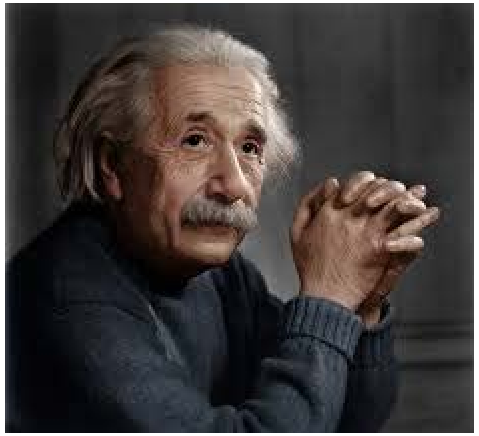
Deep Learning for Computer Vision

C.-C. Jay Kuo
University of Southern California

Part I: Big Visual Data Analytics

Face Recognition: An Example

- Face recognition problem
  - Whose face is this?
  - Humans recognize it with experience
  - The more we see, the faster we perceive
- Machine learning (ML) enables computers to automatically learn from data and convert data into useful information
- ML develops solutions and improves the performance with experience



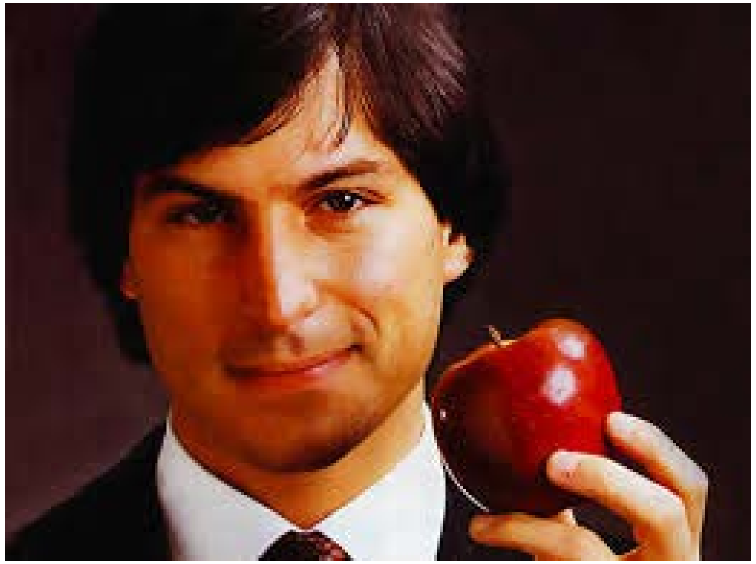
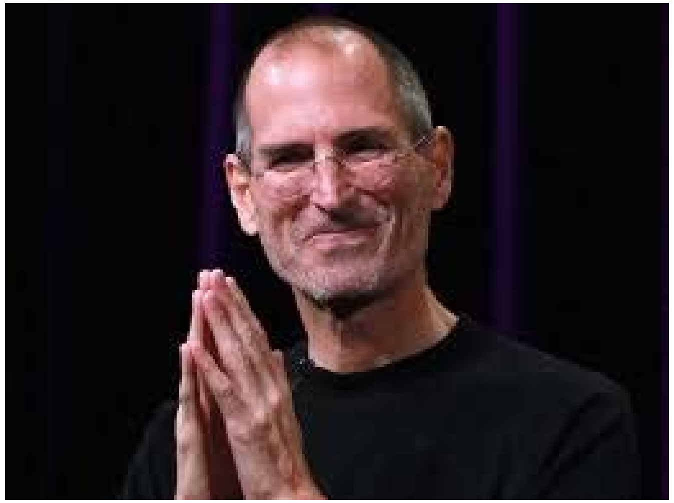
Machine Learning

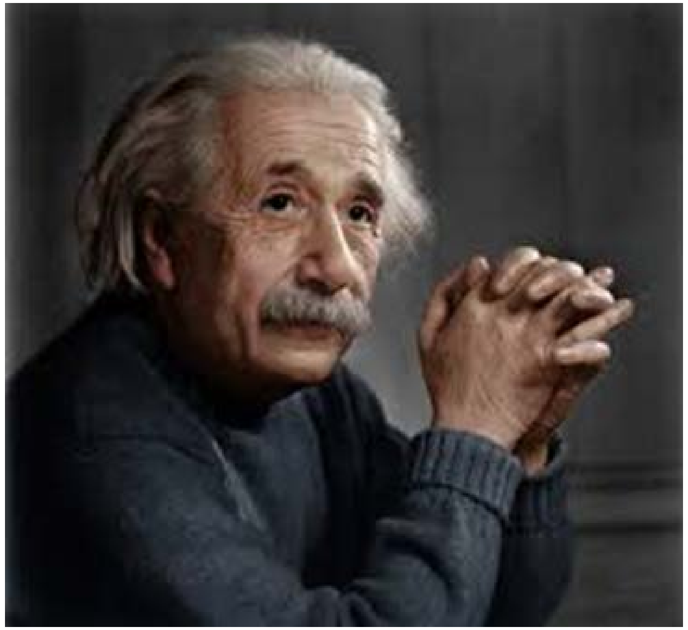
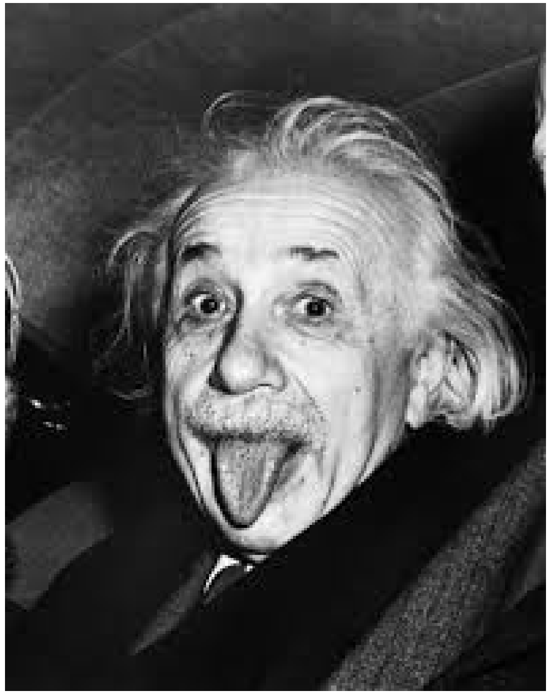
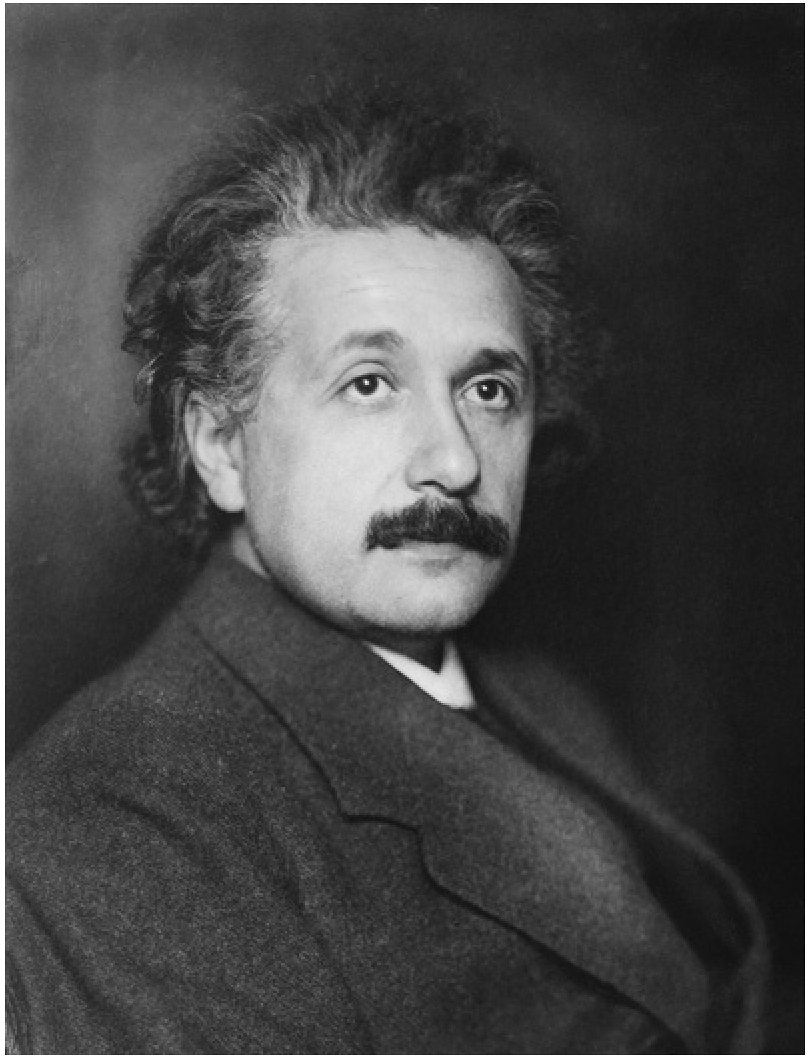
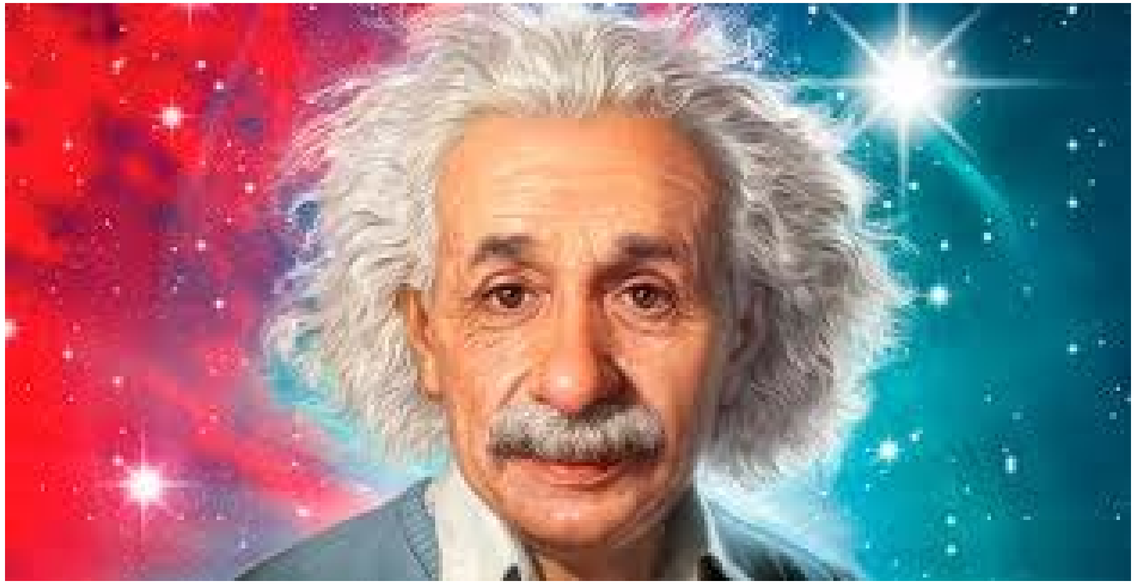
- A branch of artificial intelligence, booming in the last decade
- Definition
  - A computer program learns from experience (E) with respect to some tasks (T) and performance measure (P)
  - Its performance (P) at tasks (T) improves with experience (E)
- Examples
  - Decision tree, association rule, artificial neural nets, genetic programming, support vector machine, clustering, Bayesian networks, etc.

Generative Data Analytics

- Model-based and often weakly supervised (舉一返三)
- Examples:
  - Speech processing: Hidden Markov Model (HMM) and Gaussian Mixture Model (GMM)
  - Image/video processing: texture synthesis, face synthesis, etc.
- Advantage:
  - Highly efficient if the model is suitable (only a small amount of training data is needed)
- Limitations:
  - Phenomena could be too difficult to model based on prior knowledge
  - Models could be too complicated to manipulate due to nonlinearity




gettyimages
VCG


gettyimages
ChinaFotoPress

Discriminative Data Analytics

- Distance-based and often strongly-supervised
- Examples:
  - SVM, Decision Tree, Random Forest,
  - Convolutional/Recurrent neural networks
- Advantages:
  - Data-driven (in contrast with model-driven)
  - The underlying model could be nonlinear and very complicated
- Challenges:
  - Demanding a large amount of training data

Main Differences Between the Two

- Discriminative Model
  - Relies on a feature space and a distance metric in that space
- Generative Model
  - Know the probability distribution of the source data

ImageNet

- An image database organized according to the WordNet hierarchy (currently only nouns)
- Each node of the hierarchy is depicted by hundreds and thousands of images (500 images per node on the average)
- 1.4 million images in total
- URL: http://www.image-net.org/

