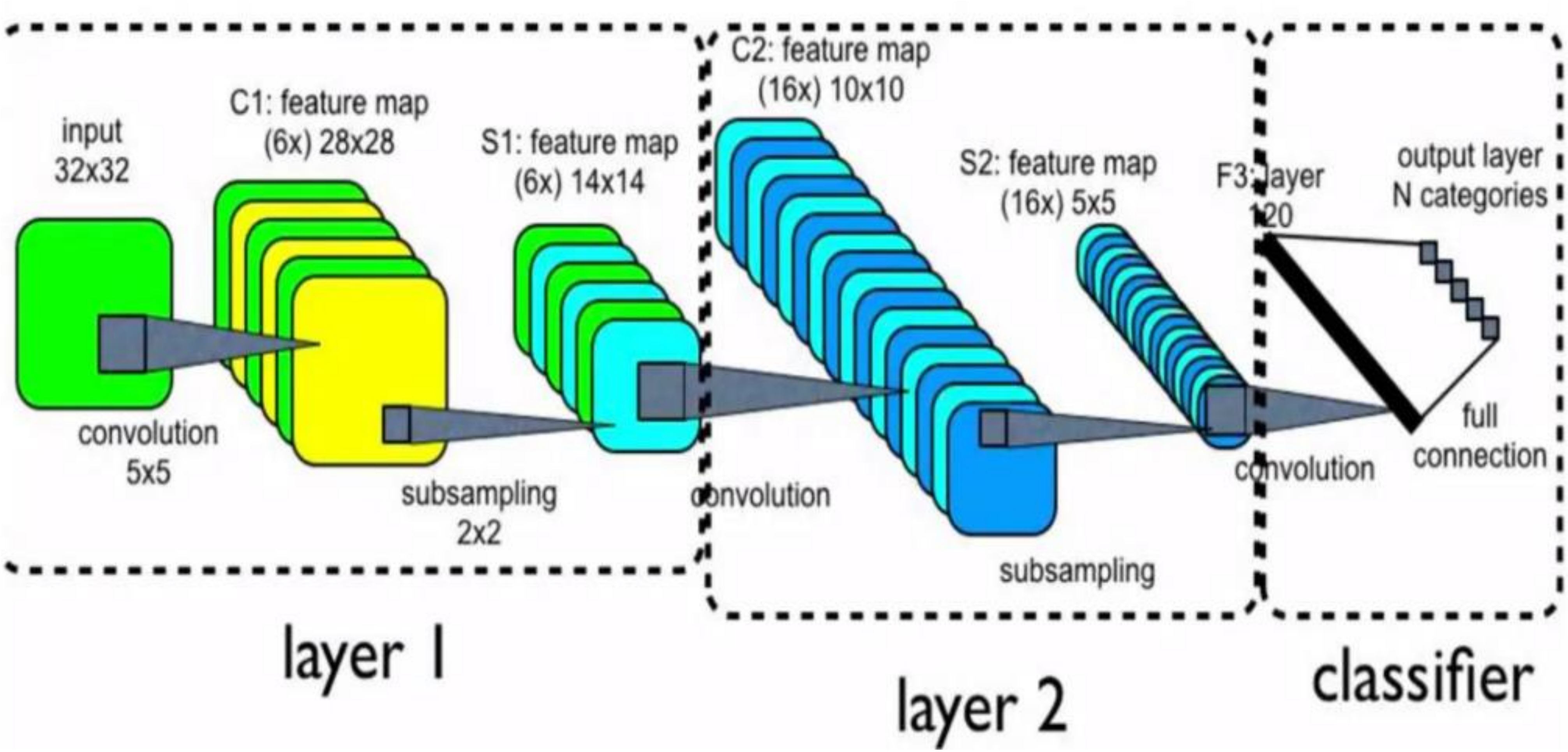
# Deep Learning Architectures

- Deep Neural Networks
- Deep Belief Networks
- Convolutional Neural Networks
- Deep Boltzmann Machines
- Deep Stacking Networks

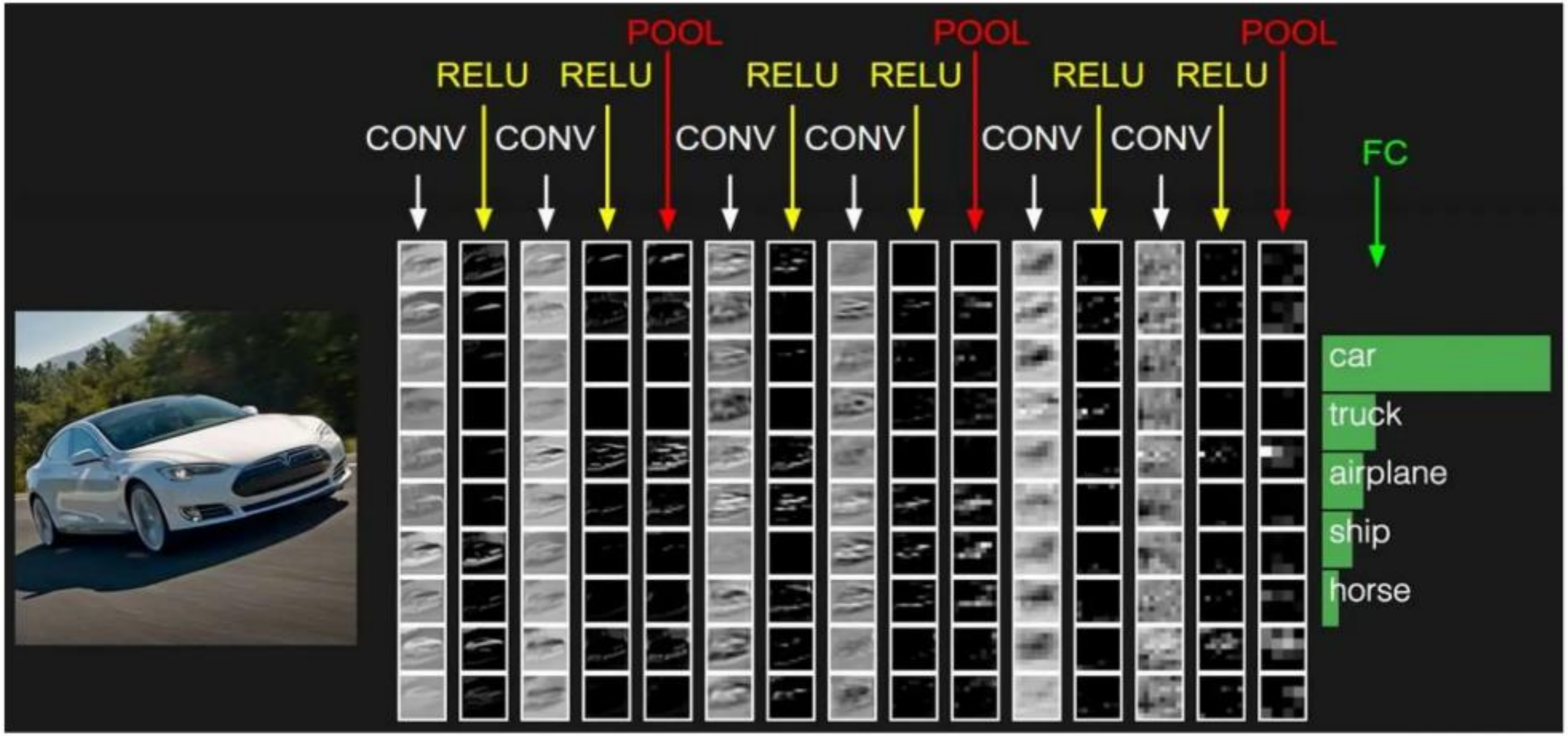
# Components of CNN

- A **CNN** consists of a number of layers:
  - Convolutional layers.
  - Pooling Layers.
  - Fully-Connected Layers.

