CS 4104
APPLIED MACHINE LEARNING

Dr. Hashim Yasin

National University of Computer and Emerging Sciences,

Faisalabad, Pakistan.

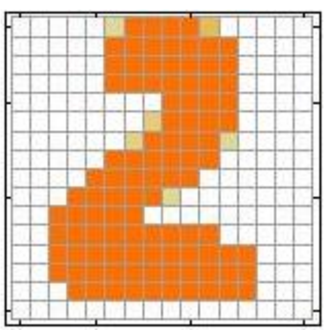
CONVOLUTIONAL NEURAL NETWORK

Why CNN

Image Classification

64x64x3

→ Cat? (0/1)

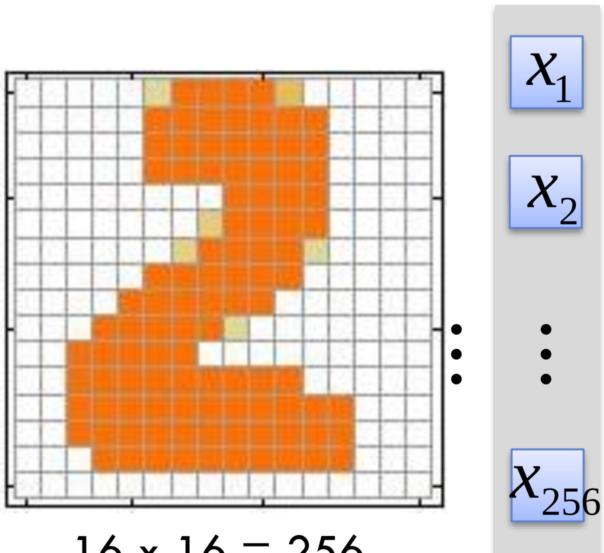


16 x 16 = 256

Text detection

Object detection

Why CNN

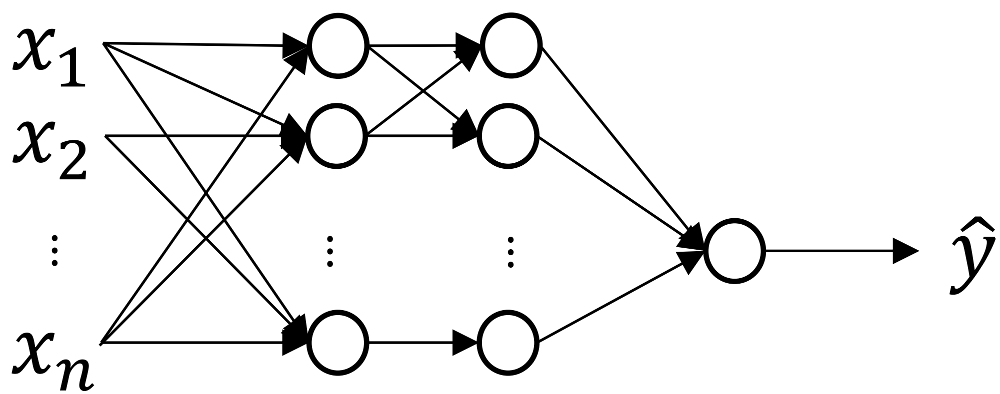


16 x 16 = 256

Ink → 1

No ink → 0

Total weights = ?



1000 1000

Why CNN

Image Classification

64x64x3

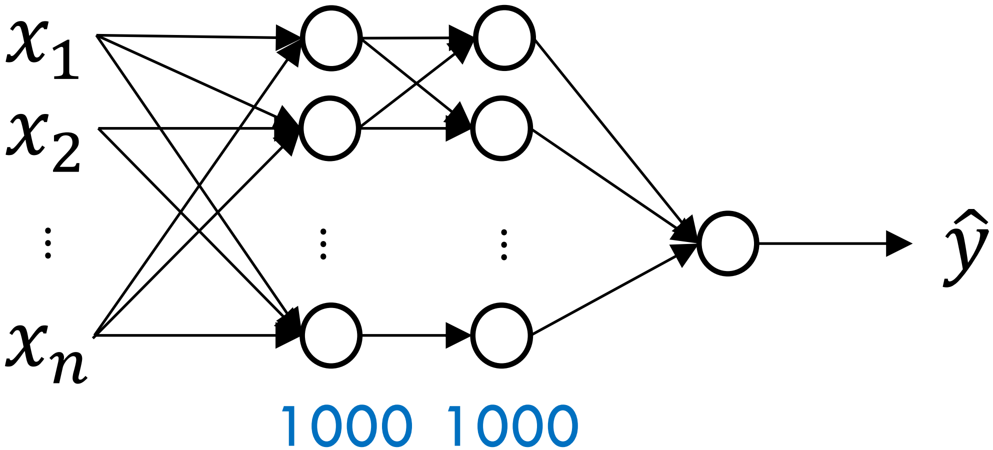
$\longrightarrow$ Cat? (0/1)

