



# Visual Search

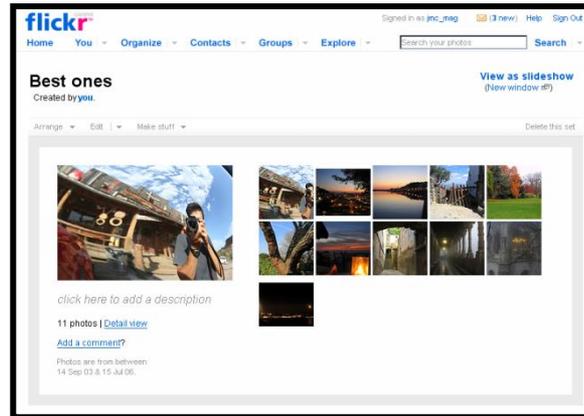
Convolutional Neural Networks

Web Data Mining and Search

# Visual search



Web content



Photographs



News/Sports/Movies

How to improve search capabilities over these types of content?

# Visual search by keyword

The screenshot shows a Flickr photo page for a night view of Lisbon. The page includes a navigation bar with options like Home, You, Organize, Contacts, Groups, and Explore. The main content area features a large photo of the city at dusk, with a toolbar above it containing actions like ADD NOTE, SEND TO GROUP, ADD TO SET, BLOG THIS, ALL SIZES, ORDER PRINTS, ROTATE, and DELETE. Below the photo is a comment box with the text "Add your comment". To the right of the photo, there is a section for "jmc\_mag's photostream" and a "Tags" section. The "Tags" section is highlighted with an orange box and contains the following tags: lisbon [x], sunset [x], buildings [x], bridge [x], sky [x], and city [x]. An orange arrow points from this "Tags" section to the text "ElasticSearch" on the right. Below the tags is an "ADD" button and instructions on how to use tags. The "Additional Information" section includes copyright information and metadata such as "Taken with a Canon PowerShot S50", "Taken on September 14, 2003", and "Viewed 27 times".

Signed in as [jmc\\_mag](#) (3 new) Help Sign Out

Home You Organize Contacts Groups Explore Search everyone's photos Search

## Lisbon

ADD NOTE SEND TO GROUP ADD TO SET BLOG THIS ALL SIZES ORDER PRINTS ROTATE DELETE



Uploaded on August 3, 2006 by [jmc\\_mag](#)

**jmc\_mag's photostream**

You are at the first photo 19 photos View as slideshow

browse more

**Tags**

- lisbon [x]
- sunset [x]
- buildings [x]
- bridge [x]
- sky [x]
- city [x]

ADD

Choose from your tags

Separate each tag with a space: *cameraphone urban moblog*. Or to join 2 words together in one tag, use double quotes: *"daily commute"*.

**Additional Information**

- © All rights reserved (privacy)
- Taken with a Canon PowerShot S50. [More properties](#)
- Taken on September 14, 2003 (edit)
- See different sizes
- Viewed 27 times (Not including you)
- [Edit](#) title, description, and tags

(Some HTML is OK.)

[Hide this photo from public searches?](#)

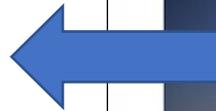
ElasticSearch

# Visual search by text

Sunset picture taken in Lisbon from Miradouro da Graça, with Ponte 25 de Abril on the landscape and with streetlights already on.



**ElasticSearch**



Signed in as [jmc\\_mag](#) (3 new) Help Sign Out

Home You Organize Contacts Groups Explore Search everyone's photos Search

## Lisbon

ADD NOTE SEND TO GROUP ADD TO SET BLOG THIS ALL SIZES ORDER PRINTS ROTATE DELETE

Uploaded on August 3, 2006 by [jmc\\_mag](#)

### jmc\_mag's photostream

You are at the first photo 19 photos [View as slideshow](#)

browse more

#### Tags

- lisbon [x]
- sunset [x]
- buildings [x]
- bridge [x]
- sky [x]
- city [x]

[ADD](#)

Choose from your tags

Separate each tag with a space:  
*cameraphone urban moblog*. Or to join 2 words together in one tag, use double quotes: "daily commute".

#### Additional Information

- © All rights reserved ([privacy](#))
- Taken with a Canon PowerShot S50. [More properties](#)
- Taken on [September 14, 2003](#) ([edit](#))
- See [different sizes](#)
- Viewed 27 times (Not including you)
- [Edit](#) title, description, and tags

[Hide this photo from public searches?](#)

(Some HTML is OK.)

# Visual search by example

User query

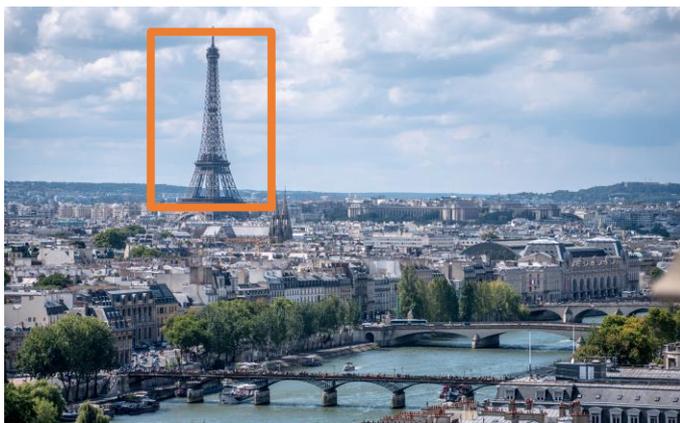


Search results

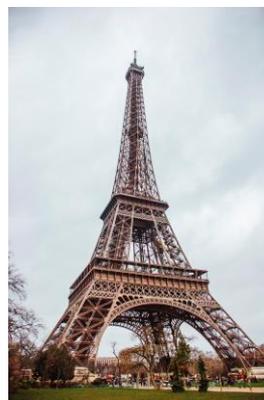
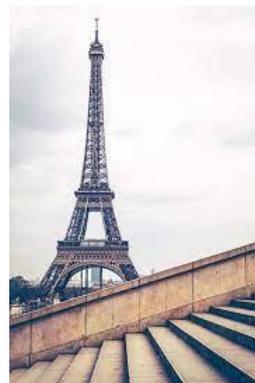


# Visual search by region

User bounded query



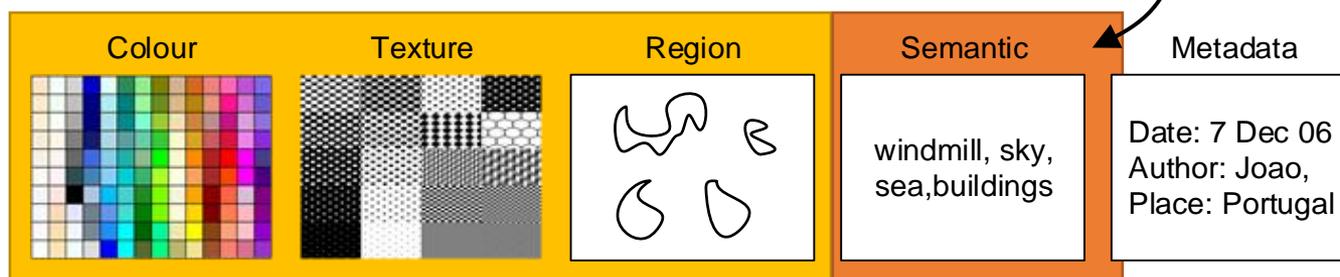
Search results



# Hand-crafted vs learned feature extraction



**Approach B:** end-to-end classifiers learn task specific data representations.



**Approach A:** classifiers learn with hand-crafted methods for feature extraction.

# Learning a keyword probability distribution

elkhound	tench	Komodo dragon	tick	wallaby	white stork	valley
otterhound	goldfish	African crocodile	centipede	koala	black stork	volcano
Saluki	great white shark	American alligator	black grouse	wombat	spoonbill	ballplayer
Scottish deerhound	tiger shark	triceratops	ptarmigan	jellyfish	flamingo	groom
Weimaraner	hammerhead	thunder snake	ruffed grouse	sea anemone	little blue	scuba diver
Staffordshire	electric ray	ringneck snake	prairie chicken	brain coral	heron	rapeseed
bullterrier	stingray	hognose snake	peacock	flatworm	American	daisy
American	cock	green snake	quail	nematode	egret	yellow lady's
Staffordshire terrier	hen	king snake	partridge	conch	bittern	slipper
Bedlington terrier	ostrich	garter snake	African grey	snail	crane	corn
Border terrier	brambling	water snake	macaw	slug	limpkin	acorn
Kerry blue terrier	goldfinch	vine snake	sulphur-crested	sea slug	European	hip
Irish terrier	house finch	night snake	cockatoo	chiton	gallinule	buckeye
Norfolk terrier	junco	boa constrictor	lorikeet	chambered	American coot	coral fungus
Norwich terrier	indigo bunting	rock python	coucal	nautilus	bustard	agaric
Yorkshire terrier	robin	Indian cobra	bee eater	Dungeness crab	ruddy	gyromitra
wire-haired fox	bulbul	green mamba	hornbill	rock crab	turnstone	stinkhorn
terrier		sea snake	hummingbird	fiddler crab	red-backed	earthstar
Lakeland terrier		horned viper	jacamar	king crab	sandpiper	hen-of-the-woods
Sealyham terrier		diamondback	toucan	American	redshank	bolete
Airedale		sidewinder	drake	lobster	dowitcher	ear
cairn			red-breasted	spiny lobster	oystercatcher	toilet tissue
			merganser	crayfish	pelican	
			goose	hermit crab	king penguin	
			black swan	isopod	albatross	
			tusker		grey whale	
			echidna		killer whale	
			platypus		dugong	
					sea lion	

# ImageNet competition

- A total of 1.43 million images annotated with 1,000 object classes
- The goal is to annotated a test sample and be as accurate as possible.
- Human error is 5.1%
- Great impact in advancing the state of the art.