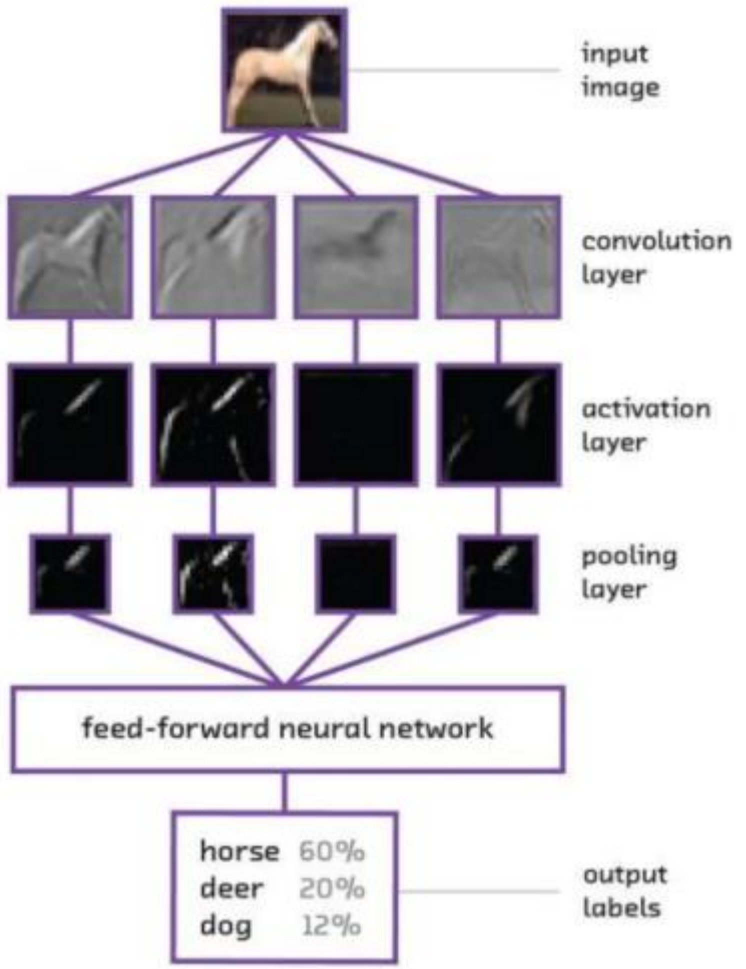
# Convolutional Neural Network



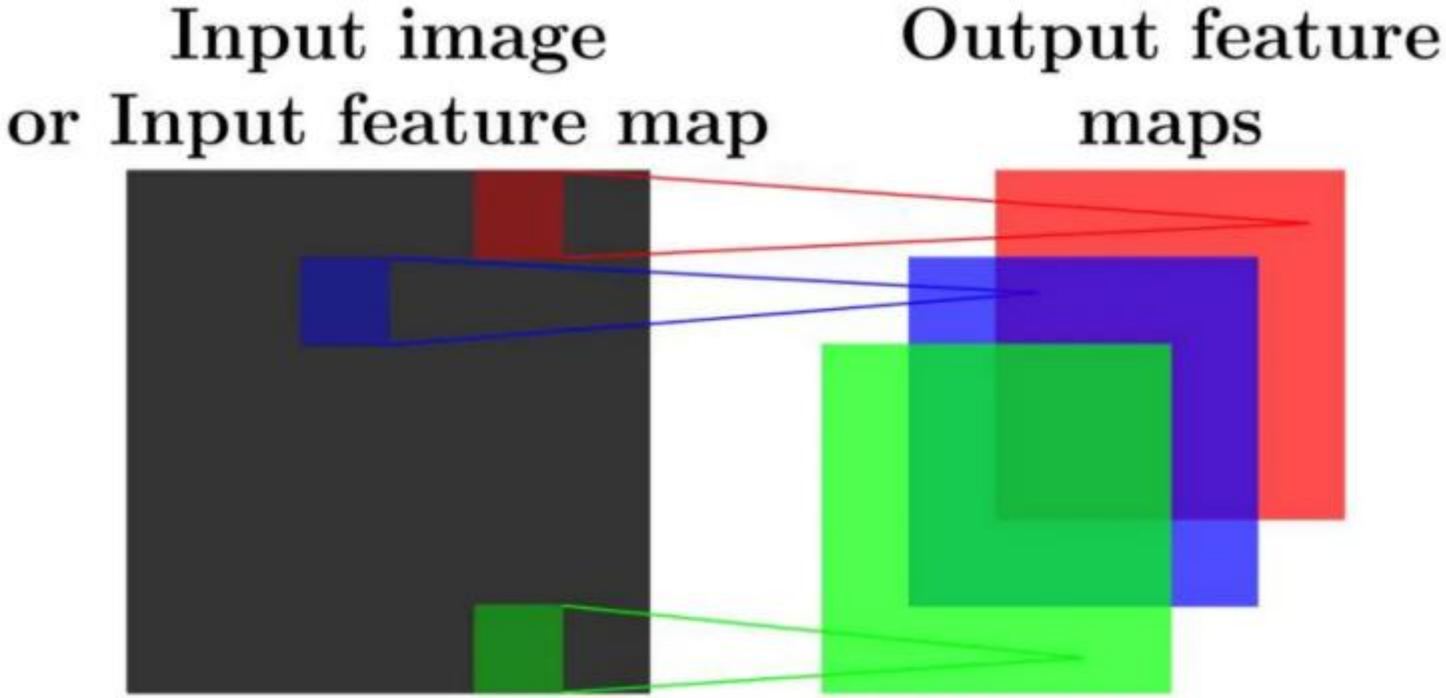
# Convolutional Neural Network



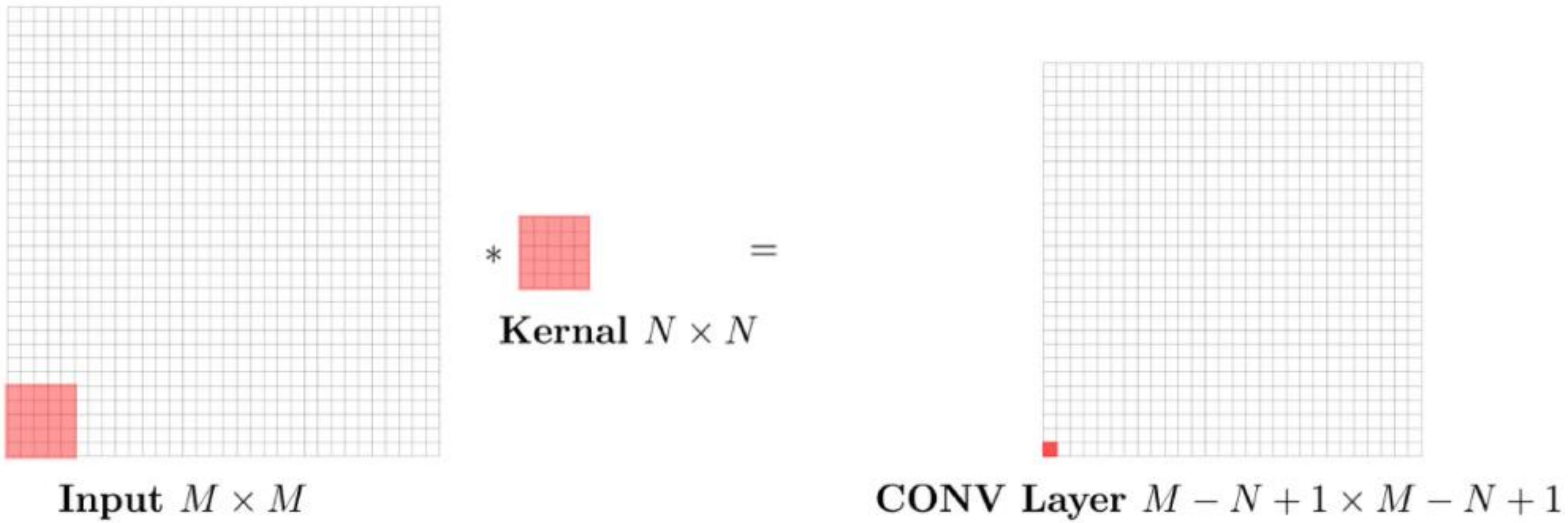
# Convolutional Neural Network



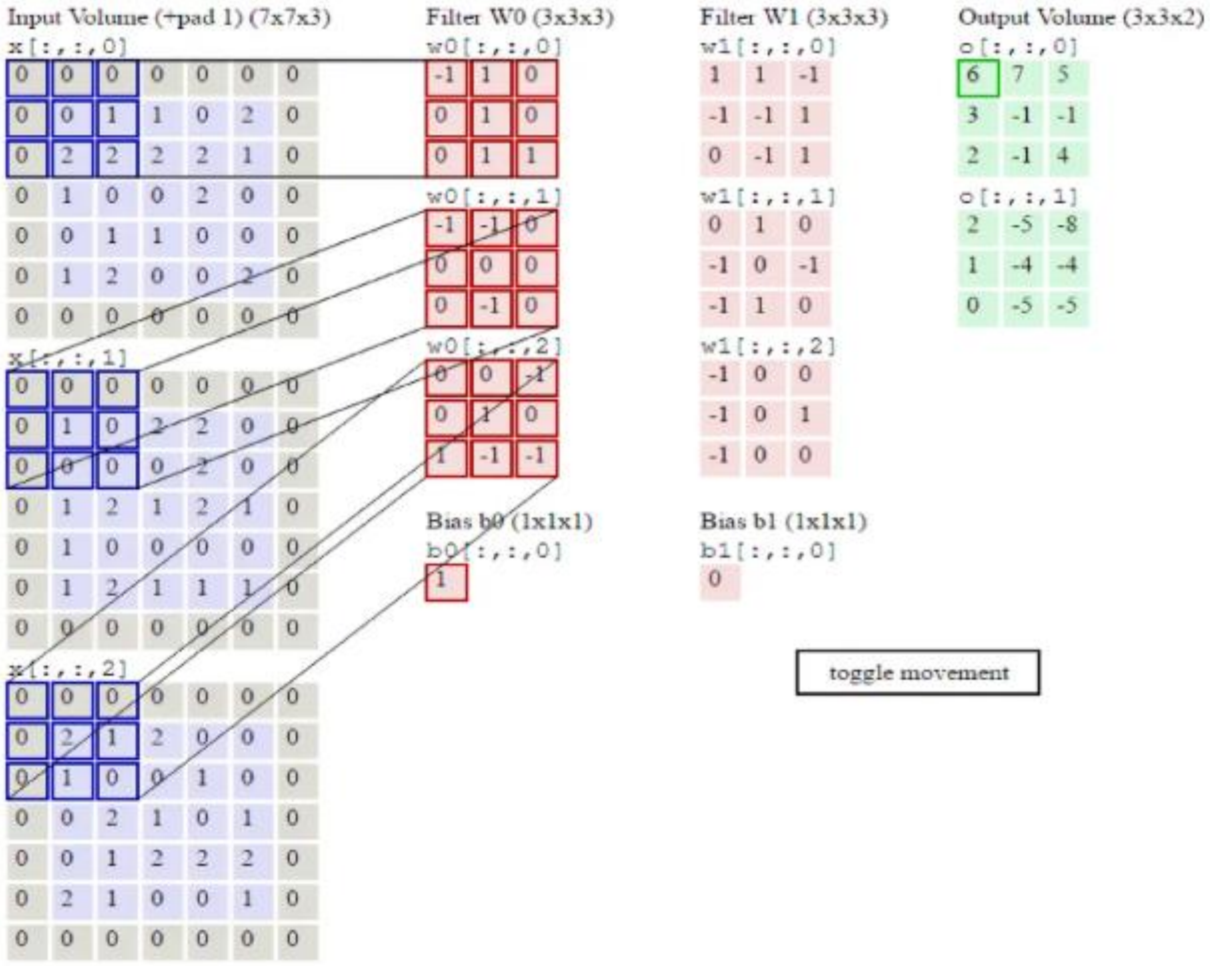
