Convolutional Neural Networks

Resana Machine Learning Workshop

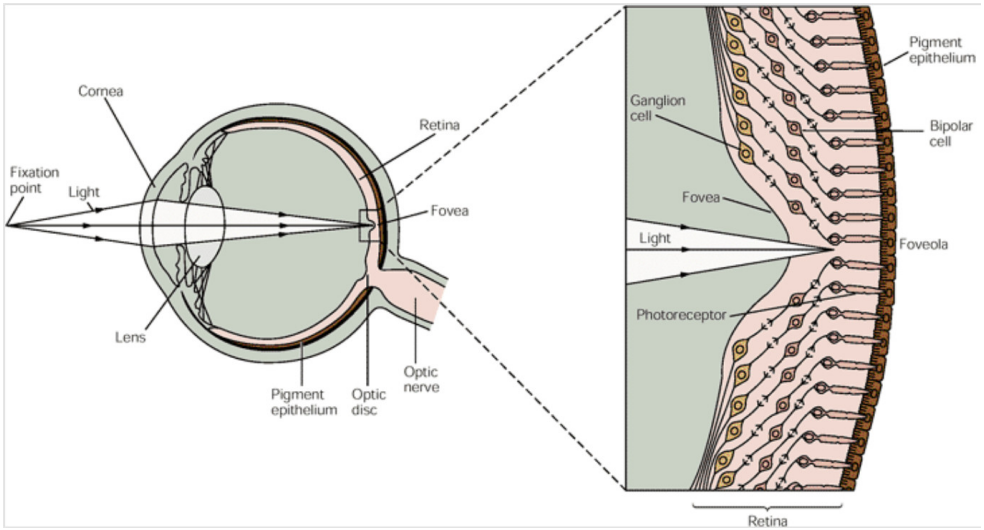
prepared by Hosein Hasani

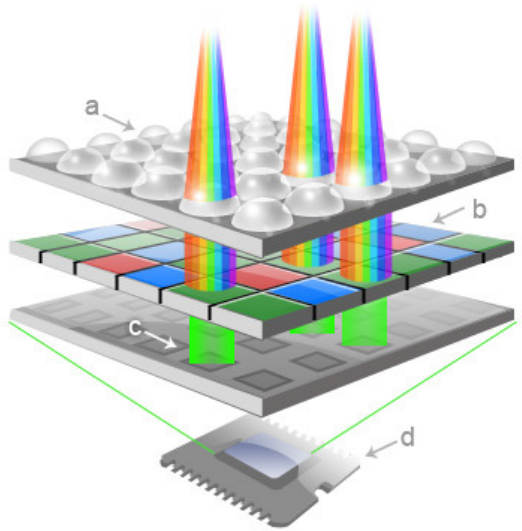
Summer 1399

Outline

- Image Formation
- Biological Inspirations
- Convolutional Neural Networks
  - Convolution Kernels
  - Convolution Layer
  - Parameters Settings
  - Convolution Layer VS FC Layer
  - Receptive Field
  - Feature Visualization
  - Pooling Layer
- Architectures for CNNs
- Transfer Learning
- Applications in Computer Vision Tasks

Image Formation





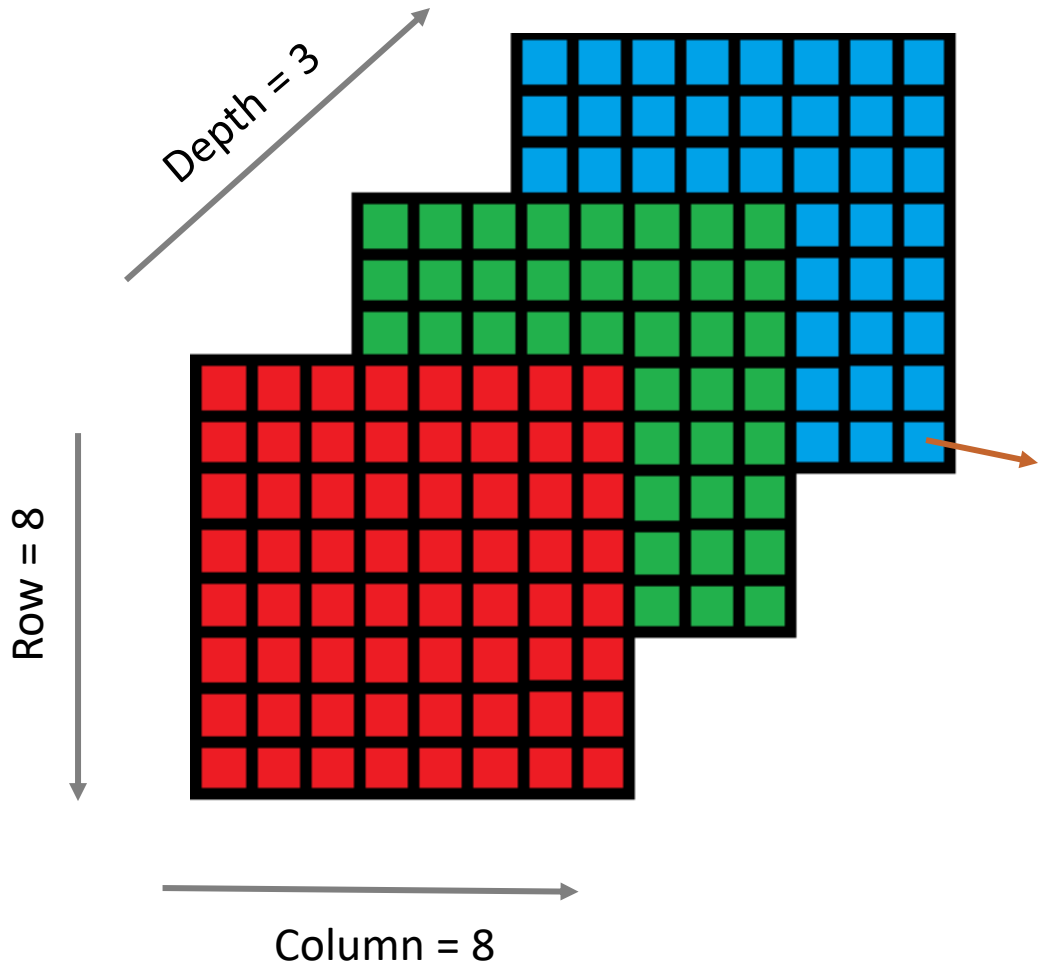
Digital camera sensors array

Photoreceptor cells → Cones: color vision

Photoreceptor cells → Rods: night vision

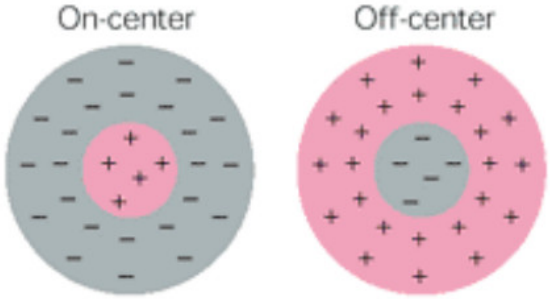
[Kandel et al. *Principles of neural science*. 4th edition.]

Image Formation

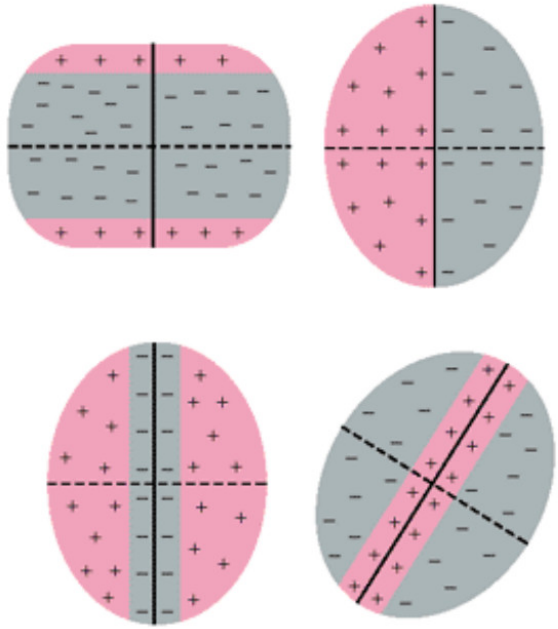


Biological Inspirations: Filtering

Receptive fields of concentric cells of retina and lateral geniculate nucleus

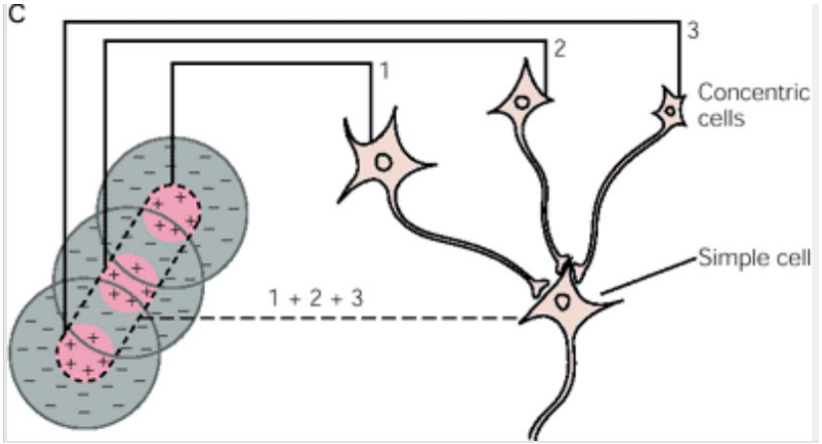


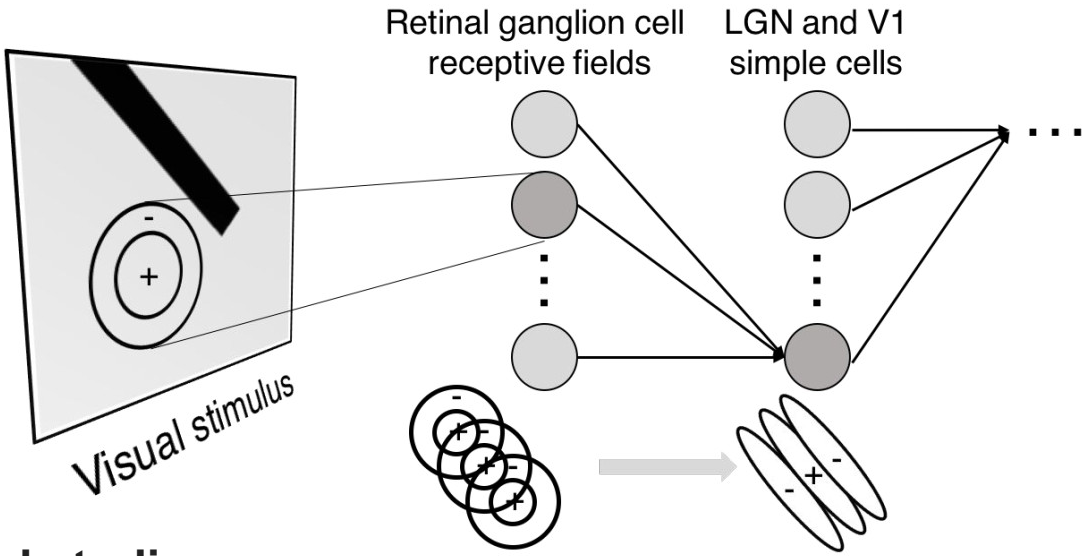
Receptive fields of simple cells of primary visual cortex



[Kandel et al. *Principles of neural science*. 4th edition.]

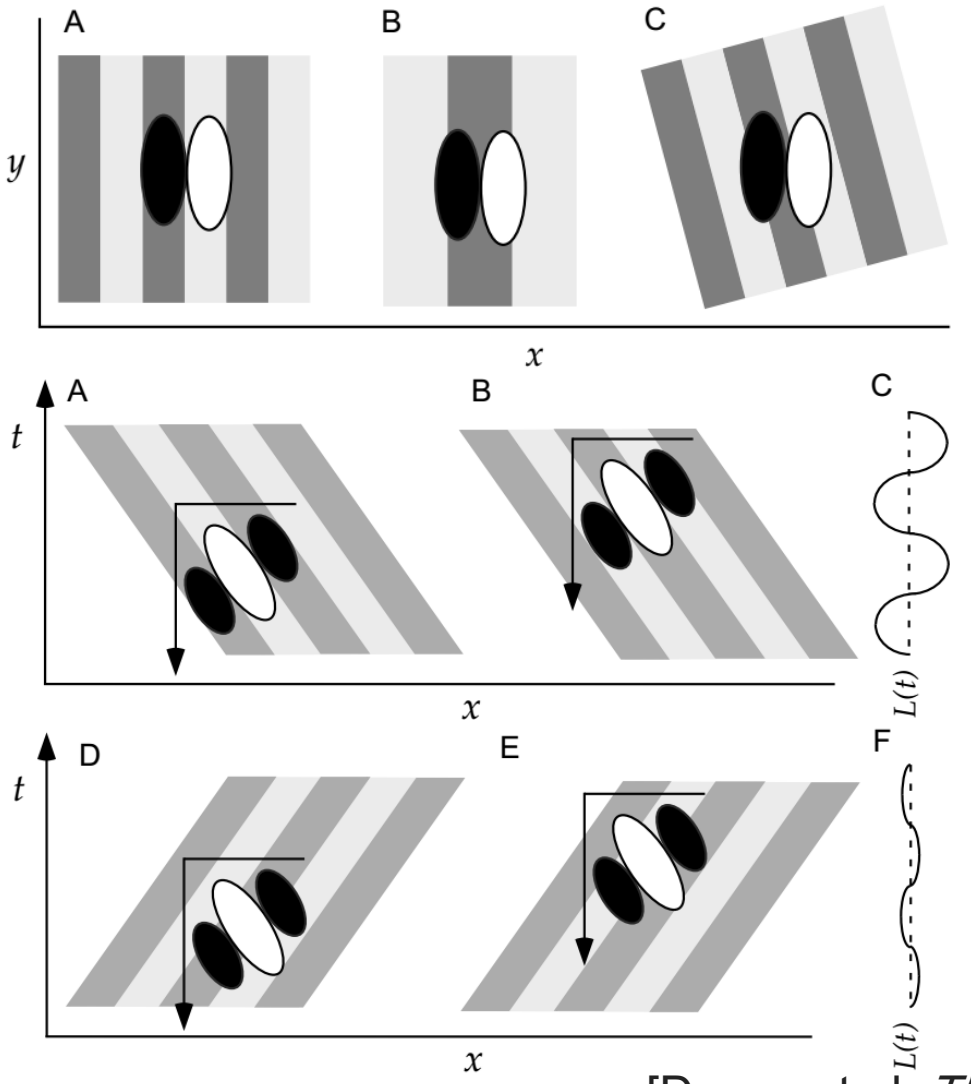
Hierarchical Receptive Fields in the Brain





Hubel and Wiesel studies

Response to Visual Patterns



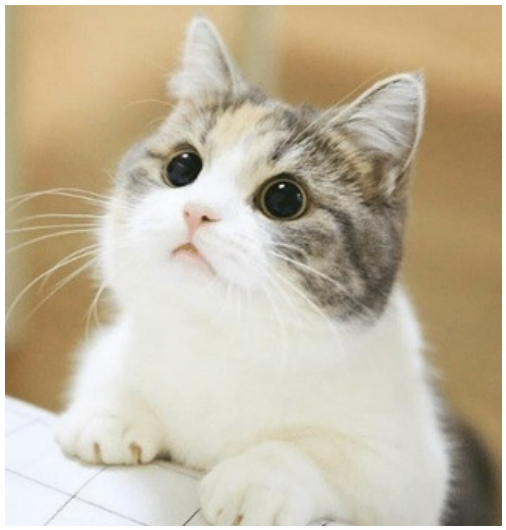
Primary visual cortex cells

[Dayan et al. *Theoretical neuroscience*.]