<http://image-net.org/explore.php>

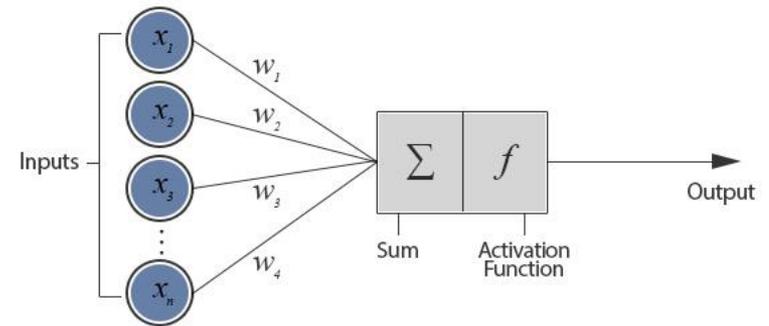
The screenshot shows the ImageNet website interface. At the top, there is a search bar and navigation links for Home, Explore, About, and Download. The main content area is titled 'Sport, athletics' and includes a description: 'An active diversion requiring physical exertion and competition'. To the right of the title, it shows '1888 pictures' and '92.64% Popularity Percentile'. Below the title, there are tabs for 'Treemap Visualization', 'Images of the Synset', and 'Downloads'. The 'Treemap Visualization' tab is active, displaying a grid of small images categorized into sub-synsets like Athletic, Contact, Outdoor, Water, Blood, Racing, Gymnast, Sledding, Cycling, Team, Skating, Funambulism, Archery, Judo, Rowing, Riding, Track, Rock, and Skiing. A sidebar on the left shows a hierarchical tree of synsets, with 'Sport, athletics (176)' selected. The footer contains copyright information for Stanford Vision Lab, Stanford University, Princeton University, and support@image-net.org.

# Perceptron: general formulation

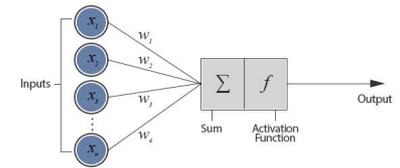
- **Binary classification:**

$$z = w_0 + w_1x_1 + \dots + w_nx_n$$

$$\hat{y} = f(z) = \begin{cases} +1 & , \text{if } z \geq 0 \\ -1 & , \text{if } z < 0 \end{cases}$$

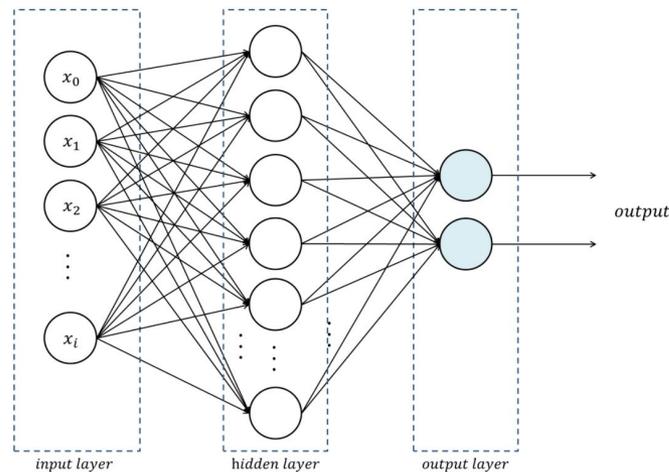


- **Input:** Vectors  $\mathbf{x}^{(j)}$  and labels  $\mathbf{y}^{(j)}$ 
  - Vectors  $\mathbf{x}^{(j)}$  are real valued where  $\|\mathbf{x}\|_2 = 1$
- **Goal:** Find vector  $\mathbf{w} = (w_1, w_2, \dots, w_d)$ 
  - Each  $w_i$  is a real number

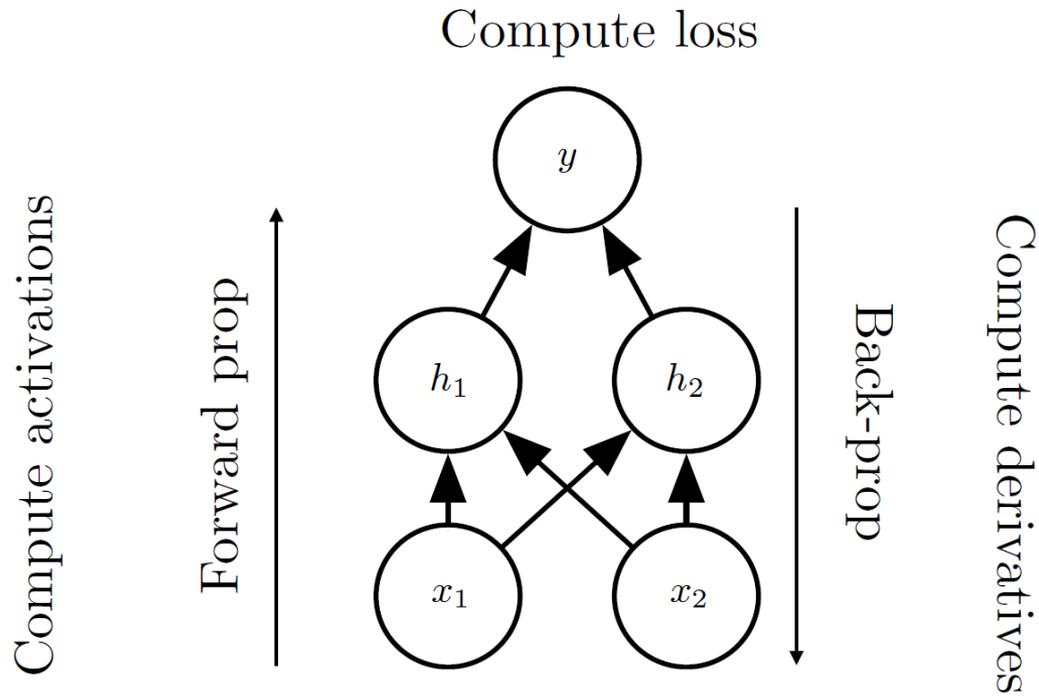
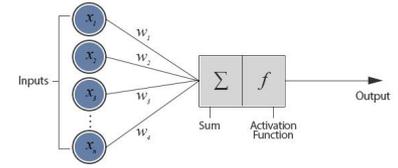


# Multi-layer classifiers

- Multi-layer classifiers allow to learn non-linear relations, i.e. complex relationships such as exclusive-OR.
- Usually one to two hidden layers produce the best results.
- Trained with the back-propagation algorithm

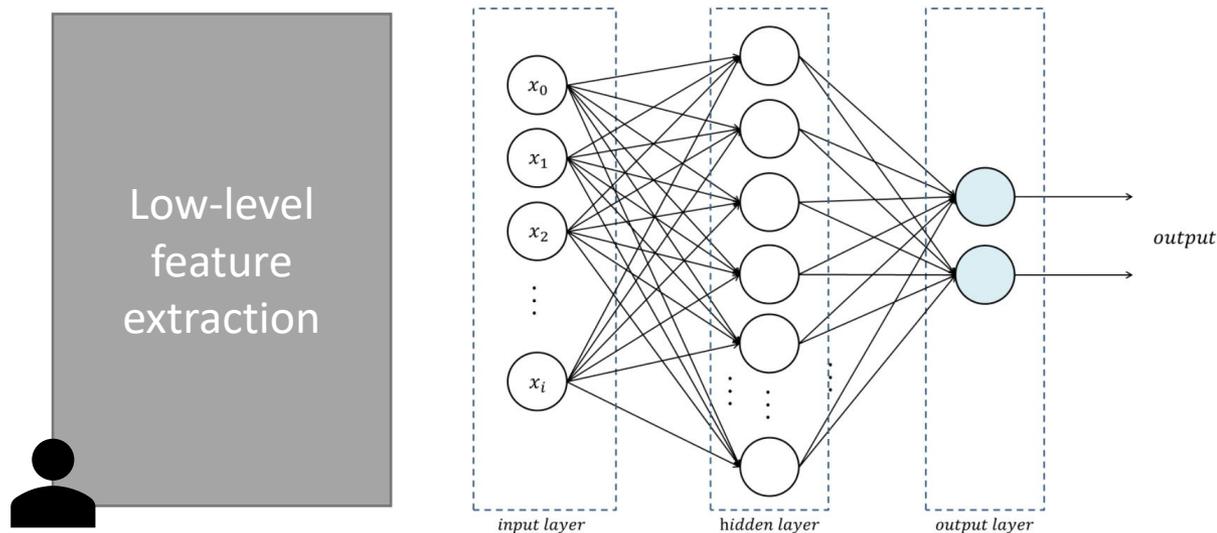


# Simple back propagation

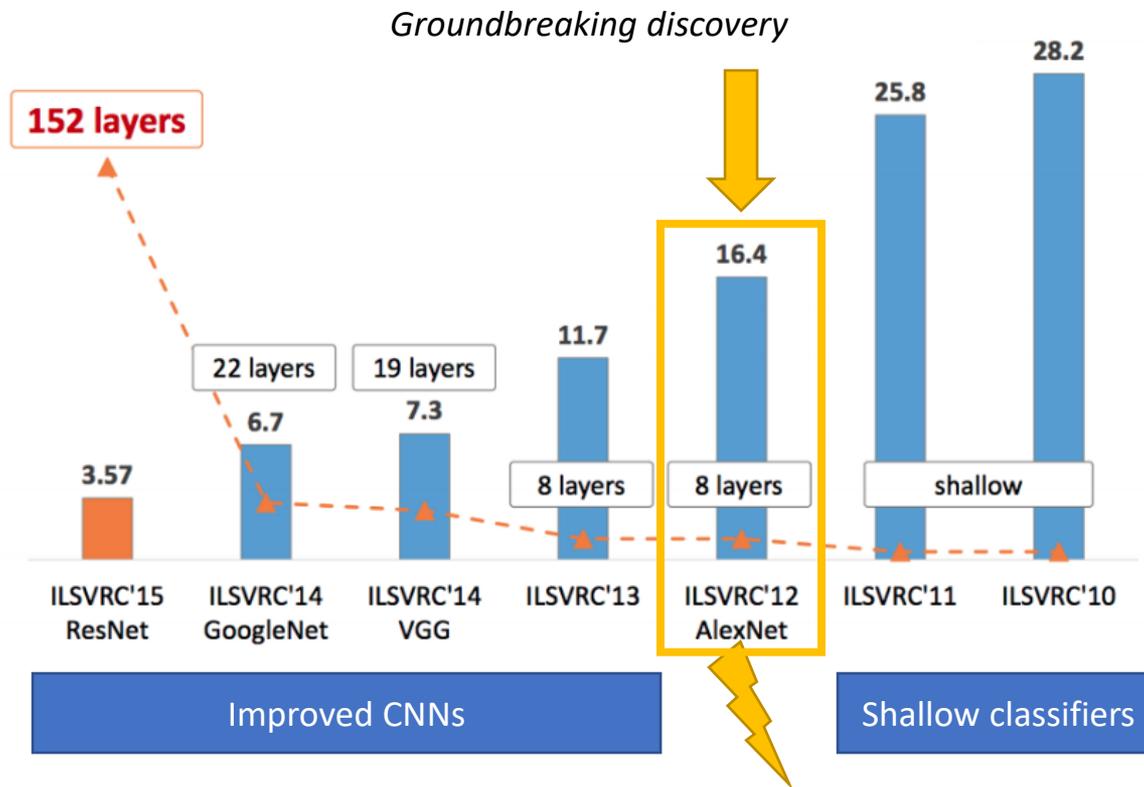


# Traditional neural network architectures

- Traditionally, neural networks receive input features that are extracted from data (text, images, etc.) and are task independent.
- This creates a bottleneck: only so much can you learn from those task independent features.