# Convolution Operation



# Convolution Operation

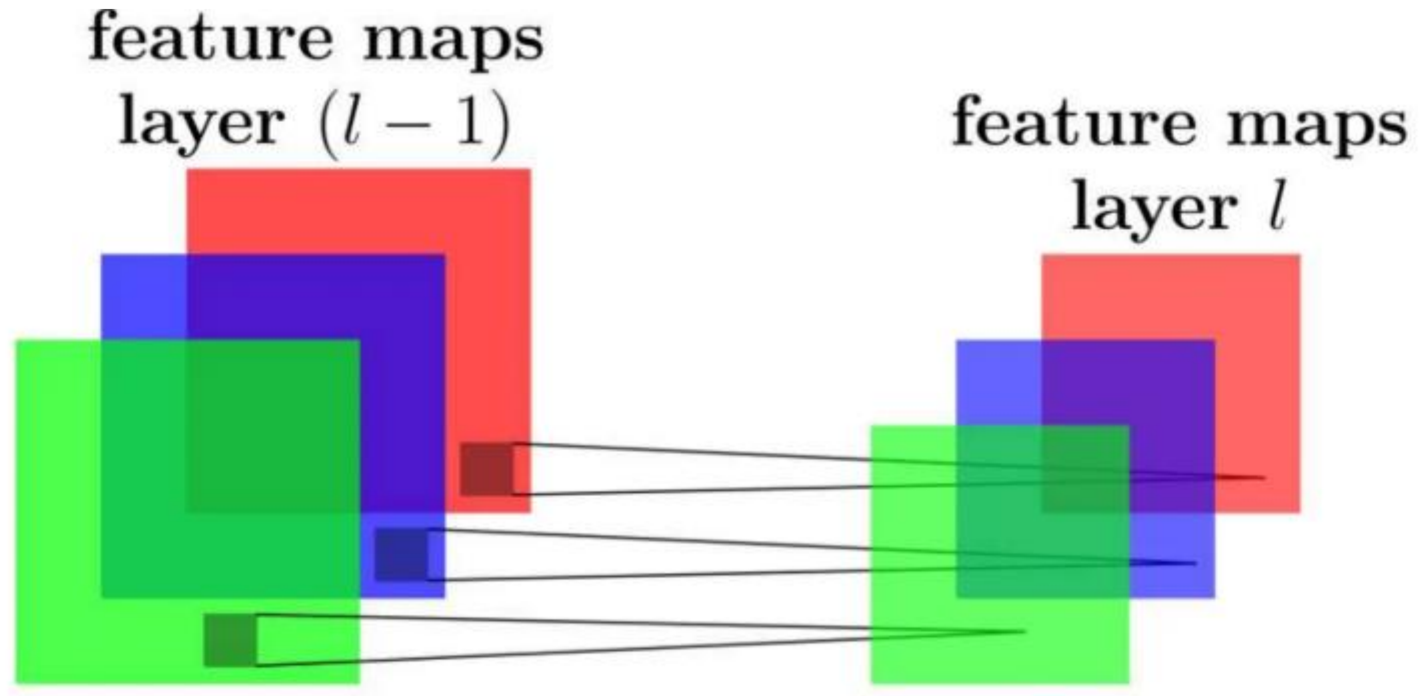


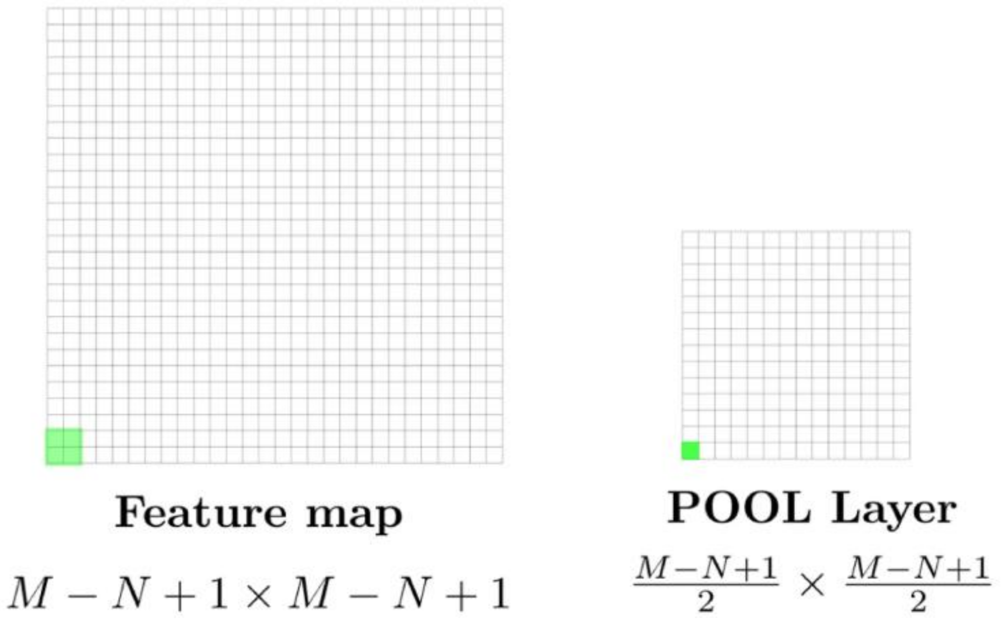
# Convolution Layers



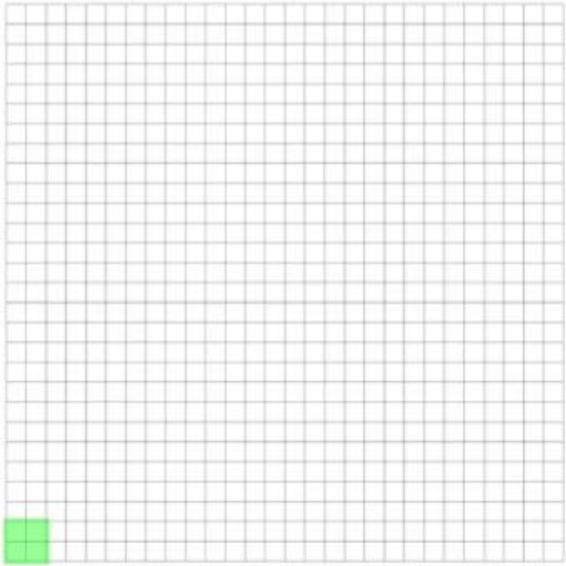
# Pooling Layer

Pooling Layer reduces the resolution of the image to reduce the precision of the translation (shift and distortion) effect.