1000x1000x3

Why CNN

Image Classification

1000x1000x3



Total weights = ?

CNN

- Neural Networks that use convolution in place of general matrix multiplication in at least one layer

- There are three types of layers in the convolutional network,

  - **Convolution layer (Conv)**
  - **Pooling layer (Pool)**
  - **Fully connected layer (FC)**

Cross-correlation

- Let f be the image,
- w be the kernel of size $m \times n$
  - where $m = 2a + 1$ and $n = 2b + 1$), a and b are the positive integers.
- g be the output image

$$g(x,y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s,t) f(x+s, y+t)$$

This is called a **cross-correlation** operation:

$$\text{g} = w \otimes f$$

Cross-correlation

$$g(x,y)=\sum_{s=-a}^{a}\sum_{t=-b}^{b}w(s,t)\,f(x+s,y+t)$$

At any point (x,y), the response g(x,y) of the filter is the sum of product of filter coefficient and the image pixels

$$\begin{aligned}g(x,y)=\ &w(-1,-1)\,f(x-1,y-1)+\\ &w(-1,0)\,f(x-1,y)+\ldots\\ &w(0,0)\,f(x,y)+\ldots\\ &w(1,1)\,f(x+1,y+1)\end{aligned}$$

Convolution

- Same as cross-correlation, except that the kernel is "*flipped*" (horizontally and vertically)

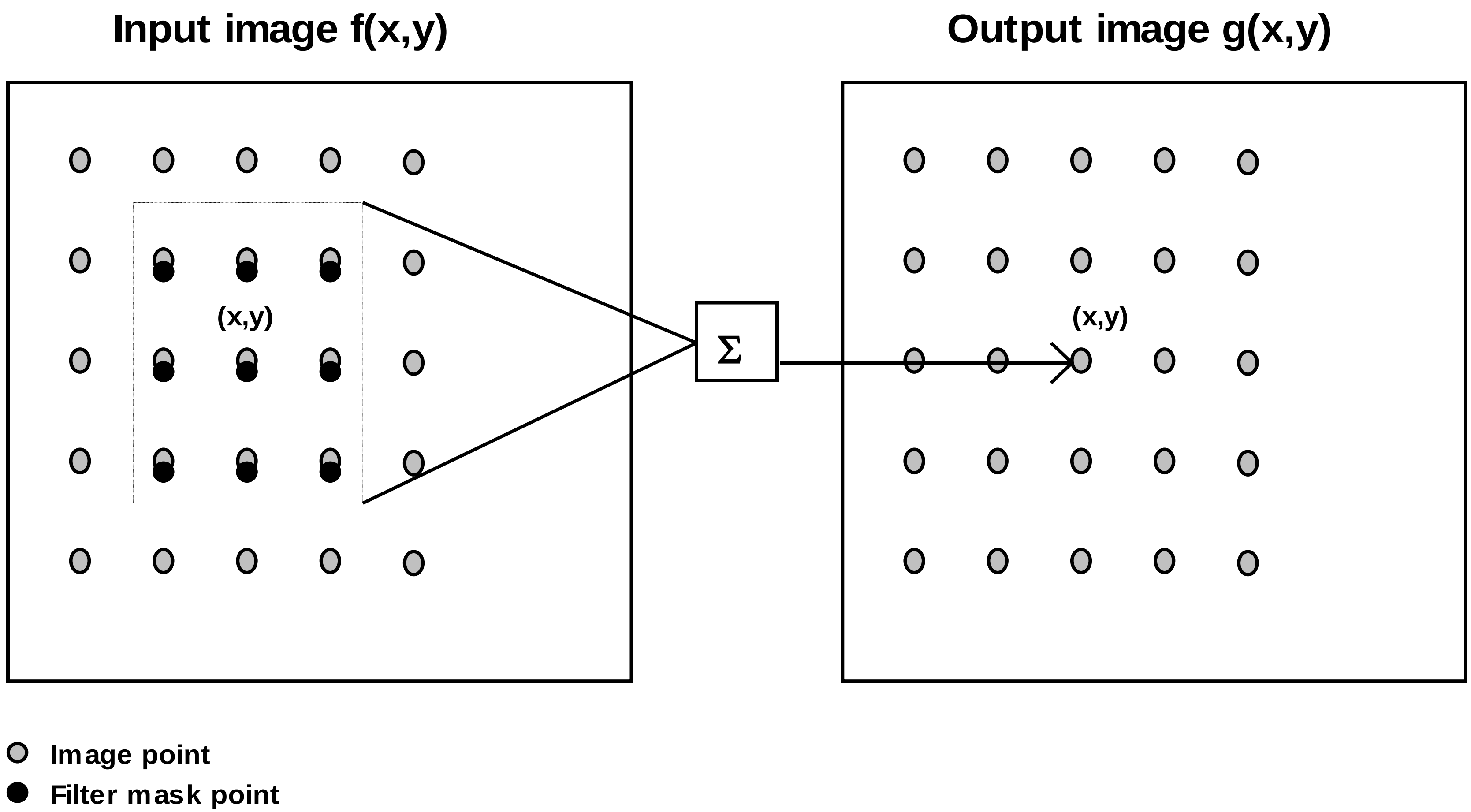
$$g(x,y)=\sum_{s=-a}^{a}\sum_{t=-b}^{b}w(s,t)\,f(x-s,y-t)$$