Flute

Strawberry

Traffic light

Backpack

Matchstick

Bathing cap

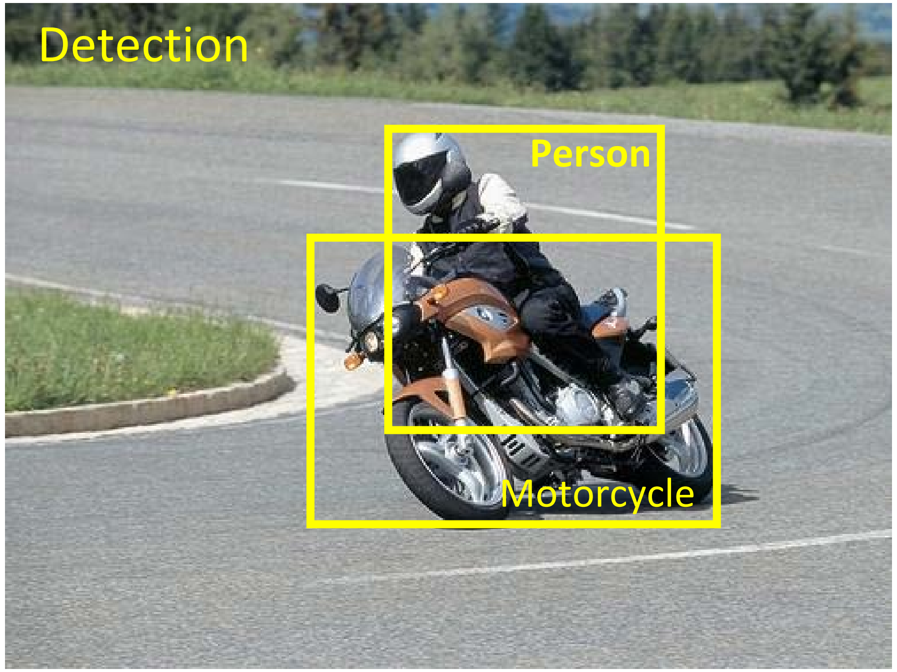
Sea lion

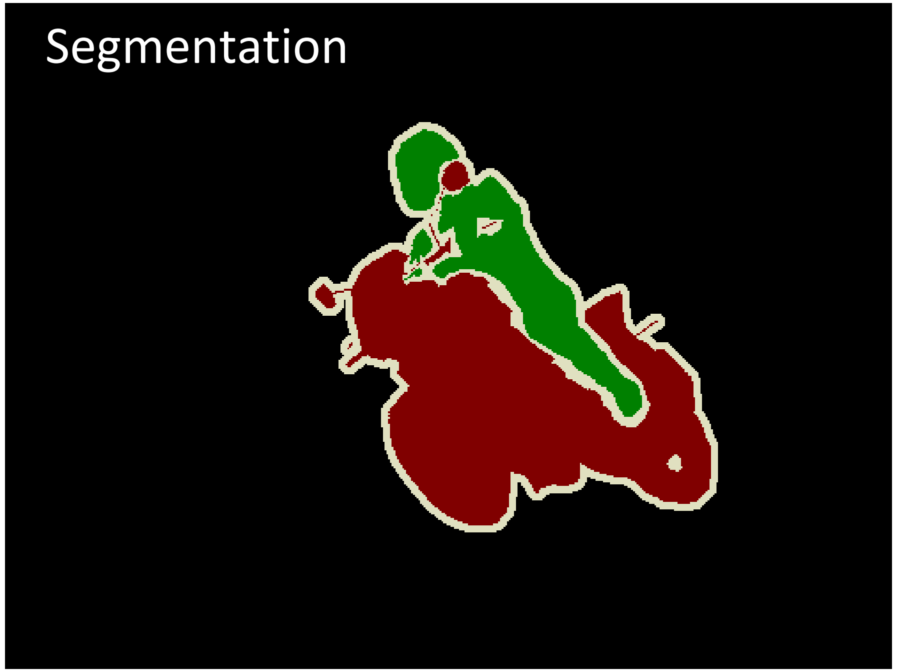
Racket

PASCAL VOC 2005-2012

20 object classes **22,591 images**

Classification: person, motorcycle





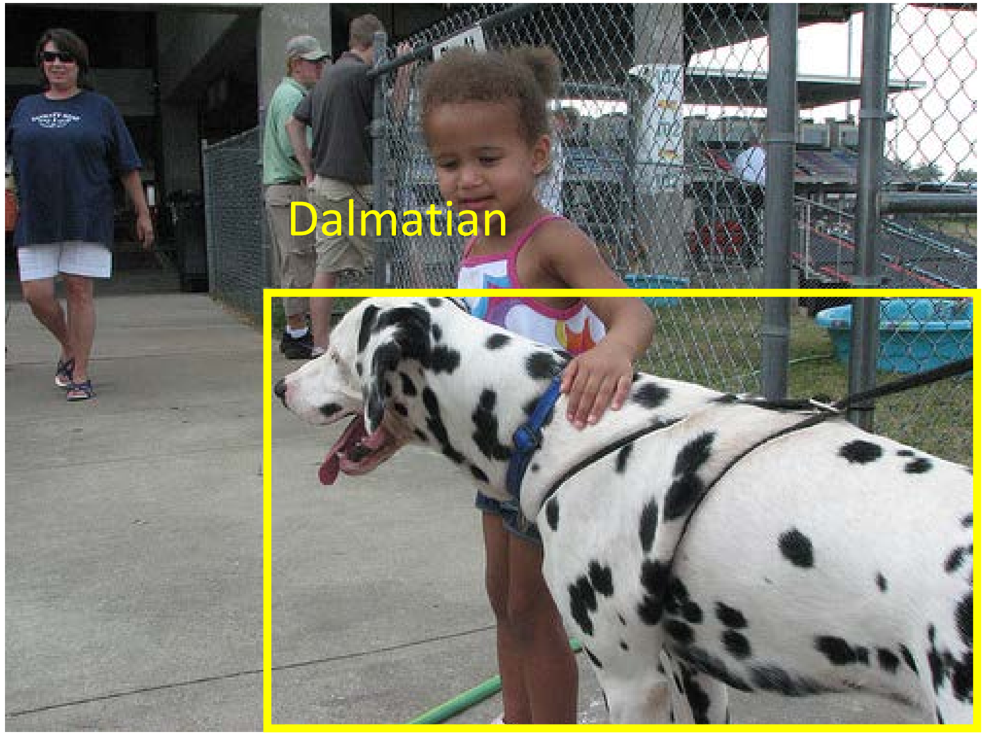
Action: riding bicycle

Everingham, Van Gool, Williams, Winn and Zisserman.
The PASCAL Visual Object Classes (VOC) Challenge. IJCV 2010.

IMAGENET

~~20 object classes~~ ~~22,591 images~~

1000 object classes **1,431,167 images**



http://image-net.org/challenges/LSVRC/{2010,2011,2012}

Variety of Object Classes

Places Database

- Scene recognition is one of the key tasks of computer vision
  - Allowing defining a context for object recognition
- A new scene-centric database
  - 205 scene categories
  - 2.5 millions of images with a category label
- URL: http://places.csail.mit.edu/

Examples

Given an image, predict which place we are in.

Bedroom

Harbor

More Examples

Bedroom

Mountain

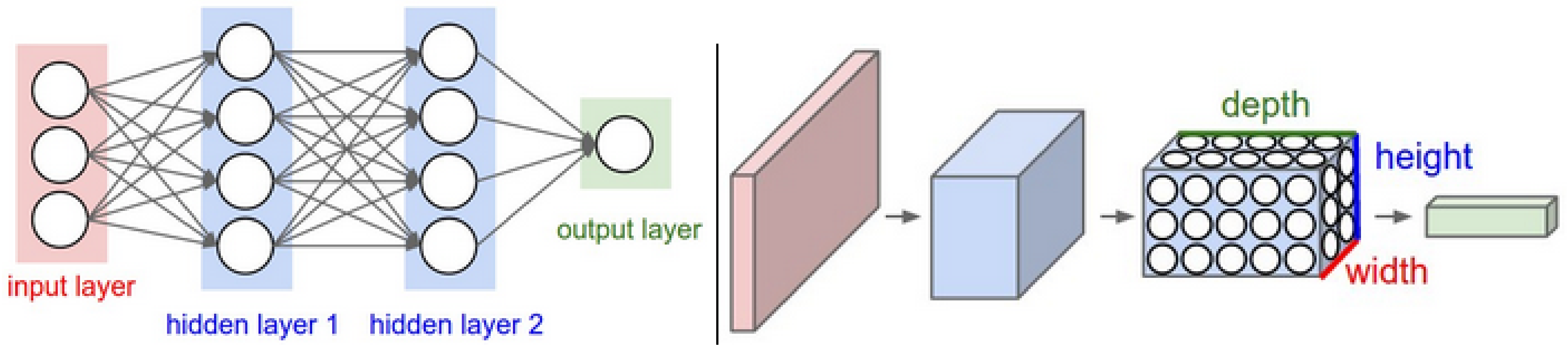
Part II: CNN Basics

Two Deep Neural Networks

- Convolutional Neural Network (CNN)
  - Finds applications in image/video processing and computer vision
    - Image/video processing: restoration, super-resolution, denoising, segmentation, etc.
    - Computer vision: object classification, object detection, object tracking, face recognition, 3D shape recognition, event detection, etc.
- Re-current Neural Network (RNN)
  - Finds applications in speech/language processing and time series processing
    - Speech/language processing: speech understanding, speaker recognition, automatic speech/language translation, etc.
    - Time series processing: EEG (for brain) and ECG (for heart) data analysis, time dynamic of social network data, etc.

From ANN to CNN

- Two illustrative diagrams

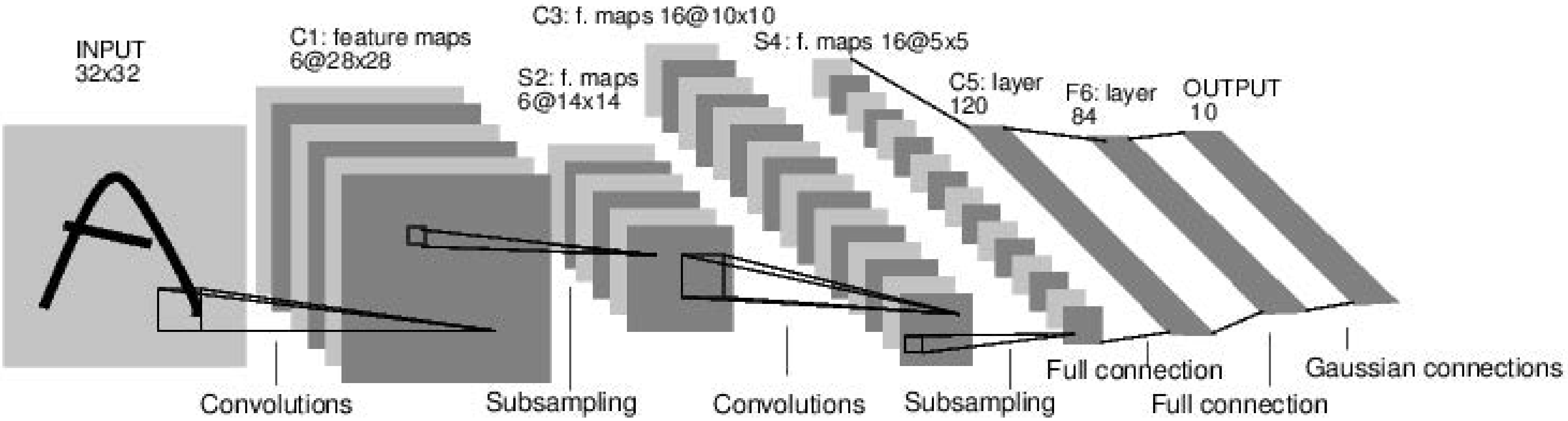


Left: A regular 3-layer Neural Network. Right: A ConvNet arranges its neurons in three dimensions (width, height, depth), as visualized in one of the layers. Every layer of a ConvNet transforms the 3D input volume to a 3D output volume of neuron activations. In this example, the red input layer holds the image, so its width and height would be the dimensions of the image, and the depth would be 3 (Red, Green, Blue channels).

Figure Credit: Stanford CS231n Course Instruction

An Exemplary CNN: LeNet-5 (1998)

Input: a 8-bit gray-scale image of size 32x32



LeCun, Bottou, Bengio, Haffner, Gradient-based learning applied to document recognition, Proc. IEEE, 1998.