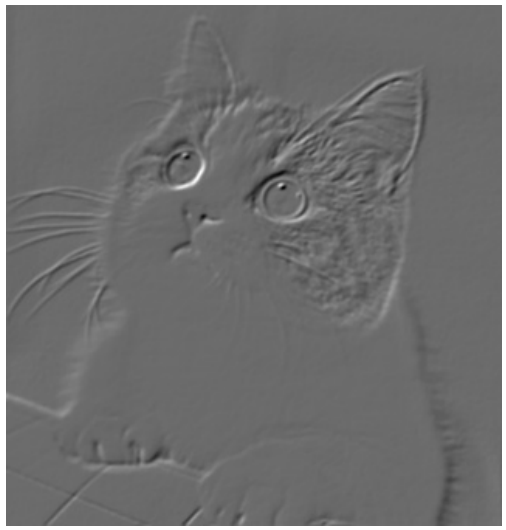
Sobel filters

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

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

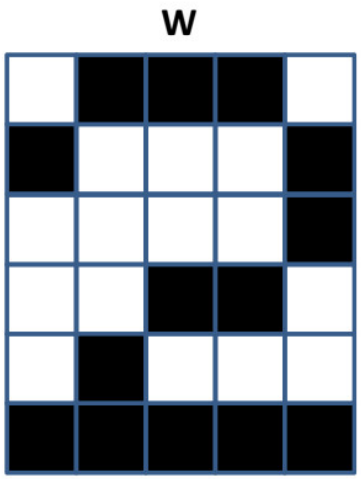


$image$

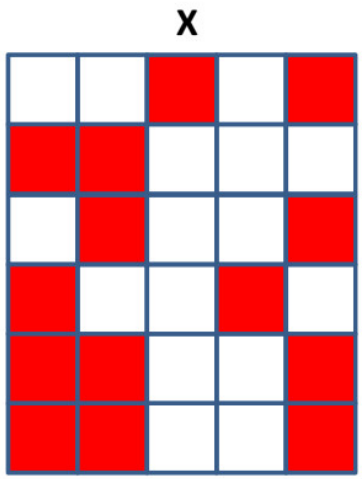


$$0.5 * \frac{d}{dx} image + 0.5 * \frac{d}{dy} image$$

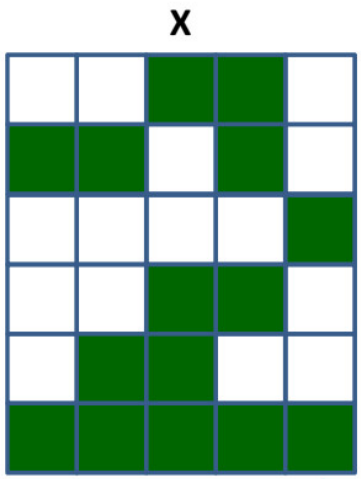
Filter (kernel) as a “template”



filter

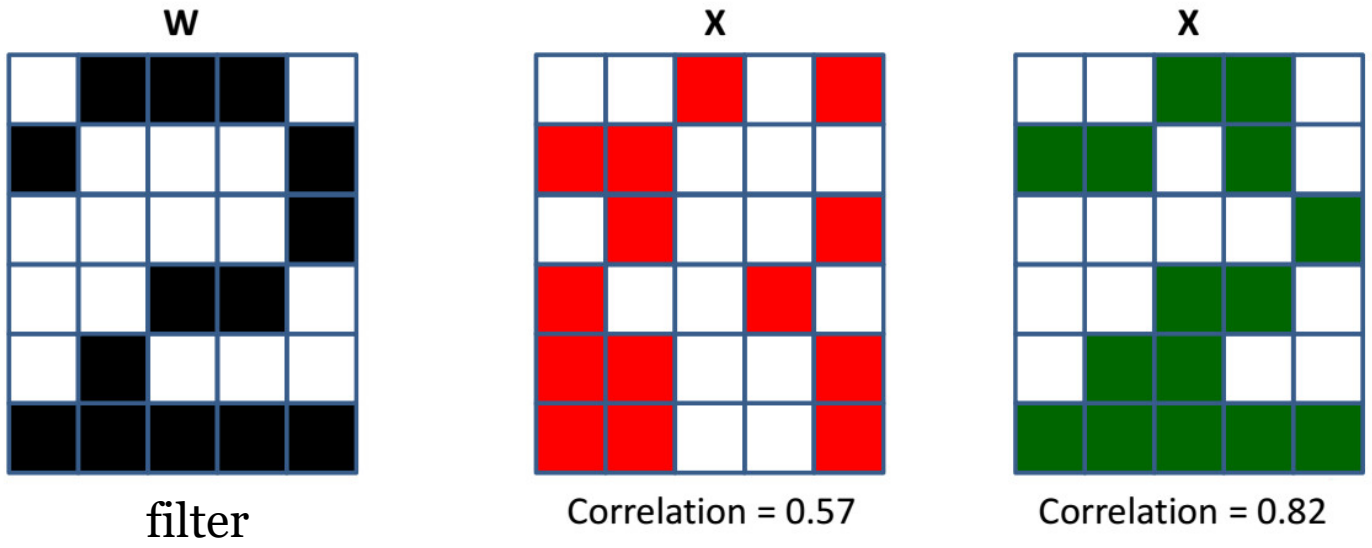


Correlation = 0.57



Correlation = 0.82

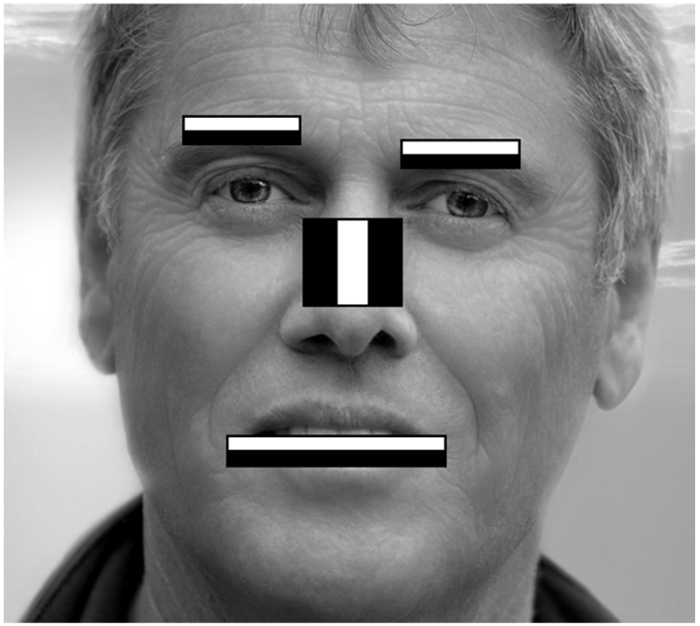
Filter (kernel) as a “template”



- The green pattern looks more like the weights pattern (black) than the red pattern
  - The green pattern is more *correlated* with the weights

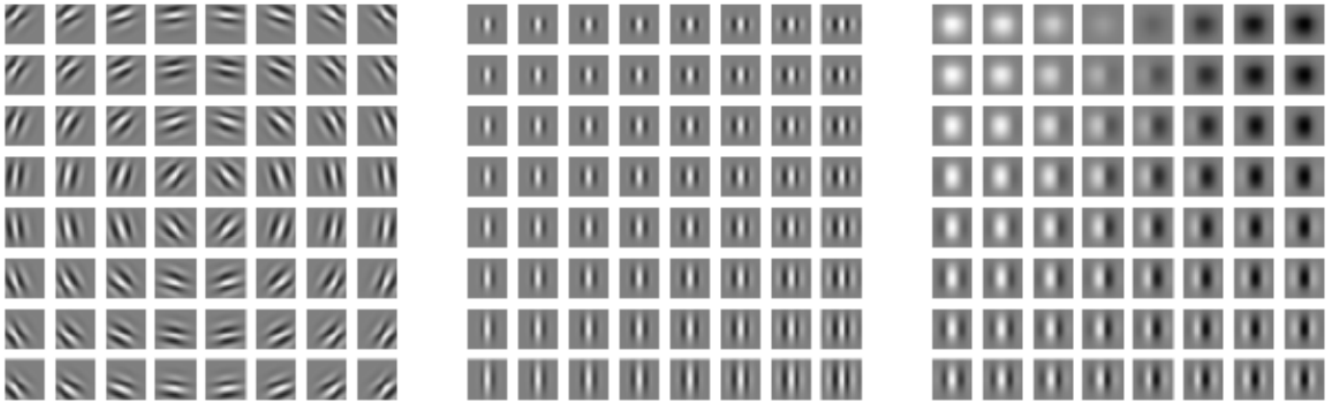
Feature Extraction

- Filters have great role in feature extraction.
- Better features provides more accurate models.



Gabor filters

- Scanning for patterns!
- Extract frequency features from local image area.



[Goodfellow et al. 2016]

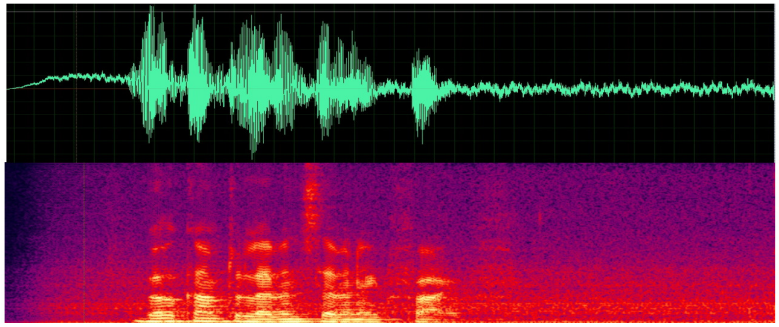
$$g(x, y; \lambda, \theta, \phi, \sigma, \gamma) = \exp\left(-\frac{x'^2 + \gamma^2 y^2}{2\sigma^2}\right) \exp(i(2\pi \frac{x'}{\lambda} + \phi)$$

$$x' = x\, cos\theta + y\, sin\theta$$

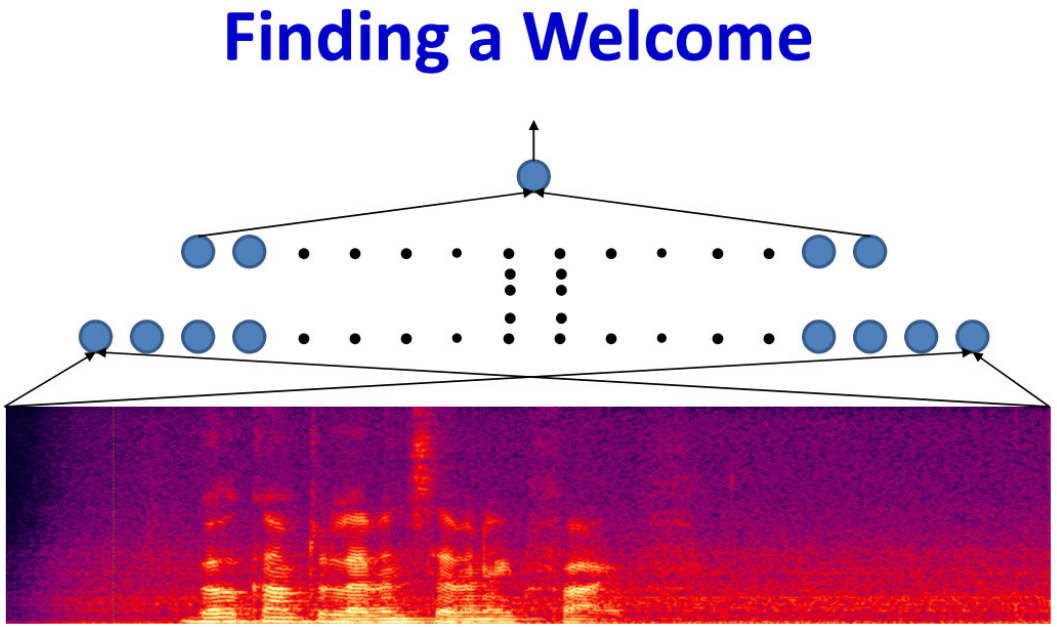
$$y' = -x\, sin\theta + y\, cos\theta$$

Problem with MLP

- How to classify high dimensional (audio, image) signals?