This is called a **convolution** operation:

$$g = w * f$$

- Convolution is **commutative** and **associative**

2D Spatial filtering



2D Spatial filtering

10	11	10	0	0	1
9	10	11	1	0	1
10	9	10	0	2	1
11	10	9	10	9	11
9	10	11	9	99	11
10	9	9	11	10	10

F

H

1/9

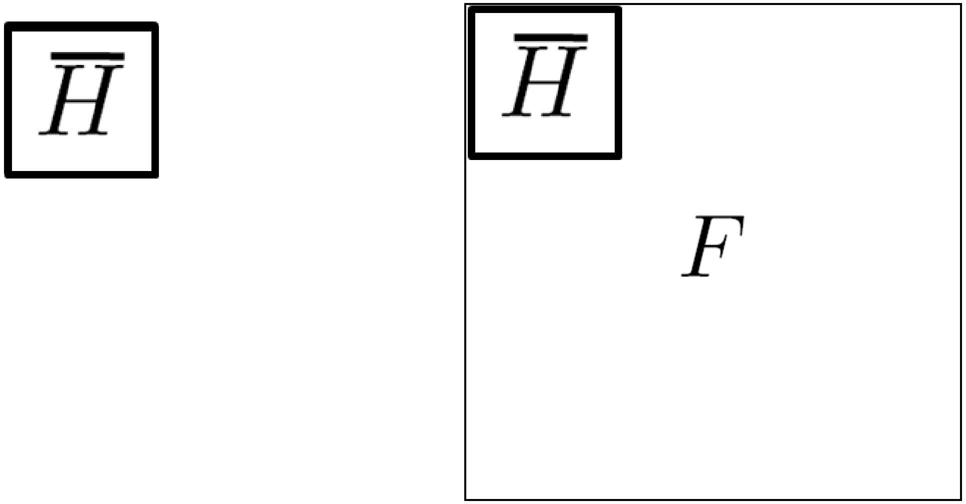
1	1	1
1	1	1
1	1	1

X	X	X	X	X	X
X	10				X
X					X
X					X
X					X
X	X	X	X	X	X

G

$$1/9.(10\times1 + 11\times1 + 10\times1 + 9\times1 + 10\times1 + 11\times1 + 10\times1 + 9\times1 + 10\times1) = 1/9.(90) = 10$$