Convolution with Filter Banks

Input: a 24-bit color image of size 32x32

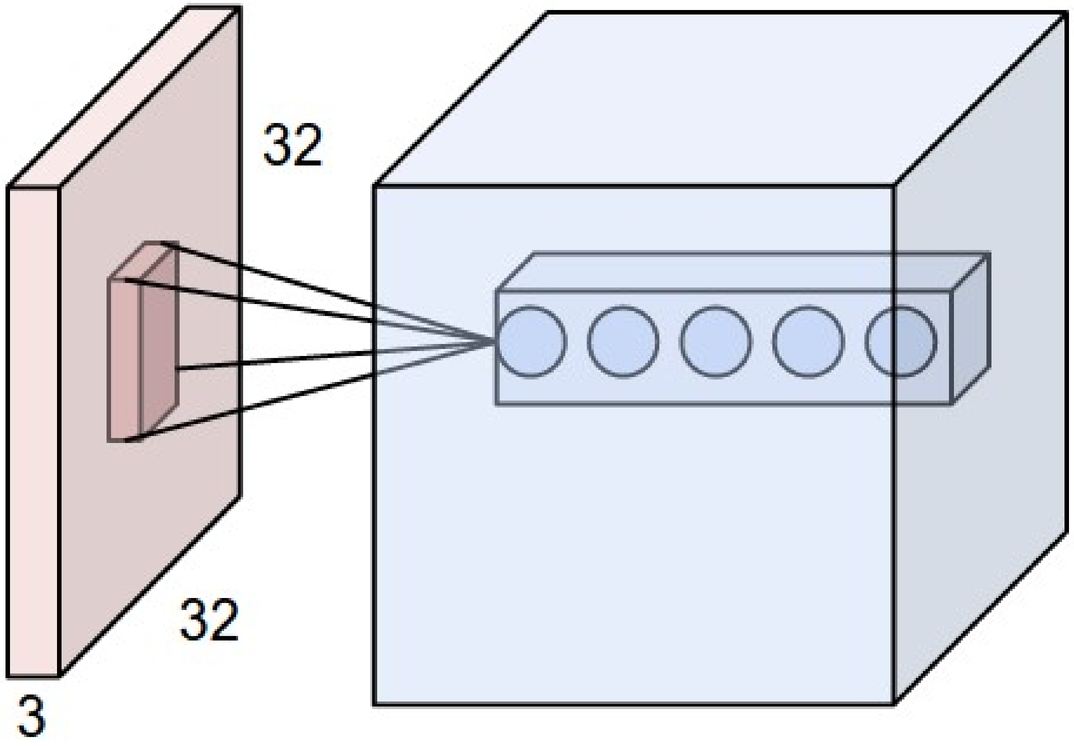


Figure Credit: Stanford CS231n Course Instruction

Convolution + Nonlinear Activation

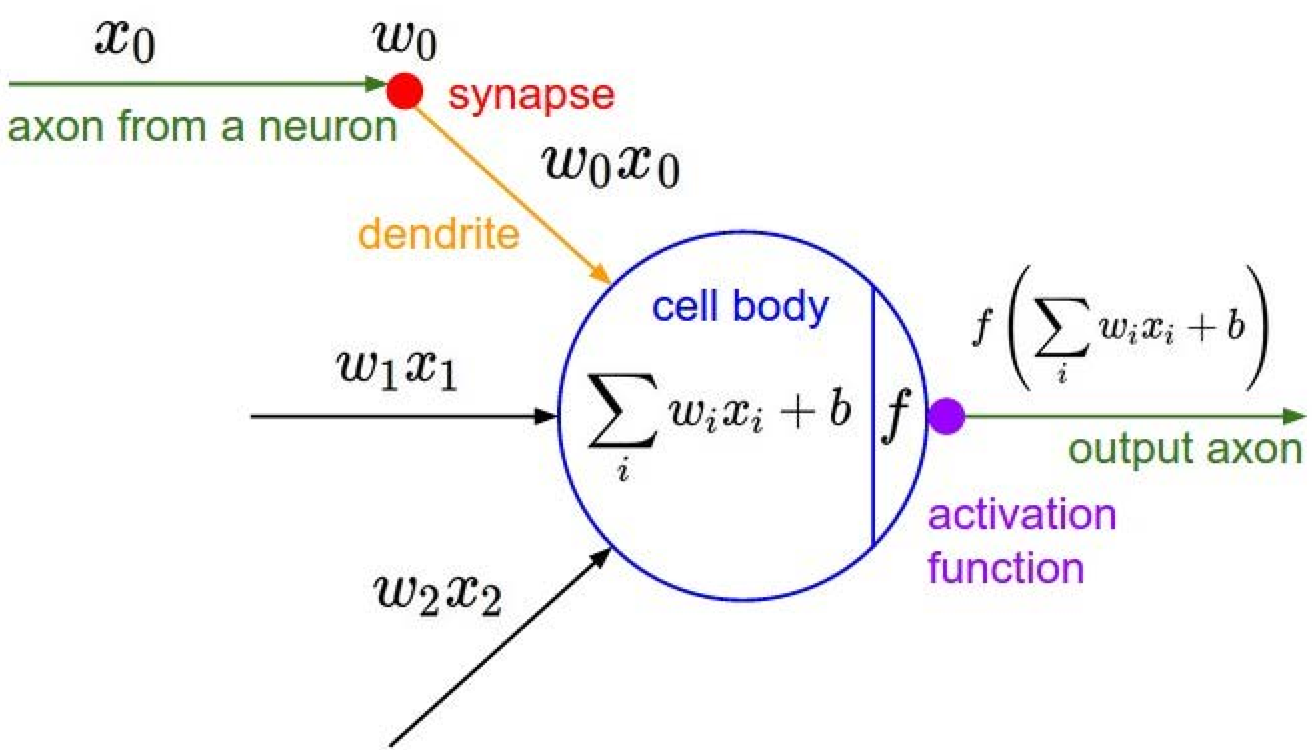
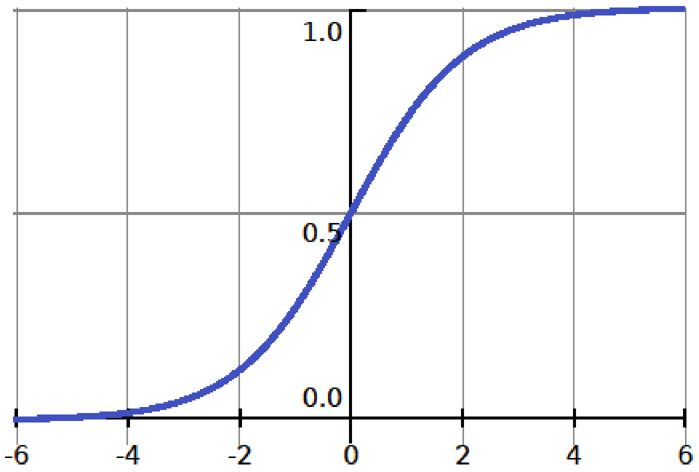
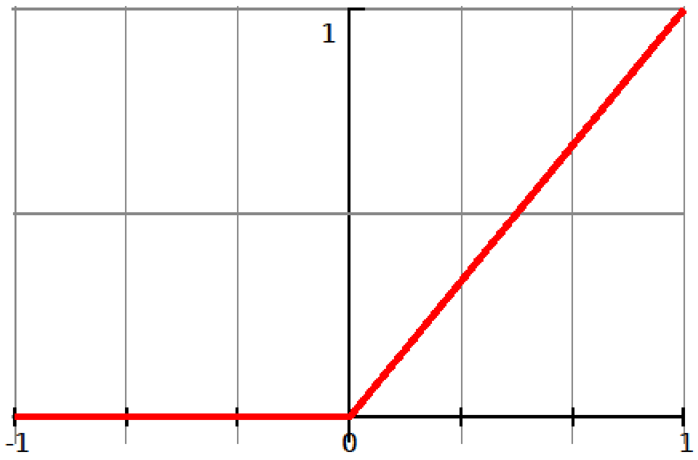


Figure Credit: Stanford CS231n Course Instruction

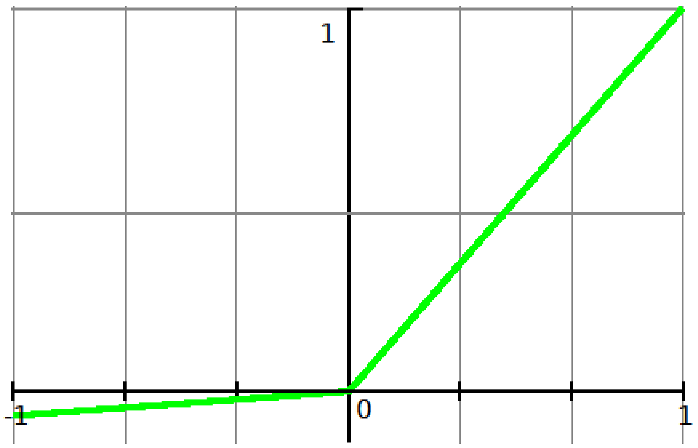
Nonlinear Activation



Sigmoid



ReLU



Leaky ReLU

Parameters in Conv Layer

For the given example

- Filter size: 5x5x3
  - 5x5: the receptive filter size
  - 3: the input filter channel size
- No. of filter weights: (5x5x3)+1=76
  - 1 denotes the bias term (b)
- No. of output filter numbers (channels):
  - 5 output channels are given in the example
- Filterbank: all output filters form a filter bank
  - Each filterbank has a center
  - The output is written to the corresponding center location
- Stride: the distance to slide the center of a filter bank
- Zero padding: pad zeros around the border region

1D Example

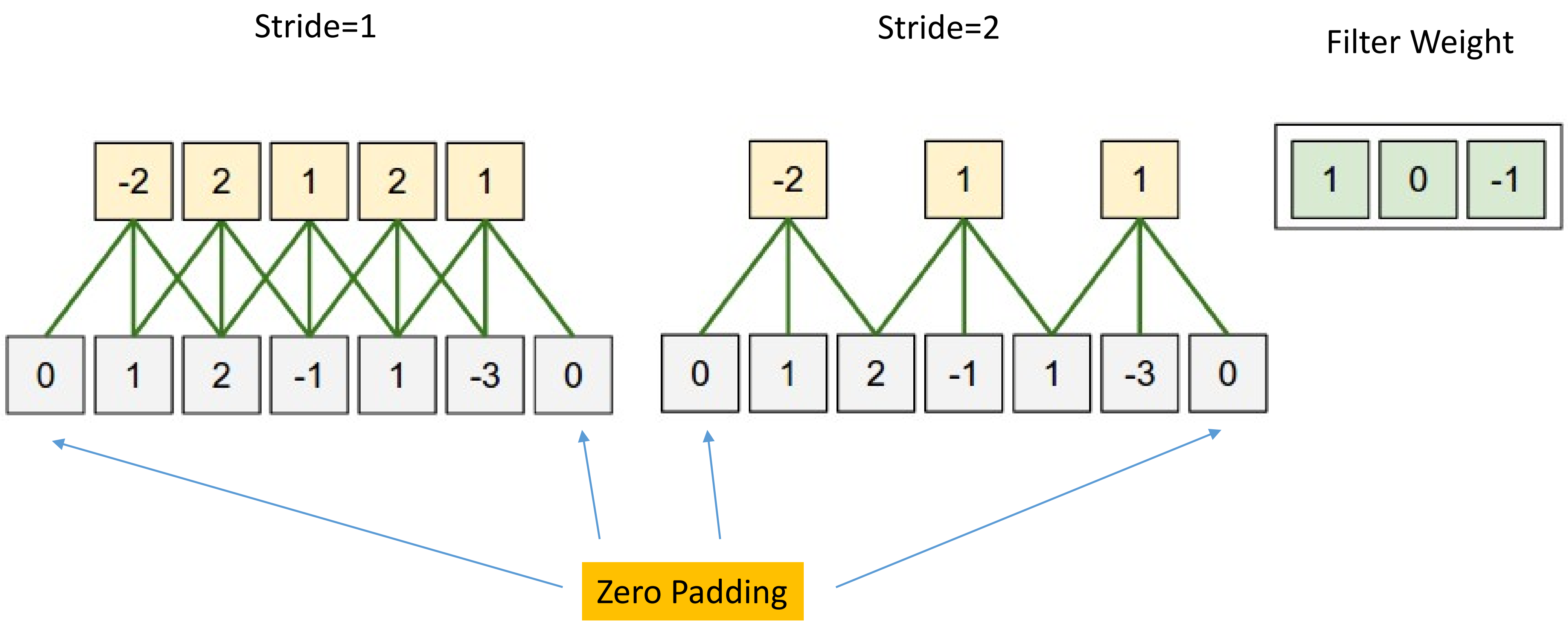
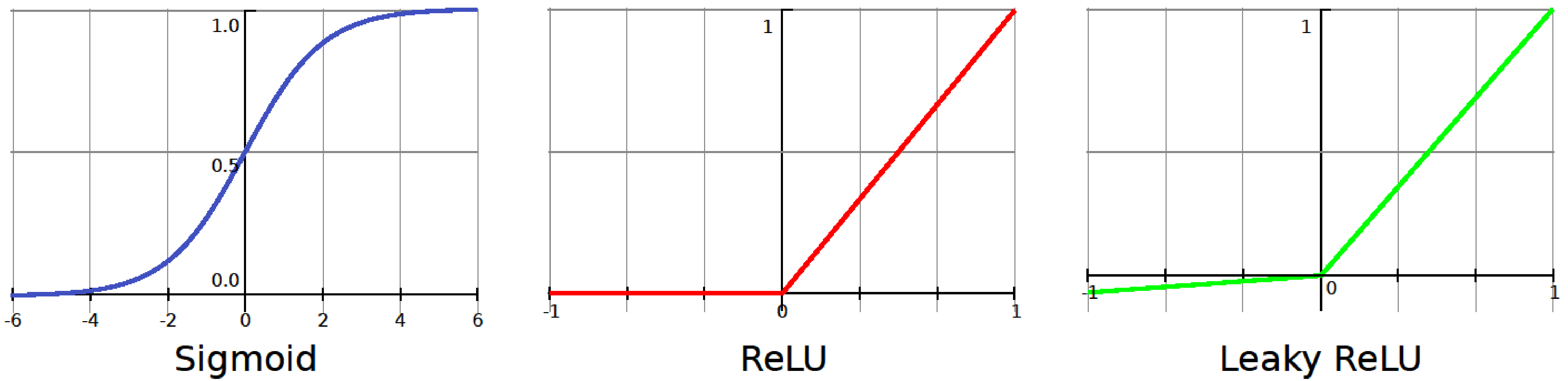


Figure Credit: Stanford CS231n Course Instruction

Nonlinear Activation



Pooling (or Down-Sampling)