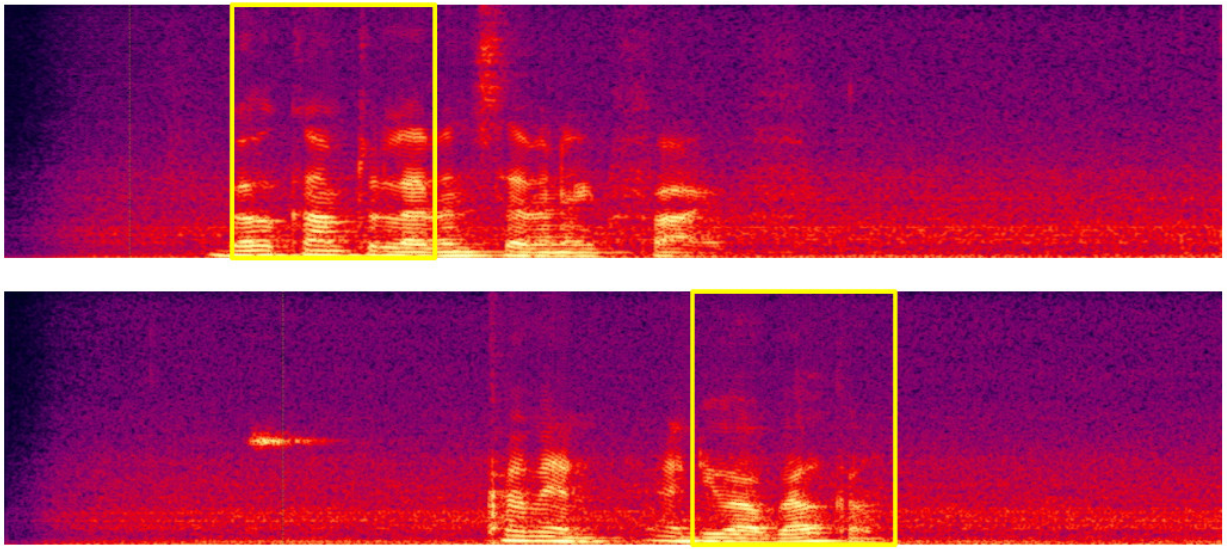


Problem with MLP



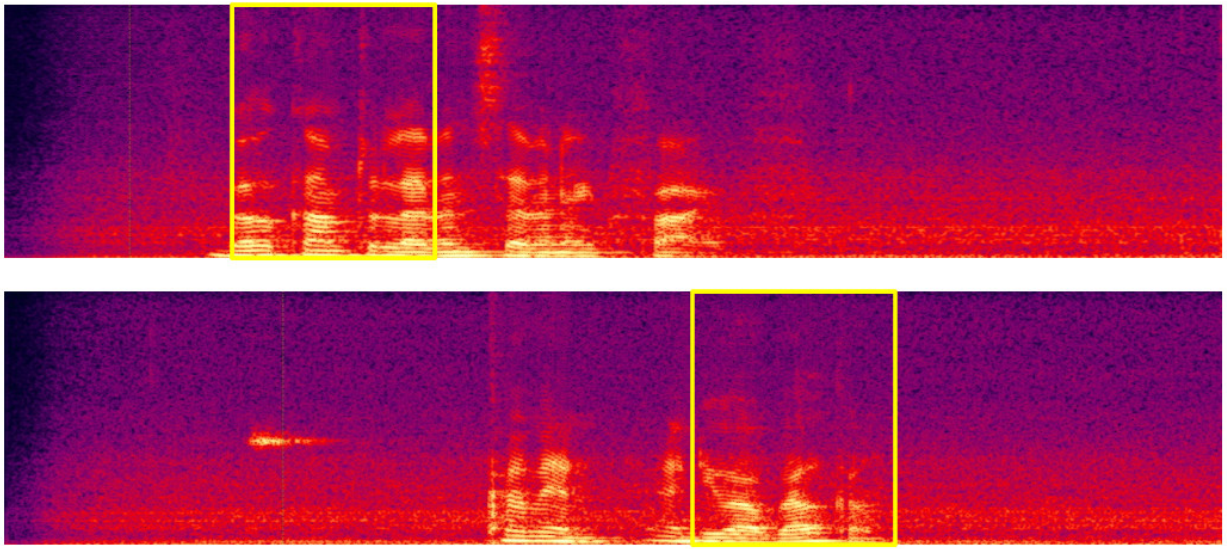
- Trivial solution: Train an MLP for the entire recording

Problem with MLP



- Problem with trivial solution: Network that finds a “welcome” in the top recording will not find it in the lower one
  – Unless trained with both
  – Will require a very large network and a large amount of training data to cover every case

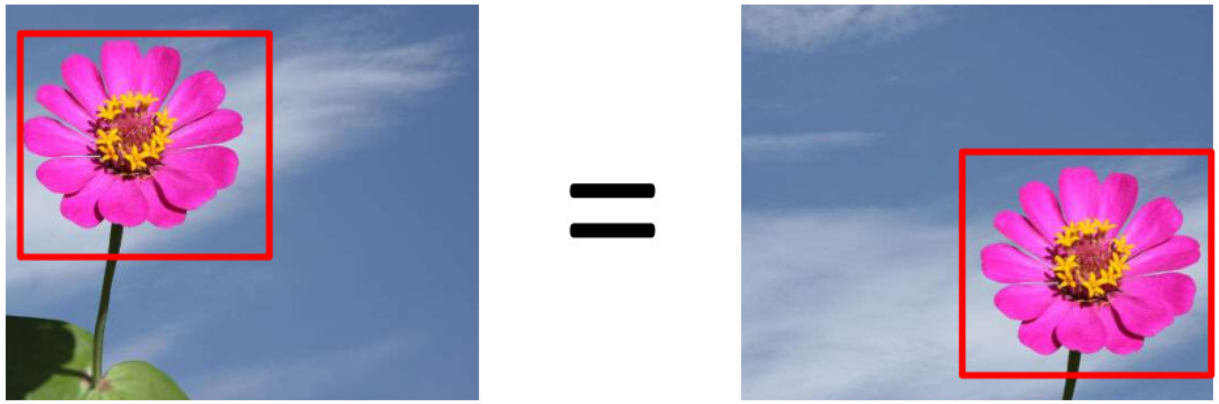
Problem with MLP



- Need a simple network that will fire regardless of the location of “Welcome”

– and not fire when there is none

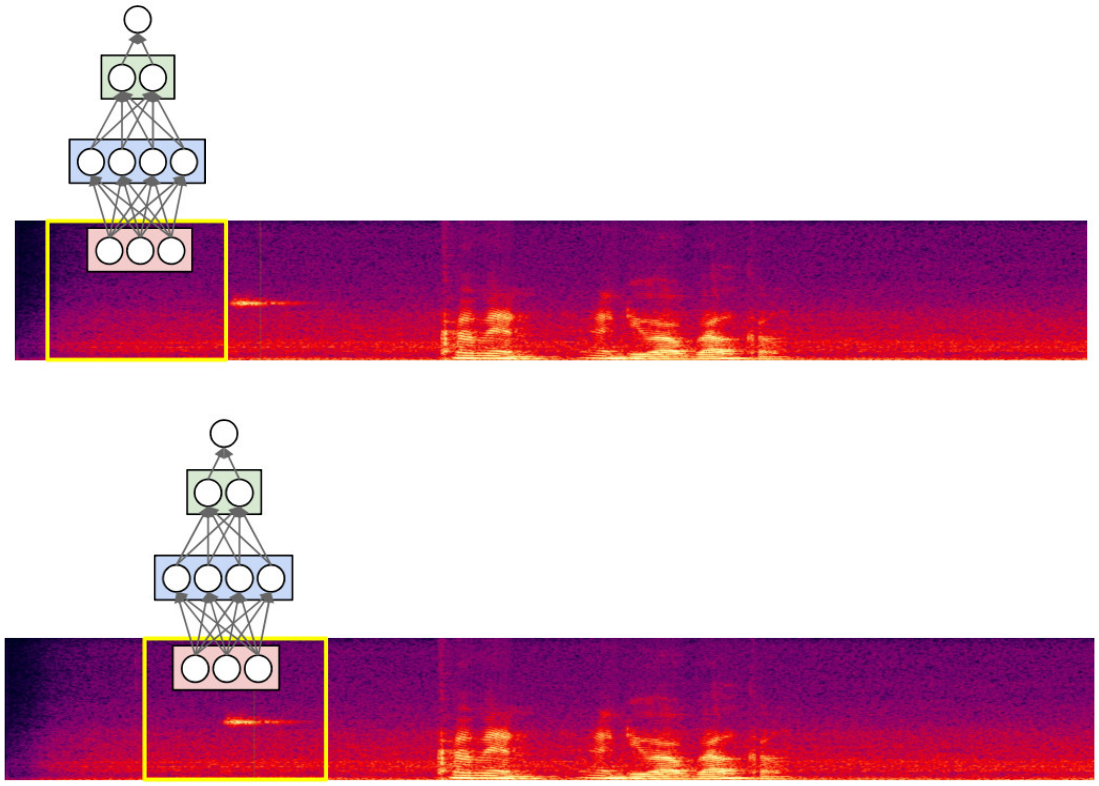
Problem with MLP



- Need a simple network that will fire regardless of the precise location of the target object
- The need for **shift invariance**

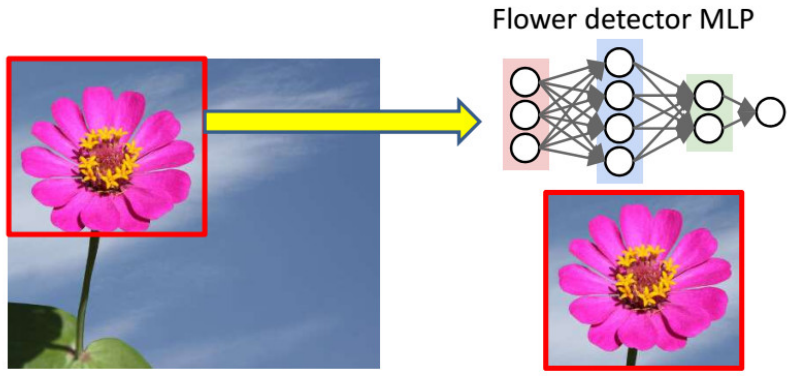
Solution: Scan!

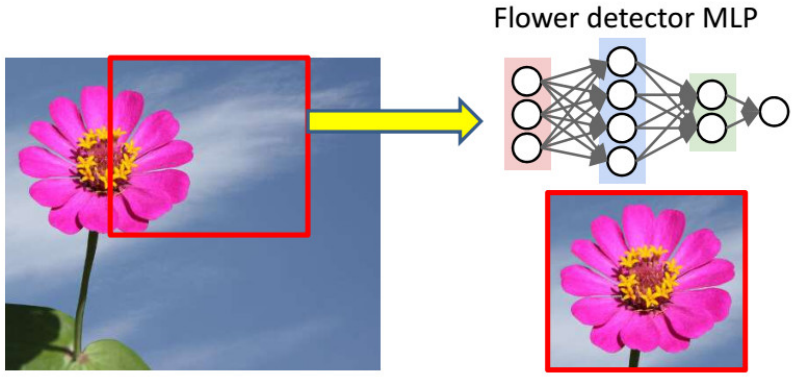
- Use same network for different windows patches



Solution: Scan!

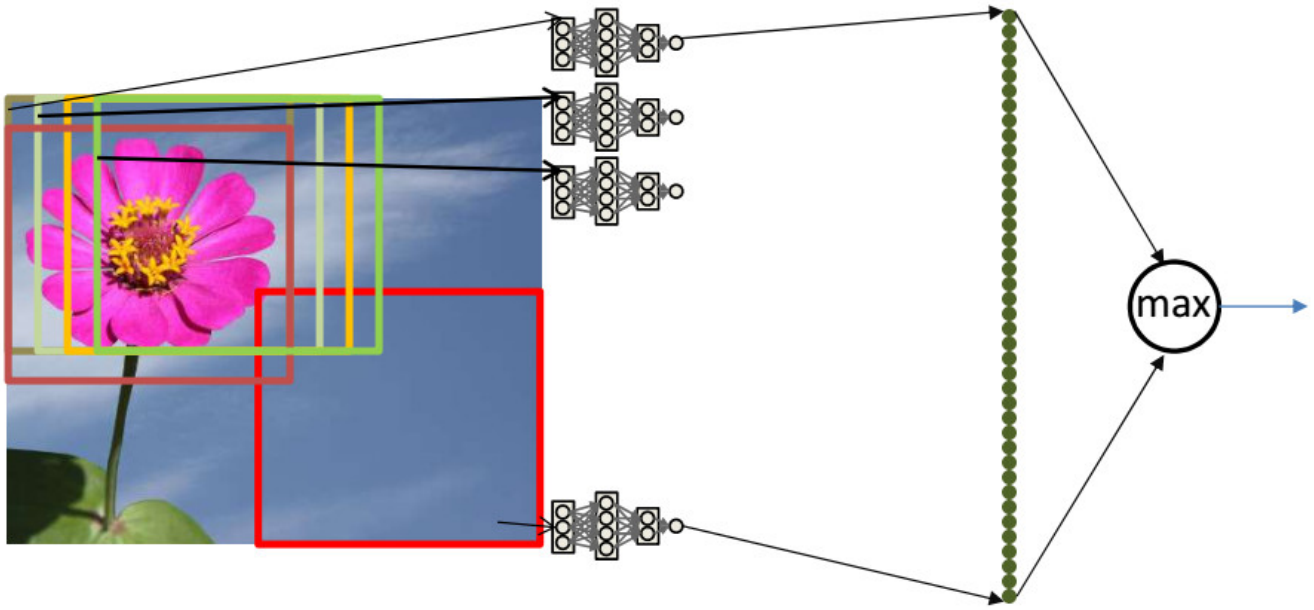
- Scan for desired pattern



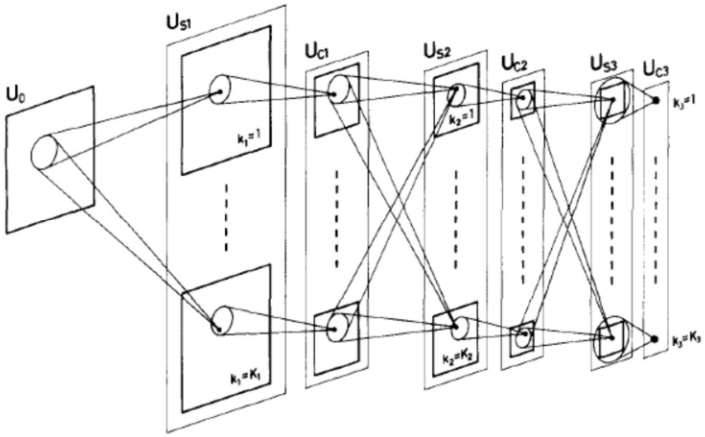


Solution: Scan!