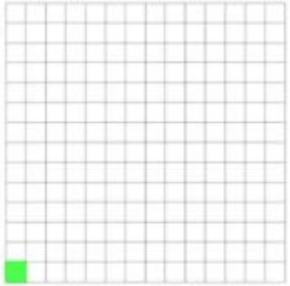



# Pooling Layer



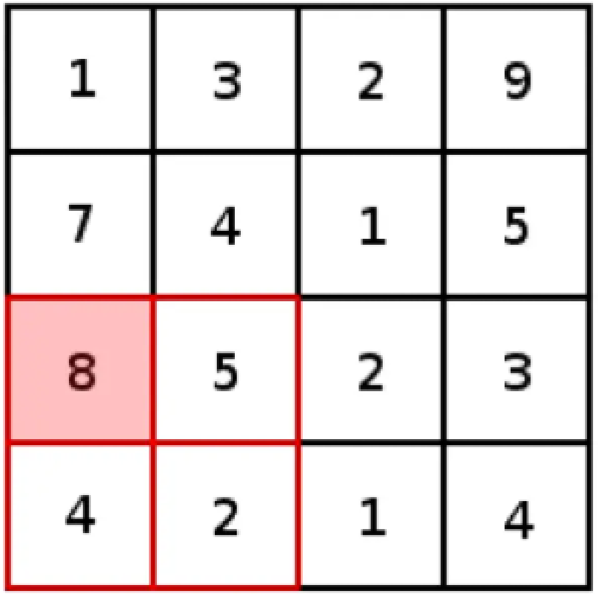
**Feature map**

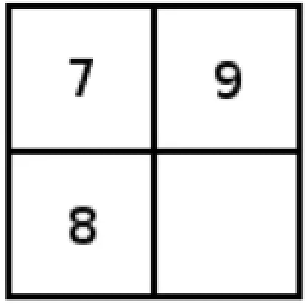
$$M - N + 1 \times M - N + 1$$



**POOL Layer**

$$\frac{M-N+1}{2} \times \frac{M-N+1}{2}$$

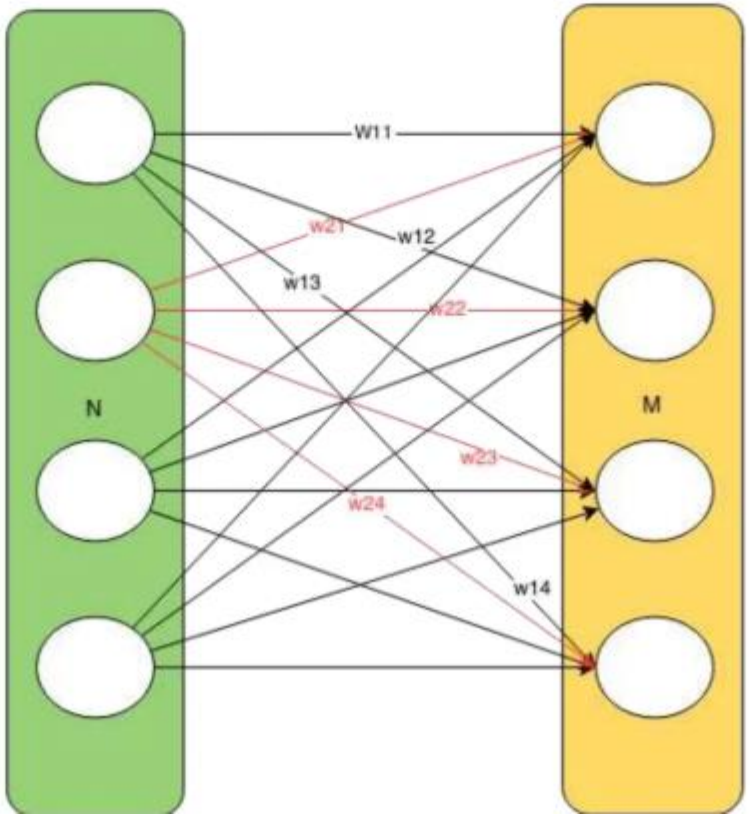


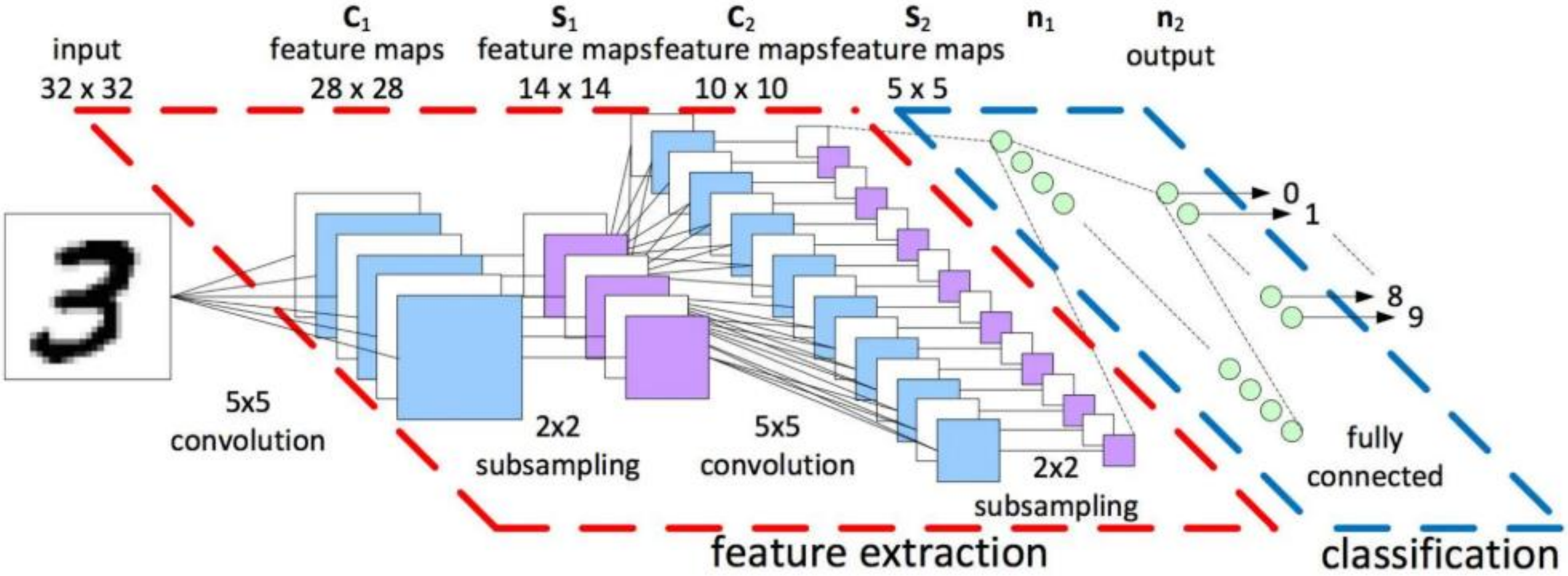



# Fully Connected Layer

Fully connected layer have full connections to all activations of the previous layer.

**Fully connected layer act as classifier.**





input
32 x 32
$C_1$
feature maps
28 x 28
$S_1$
feature maps
14 x 14
$C_2$
feature maps
10 x 10
$S_2$
feature maps
5 x 5
$n_1$
$n_2$
output
0
1
8
9
5x5
convolution
2x2
subsampling
5x5
convolution
2x2
subsampling
fully
connected
feature extraction
classification