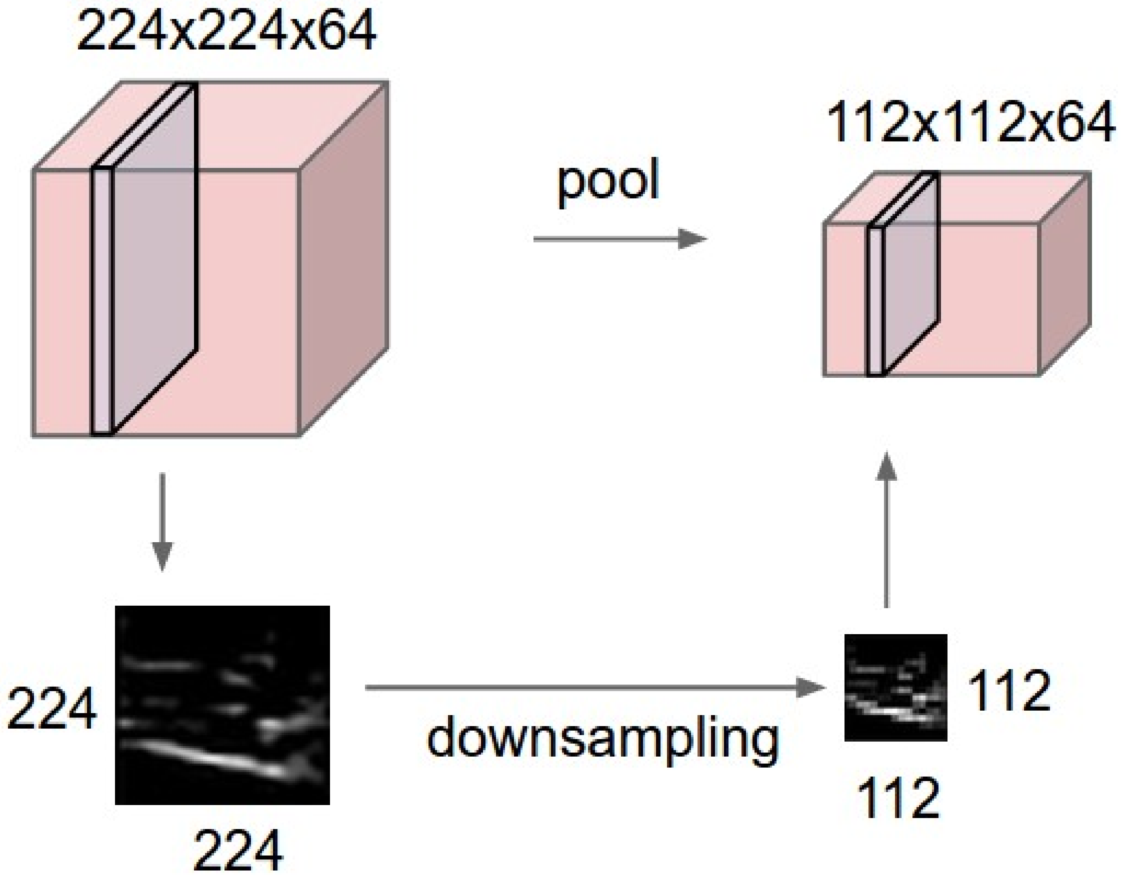


Figure Credit: Stanford CS231n Course Instruction

Maximum Pooling

- A common practice today

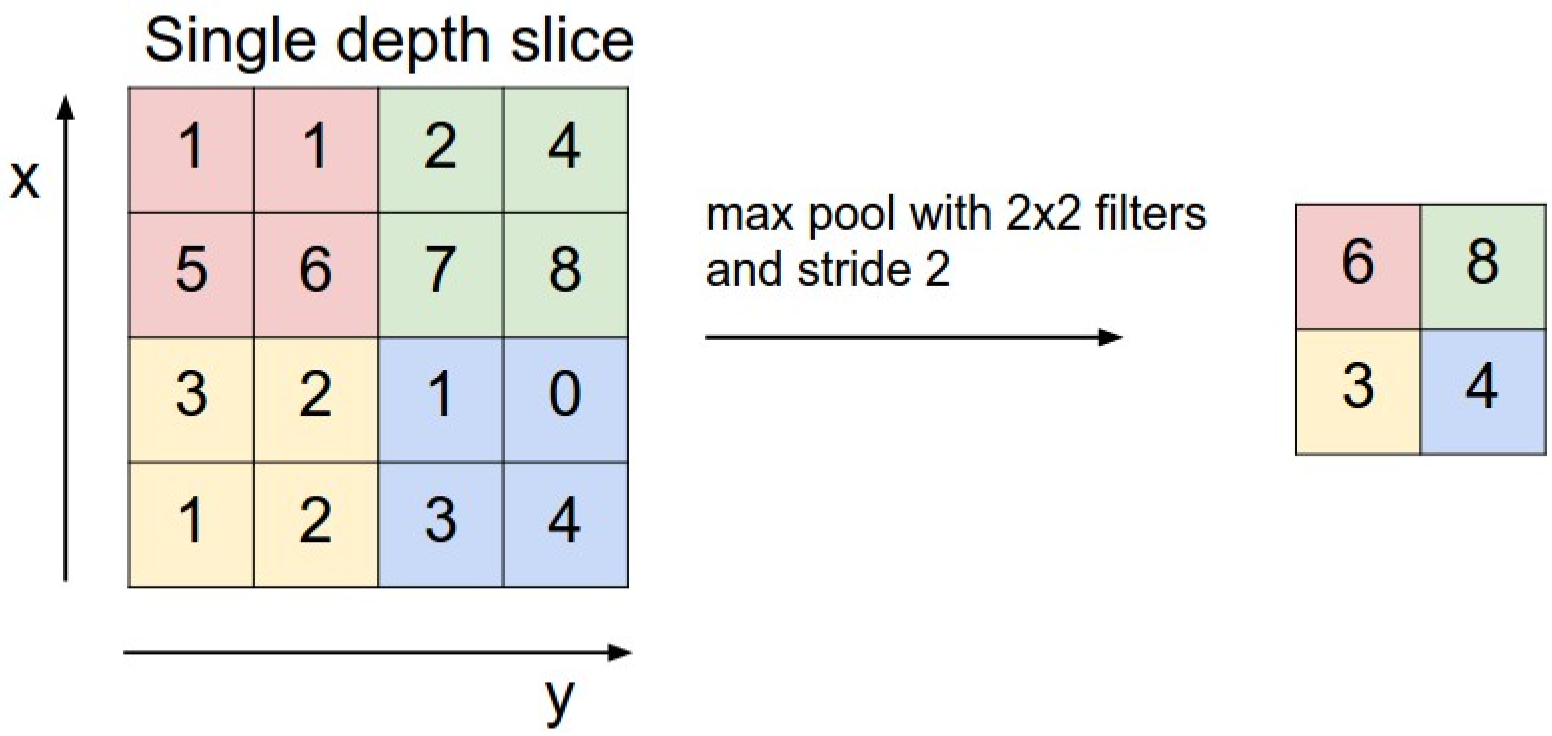
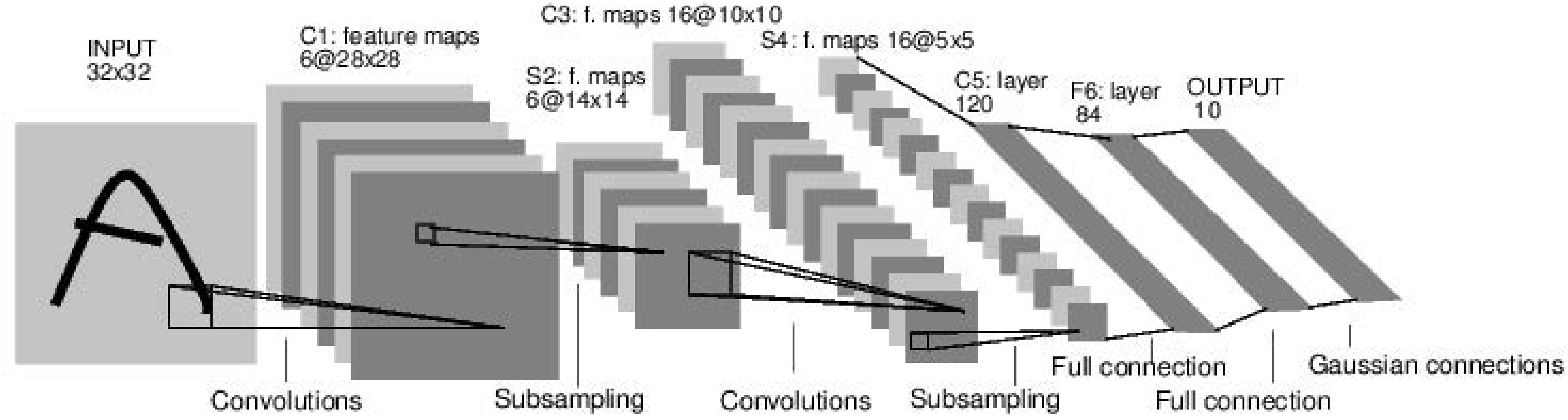


Figure Credit: Stanford CS231n Course Instruction

LeNet-5 Revisited (1)



Input → Convolution/Nonlinear Activation → C1

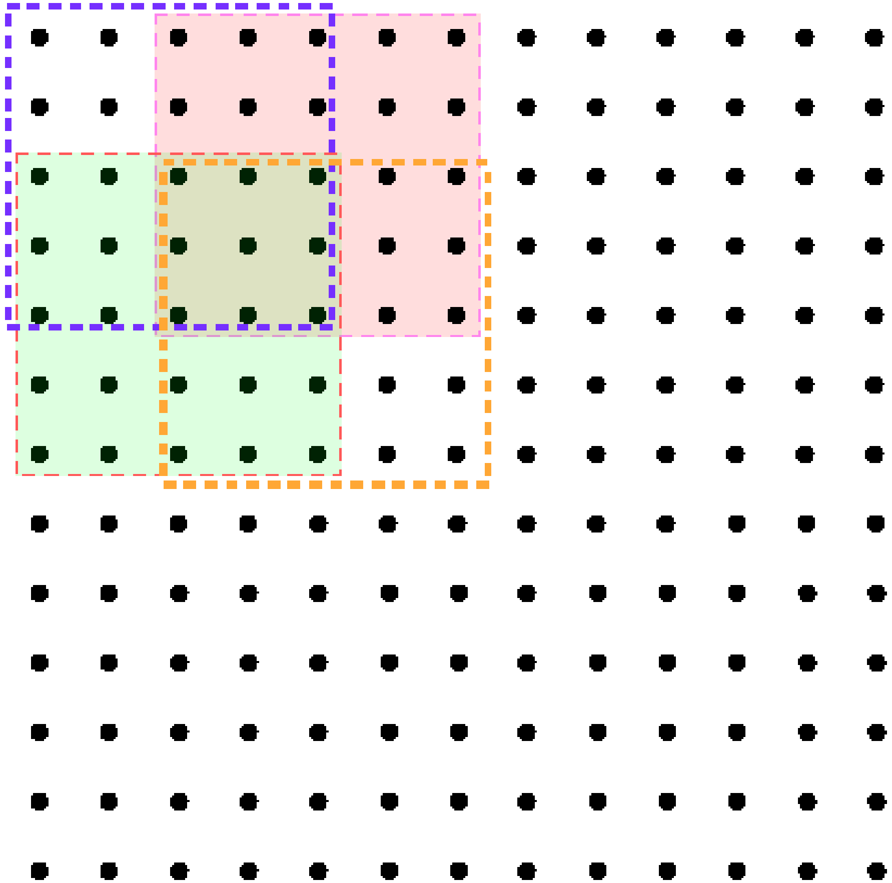
C1 → Pooling → S2

S2 → Convolution/Nonlinear Activation → C3

C3 → Pooling → S3

LeCun, Bottou, Bengio, Haffner, Gradient-based learning applied to document recognition, Proc. IEEE, 1998.

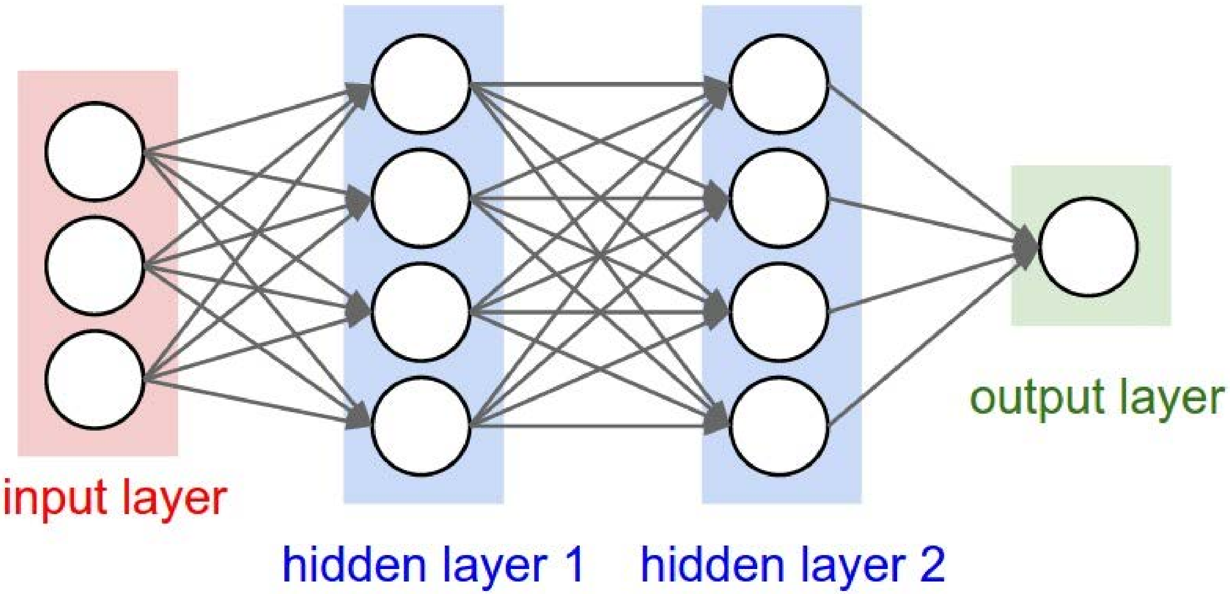
LeNet-5 Revisited (2)



Last Two Layers: Fully Connected (FC) Layers

- C5: Merge all spatial and spectral information by their weighted sum
  - 120 filters mean 120 different weighting schemes
  - Generating 120 filter responses
- F6: Dimension reduction from 120-D to 84-D
- Why these two FC layers?
  - To be further investigated
  - An analogy with a classical 3-layer artificial neural network (ANN)

A Classic 3-layer Artificial Neural Network (ANN)



C5 and F6 play the roles similar to hidden layer 1 and hidden layer 2, respectively.

Differences between LeNet-5 and ANN

- The Input to ANN is the data samples
- The Input to layer C5 in LeNet-5 is the output from S5 (a hybrid spatial-spectral feature space)
- Generally, a CNN consists of two sub-networks
  - Feature extraction sub-network (a feature vector extractor)
  - Decision sub-network (a classifier)
  - They are inter-connected
- Traditionally, a pattern recognition system is composed by two independent modules
  - Feature extraction module
  - Classification module

Output Layer

- A 10-D probability vector
- The desired output
  - If the input is “0”, set the desired output to [1,0,0,…,0]^T
  - If the input is “1”, set the desired output to [0,1,0,…,0]^T
  - If the input is “2”, set the desired output to [0,0,1,…,0]^T

  etc.
- If the output of a training sample is <u>V</u> with ground-truth of digit k, we can define the cost function as the Euclidean distance between <u>V</u> and its corresponding unit vector
- If the output of a testing sample is <u>W</u> whose length is normalized to one, we choose its element that has the highest probability and assign it to its associated digit
  - Example: <u>W</u> = [0, 0, 0.65, 0.1, 0, 0, 0, 0.2, 0, 0.05], then the answer is “2”.

Part III: CNN Training

CNN Training (1)

- Training data: labeled data samples
  - MNIST has 60,000 training samples
- Testing data: data samples with unknown labels for performance evaluation
  - MNIST has 10,000 testing samples
- How to train a CNN using the training data set?
  - Specify a network architecture
  - Initialize the filter weights
  - Search for the optimal filter weights to minimize a cost function

CNN Training (2)

- Define the cost function
- Initialization
  - Random initialization is often used
  - Could be tricky since poor initialization may lead to vanishing gradients (i.e. the weights can be changed)
- Minimizing the error (or the cost function) by backpropagation
  - Basically, the chain rule in calculus
  - Stochastic gradient descent (SGC)
    - The process is mechanical, check the following video clips:
    - https://www.youtube.com/watch?v=aVId8KMsdUU
    - https://www.youtube.com/watch?v=zpykfC4VnpM#t=19.618193

CNN Training (3)

- Standard gradient descent algorithm

It updates parameters θ of the objective J(θ) as,

$$\theta = \theta - \alpha \nabla\theta \, E[J(\theta)]$$

where the expectation in the above equation is approximated by evaluating the cost and gradient over the full training set.

- Stochastic Gradient Descent (SGD)

It simply does away with the expectation in the update and computes the gradient of the parameters using only a few training examples (called a minibatch). The new update is given by,

$$\theta = \theta - \alpha \nabla\theta \, J(\theta; x(i), y(i))$$

where (x(i),y(i)) are from the same minbatch and α is the learning rate . A typical minibatch size is 256.

CNN Training (4)

- Choosing the proper learning rate and schedule (i.e. changing the value of the learning rate as learning progresses) can be difficult
  - One standard method is to use a small enough constant learning rate that gives stable convergence in the initial epoch (full pass through the training set) or two of training and then halve the value of the learning rate as convergence slows down
  - Another approach is to evaluate a held out set after each epoch and anneal the learning rate when the change in objective between epochs is below a small threshold. This tends to give good convergence to a local optima
- Training data shuffling
  - If the data is given in some meaningful order, this can bias the gradient and lead to poor convergence. Generally a good method to avoid this is to randomly shuffle the data prior to each epoch of training.

CNN Training (5)

- If the objective has the form of a long shallow ravine leading to the optimum and steep walls on the sides, standard SGD will tend to oscillate across the narrow ravine since the negative gradient will point down one of the steep sides rather than along the ravine towards the optimum. The standard SGD can lead to very slow convergence particularly after the initial steep gains.
- Momentum is one method for pushing the objective more quickly along the shallow ravine.

CNN Training (6)

The momentum update is given by

$$v = \gamma\, v + \alpha\, \nabla\theta\, J(\theta; x(i), y(i))$$

$$\theta = \theta - v$$

where

- v is the current velocity vector which is of the same dimension as the parameter vector θ
- α is the learning rate although when using momentum α may need to be smaller since the magnitude of the gradient will be larger.
- $\gamma \in (0,1]$ determines for how many iterations the previous gradients are incorporated into the current update. Generally γ is set to 0.5 until the initial learning stabilizes and then is increased to 0.9 or higher.