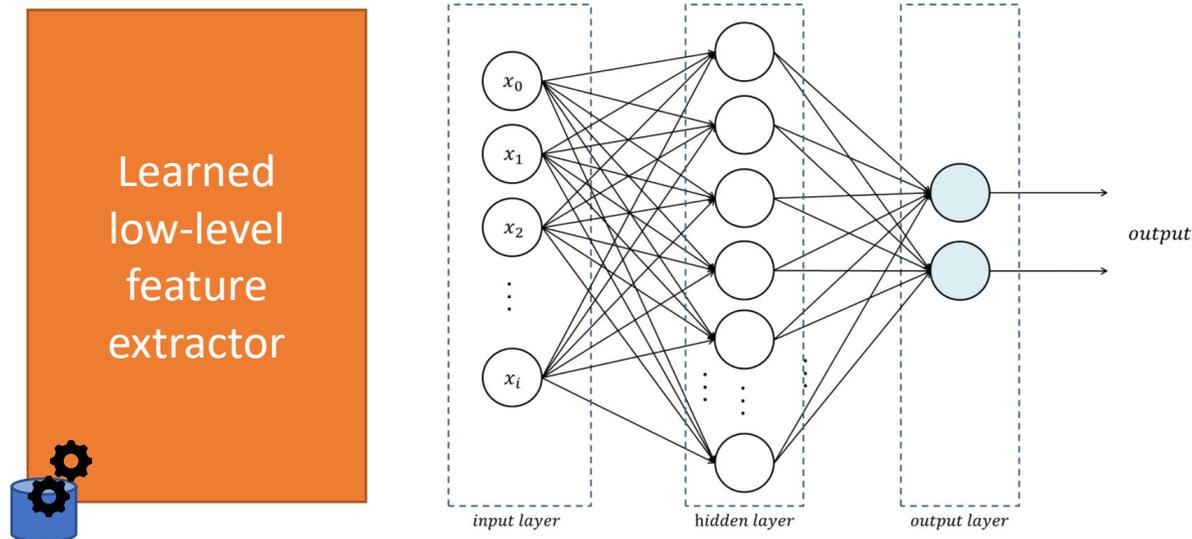
# ImageNet Challenge top-5 error





# Visual representation learning

- Deep architectures were introduced to learn data representations that were better suited to each task.
- Deep architectures look at the most basic data element, i.e., an image pixel or a text character, to learn new data representations.





# Convolution filters

- A convolution filter applies a kernel to the all image by performing the convolution operation.

$$h * A = g(x, y) = \sum_{j=-M}^M \sum_{i=-M}^M h(i, j) \cdot A(x + i, y + j)$$

$$h(i, j) = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}$$

1 <sub>x1</sub>	1 <sub>x0</sub>	1 <sub>x1</sub>	0	0
0 <sub>x0</sub>	1 <sub>x1</sub>	1 <sub>x0</sub>	1	0
0 <sub>x1</sub>	0 <sub>x0</sub>	1 <sub>x1</sub>	1	1
0	0	1	1	0
0	1	1	0	0

Image

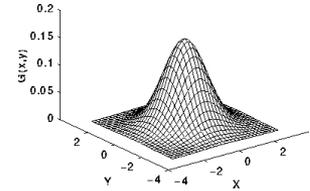
4		

Convolved  
Feature



# Low-pass convolution filters

- The low-pass convolution filter applies a gaussian filter to the input image.
- The Gaussian filter is approximated by a kernel with a given width.
- Example:



$$h_0(x, y) = \frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

$$\text{Input image } A(x, y) = \begin{bmatrix} 255 & 255 & 0 & 0 \\ 255 & 255 & 0 & 0 \\ 255 & 255 & 0 & 0 \\ 255 & 255 & 0 & 0 \\ 255 & 255 & 0 & 0 \end{bmatrix}$$

$$\text{Output image } g(x, y) = \begin{bmatrix} 255 & 191 & 64 & 0 \\ 255 & 191 & 64 & 0 \\ 255 & 191 & 64 & 0 \\ 255 & 191 & 64 & 0 \\ 255 & 191 & 64 & 0 \end{bmatrix}$$

# Example

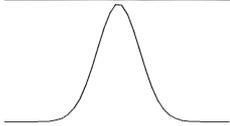
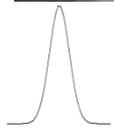
$$h_0(x, y) = \frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

3x3

5x5

7x7

9x9





# High-pass convolution filters

- High pass filters aim to detect the image edges
- Different kernels are used to detect such edges at different scales and orientations.

Input image

$$A(x, y) = \begin{bmatrix} 255 & 255 & 0 & 0 \\ 255 & 255 & 0 & 0 \\ 255 & 255 & 0 & 0 \\ 255 & 255 & 0 & 0 \end{bmatrix}$$

Output image after applying horizontal filter

Horizontal filter

$$h_h(i, j) = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$$
$$g_h(x, y) = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Output image

Output image after applying vertical filter

Vertical filter

$$h_v(i, j) = \begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix}$$
$$g_v(x, y) = \begin{bmatrix} 0 & 255 & 255 & 0 \\ 0 & 255 & 255 & 0 \\ 0 & 255 & 255 & 0 \\ 0 & 255 & 255 & 0 \\ 0 & 255 & 255 & 0 \end{bmatrix}$$

Output image

# Example



$$h_v(i, j) = \begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix}$$

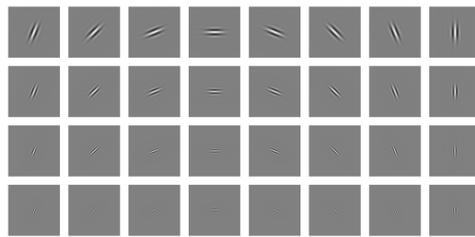


$$h_h(i, j) = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$$

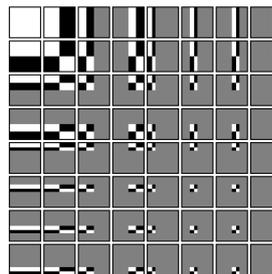


# Convolution filter kernels

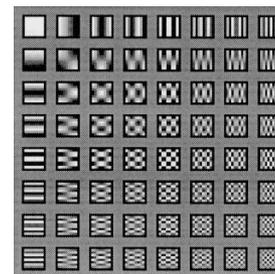
There are many different convolution filter kernels that were studied over decades in the past.



Gabor



Haar



DCT

**Can we learn the convolution kernels?**

**Yes, we can!**

# Deep CNNs

Image annotation and feature extraction

# Convolutional Networks

- Scale up neural networks to process very large images / video sequences
  - Sparse connections
  - Parameter sharing
- Automatically generalize across spatial translations of inputs
- Applicable to any input that is laid out on a grid (1-D, 2-D, 3-D, ...)