**LeNet :**The first successful applications of CNN

**AlexNet:** The **ILSVRC** 2012 winner

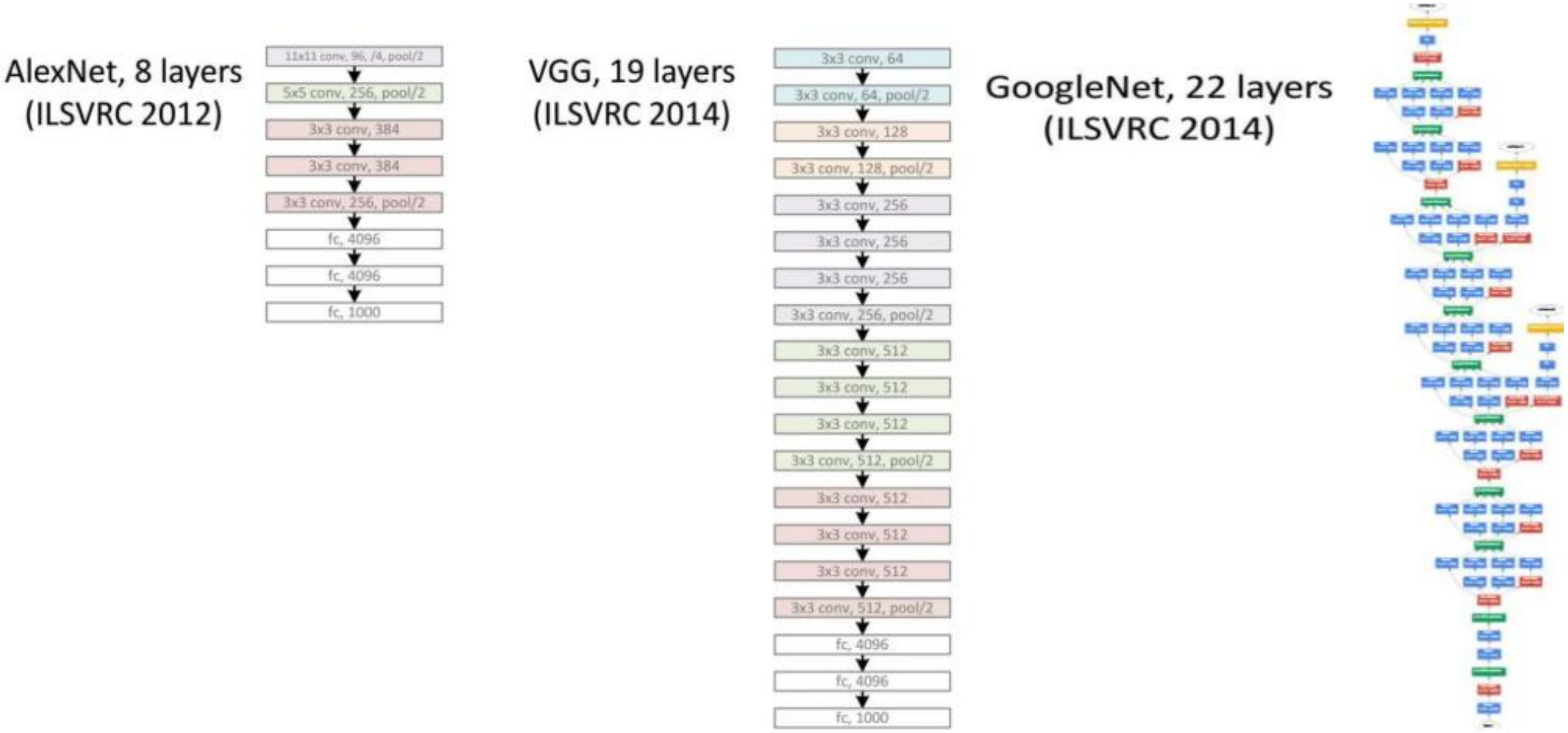
**ZFNet:** The **ILSVRC** 2013 winner

**GoogLeNet:** The **ILSVRC** 2014 winner

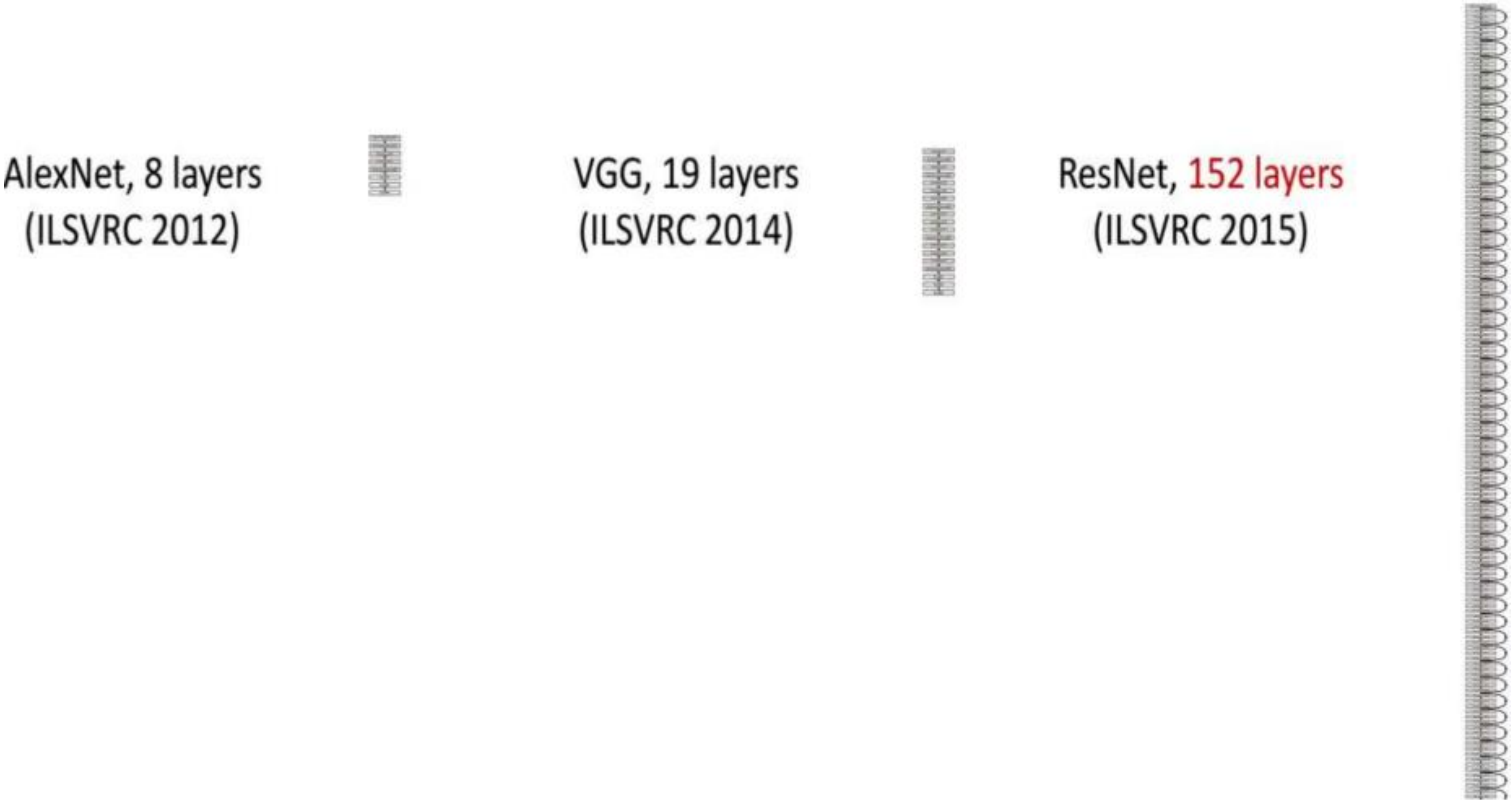
**VGGNet:** The runner-up in **ILSVRC** 2014

**ResNet:** The winner of **ILSVRC** 2015

# Depth of CNN


AlexNet, 8 layers
(ILSVRC 2012)
11x11 conv, 96, /4, pool/2
5x5 conv, 256, pool/2
3x3 conv, 384
3x3 conv, 384
3x3 conv, 256, pool/2
fc, 4096
fc, 4096
fc, 1000
VGG, 19 layers
(ILSVRC 2014)
3x3 conv, 64
3x3 conv, 64, pool/2
3x3 conv, 128
3x3 conv, 128, pool/2
3x3 conv, 256
3x3 conv, 256
3x3 conv, 256
3x3 conv, 256, pool/2
3x3 conv, 512
3x3 conv, 512
3x3 conv, 512
3x3 conv, 512, pool/2
3x3 conv, 512
3x3 conv, 512
3x3 conv, 512
3x3 conv, 512, pool/2
fc, 4096
fc, 4096
fc, 1000
GoogleNet, 22 layers
(ILSVRC 2014)

# Depth of CNN


AlexNet, 8 layers
(ILSVRC 2012)
VGG, 19 layers
(ILSVRC 2014)
ResNet, 152 layers
(ILSVRC 2015)

# Datasets Benchmark

**MNIST** Handwritten digits – 60000 Training + 10000 Test Data

**Google House Numbers** from street view - 600,000 digit images

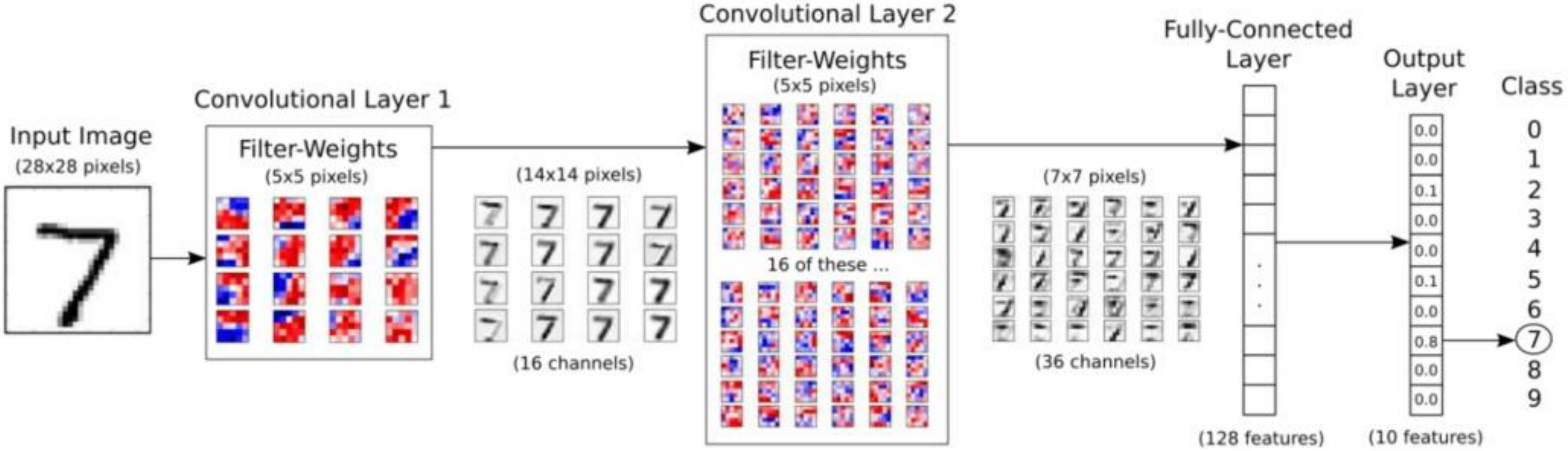
**CIFAR-10** 60000 32x32 colour images in 10 classes