Summary of Training Parameters

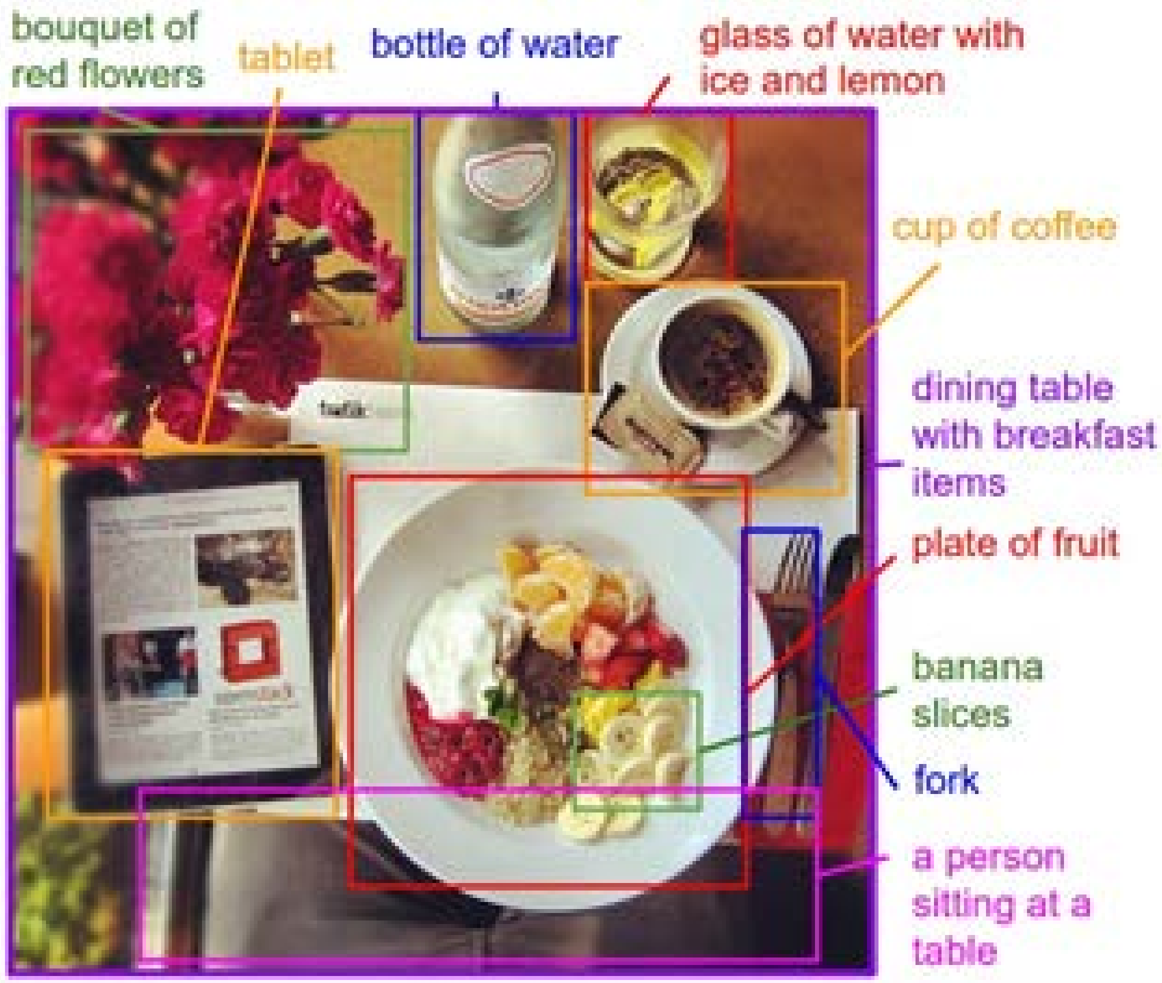
- Learning rate
- Mini-Batch
- Epoch
- Momentum

Part IV: Understanding CNNs

Bottlenecks of Today's Deep Learning

- No theory
  - Ad hoc engineering work (try and error) rather than science
- Heavily supervised learning
  - Where to get high quality labeled data
  - Expensive and tedious

A Labeled Image Example



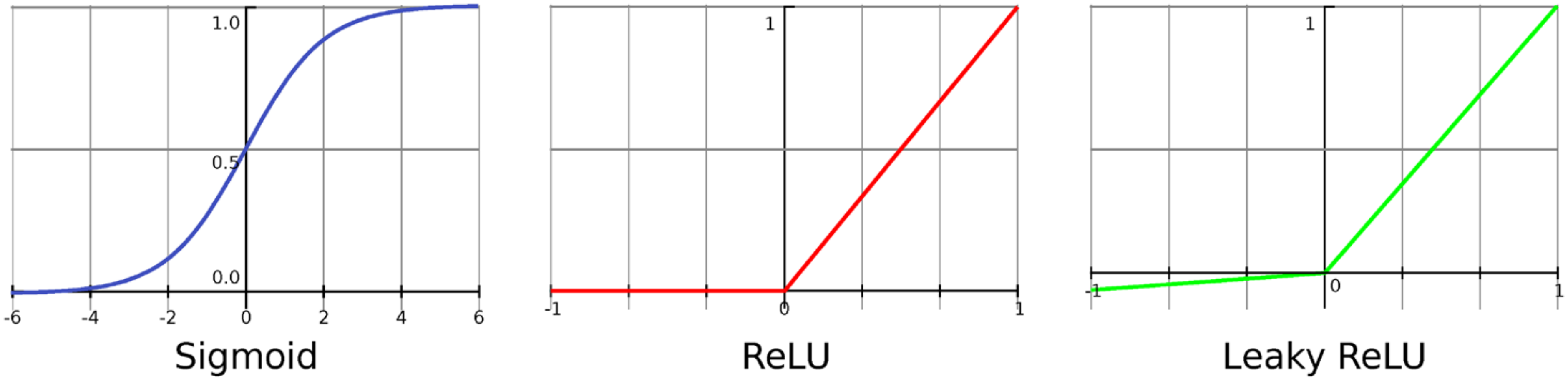
More on CNN's Limitations

- Poor localization
- No good solution to small object detection
  - Partially solved by localization

CNN Theory

- Two key questions
  - Why nonlinear activation is essential?
  - Why a cascade of two convolution layers is better than one single layer?
- A recent paper to address these two questions:
  - C.-C. Jay Kuo, "Understanding Convolutional Neural Networks with A Mathematical Model", arXiv:1609.04112

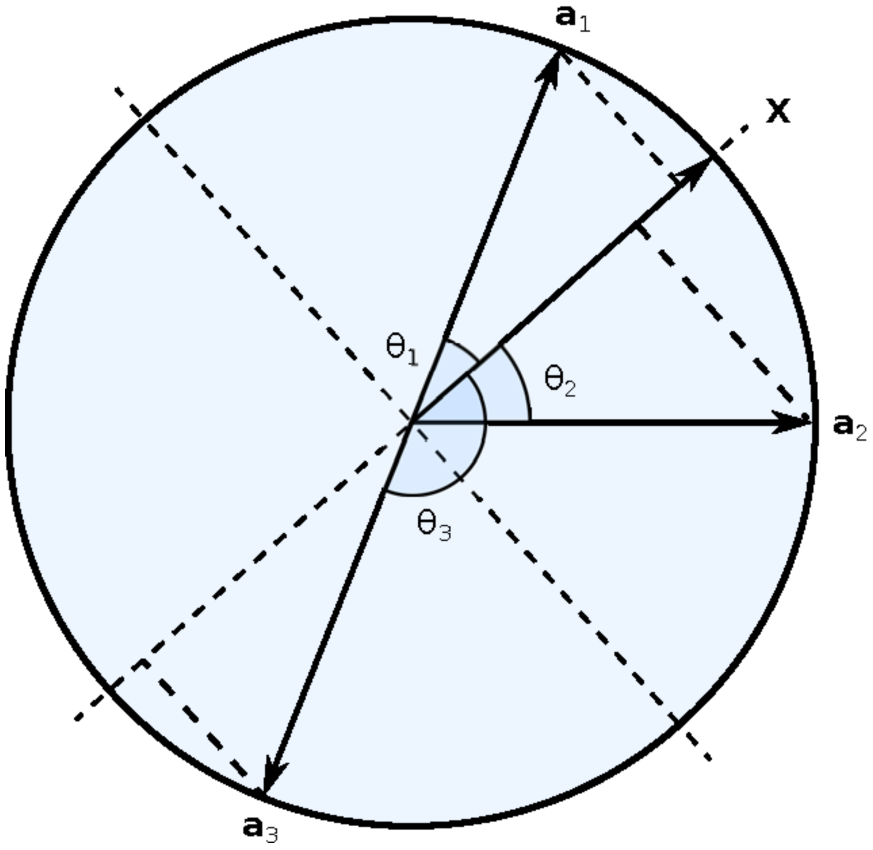
Nonlinear Activation Functions - Revisited



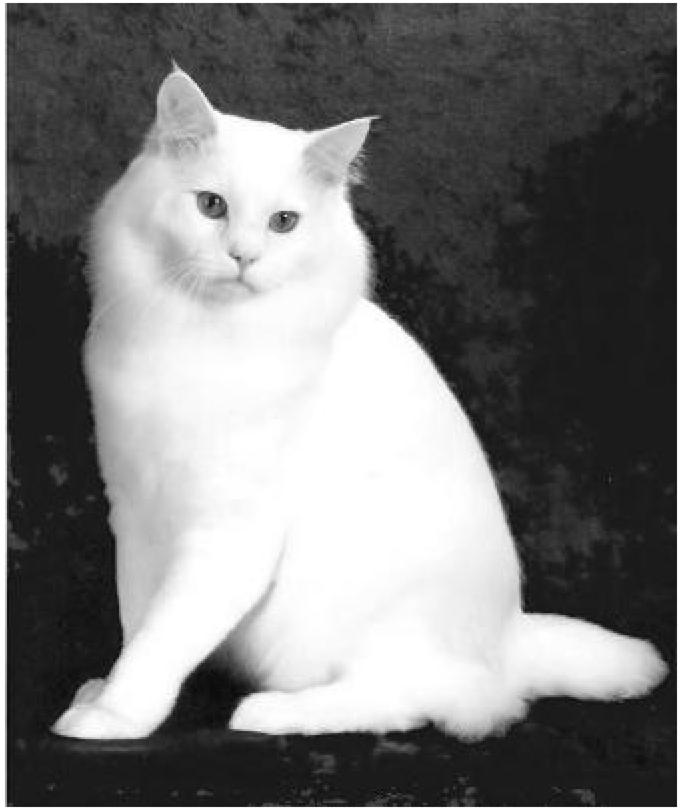
How CNN Stores "Learned Results"?

- All training/learning results are summarized in filter weights
  - Filter weights play a critical role in understanding CNN
  - Called the "anchor vectors"

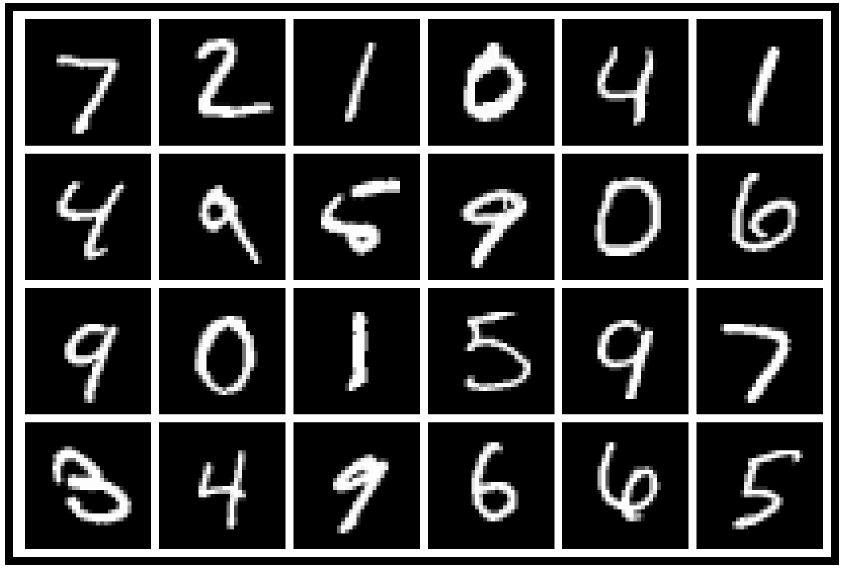
REctified COrrelation on a Sphere (RECOS) Model



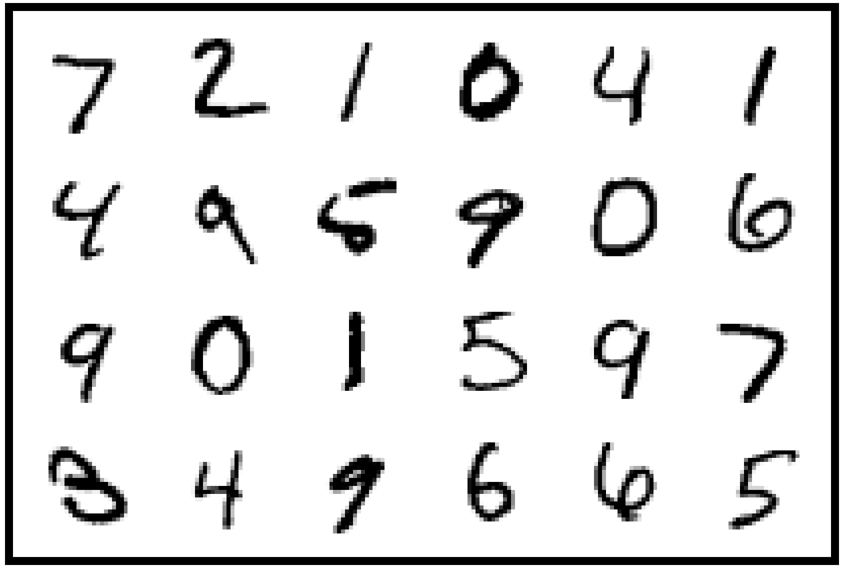
Comparison of Positive & Negative Correlations



Experiments on MNIST



Original



Negative

Performance of LeNet-5

- Original: 98.94%
- Negative: 37.36%

Other Related Questions

- How many convolutional layers?
- How many filters at each layer?
- What is the filter size?

    ….

  Actually, it is a matter of design trade-off

Part V: CNN Architectures

Pioneering Work

- McClulloch and Pitts (M-P) neuron model (1943)
  - “all-or-none” characteristics (logic unit)

$$y = \operatorname{sgn}(wx - \varphi)$$

$x = (x_1, x_2, \cdots, x_n)^T$—an input vector

$w = (w_1, w_2, \cdots, w_n)$—a weight vector

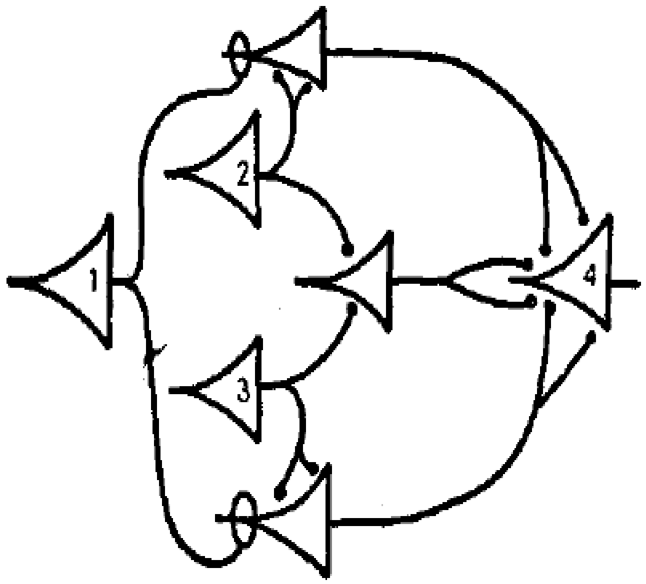
φ—a threshold

$$\operatorname{sgn}(v) = \begin{cases} 1, & v > 0 \\ -1, & v \leq 0 \end{cases}$$

McCulloch WS, Pitts W. A logical calculus of the ideas immanent in nervous activity. The bulletin of mathematical biophysics. 1943 Dec 1;5(4):115-33.