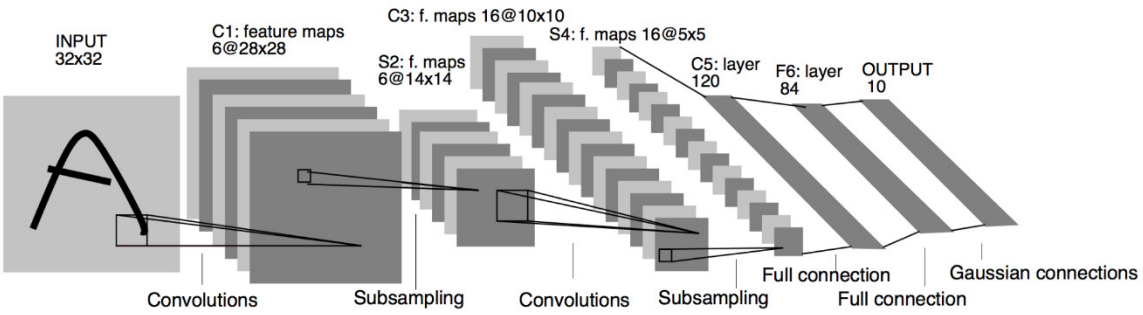
- Scan for desired pattern



Convolutional Neural Network: History

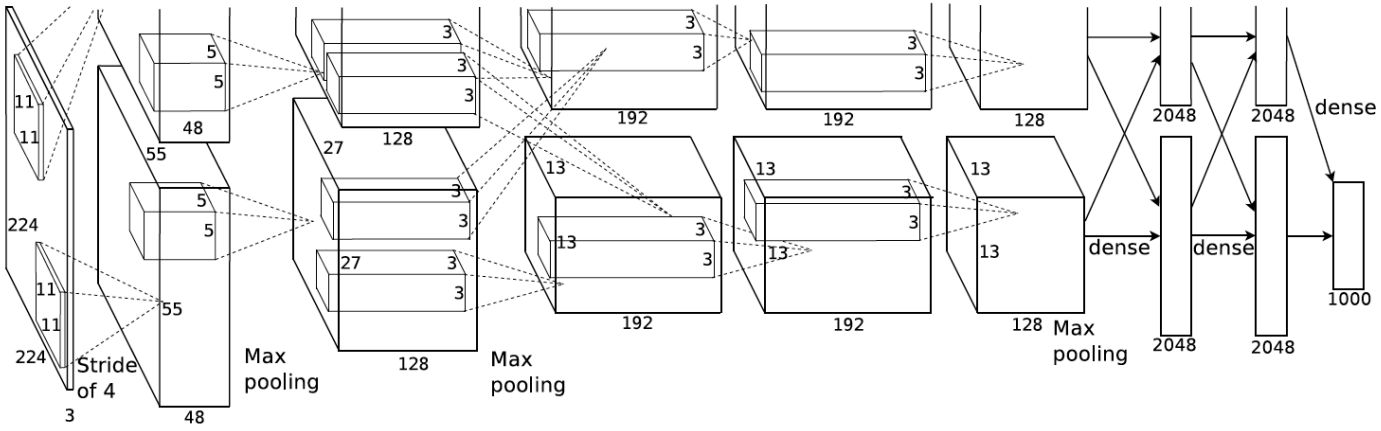


Neocognitron [Fukushima 1980]



LeNet-5 [LeCun et al. 1998]

Convolutional Neural Network: History

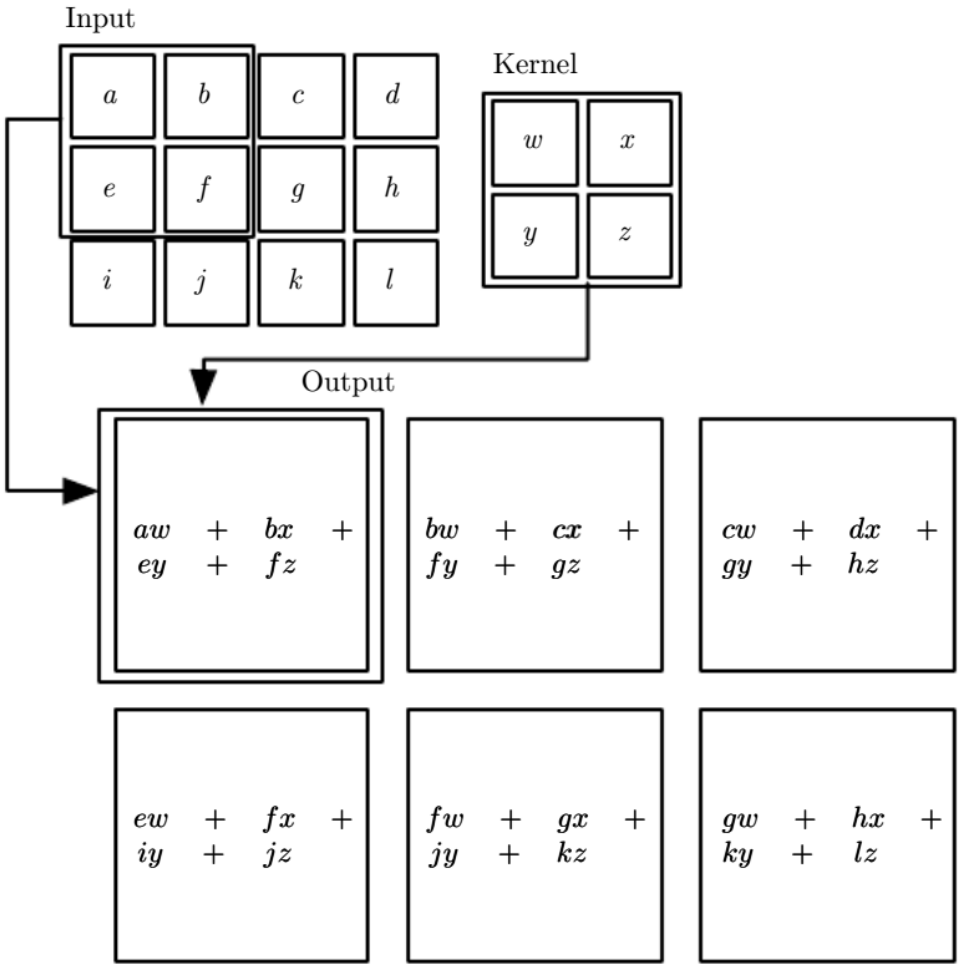


AlexNet [Krizhevsky et al. 2012]

Convolutional Neural Network

- Three main types of layers
  - **Convolutional Layer**
    - Extract useful features
    - Output of neurons are connected to local regions in the input
    - Appling the same filter on the whole image (weight sharing)
  - **Pooling Layer**
    - Perform a down-sampling operation along the spatial dimensions
    - Increase feature invariance
  - **Fully-Connected Layer (typical layer in MLPs)**
    - Classification or regression based on extracted features

Convolution Kernels



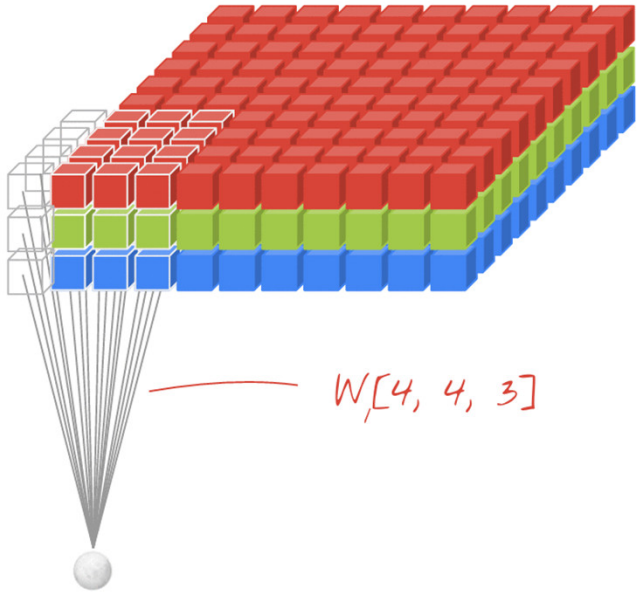
[Goodfellow et al. 2016]

Convolution Kernels

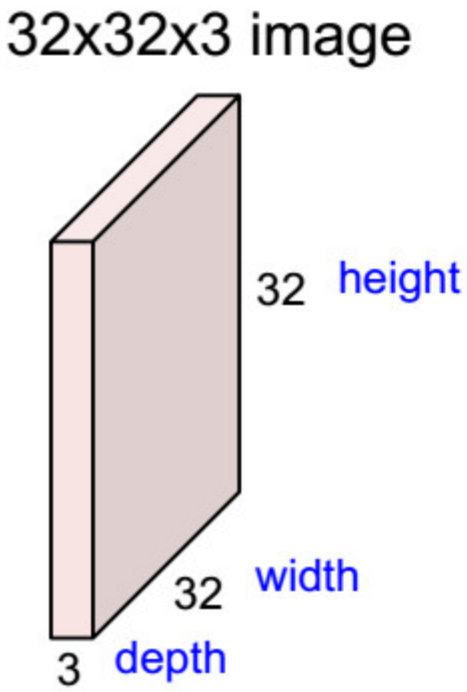
We call the layer convolutional because it is related to convolution of two signals:

$$f[x,y] * g[x,y] = \sum_{n_1=-\infty}^{\infty} \sum_{n_2=-\infty}^{\infty} f[n_1,n_2] \cdot g[x-n_1, y-n_2]$$

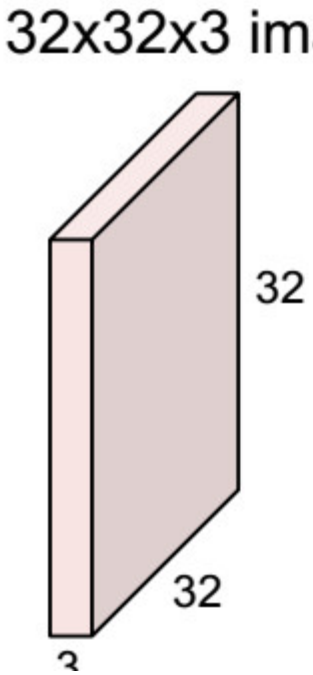
elementwise multiplication and sum of a filter and the signal (image)