**IMAGENET** 1.2 million images, >150 GB

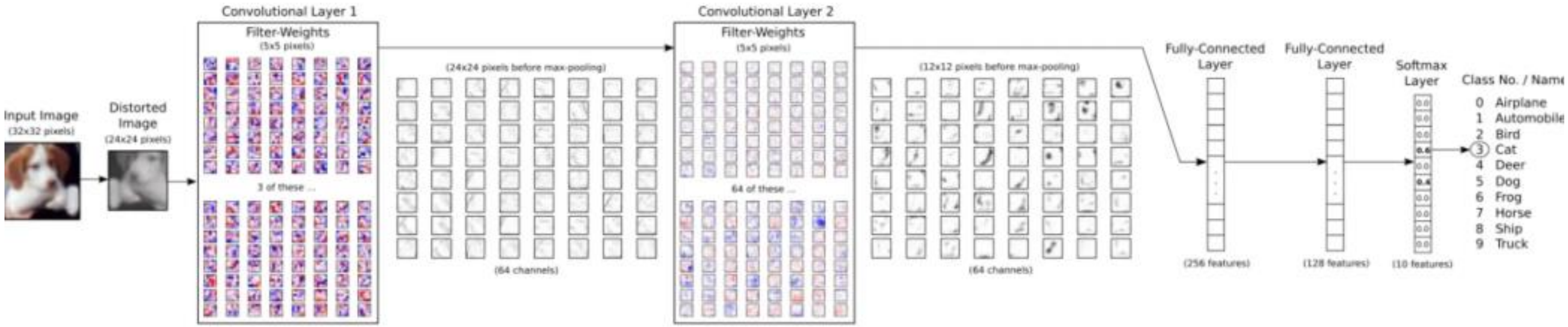
**Tiny Images** 80 Million tiny images

**Flickr Data** 100 Million Yahoo dataset

# ❑ CNN on MNIST Dataset

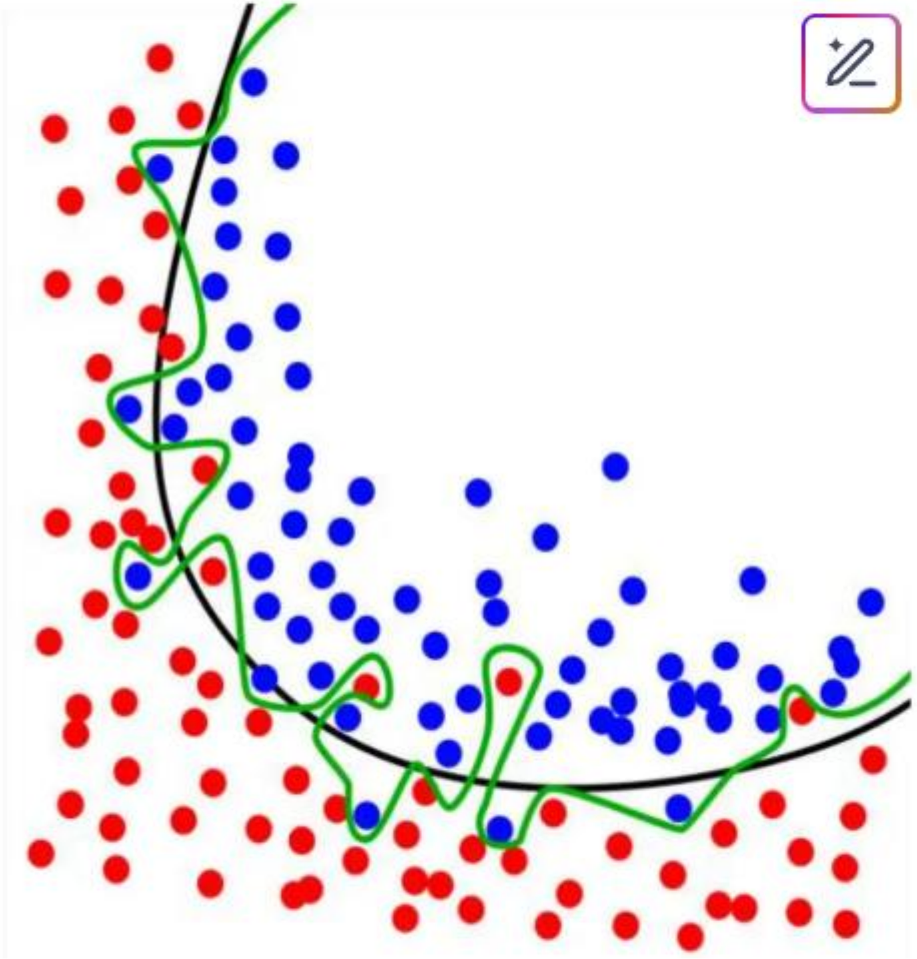


# ❑ CNN on CIFAR-10 Dataset



# CNN Optimization

- Overfitting Problem
  - Larger network have a lots of weights this lead to high model complexity
  - Network do excellent on training data but very bad on validation data



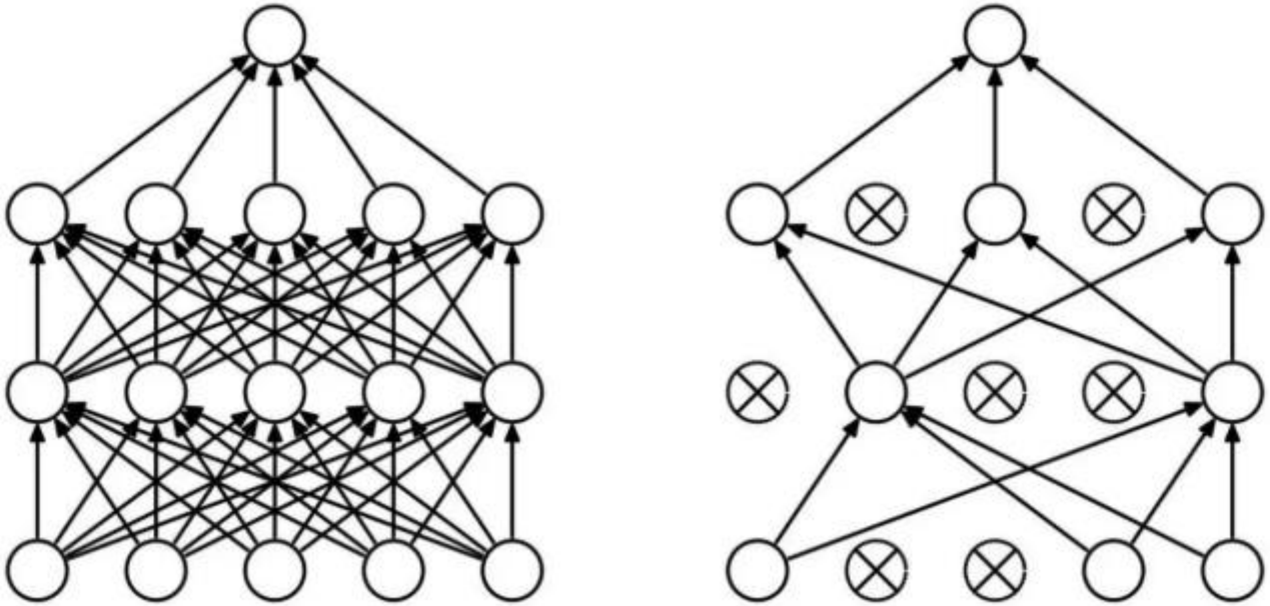
# CNN Optimization

- CNN Optimization used to reduce the overfitting problem in CNN

1) Dropout
2) L2 Regularization
3) Mini-batch
4) Gradient descent algorithm
5) Early stopping
6) Data augmentation

# CNN Optimization

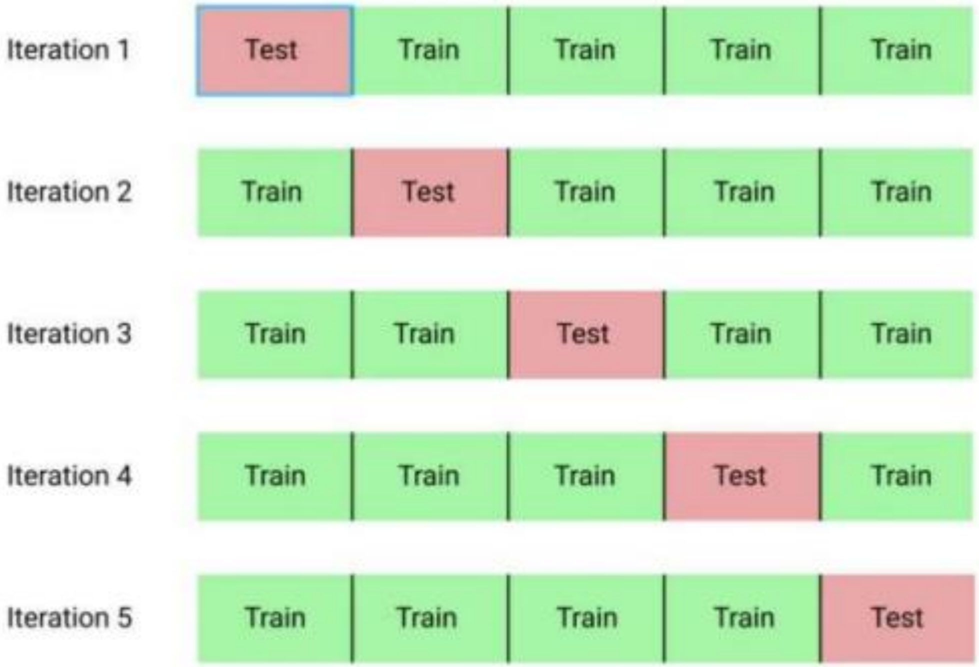
❑ **Dropout** is a technique of reducing overfitting in CNN.



(a) Standard Neural Net (b) After applying dropout.