# Convolutional Network Components

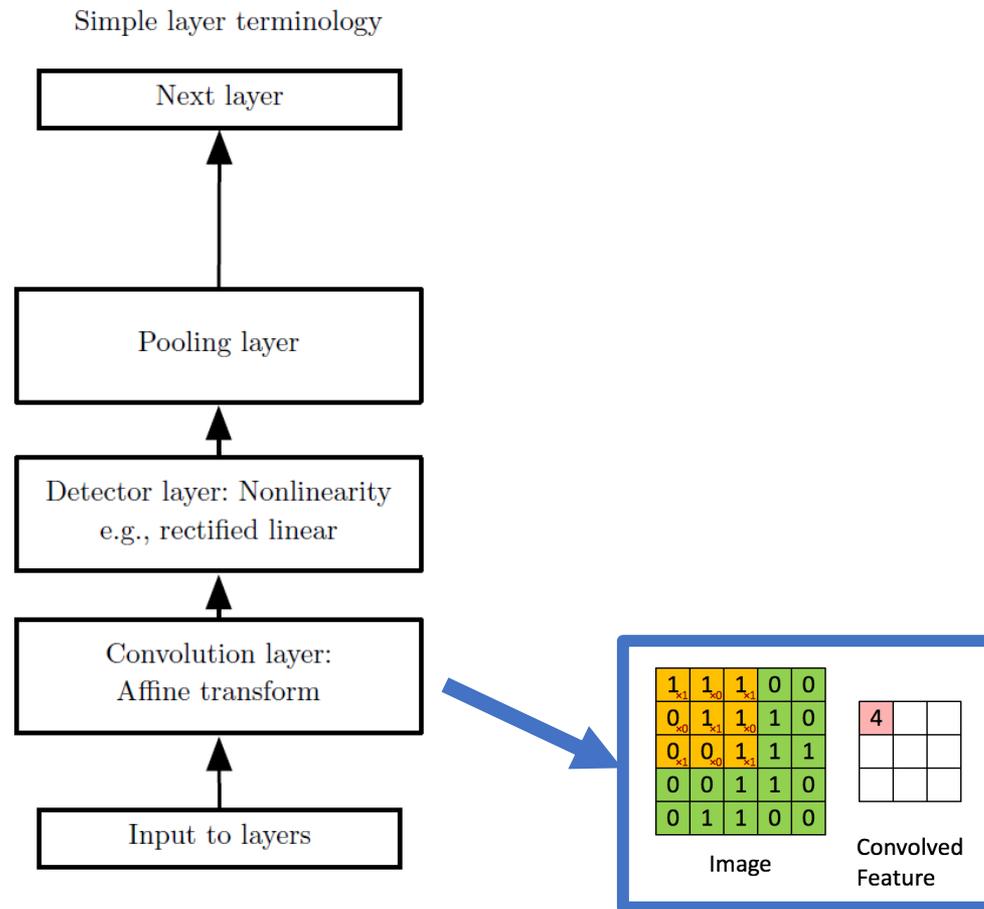


Figure 9.7



# 2D Convolution

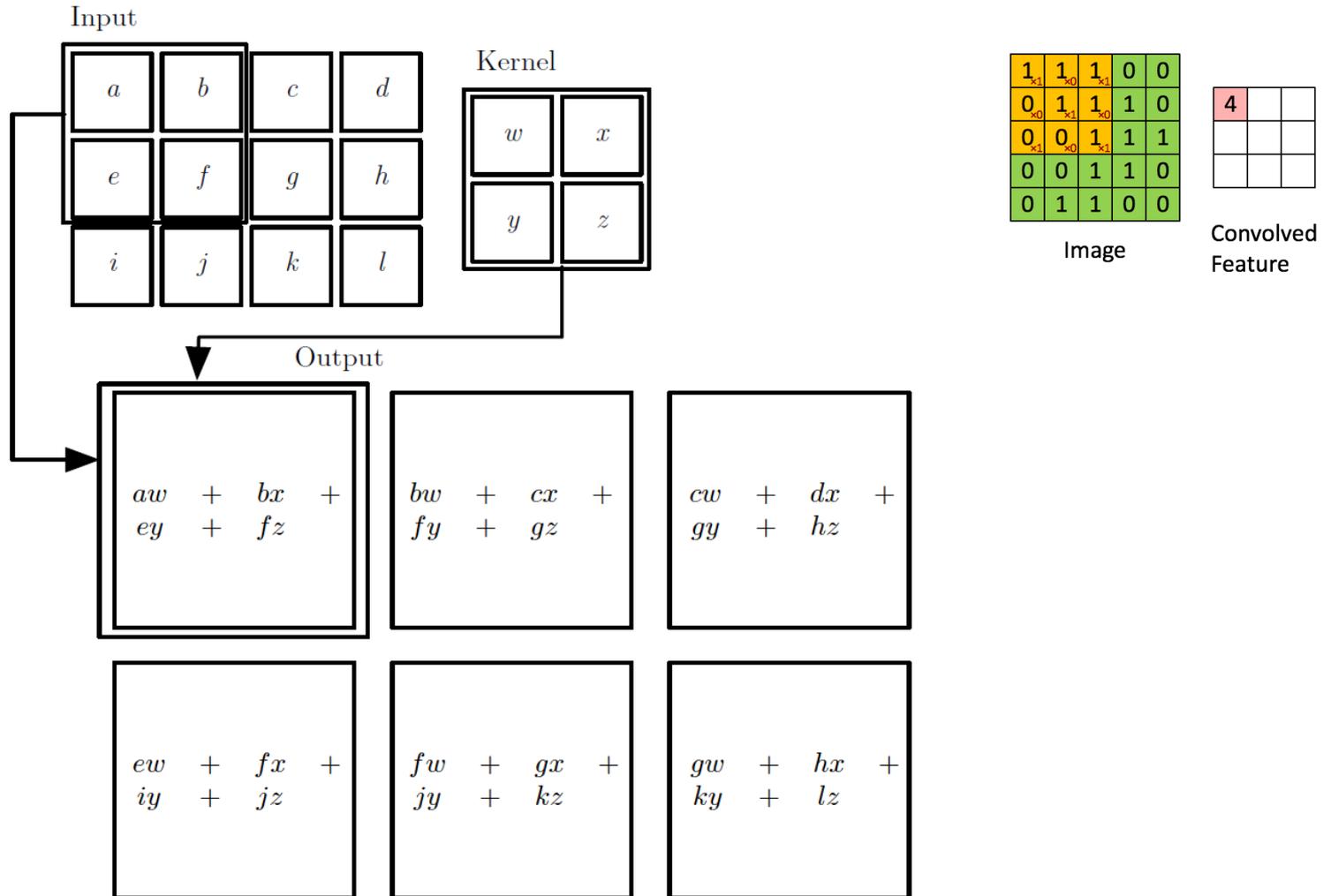


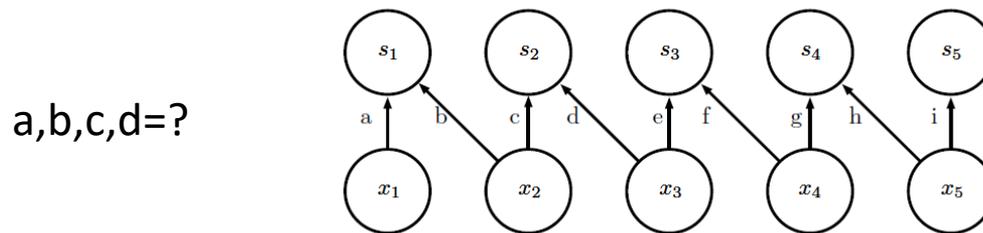
Figure 9.1



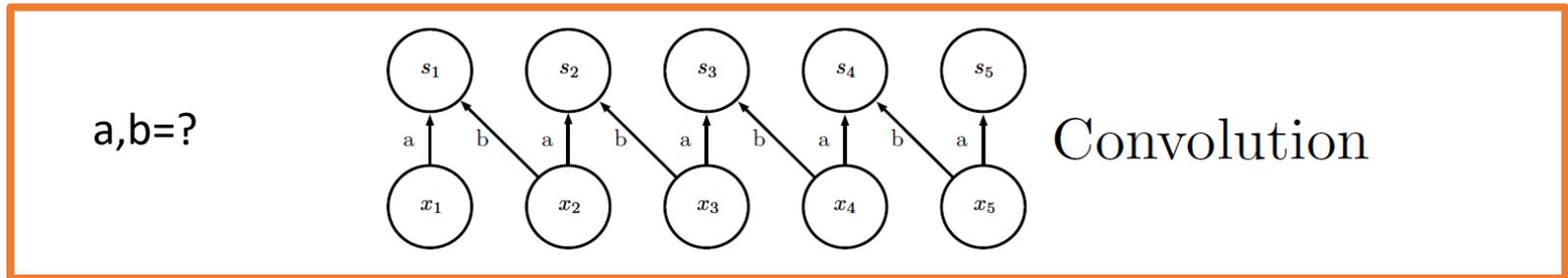
# Types of connectivity

Kernel: 

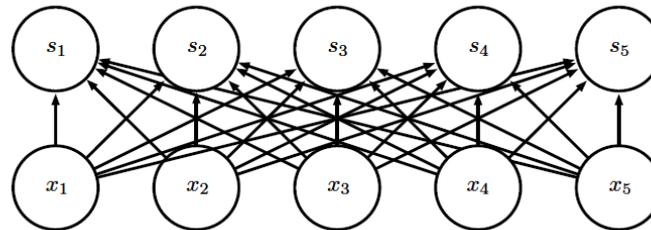
1	-1
---	----



Local connection:  
like convolution,  
but no sharing

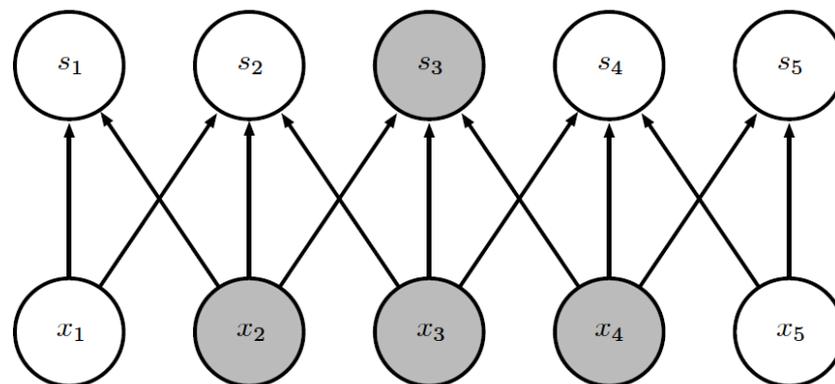
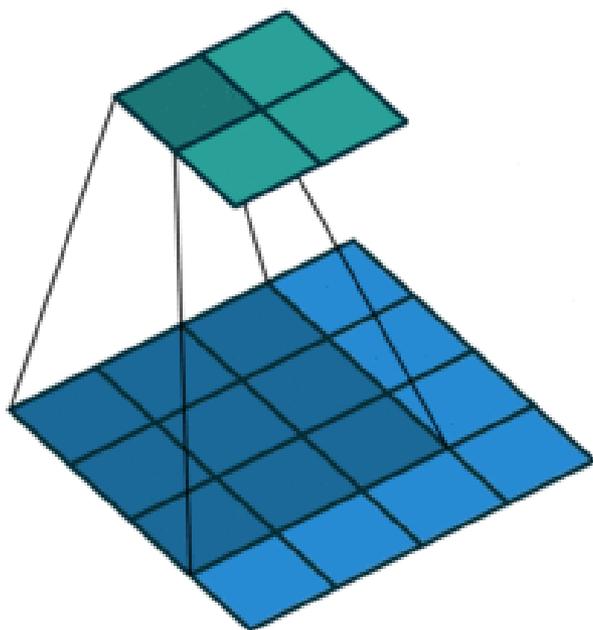