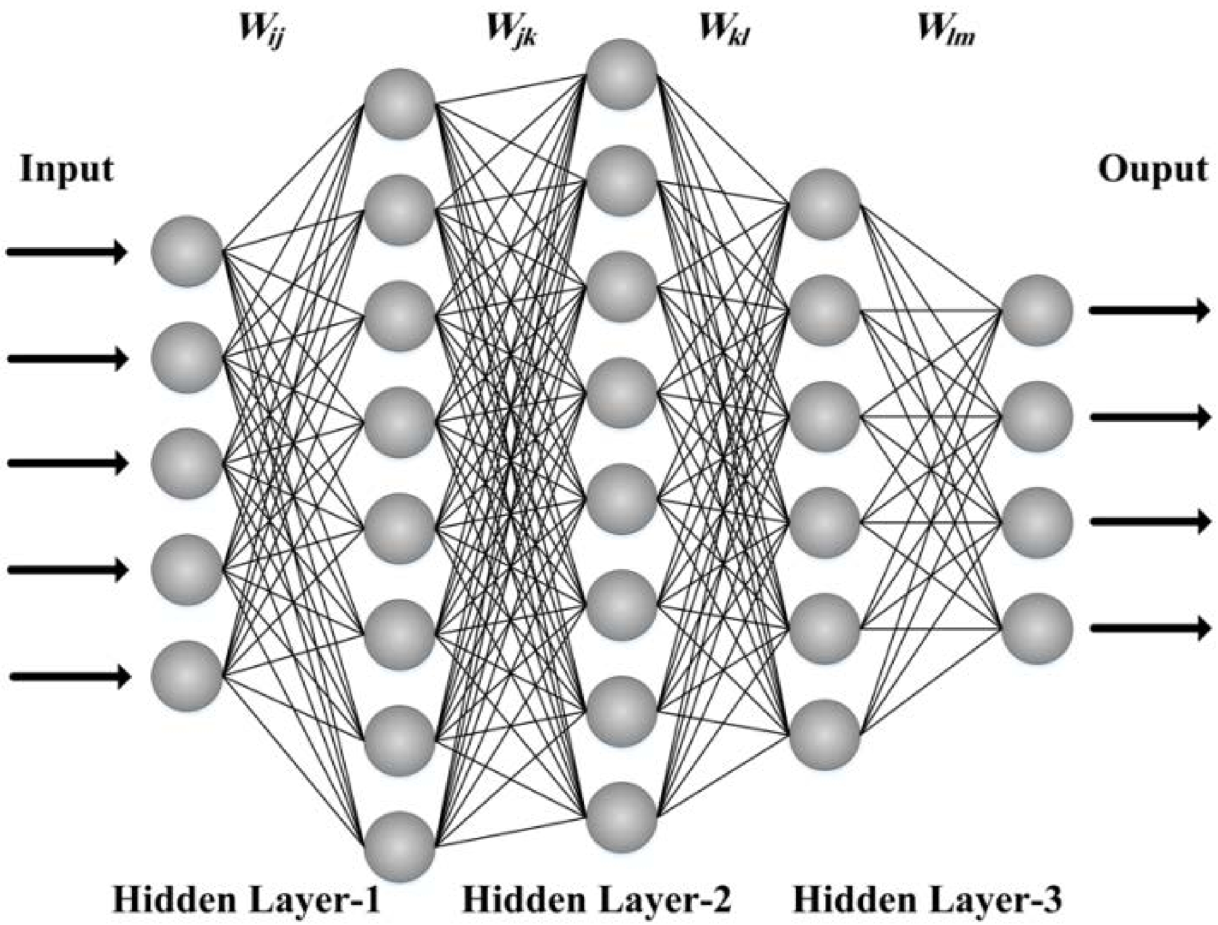
M-P Model

- It contains some basic elements
  - convolution operation
  - bias term
  - nonlinear activation
- What M-P model does not have?
  - No parallelism
  - No training (a feedforward network)



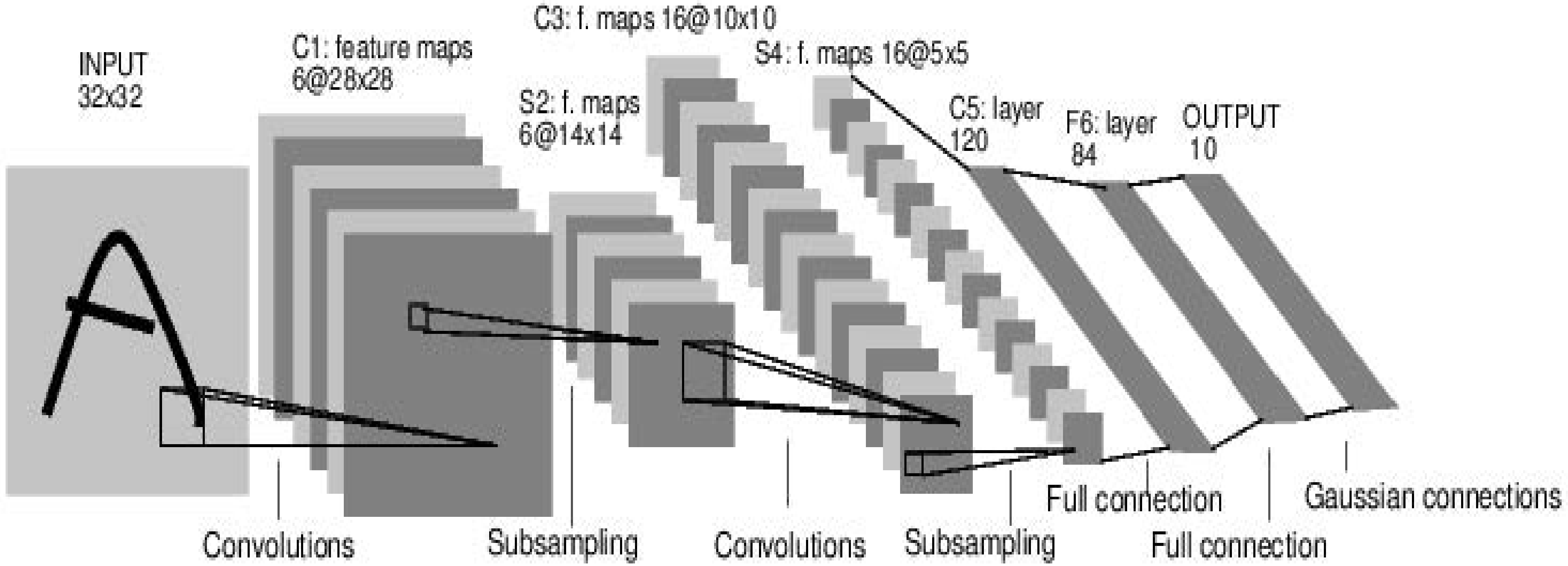
Multilayer Perceptron (MLP) with Backpropagation (BP) Training

- Highly parallelism
- Fully connection between every two adjacent layers
- No connection between neurons at the same layer
- Supervised learning by BP
- Artificial neural network (ANN) – late 80s and early 90s



Modern Convolutional Neural Network (CNN)

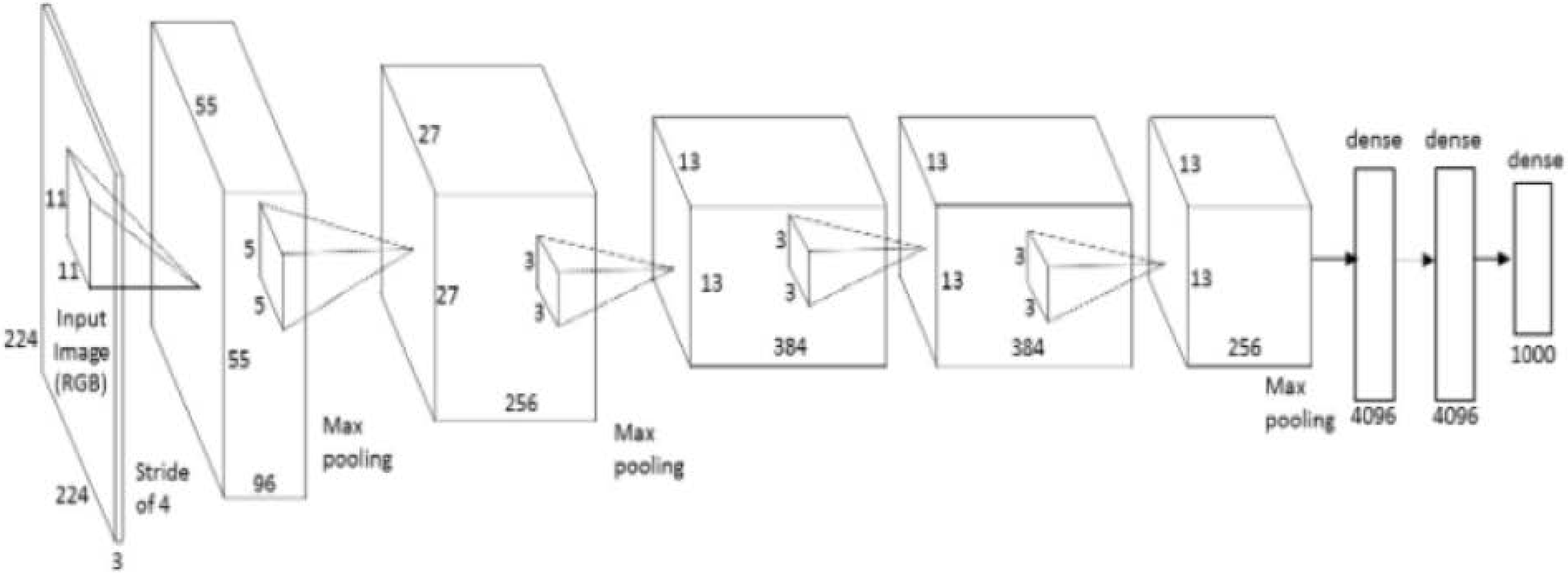
- LeNet-5



- Can handle image input by block partitioning
- Convolutional layers -> feature extraction module
- Fully connected layers -> decision module
- Two modules are back-to-back connected

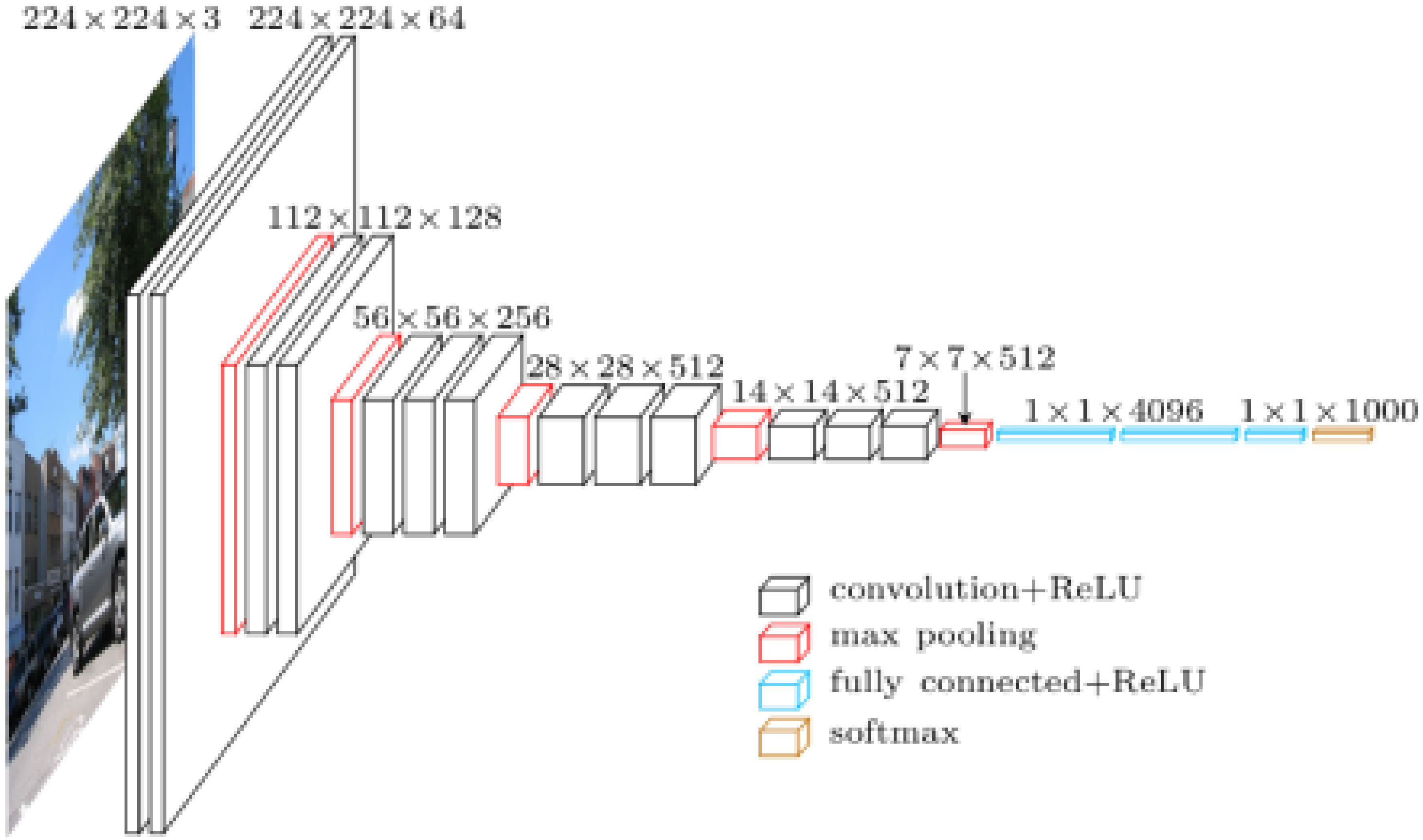
Pyramidal CNNs

- AlexNet



Pyramidal CNNs

- VGG-16

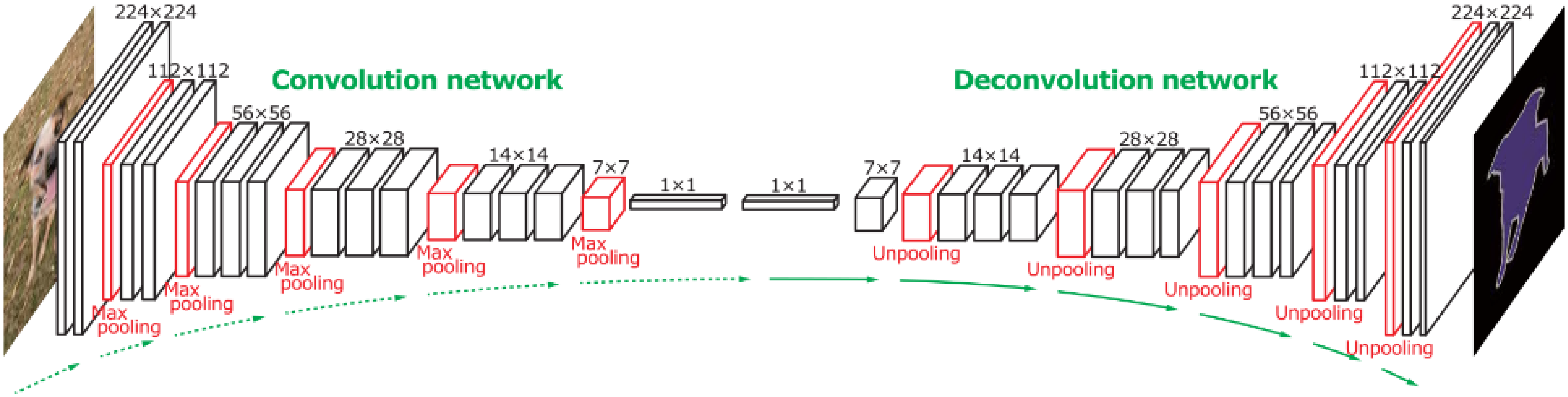


Comments

- Pyramidal CNNs
  - Input – an image
  - Output – a class label
  - Not suitable for image processing (where the input and the output are both images)
- How to design CNN for the image processing purpose?