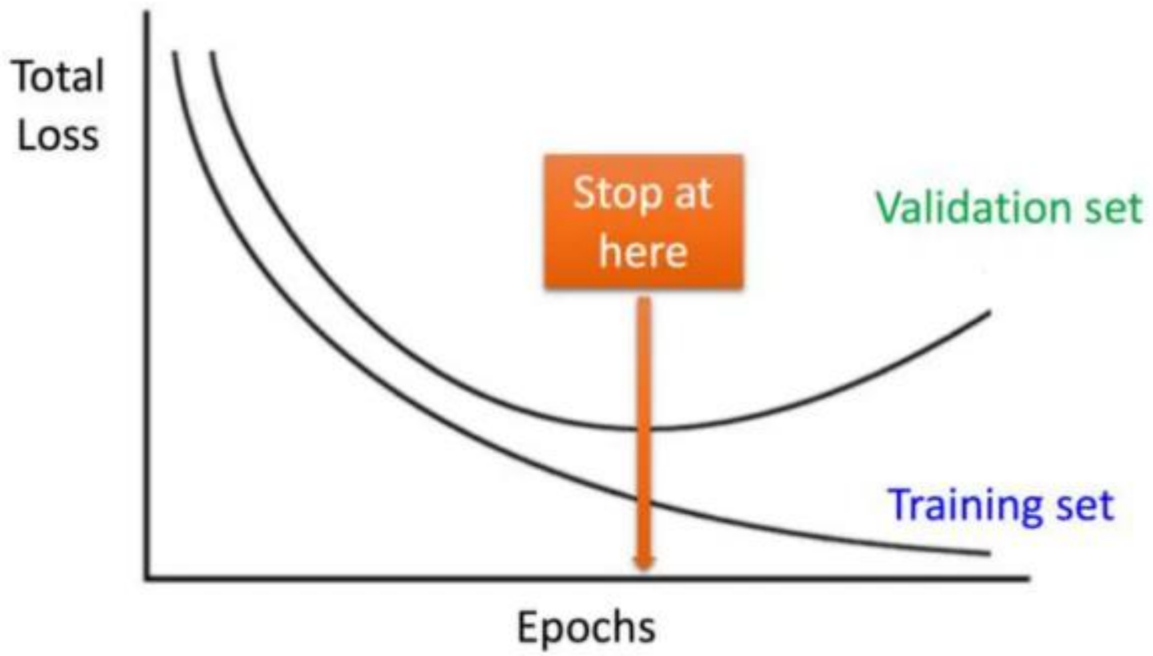
# CNN Optimization

❑ **Mini-batch** is to divide the dataset into small batches examples, compute the gradient using a single batch, make an update, then move to the r



# CNN Optimization

- **Early stopping** monitoring the deep learning process of the network from overfitting.
- If there is no more improvement, or worse, the performance on the test set degrades, then the learning process is aborted



# CNN Optimization

- Data augmentation means increasing the number of dataset.

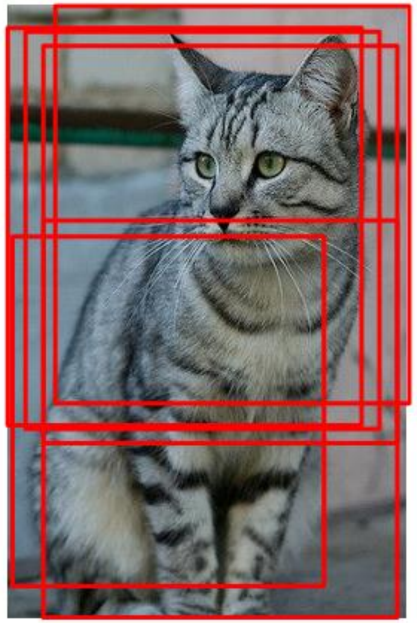
# Data Augmentation

## Random crops and scales

**Training**: sample random crops / scales

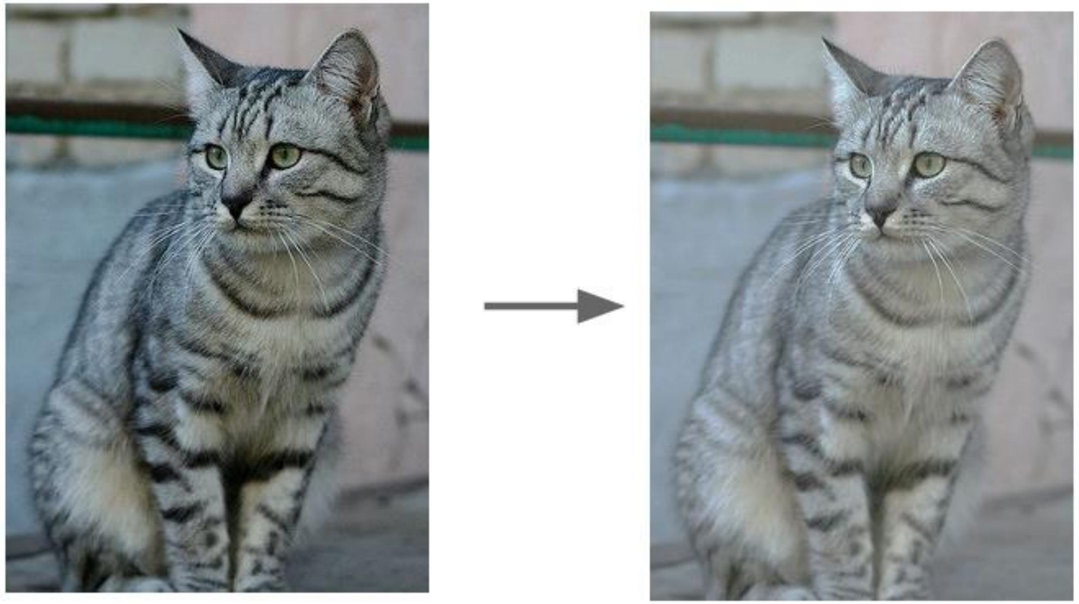
ResNet:

1. Pick random L in range [256, 480]
2. Resize training image, short side = L
3. Sample random 224 x 224 patch



# Data Augmentation
## Color Jitter

Simple: Randomize contrast and brightness



# Data Augmentation

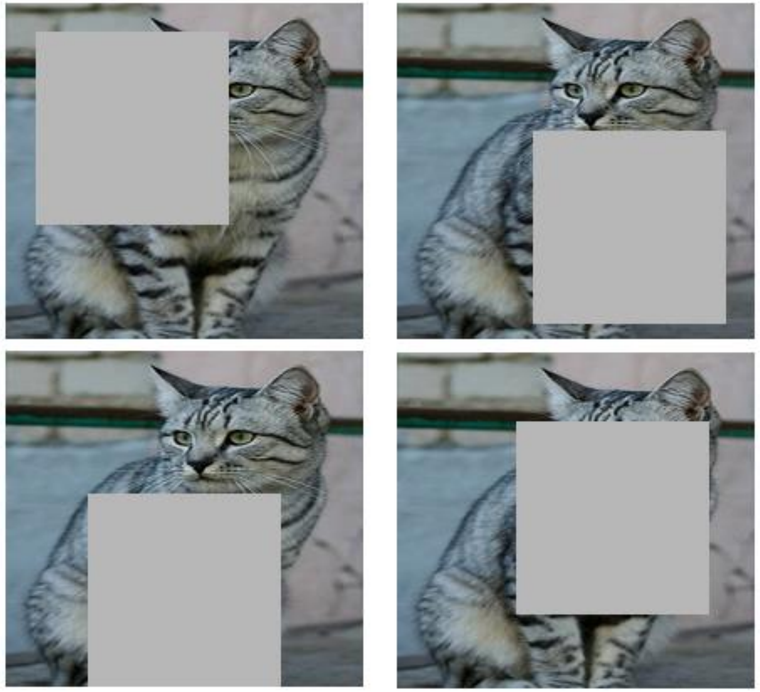
Get creative for your problem!

Examples of data augmentations:

- translation
- rotation
- stretching
- shearing,
- lens distortions, … (go crazy)

# Random Cutout

**Training**: Set random image regions to zero
**Testing**: Use full image

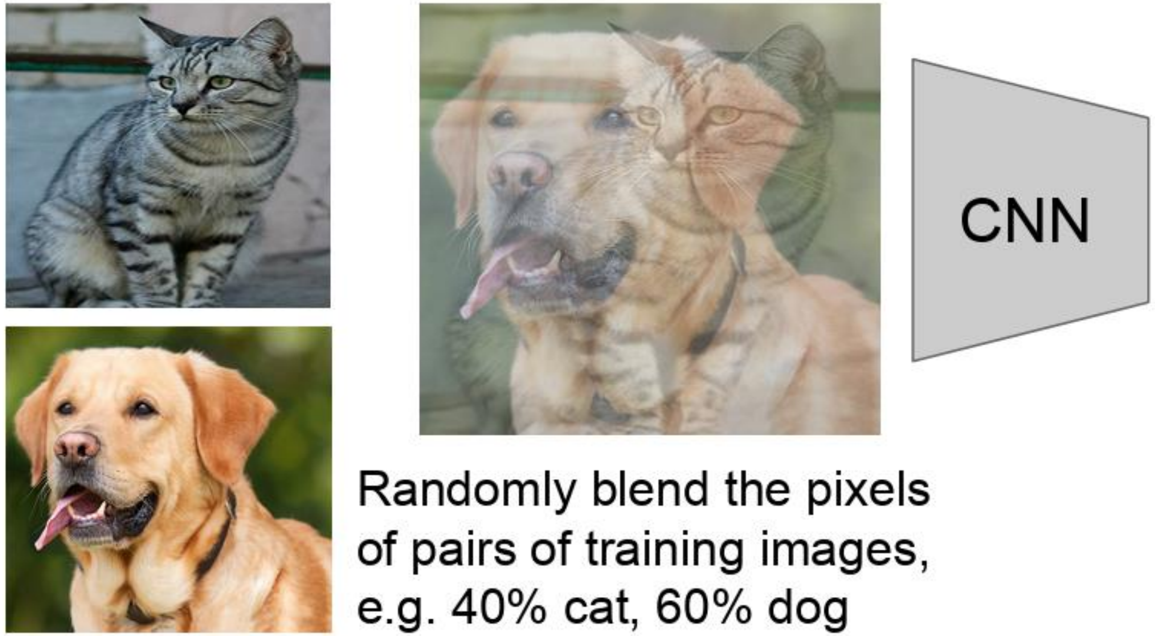


Works very well for small datasets like CIFAR, less common for large datasets like ImageNet