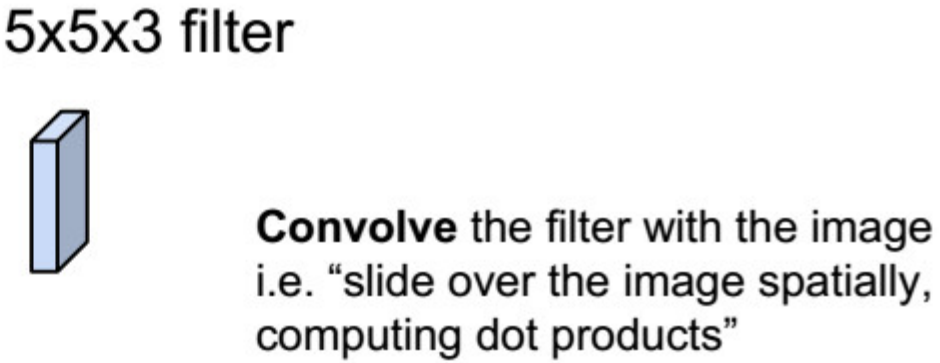


Convolution Layer

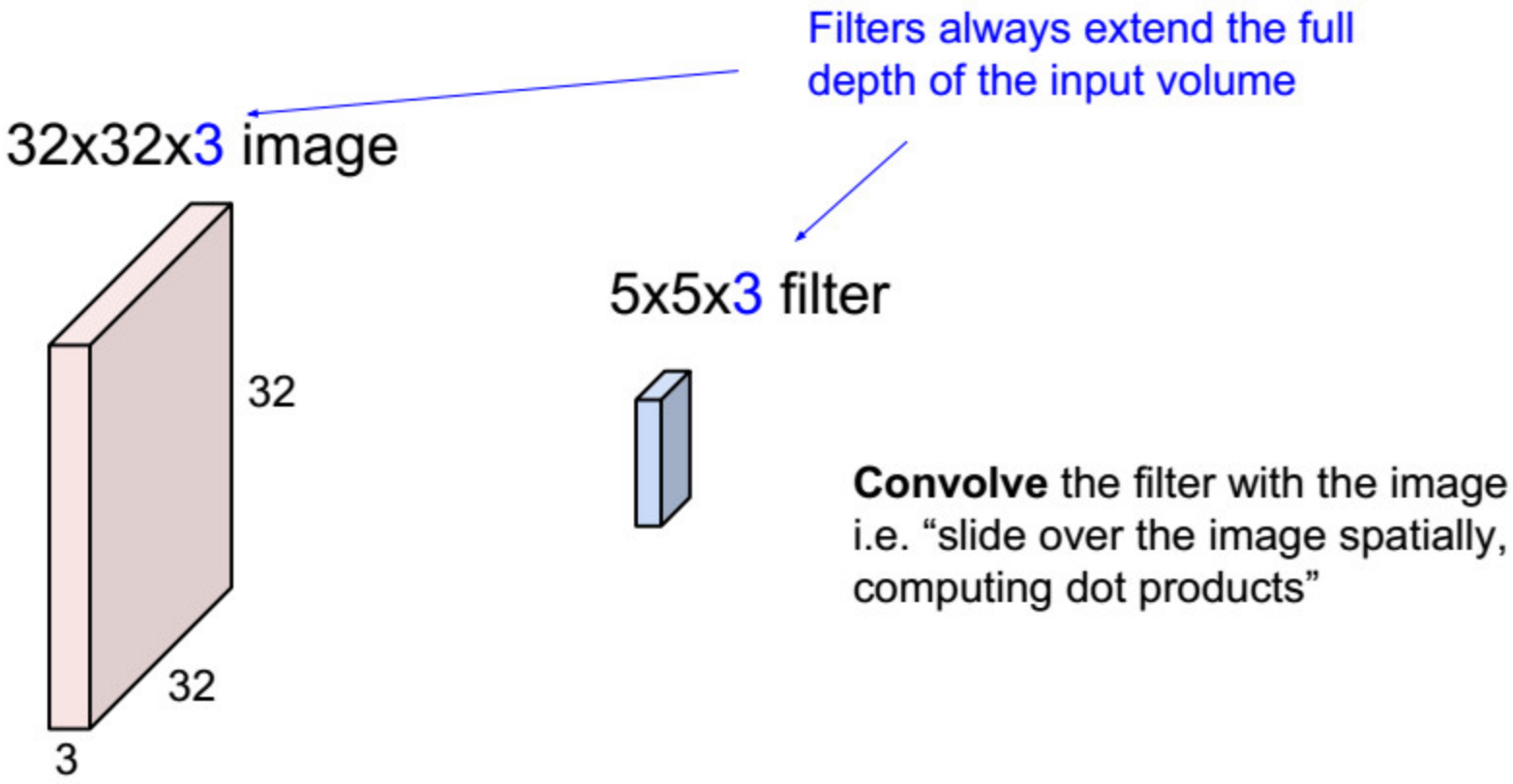


Convolution Layer

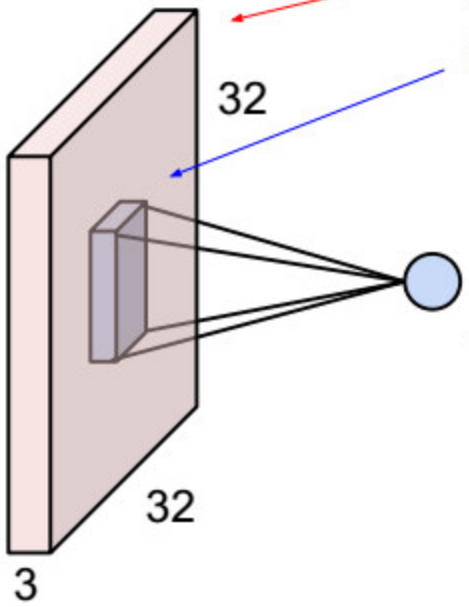




Convolution Layer



Convolution Layer



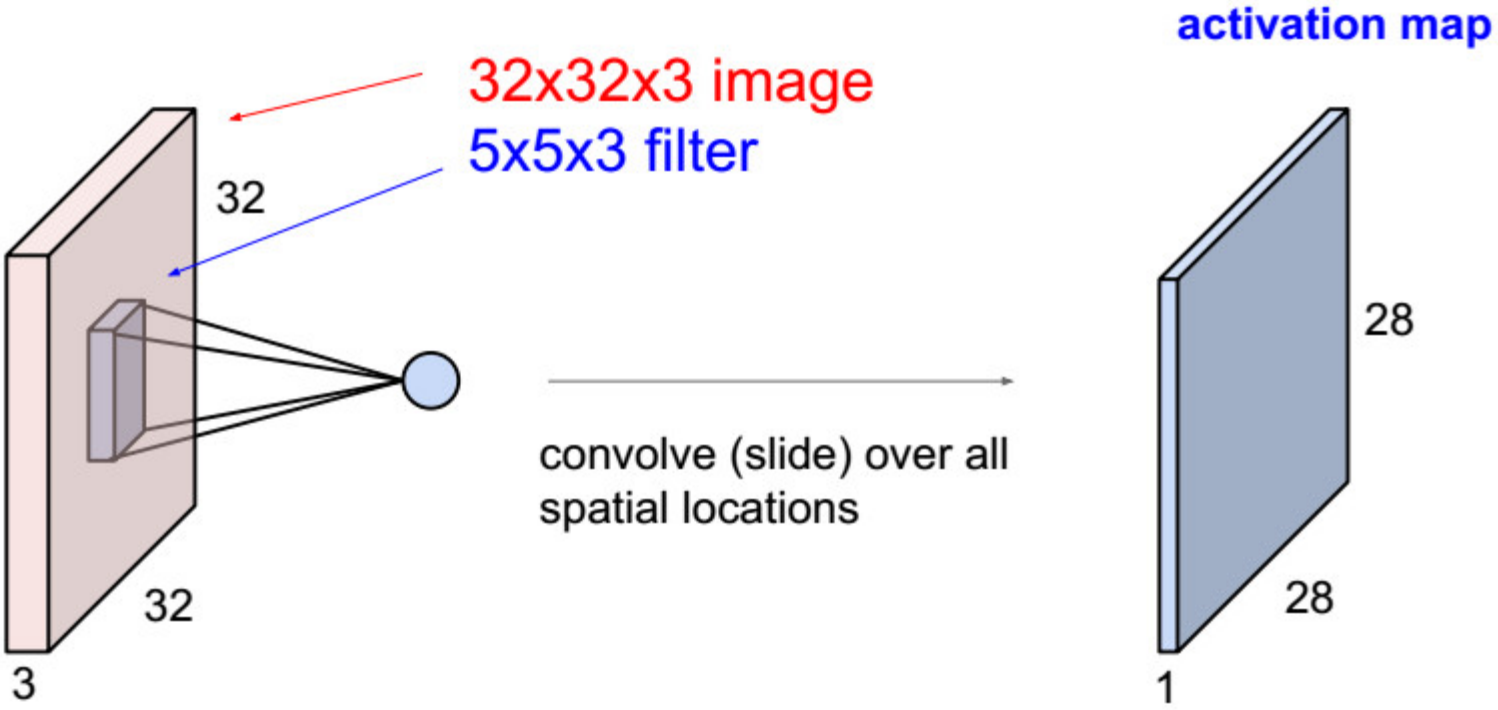
32x32x3 image

5x5x3 filter w

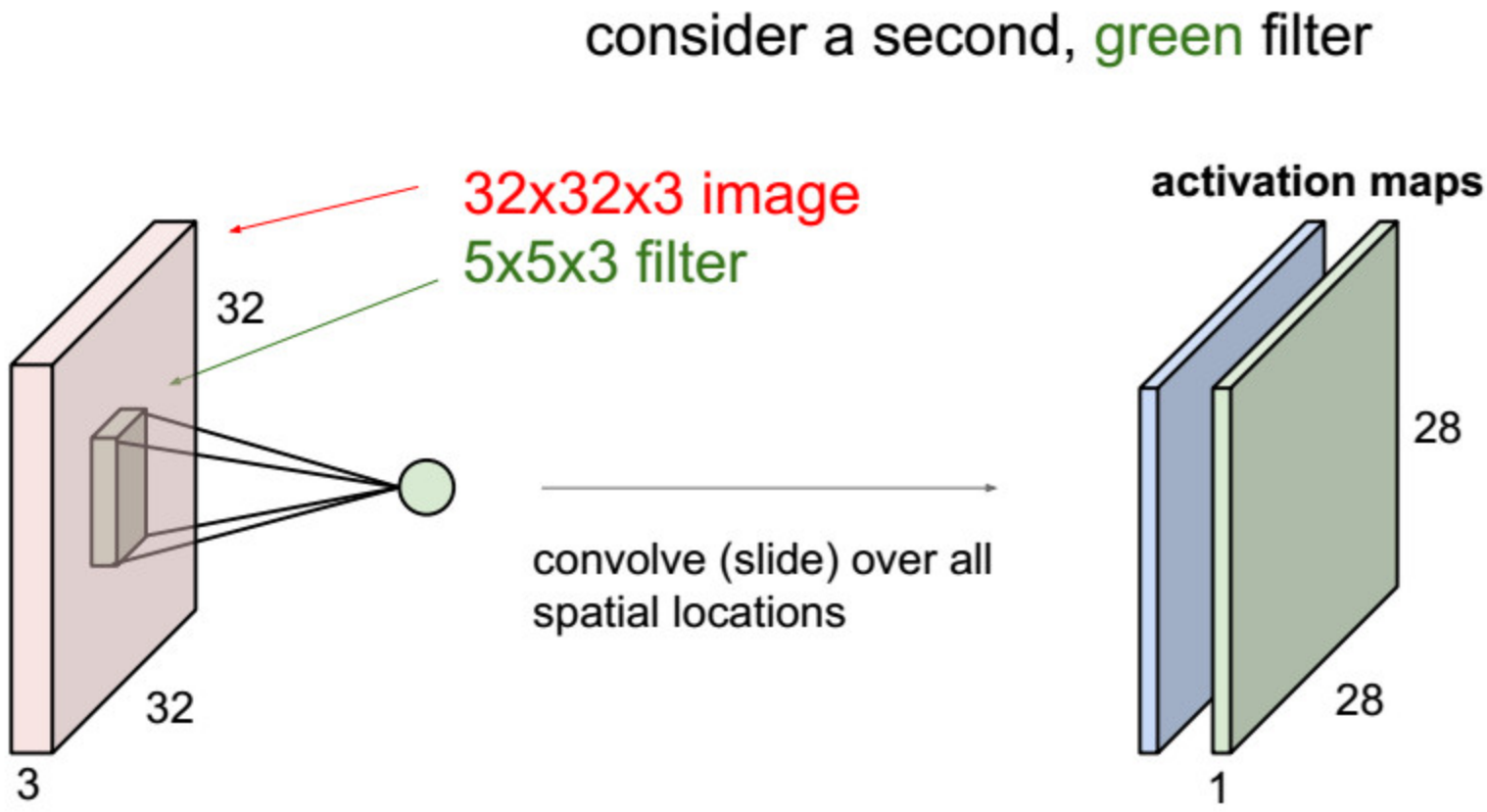
1 number:
the result of taking a dot product between the filter and a small 5x5x3 chunk of the image
(i.e. 5*5*3 = 75-dimensional dot product + bias)