Convolution



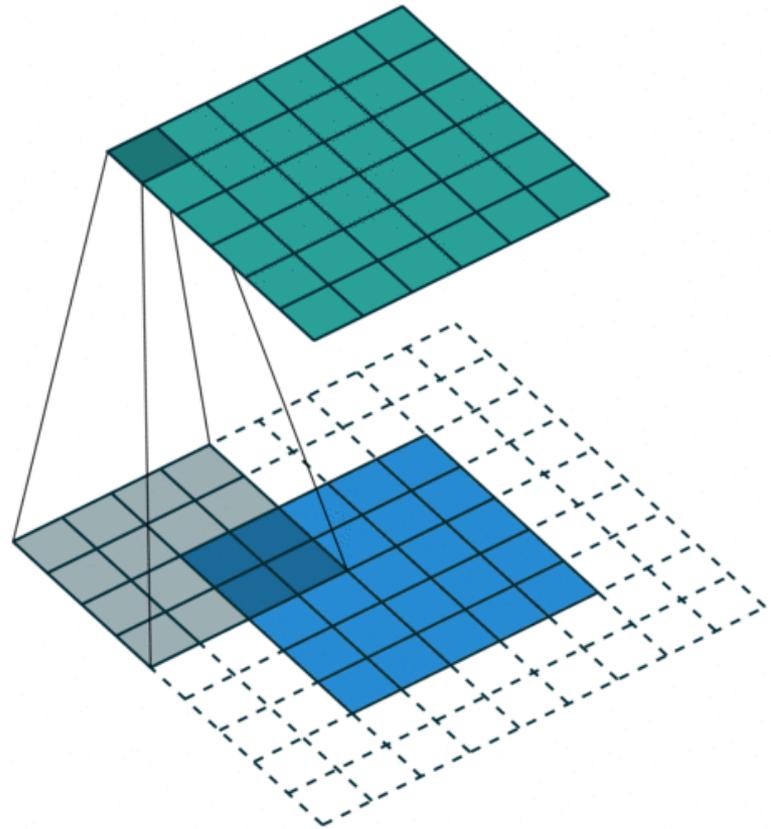
Fully connected

# Convolution as a NN



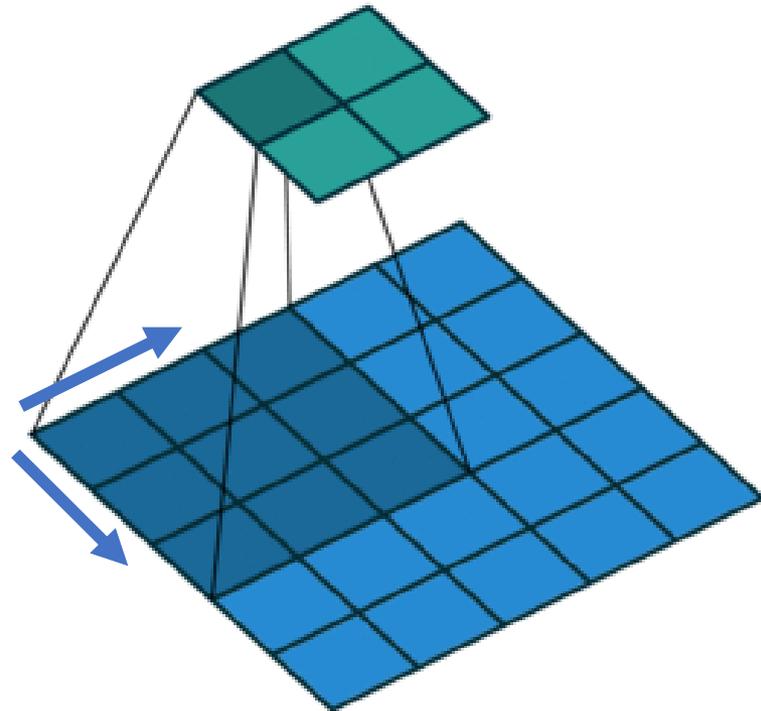
# Padding

- Padding extends an image with adds zeros to let the convolution center run over the entire image.

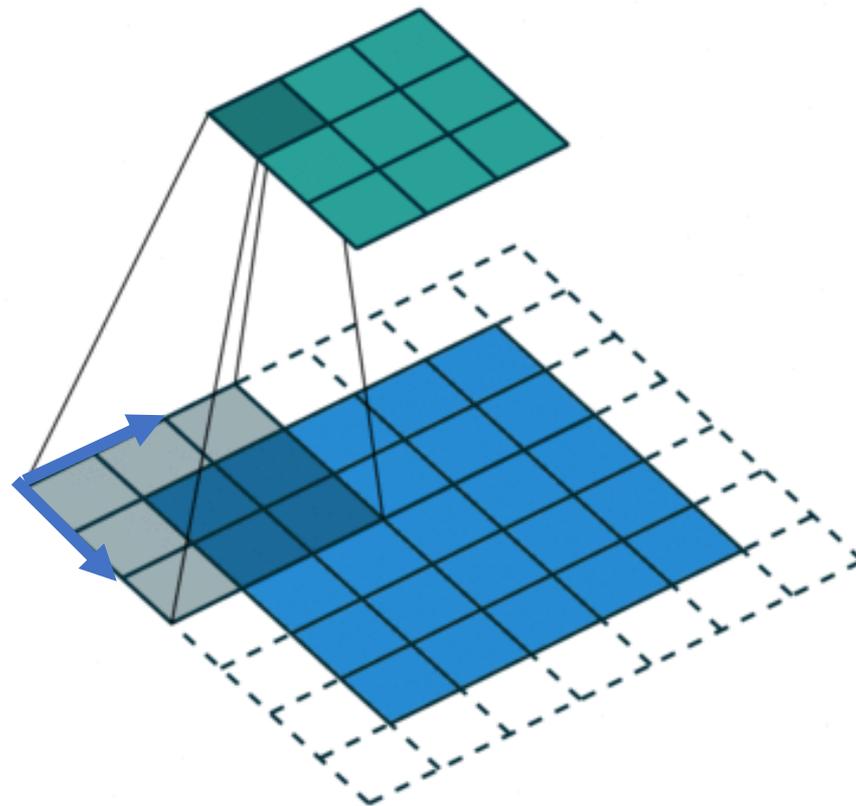


# Strides

- Defines the number of pixels a convolution moves in each direction.



# Padding with strides



# Convolutional Network Components

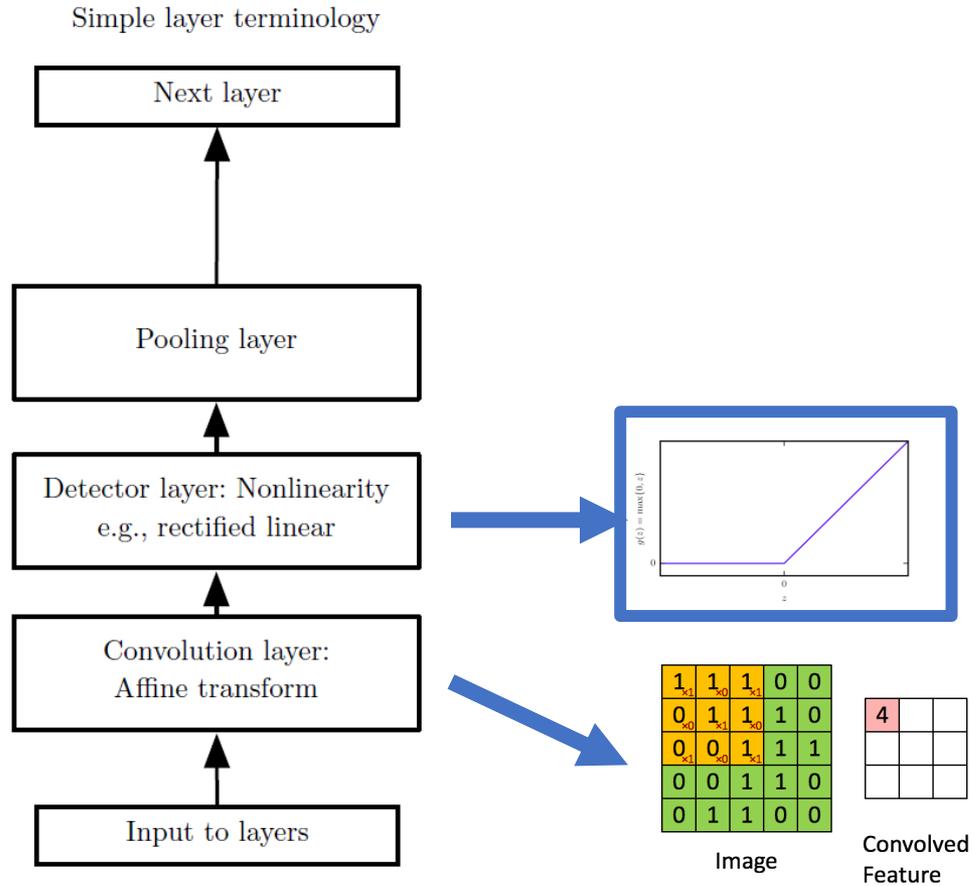
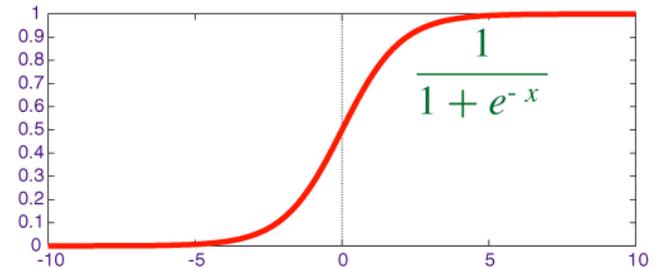


Figure 9.7

# Softmax

- The softmax function was quite popular as the activation function of neural networks.
- It is differentiable in all points
  - It is convenient from a mathematical point of view
- It can easily saturate for high values of inputs
  - Prevents passing information between layers



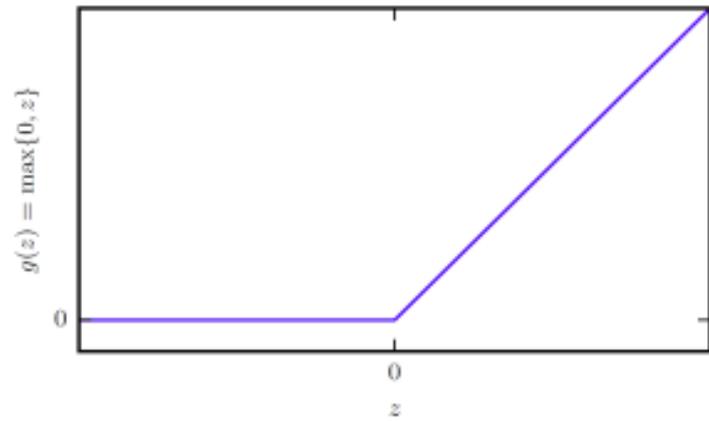


# Rectified linear unit (ReLU)

- Rectified linear activation:

$$g(z) = \max\{0, z\}$$

- Brings several advantages over traditional softmax for hidden layers:
  - Never saturates, i.e. never loses information between layers
  - Gradient is constant, i.e. faster training
  - Forces sparsity, thus removes contribution from noisy units



# Convolutional Network Components

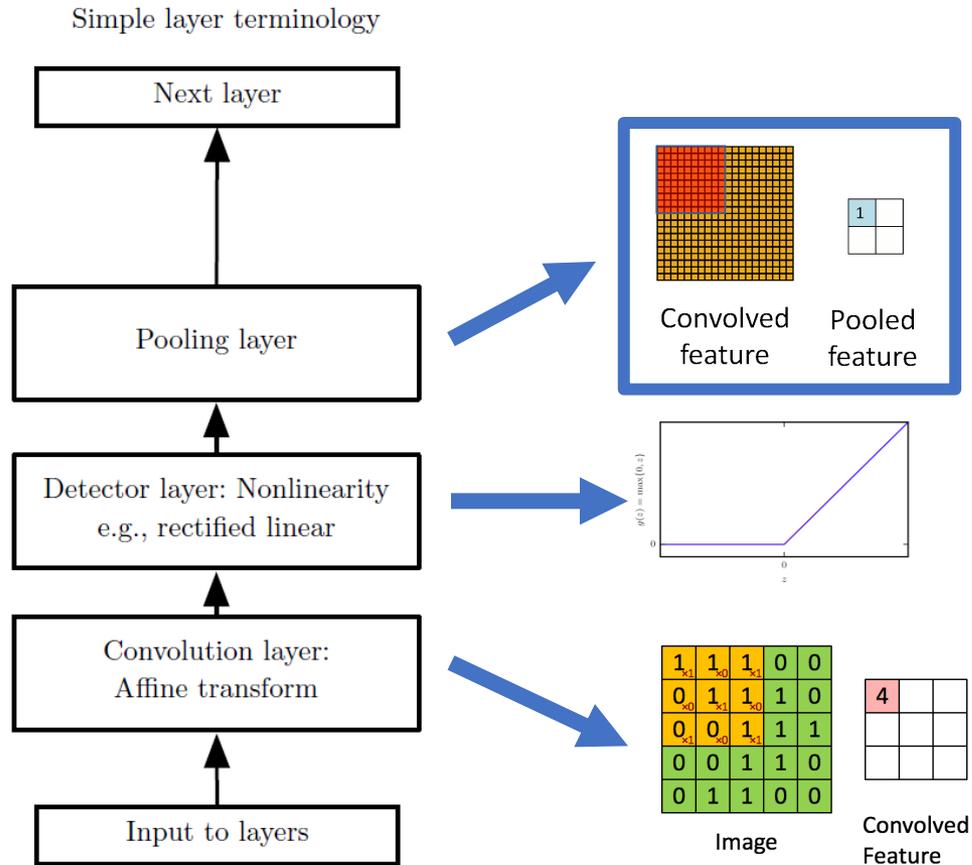
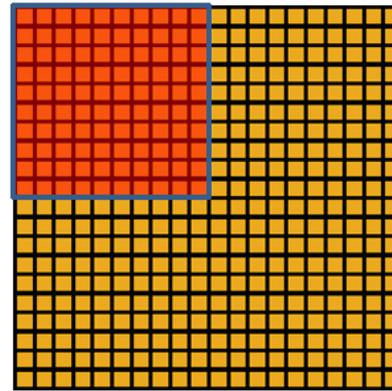