DeVries and Taylor, "Improved Regularization of Convolutional Neural Networks with Cutout", arXiv 2017

# Mixup

**Training**: Train on random blends of images
**Testing**: Use original images




Target label:
cat: 0.4
dog: 0.6

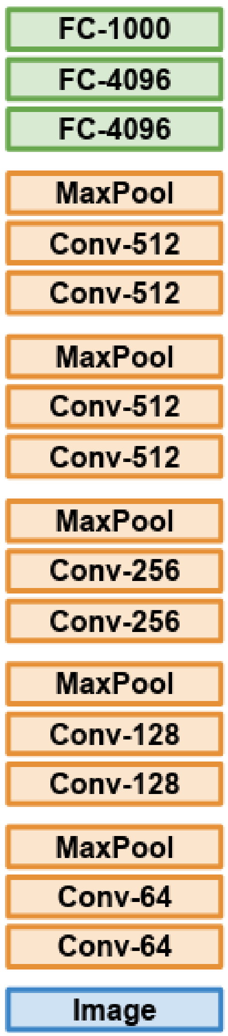
Randomly blend the pixels of pairs of training images, e.g. 40% cat, 60% dog

Zhang et al, "*mixup*: Beyond Empirical Risk Minimization", ICLR 2018

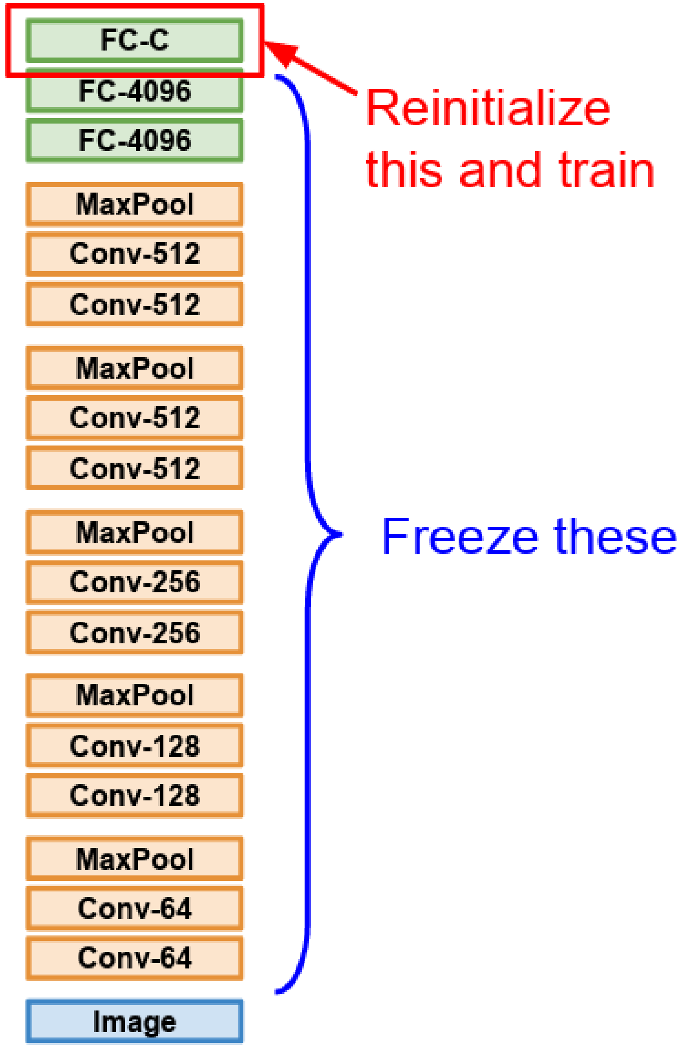
# Transfer Learning with CNNs

Donahue et al, "DeCAF: A Deep Convolutional Activation Feature for Generic Visual Recognition", ICML 2014
Razavian et al, "CNN Features Off-the-Shelf: An Astounding Baseline for Recognition", CVPR Workshops 2014

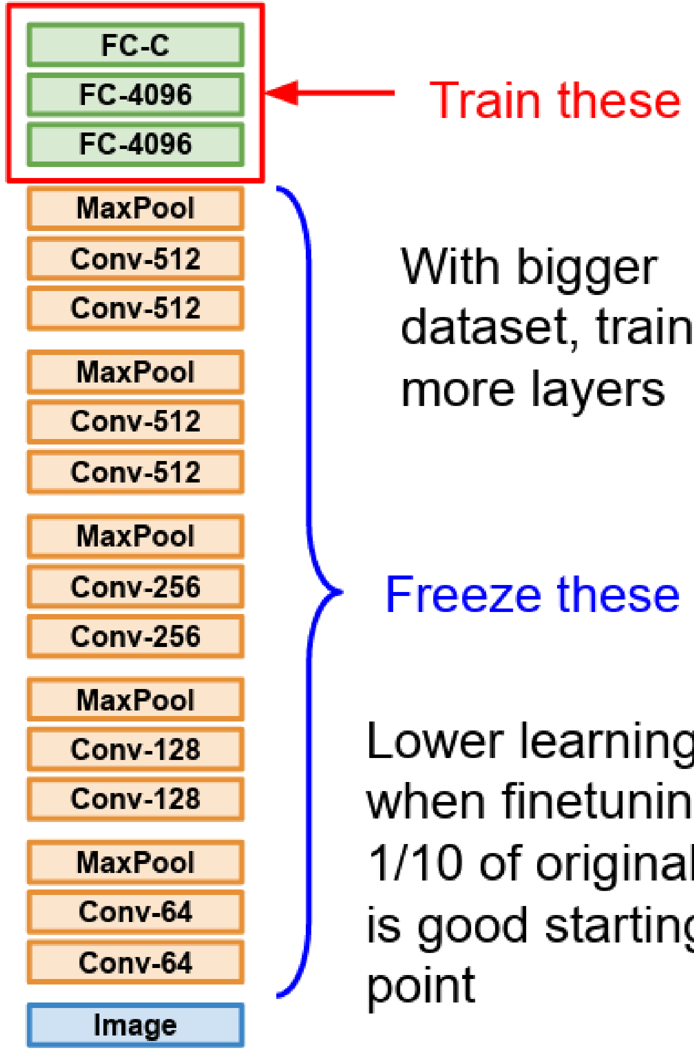
1. Train on Imagenet




2. Small Dataset (C classes)

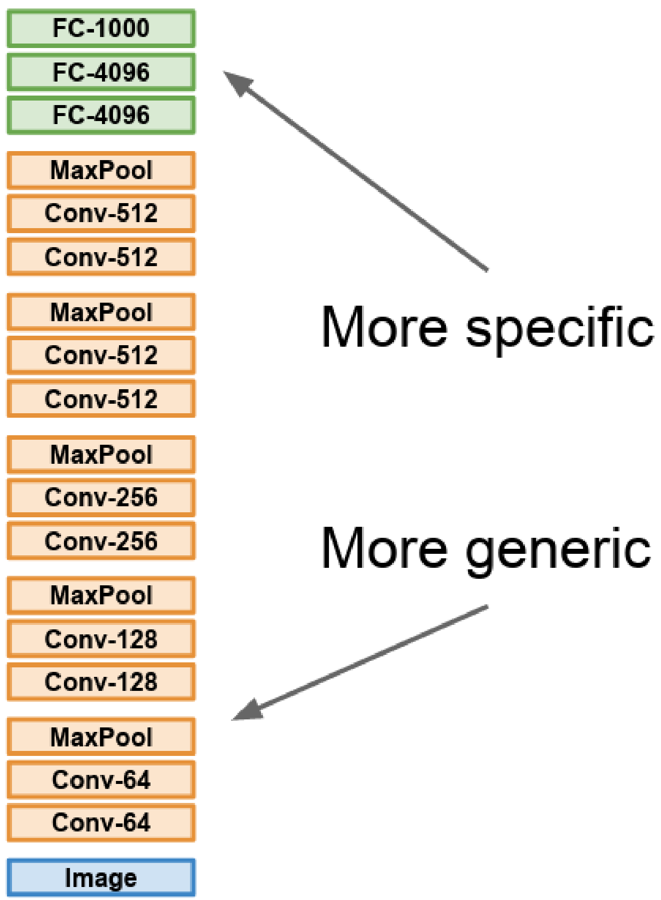



3. Bigger dataset




With bigger dataset, train more layers

Lower learning rate when finetuning; 1/10 of original LR is good starting point




| | **very similar dataset** | **very different dataset** |
|---|---|---|
| **very little data** | Use Linear Classifier on top layer | You're in trouble… Try linear classifier from different stages |
| **quite a lot of data** | Finetune a few layers | Finetune a larger number of layers |

# Takeaway for your projects and beyond:

Transfer learning be like


Source: AI & Deep Learning Memes For Back-propagated Poets

## Summary

- **Deep learning** is a class of machine learning algorithms.

- Harder problems such as video understanding, image understanding , natural language processing and Big data will be successfully tackled by **deep learning algorithms.**