$$w^T x + b$$

Convolution Layer

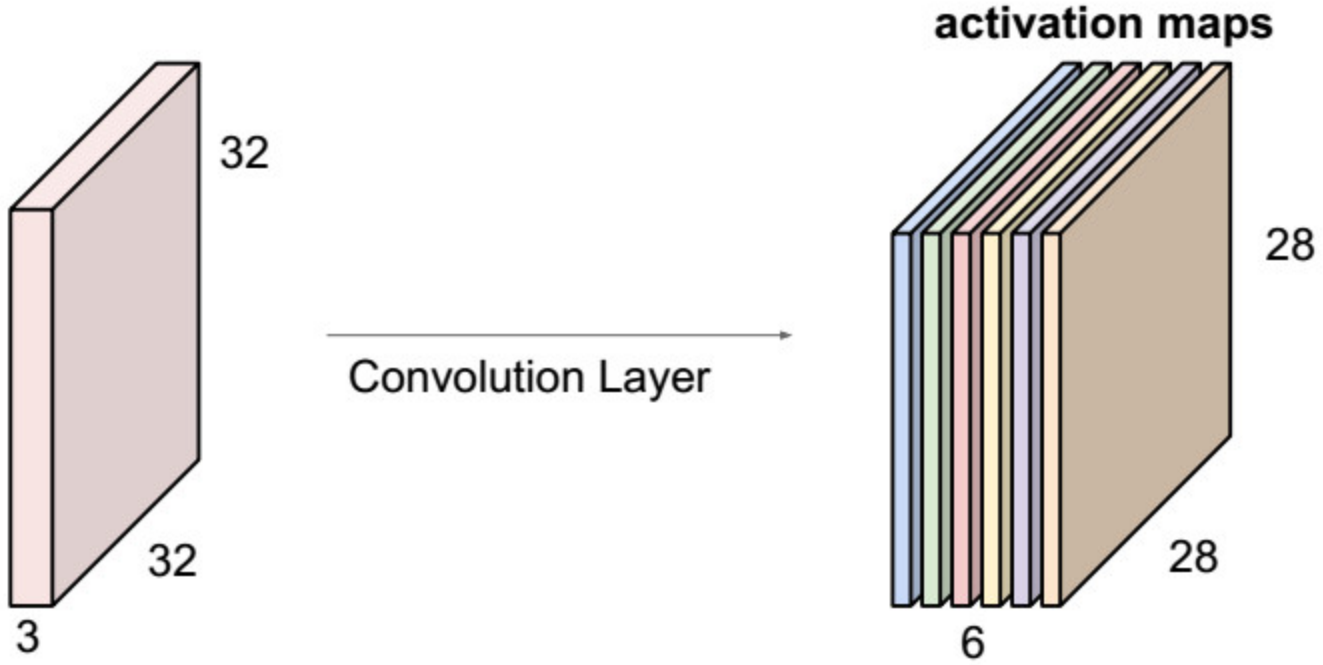


Convolution Layer



Convolution Layer

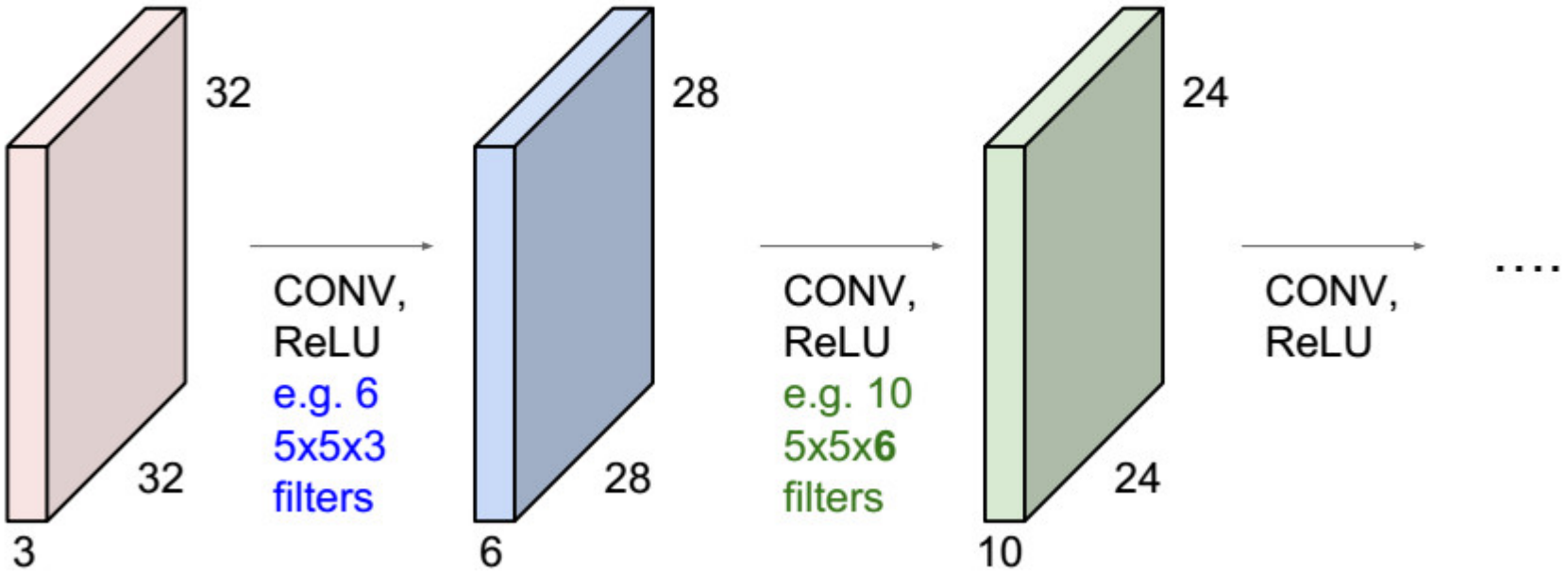
For example, if we had 6 5x5 filters, we'll get 6 separate activation maps:



We stack these up to get a "new image" of size 28x28x6!

Convolution Layer

Preview: ConvNet is a sequence of Convolutional Layers, interspersed with activation functions



Parameters Settings

- Accepts a volume of size $W_1 \times H_1 \times D_1$
- Requires four hyper-parameters:
  - Number of filters K
  - Their spatial extent F
  - The stride S
  - The amount of zero padding P
- Produces a volume of size $W_2 \times H_2 \times D_2$ where:
  - $W_2 = \frac{(W_1 - F + 2P)}{S} + 1$
  - $H_2 = \frac{(H_1 - F + 2P)}{S} + 1$
  - $D_2 = K$

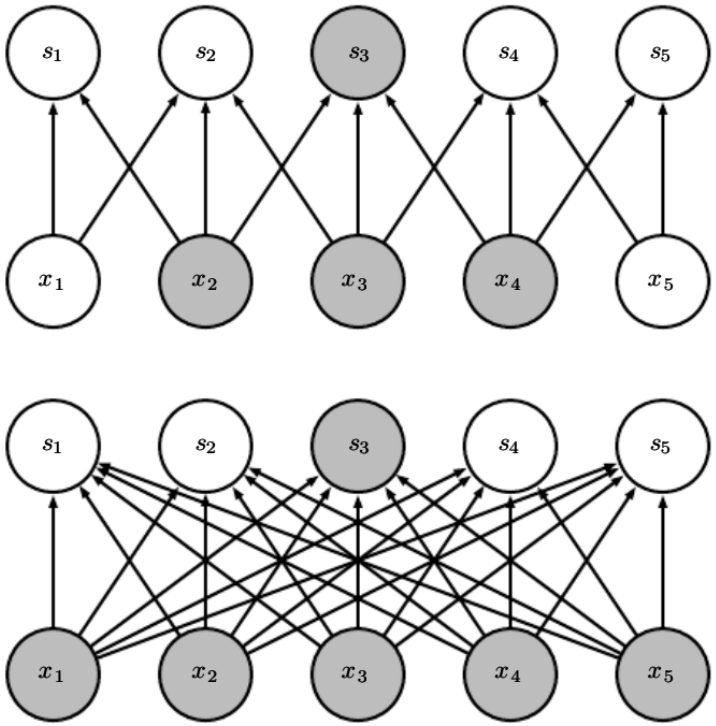
Common settings:
K = powers of 2 (e.g. 32, 64, …)
F = 3, S = 1F = 5, S = 1

Convolutional layer VS fully connected layer

- Convolutional layers use receptive fields, so they have locality in feature extraction.
- Convolutional layers use parameter sharing, so they can be used in deeper architectures.

Convolutional layer VS fully connected layer

- **locality**



convolution

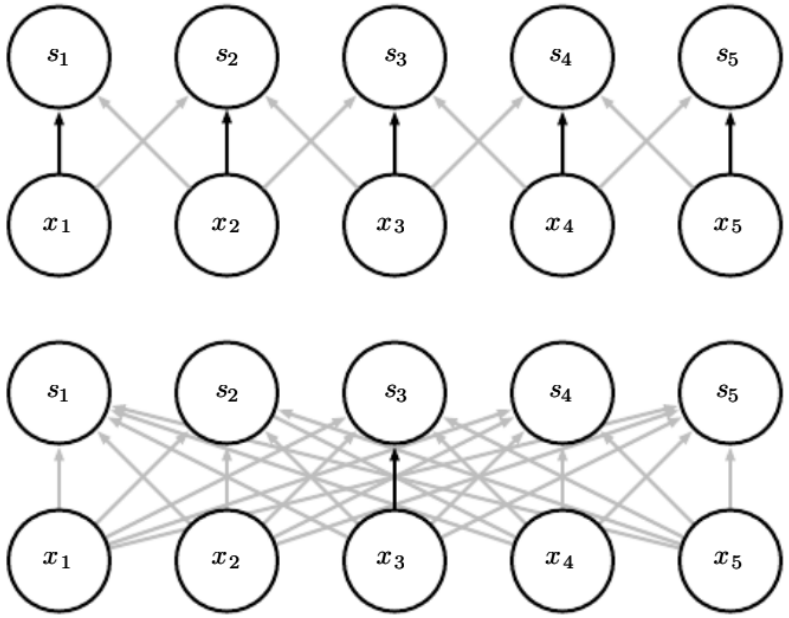
fully connected

[Goodfellow et al. 2016]

Convolutional layer VS fully connected layer

- **Parameter sharing**

(Black arrows indicate the connections that use a particular parameter in two different models)



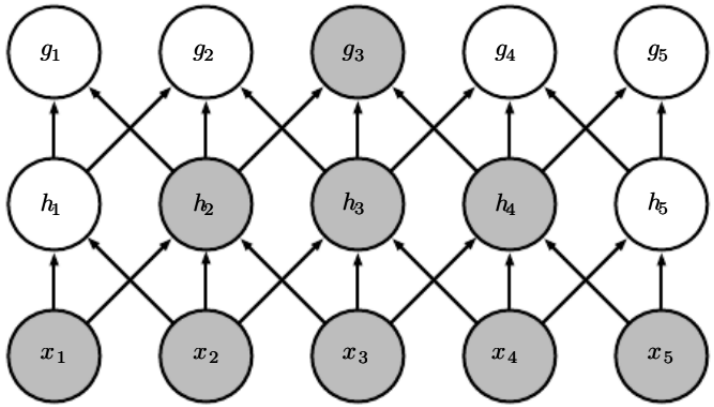
convolution

fully connected

[Goodfellow et al. 2016]

Receptive Field

How big of a region in the input does a neuron on the second conv-layer see?



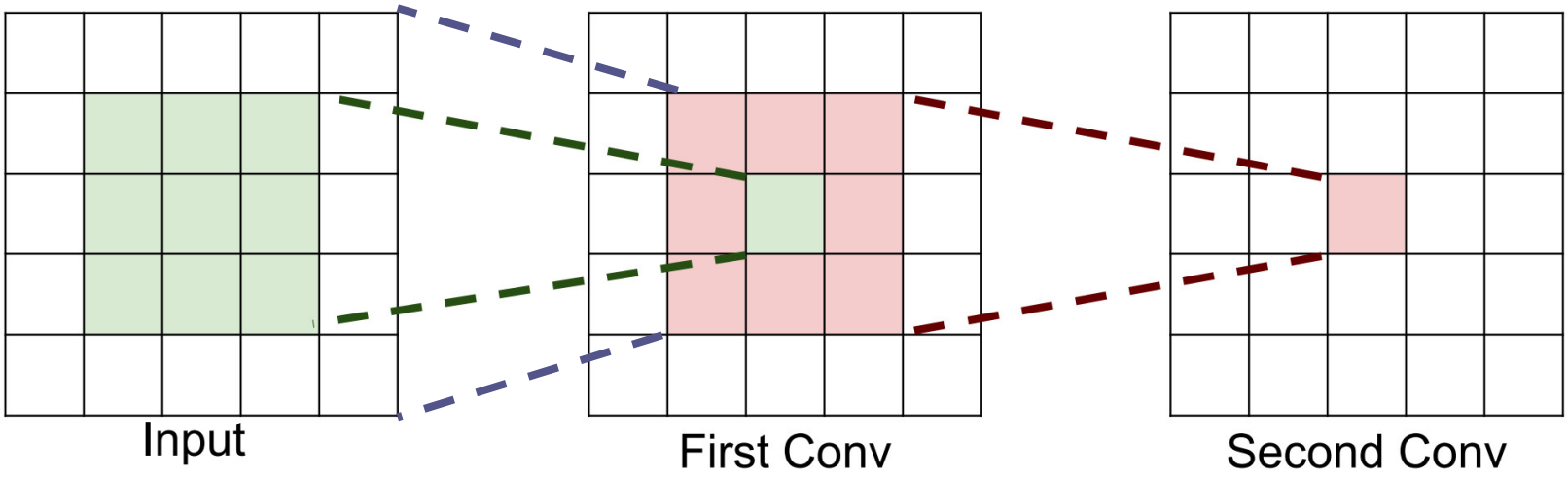
units in the deeper layers can be indirectly connected to all or most of the input image.

[Goodfellow et al. 2016]

Receptive Field

How big of a region in the input does a neuron on the second conv-layer see?

Two 3x3 filters together perform like one 5x5 filter (same receptive field)



Power of Small Filters

Suppose input is H x W x C and we use convolutions with C filters to preserve depth (stride 1, padding to preserve H, W)

one CONV with 7 x 7 filters

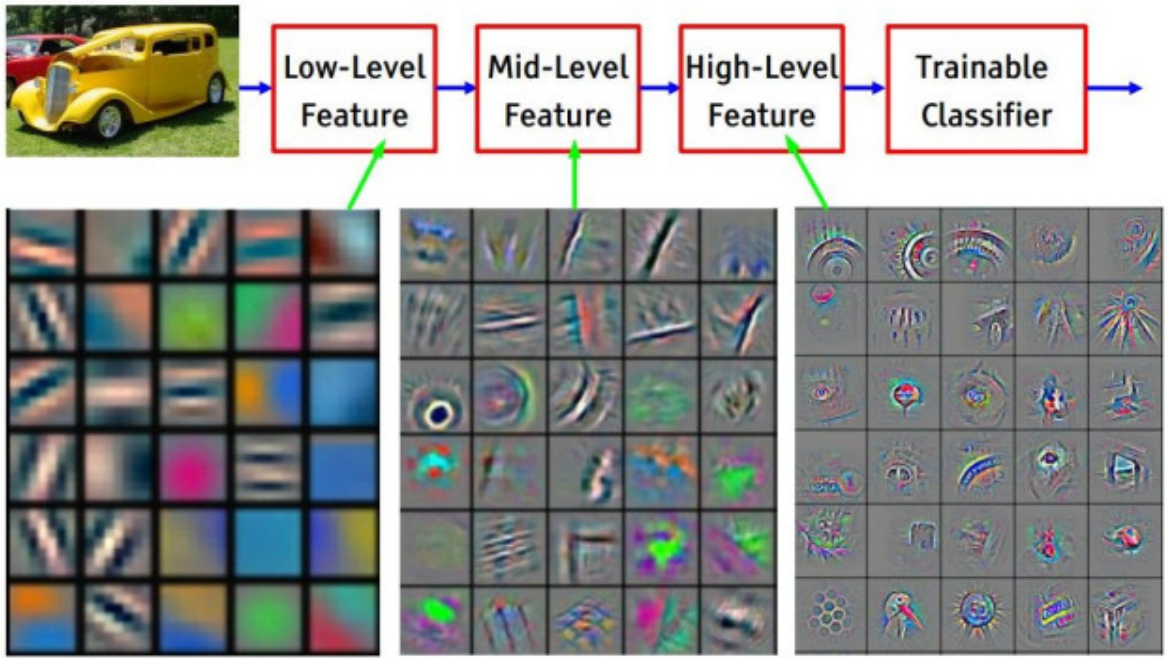
Number of weights:
= C x (7 x 7 x C) = **49 C^2**

three CONV with 3 x 3 filters

Number of weights:
= 3 x C x (3 x 3 x C) = **27 C^2**

Fewer parameters, more nonlinearity = GOOD

Feature Visualization

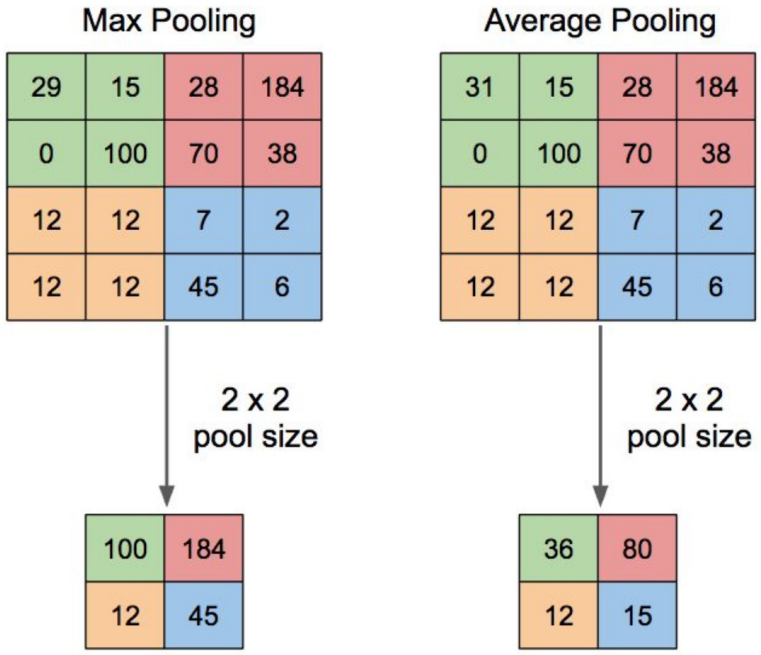


reminiscent of the Gabor filters!