Figure 9.7

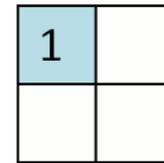


# Pooling

- Goals of pooling:
  - Downsampling
  - Translation invariant
  - Feature extraction
- Pooling strategies:
  - Max pooling
  - Min pooling
  - Average pooling



Convolved  
feature



Pooled  
feature

Figure 9.8

# Convolutional Network Components

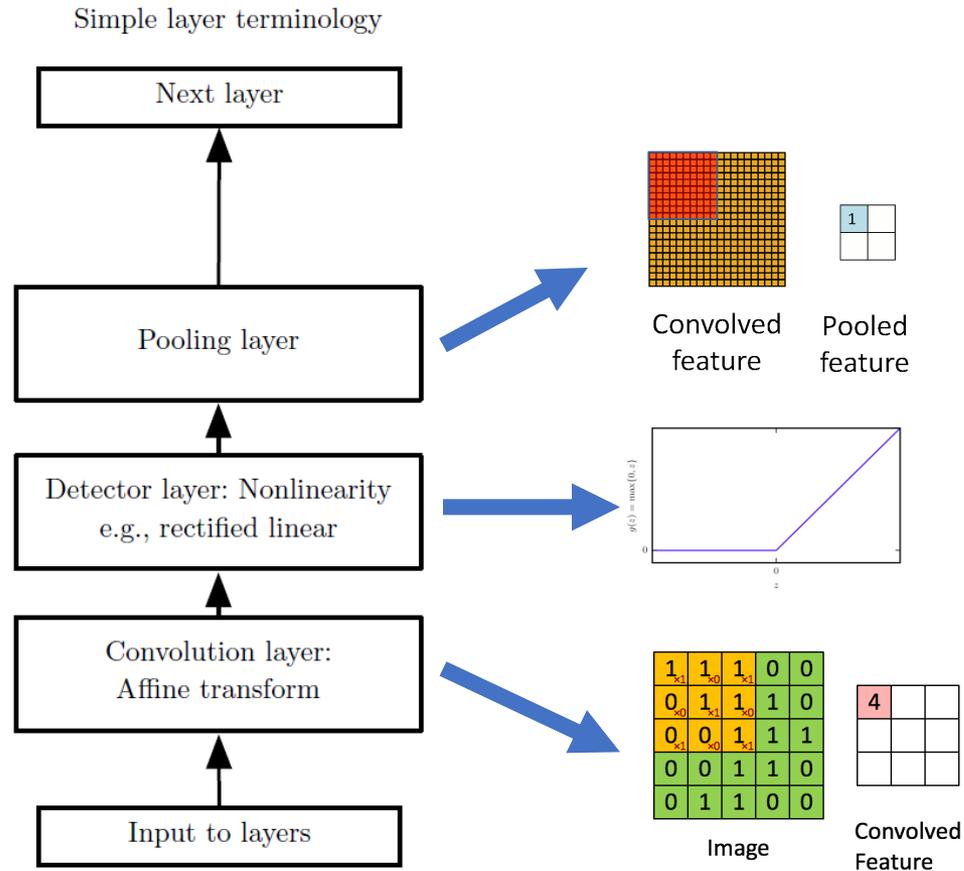
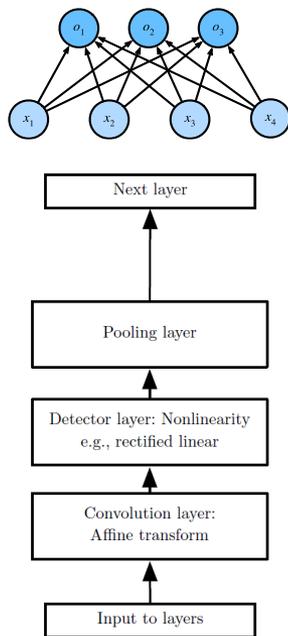


Figure 9.7

# Linear layer and softmax layer

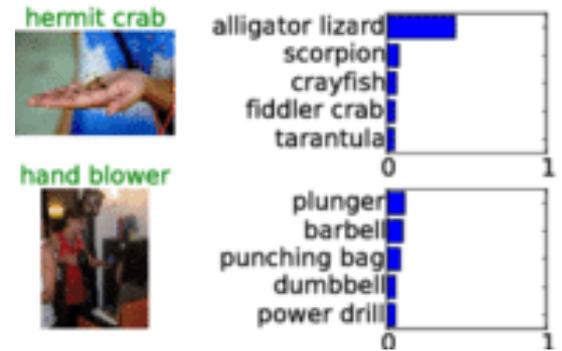
- The output of the pooling layer is flattened into a vector
- The output of the linear layer is then fed into a softmax function



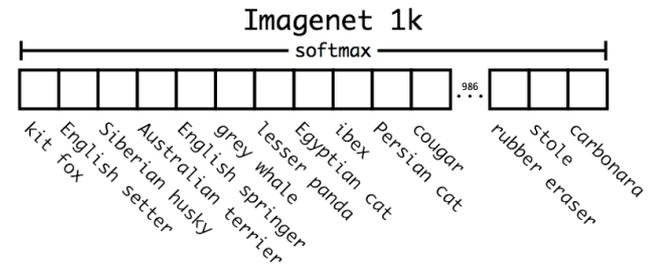
$$\text{softmax}([z_1, z_2, \dots, z_n]) = \left[ \frac{e^{z_1}}{\sum_i e^{z_i}}, \frac{e^{z_2}}{\sum_i e^{z_i}}, \dots, \frac{e^{z_n}}{\sum_i e^{z_i}} \right]$$

# Learning a keyword probability distribution

- The softmax will return a probability distribution over all keywords for each image

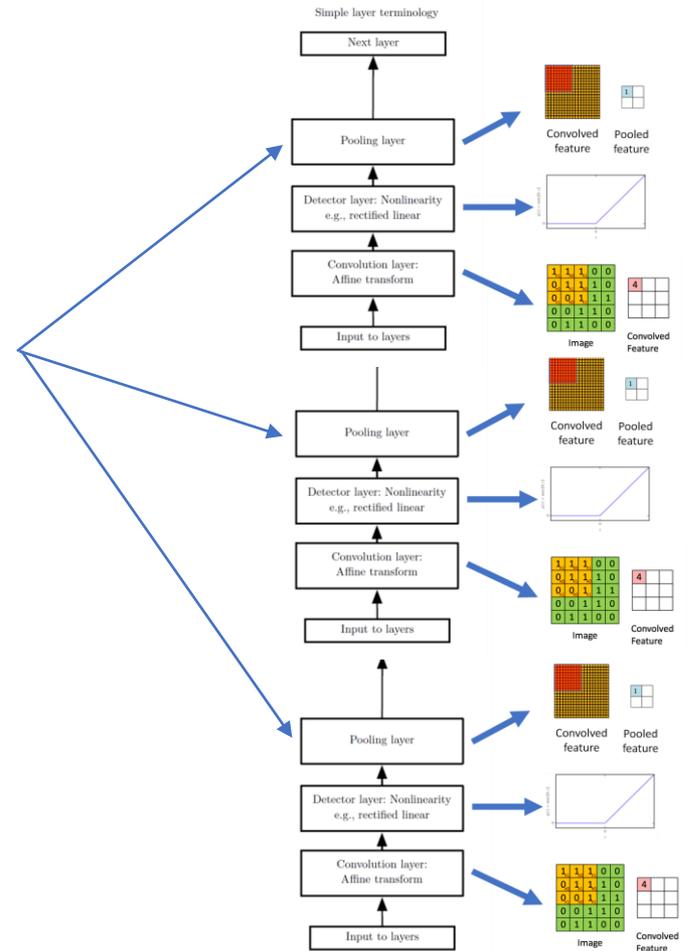


$$\text{softmax}([dog, cat, \dots, bird]) = \left[ \frac{e^{dog}}{\sum_i e^{z_i}}, \frac{e^{cat}}{\sum_i e^{z_i}}, \dots, \frac{e^{bird}}{\sum_i e^{z_n}} \right]$$