Convolution



Convolution Examples-Mean filtering

$\frac{1}{9}$

1	1	1
1	1	1
1	1	1

$*$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

$=$

	0	10	20	30	30	30	20	10	
	0	20	40	60	60	60	40	20	
	0	30	60	90	90	90	60	30	
	0	30	50	80	80	90	60	30	
	0	30	50	80	80	90	60	30	
	0	20	30	50	50	60	40	20	
	10	20	30	30	30	30	20	10	
	10	10	10	0	0	0	0	0	

$$w \quad * \quad f \quad = \quad g$$

Linear filters: examples

Original

*

0	0	0
0	1	0
0	0	0

=

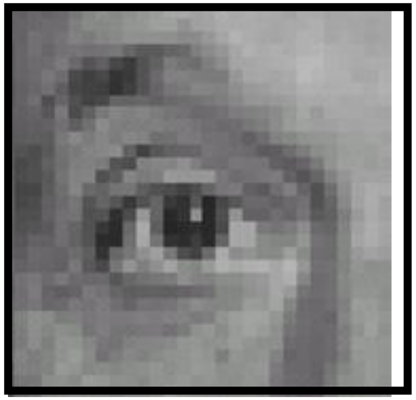
Identical image

Source: D. Lowe

Linear filters: examples

Original

$$* \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} =$$



Shifted left
By 1 pixel

Source: D. Lowe

Smoothing Spatial Filters

$$\frac{1}{9} \times \begin{array}{|c|c|c|} \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array} \qquad \frac{1}{16} \times \begin{array}{|c|c|c|} \hline 1 & 2 & 1 \\ \hline 2 & 4 & 2 \\ \hline 1 & 2 & 1 \\ \hline \end{array}$$

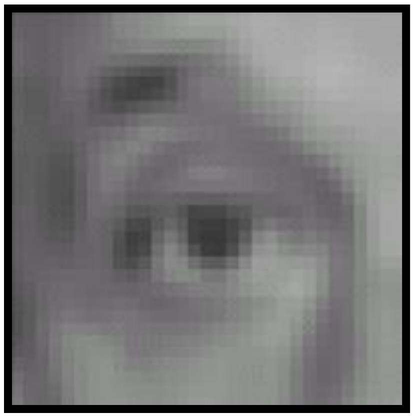
Box Filter **Weighted Average**

the center is the most important and other pixels are inversely weighted as a function of their distance from the center of the mask