[Zeiler et al. 2013]

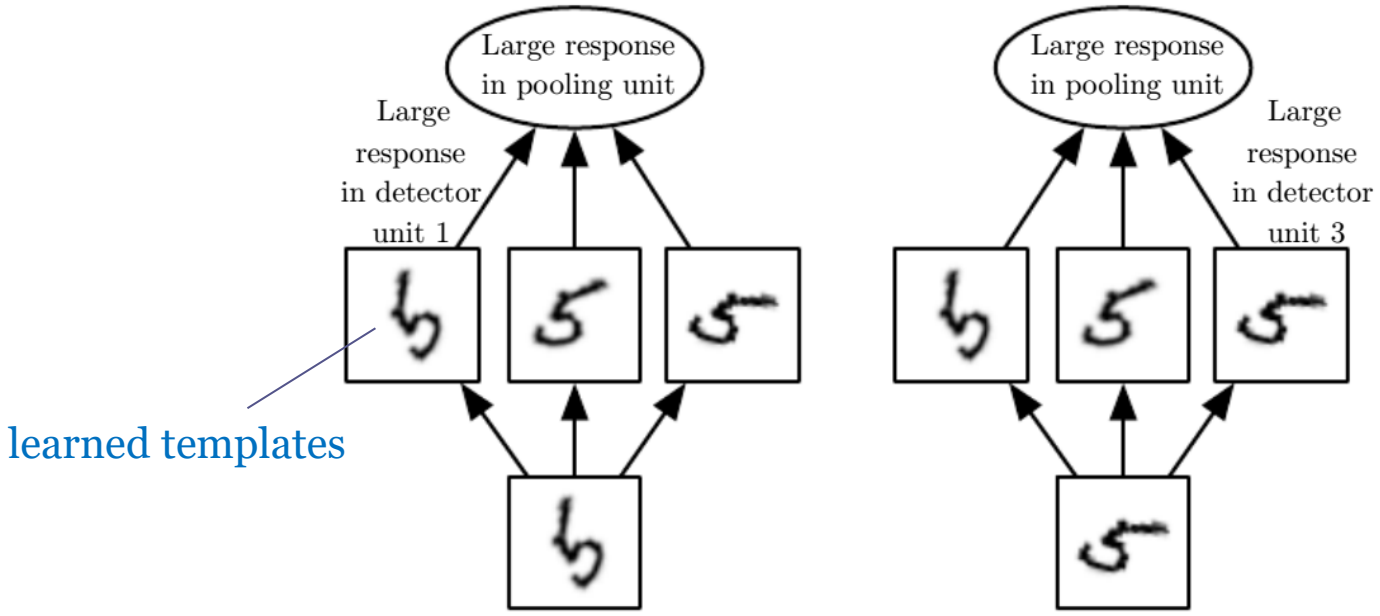
Pooling Layer

- Perform a down-sampling operation along the spatial dimensions (independent of depth slice)
- Reduce the spatial size of the representation
  - Reduce the amount of parameters and computation and control overfitting



Max Pooling

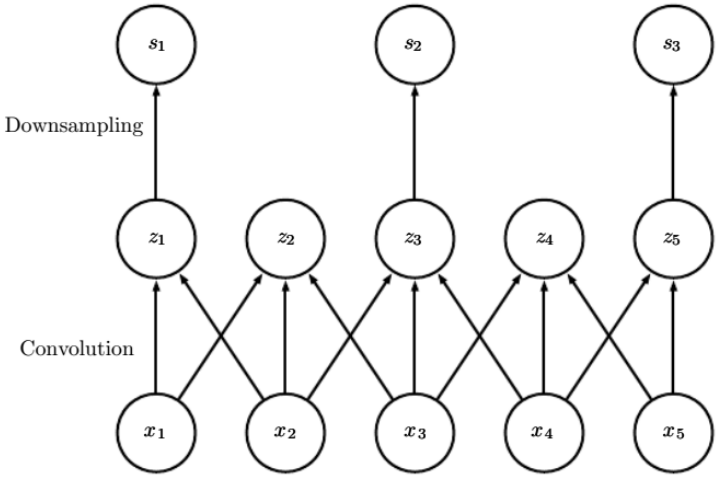
- Max pooling introduces invariance.

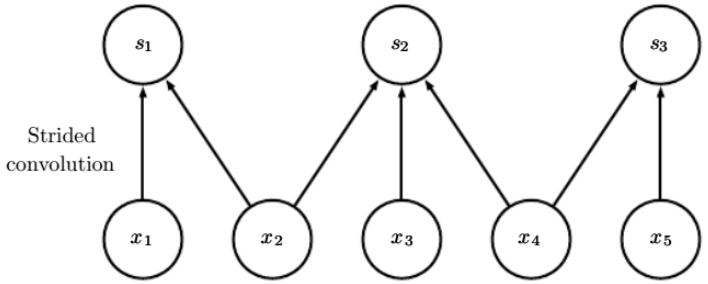


[Goodfellow et al. 2016]

Down-sampling VS Strided Convolution

- Convolution with a stride greater than one pixel is mathematically equivalent to convolution with unit stride followed by down-sampling.





S=2

[Goodfellow et al. 2016]

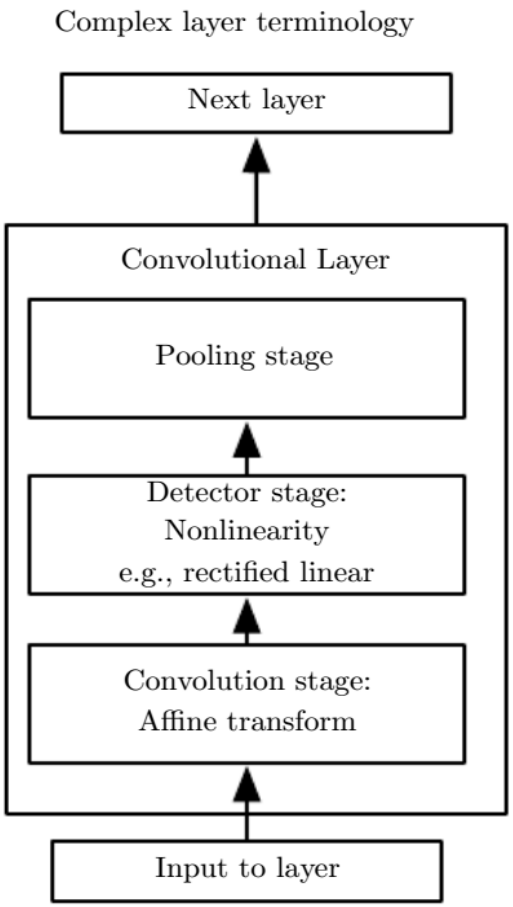
Typical architecture for CNNs

- [(CONV-RELU)*N – POOL?]*M – (FC-RELU)*K – SOFTMAX
  - Where N is usually up to ~5
  - M is large
  - 0 <= K <= 2

Typical architecture for CNNs

- [(CONV-RELU)*N – POOL?]*M – (FC-RELU)*K – SOFTMAX
  - Where N is usually up to ~5
  - M is large
  - 0 <= K <= 2

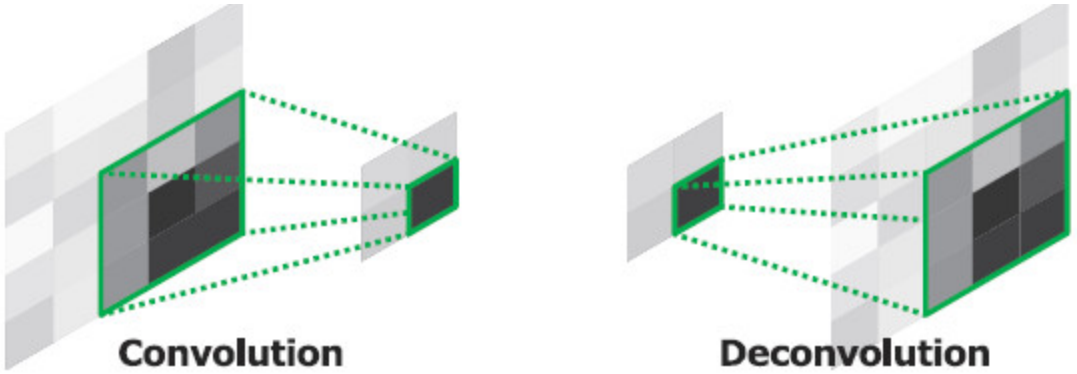
But recent advances such as ResNet and GoogleNet challenge this paradigm!



[Goodfellow et al. 2016]

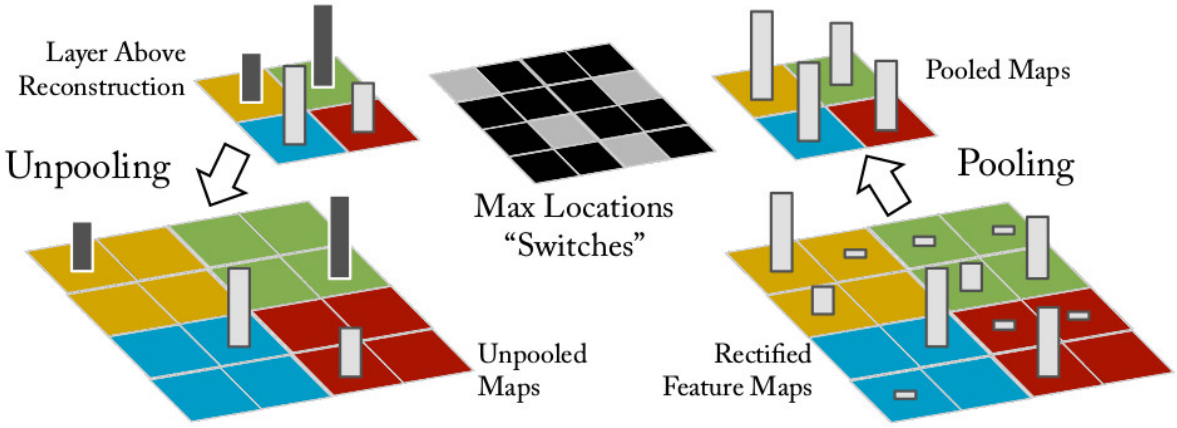
Other Type of Layers: Deconvolutional Layers

- Enlarge the size of feature map
- Densify sparse activations
- Good option for inverse mapping (from feature to image)



Other Type of Layers: Unpooling Layers

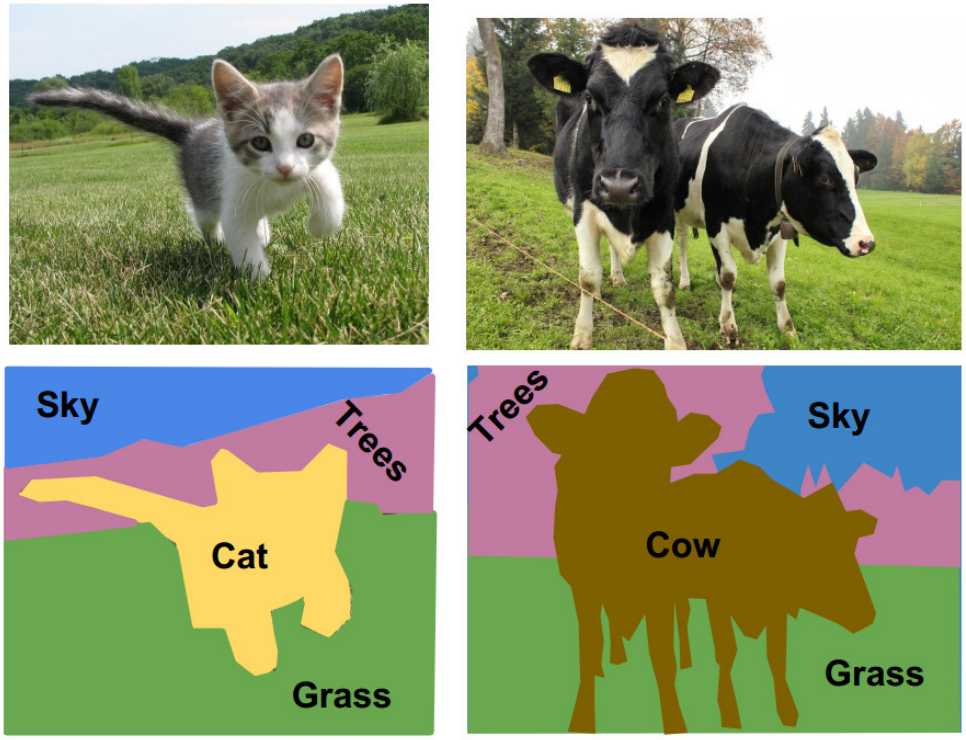
- Approximate inverse: Max pooling operation is non-invertible
- Switch variables: record the locations of maxima