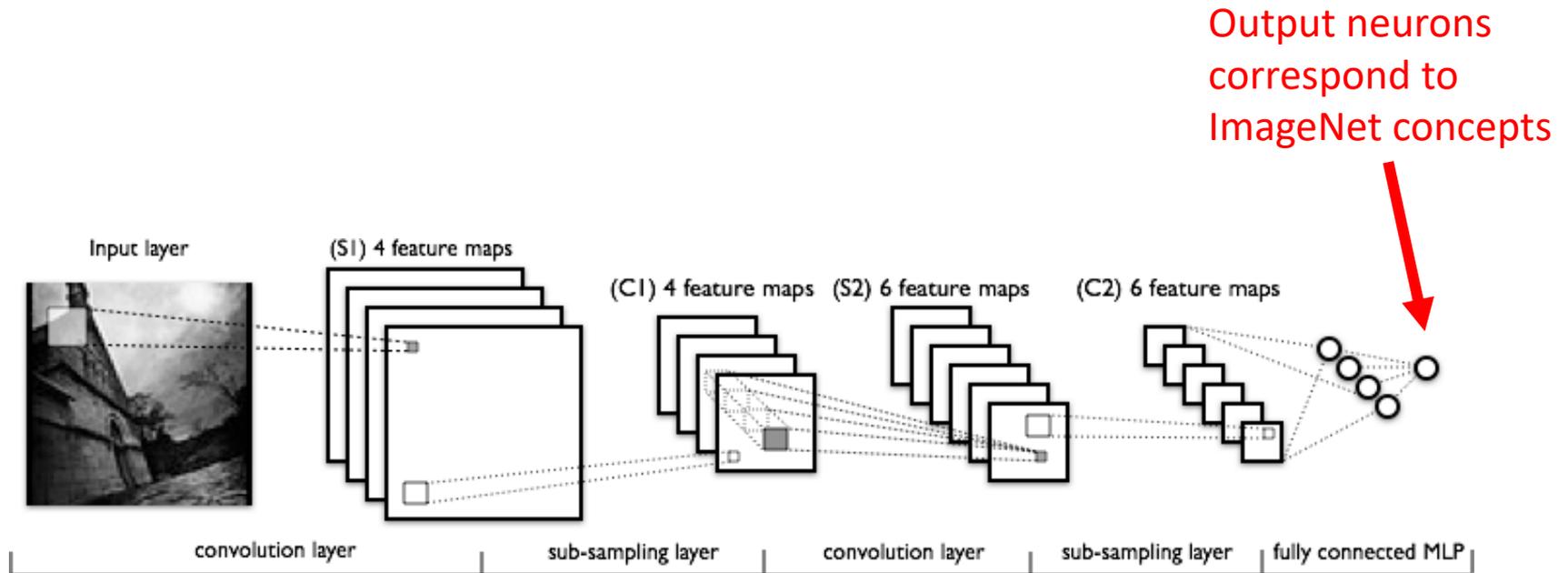


# Extracting image features for visual search

- Deep learning architectures learn hierarchies of data representations
- On each layer, we can extract the data, do a greedy pooling and then flatten the data.
  - This creates an image feature vector that we can use for searching.



# Examples of CNN architectures



# Seminal CNN architecture - 1998

PROC. OF THE IEEE, NOVEMBER 1998

7

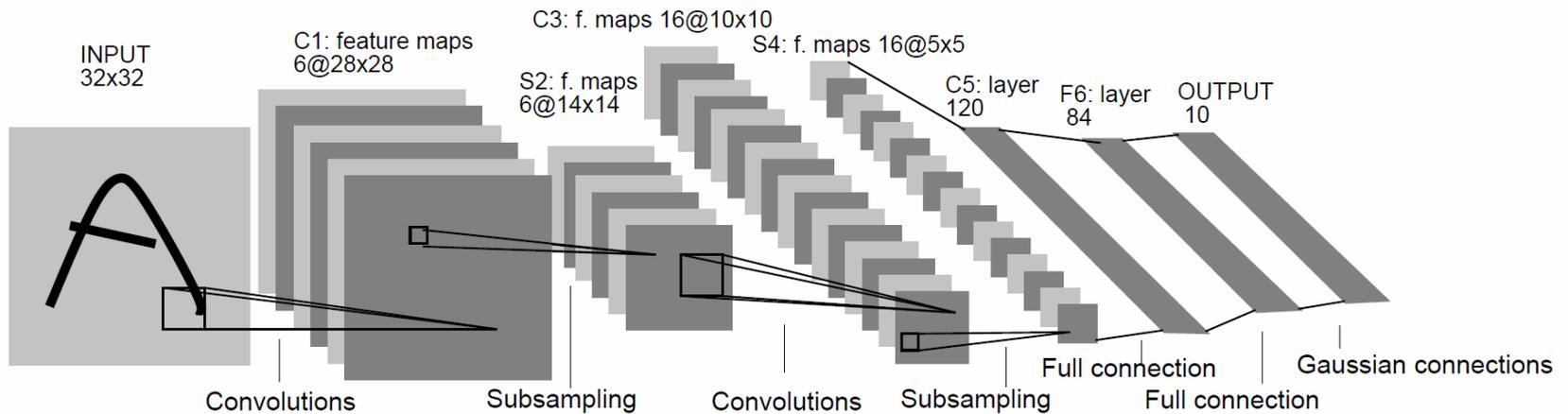
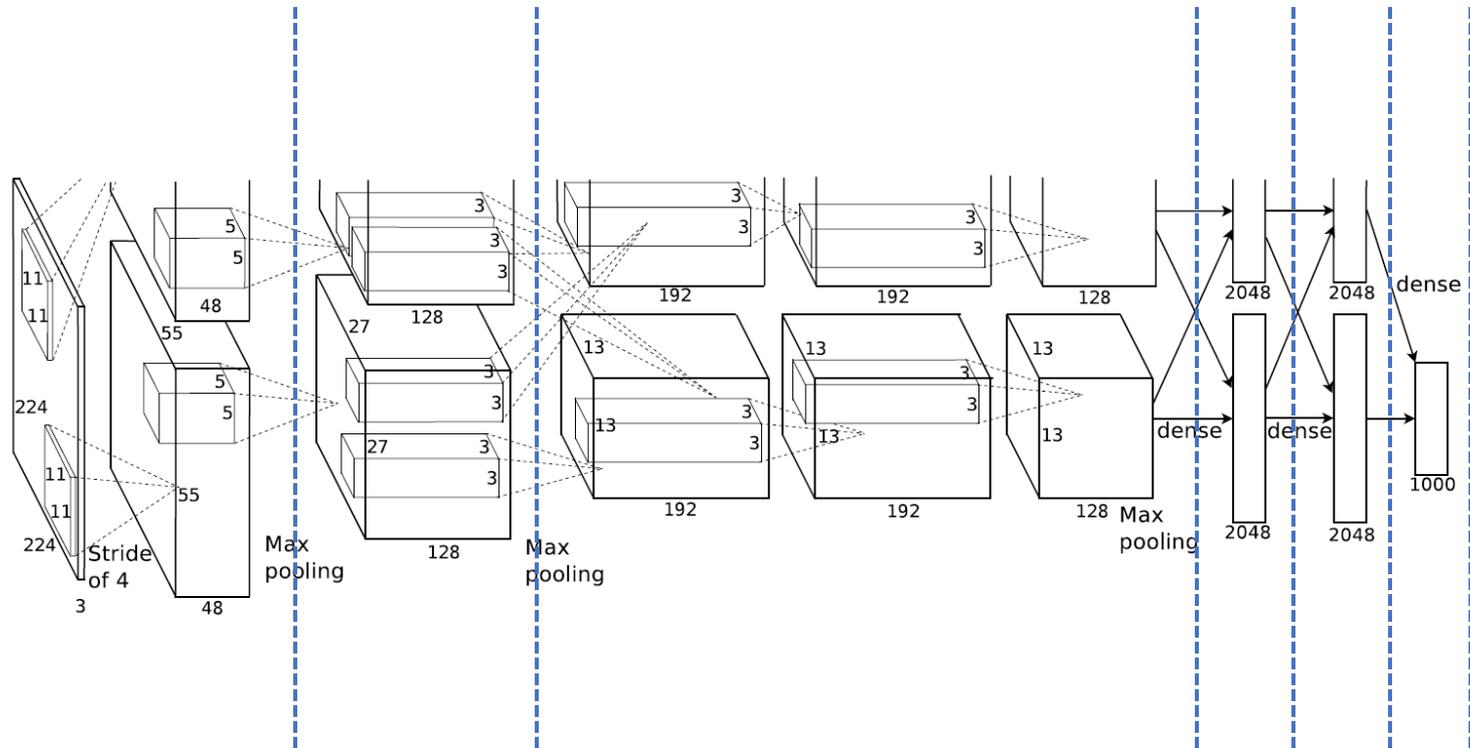


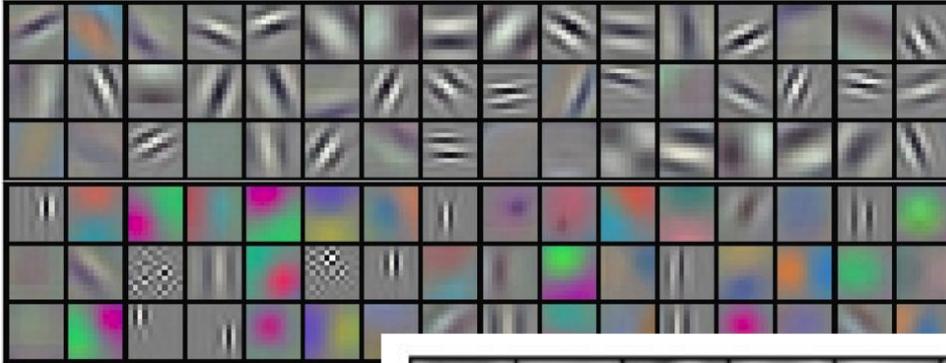
Fig. 2. Architecture of LeNet-5, a Convolutional Neural Network, here for digits recognition. Each plane is a feature map, i.e. a set of units whose weights are constrained to be identical.

# AlexNet 2012



Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems* (pp. 1097-1105).