A Straightforward Architecture

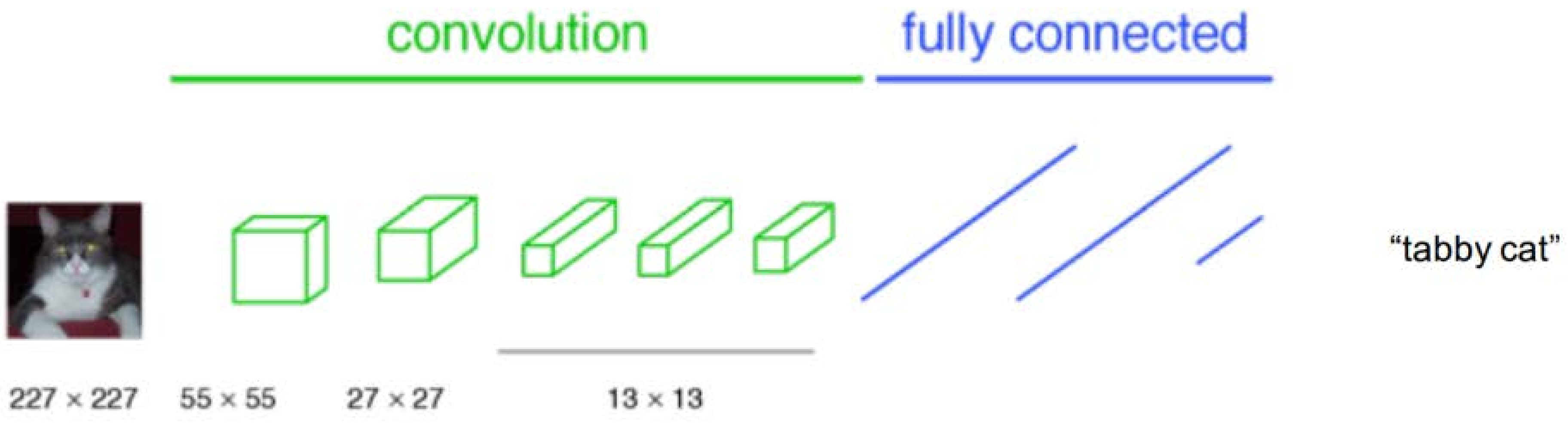
DeconvNet



Fully Convolutional Network (FCN)

- Evolution

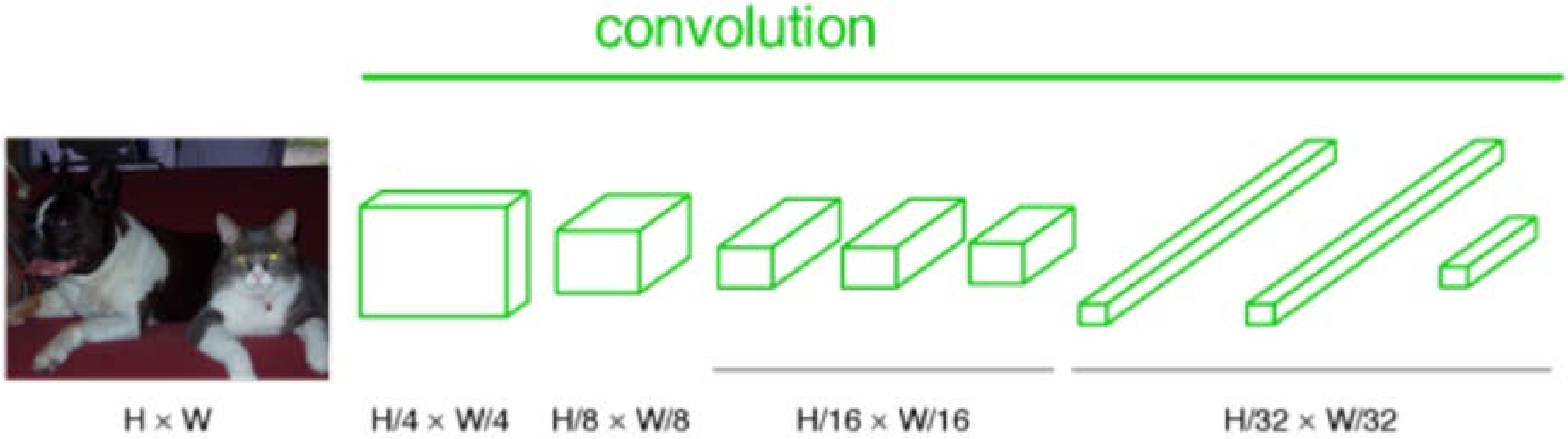
Step 1:



Fully Convolutional Network (FCN)

- Evolution

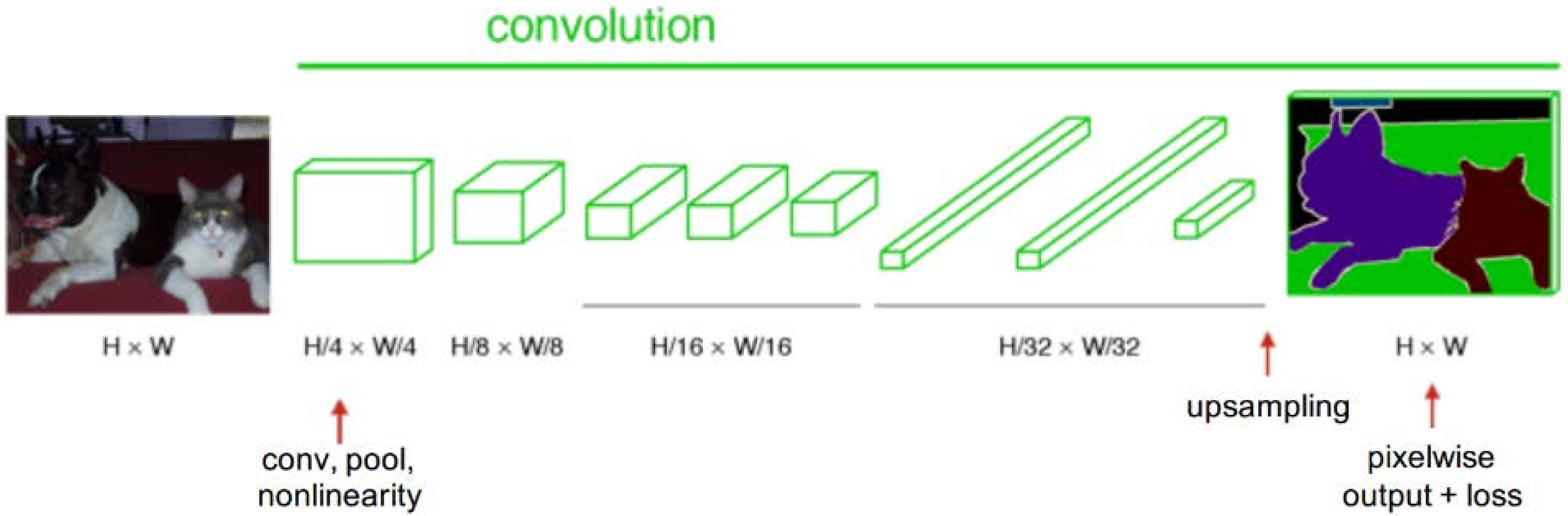
Step 2:



Fully Convolutional Network (FCN)

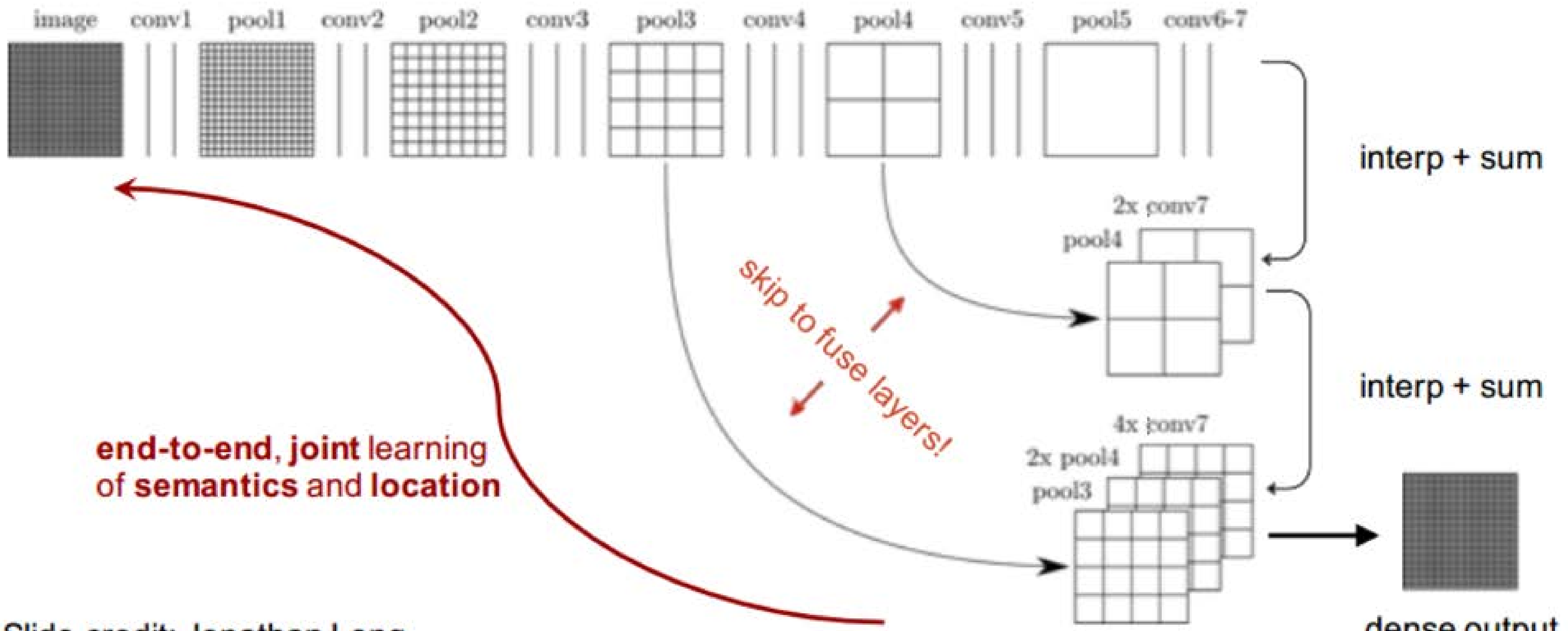
- Evolution

Step 3:



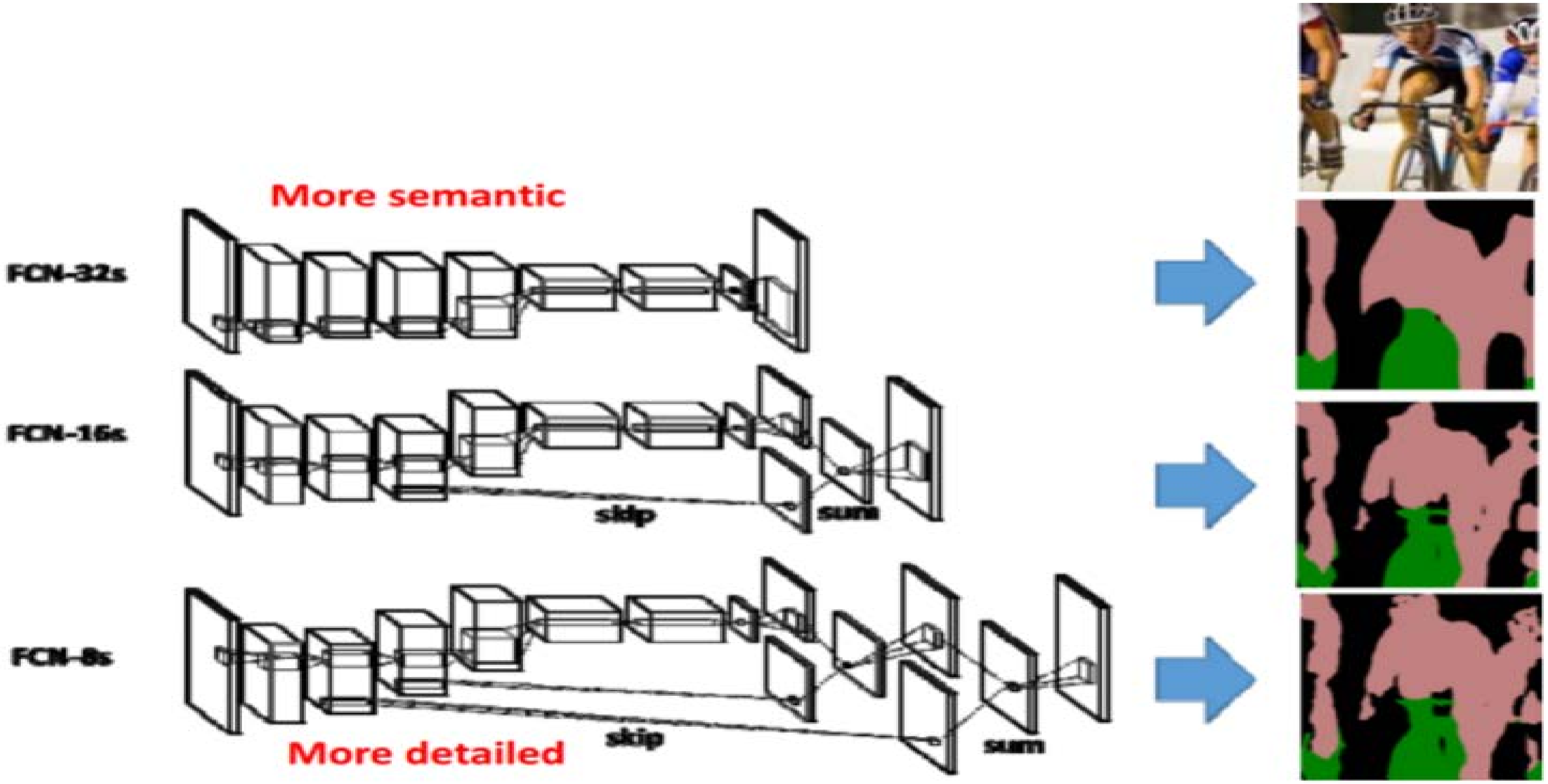
Fully Convolutional Network (FCN)