Semantic Segmentation

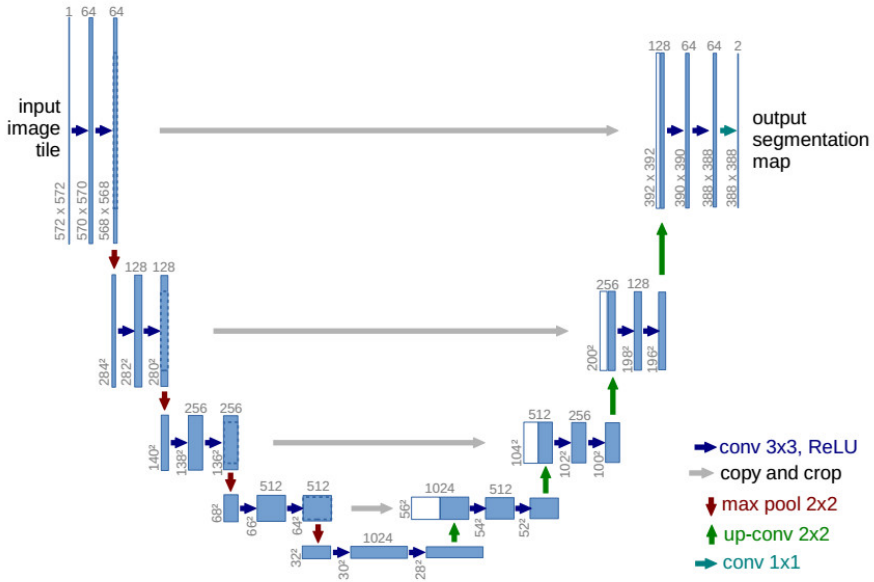
Classification: assign label to whole image

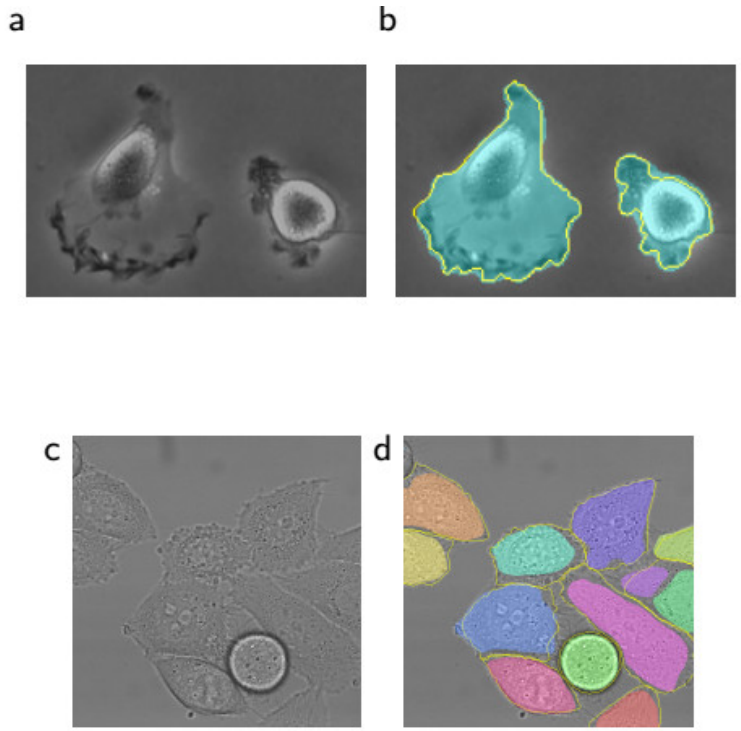
Semantic Segmentation: assign label to each pixel



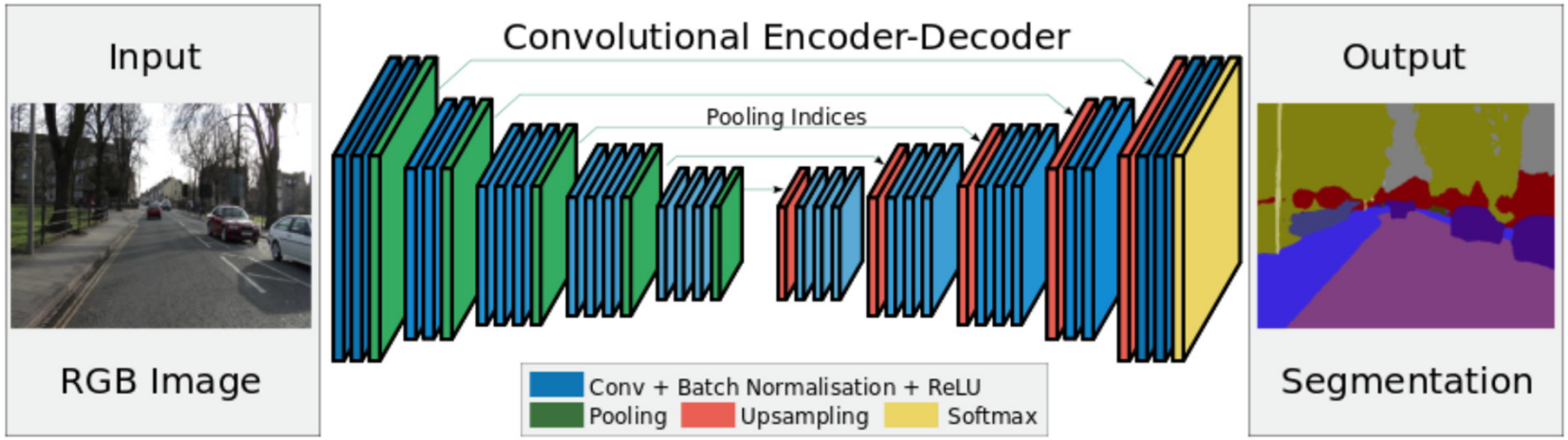
Semantic Segmentation: U-Net

[Ronneberger et al, 2015]





Semantic Segmentation: U-Net

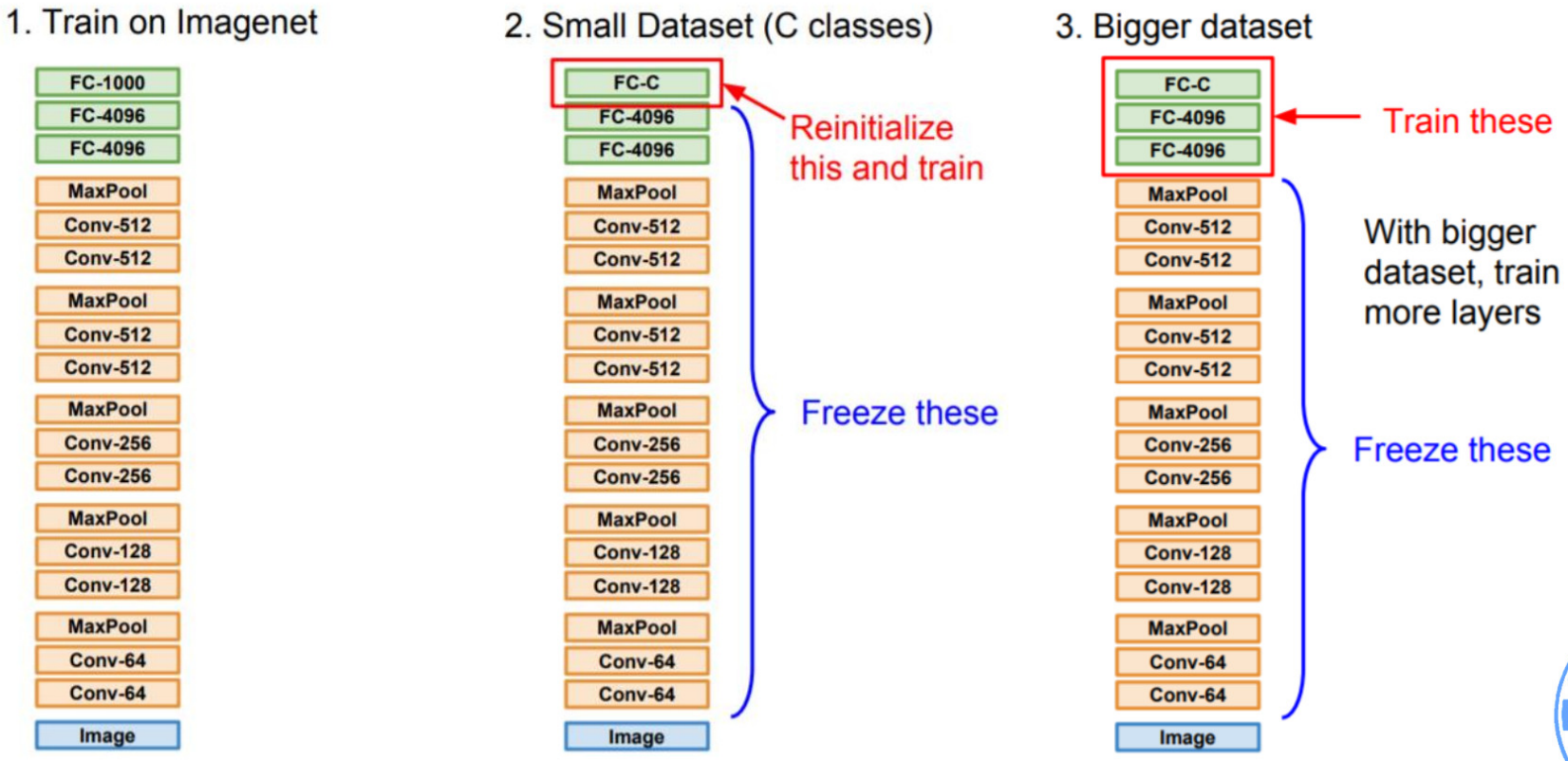


Transfer Learning

- We need a lot of a data if we want to train CNNs.
- By transfer learning you can use the learned features for a task (using a large amount of data) in another related task

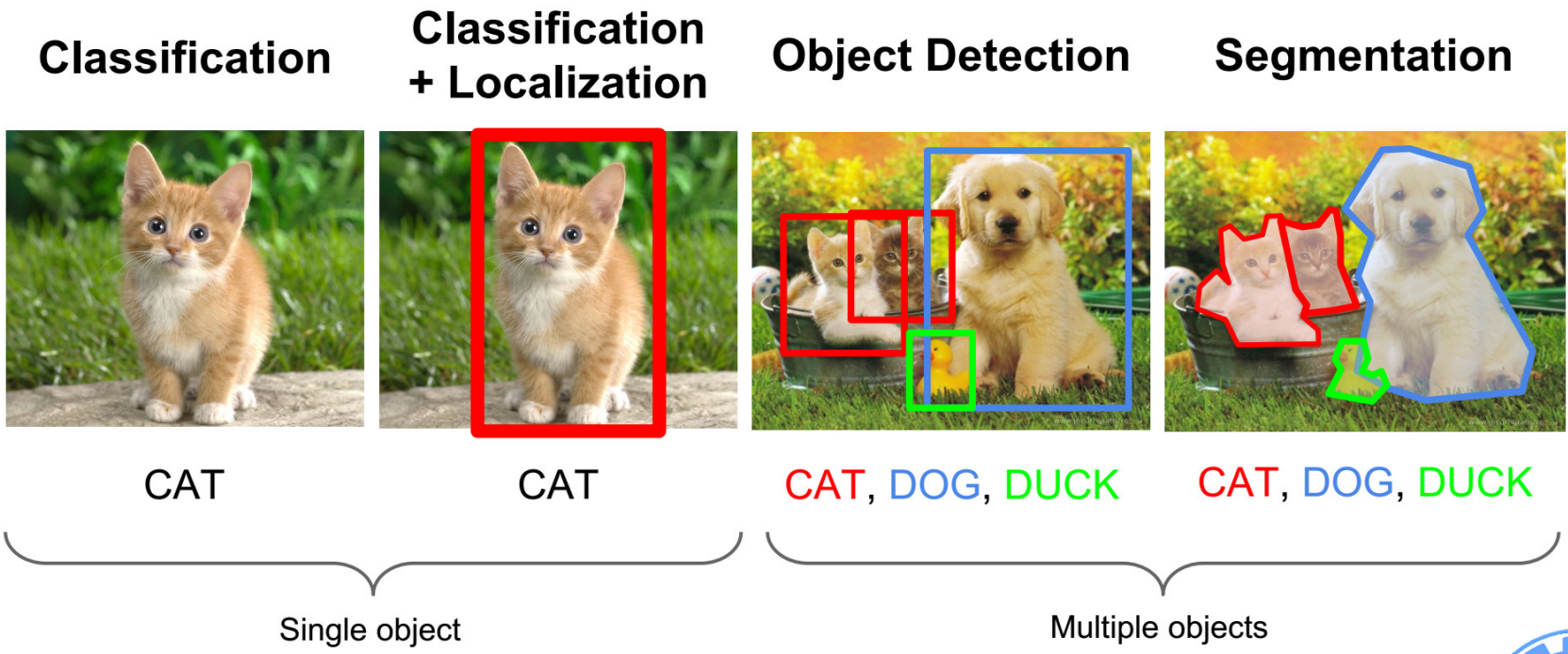
Transfer Learning

- We need a lot of a data if we want to train CNNs.
- By transfer learning you can use the learned features for a task (using a large amount of data) in another related task



Computer Vision Tasks

- CNNs achieved promising results in many vision tasks.



CNNs …

- Stacked of CONV, POOL, FC layers
- learn patterns in a hierarchical manner: simple to complex
- Perform well in classification, localization, segmentation, …
- Trend towards smaller filters and deeper architectures

Main Resources

- 11-785 **Introduction to Deep Learning**, CMU, spring 2019
- http://cs231n.stanford.edu/
- Goodfellow, et al. Deep learning. (chapter 9)