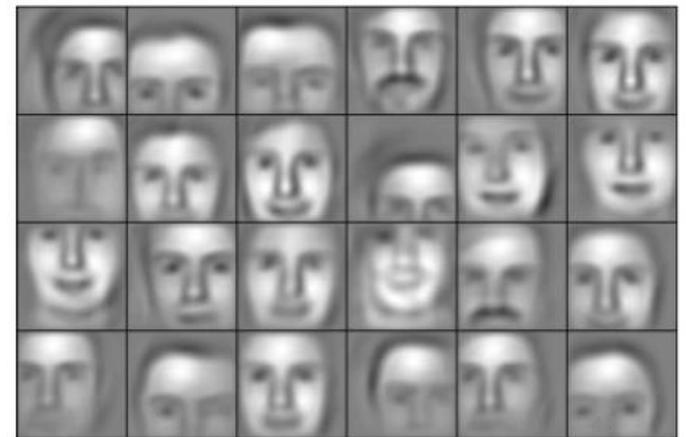
Low level CNN kernels



Mid level CNN kernels



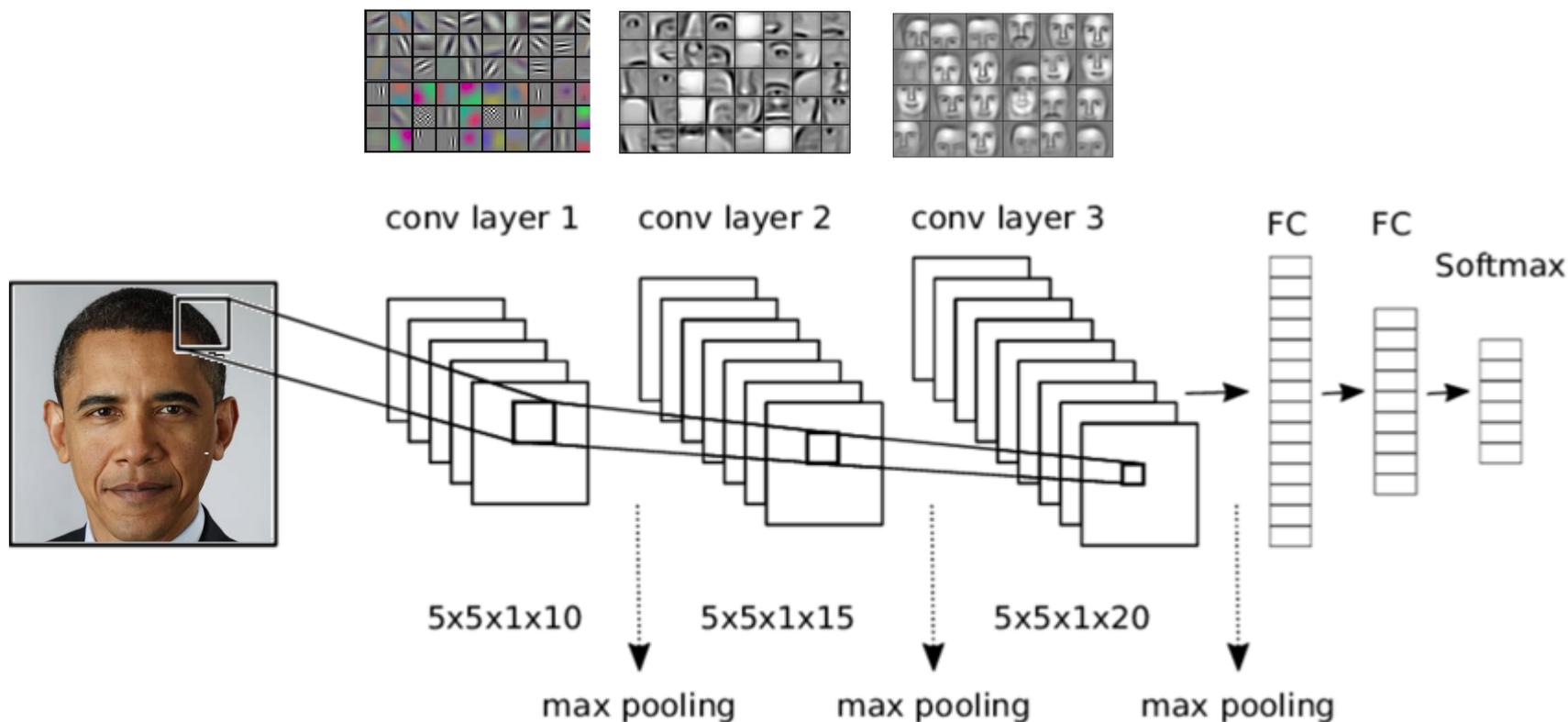
High level CNN kernels



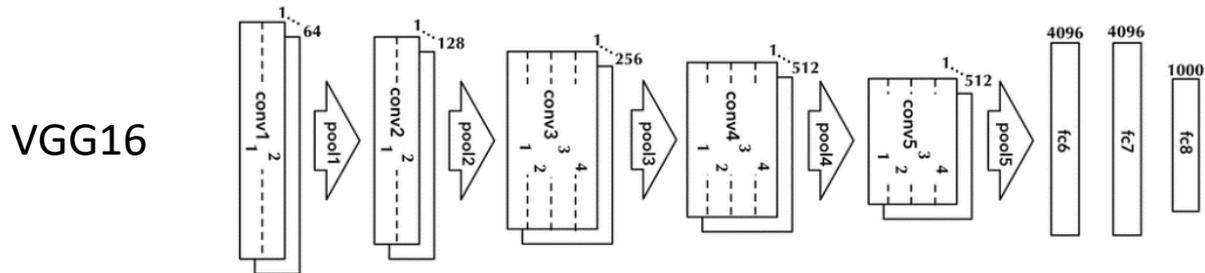
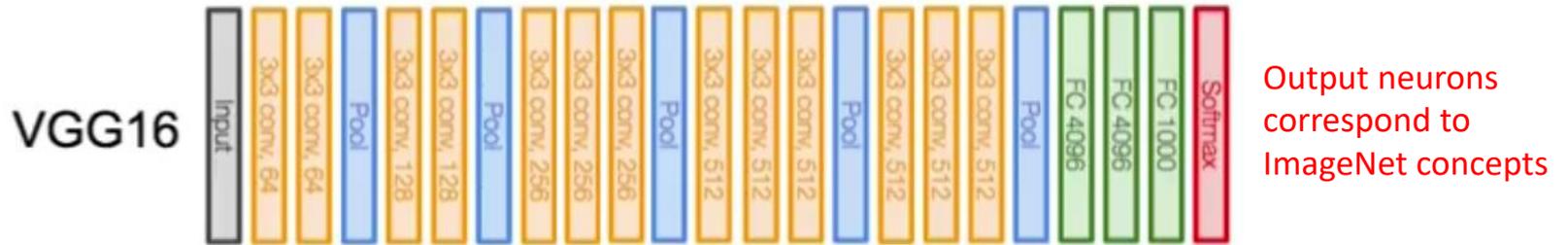
Example for  
face detection

Lee, H., Grosse, R., Ranganath, R., & Ng, A. Y. (2011). Unsupervised learning of hierarchical representations with convolutional deep belief networks. *Communications of the ACM*, 54(10), 95-103.

# Each CNN filter kernel locates a pattern



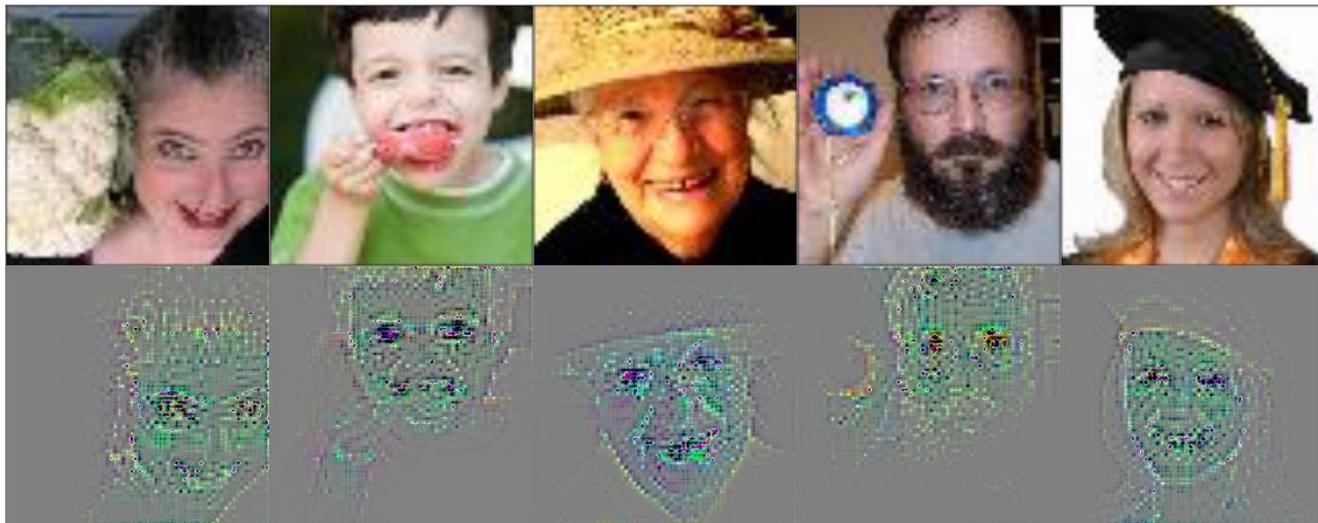
# VGG 16 architecture

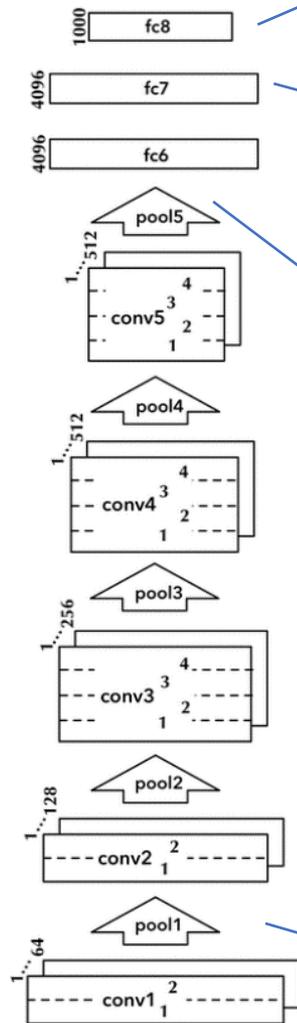


# Visualizing VGG16

[https://github.com/yosuah/vgg\\_deconv\\_vis](https://github.com/yosuah/vgg_deconv_vis)

**High level neuron from the fifth convolution block**

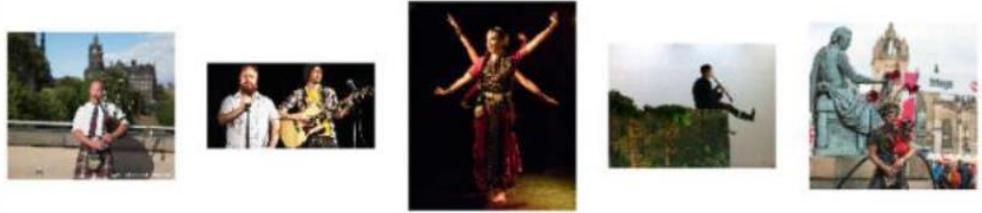




VGG16: Softmax output – 500 imagens



VGG16-fc7: **metric="euclidean"** – distâncias superiores a 0,98:



VGG16-pool5: **metric="euclidean"** – distâncias superiores a 1:



VGG16-pool1: **metric="euclidean"**:



# Summary and readings

- Learning data representations
  - Convolution operation
  - ReLU activation
  - Pooling
  - Residual Networks
- Understand visual data representations:
  - low-level layers, mid-level layers and high-level layers
- Bibliography:
  - [http://d2l.ai/chapter\\_convolutional-neural-networks/index.html](http://d2l.ai/chapter_convolutional-neural-networks/index.html)
  - [http://d2l.ai/chapter\\_convolutional-modern/index.html](http://d2l.ai/chapter_convolutional-modern/index.html)