- How to do up-sampling

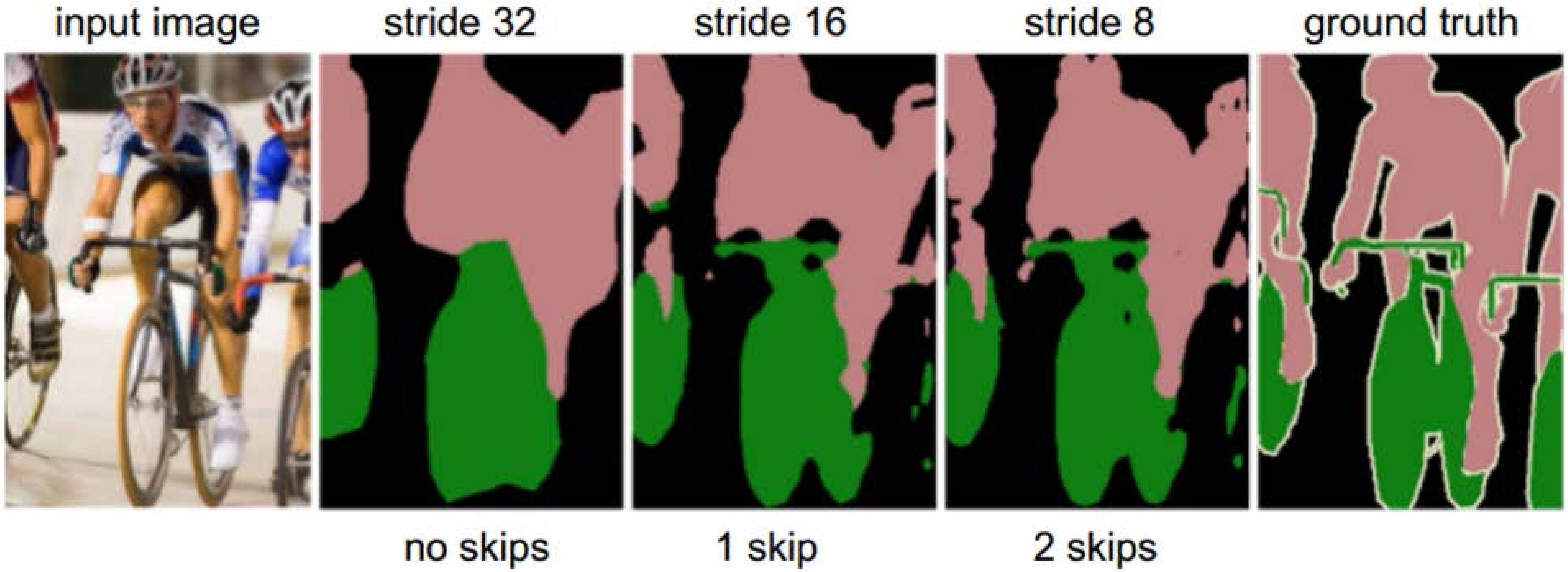


Slide credit: Jonathan Long

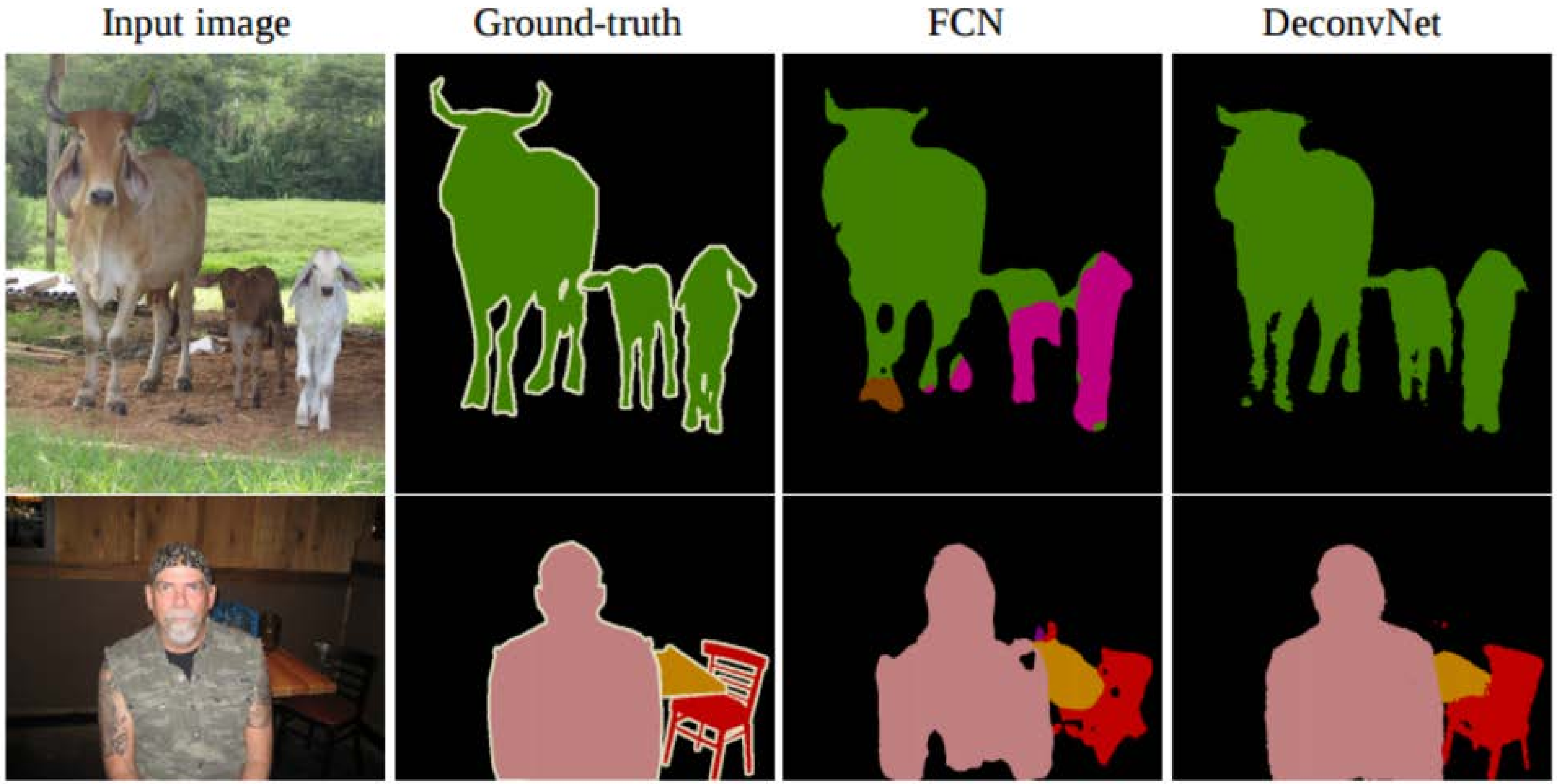
FCN with Skips



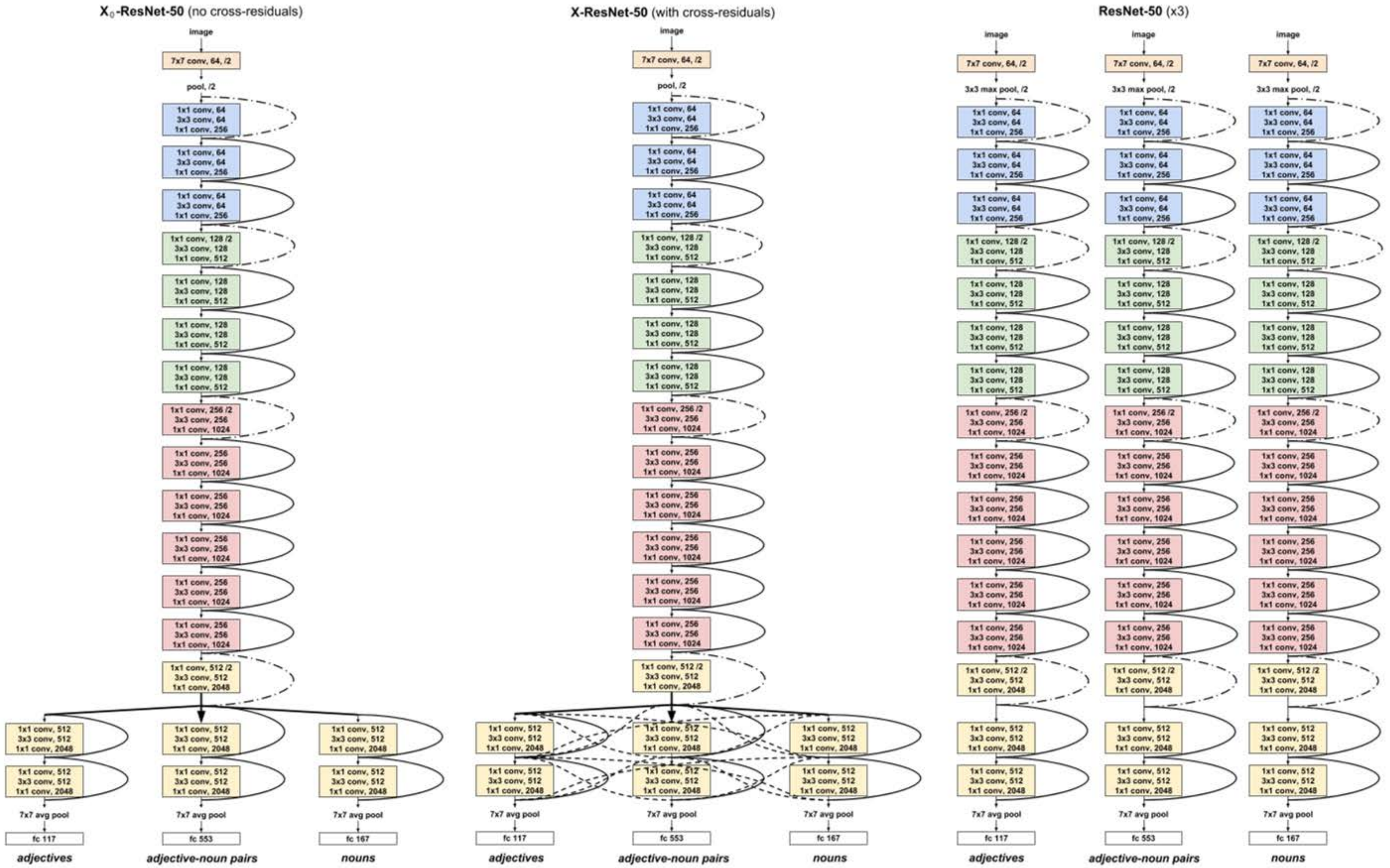
FCN Semantic Segmentation Results



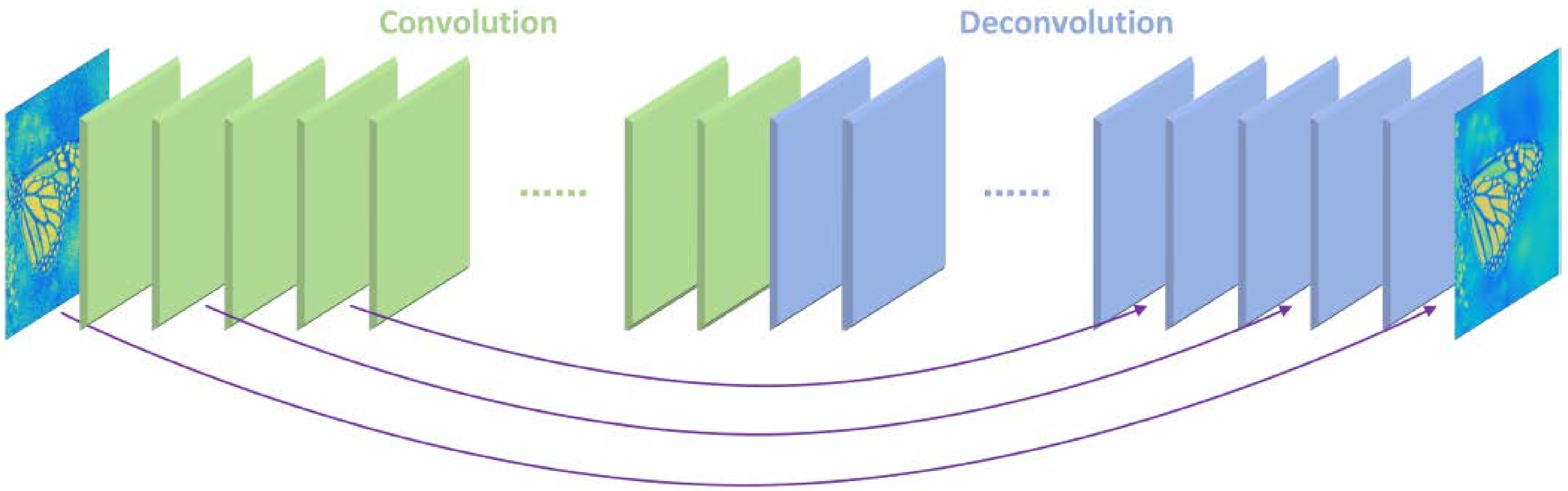
Performance Comparison between DeconvNet and FCN



Residual Networks (or ResNet)



Residual Networks



Why residual networks?

- Allow both shallow and deep networks to co-exist
- Local errors can be corrected by shallow networks
- Global errors should be corrected by deep networks
- One application scenario: super-resolution