Linear filters: examples

Original

$$* \; \frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \; =$$

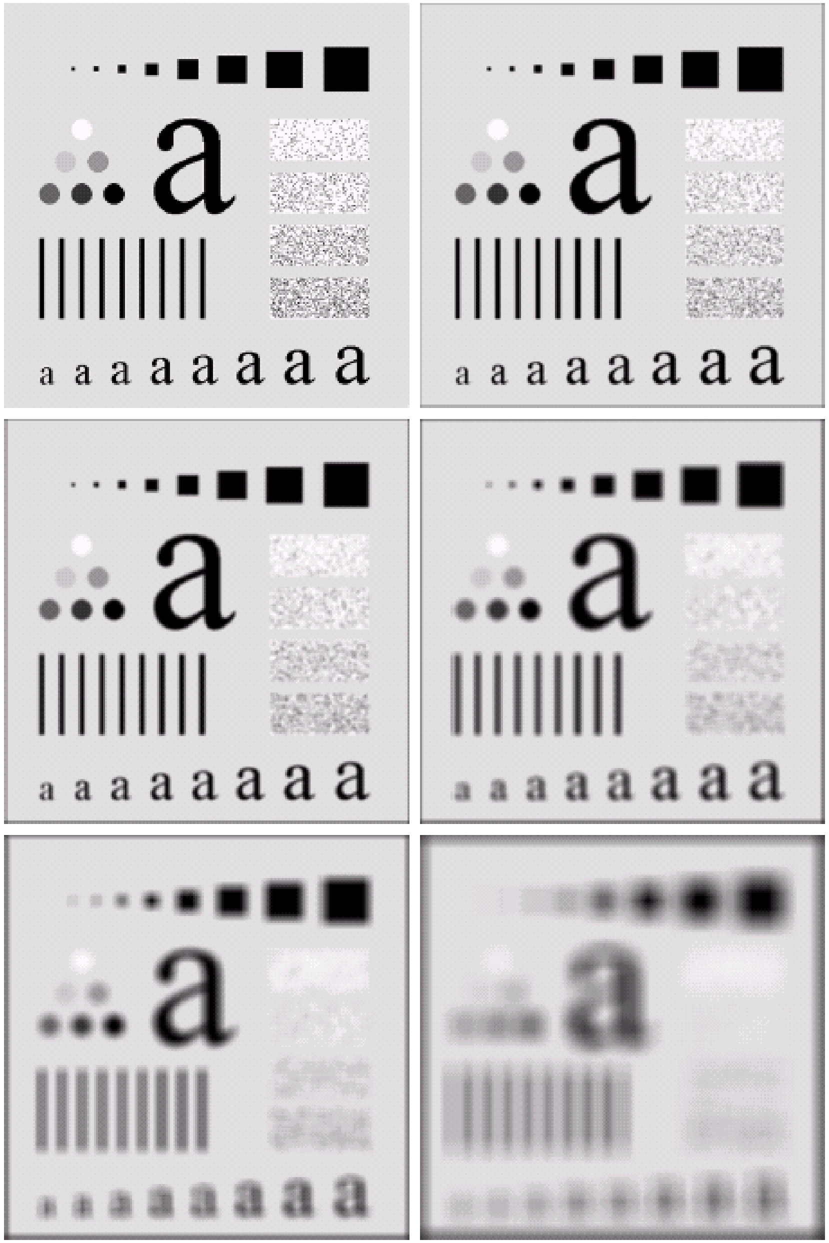


Blur (with a mean filter)

Source: D. Lowe

Smoothing Filters

- (a) is the original image of size 500x500 pixel
- (b)-(f) results of smoothing with square averaging filter masks of size n = 3, 5, 9, 15 and 35, respectively.
- Note:
  - The *big mask is used to eliminate small objects from an image*.
  - The *size of the mask establishes the relative size of the objects* that will be blended with the background.



Gradient Operator

- The first derivatives are implemented using the **magnitude of the gradient**

$$\nabla \mathrm{f} = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \dfrac{\partial f}{\partial x} \\ \dfrac{\partial f}{\partial y} \end{bmatrix}$$

$$M(x, y) = mag(\nabla \mathrm{f}) = [G_x^2 + G_y^2]^{1/2}$$

$$= \left[\left(\frac{\partial f}{\partial x} \right)^2 + \left(\frac{\partial f}{\partial y} \right)^2 \right]^{1/2}$$

Gradient Operator

$$M(x,y) = mag(\nabla \mathrm{f}) = [G_x^2 + G_y^2]^{1/2}$$

$$= \left[\left(\frac{\partial f}{\partial x} \right)^2 + \left(\frac{\partial f}{\partial y} \right)^2 \right]^{1/2}$$

commonly approx.

the magnitude becomes nonlinear

$$M(x,y) \approx |G_x| + |G_y|$$

Sobel Operators

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

- Sobel operators, 3x3

$$G_x = (z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3)$$

$$G_y = (z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7)$$

$$M(x, y) \approx |G_x| + |G_y|$$

−1	−2	−1
0	0	0
1	2	1

−1	0	1
−2	0	2
−1	0	1

the weight value 2 is to achieve smoothing by giving more important to the center point

Sobel Operators

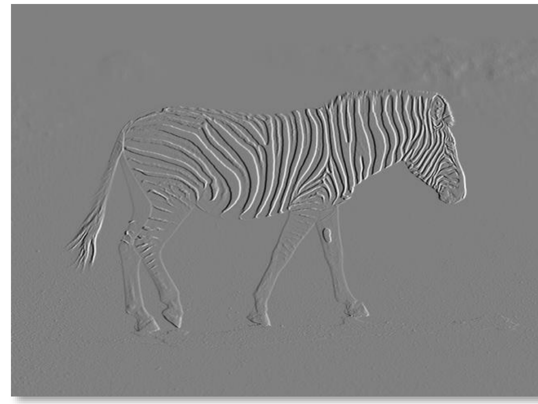
- The *summation of coefficients in all masks equals 0*, indicating that they would give a response of 0 in an area of constant gray level.

−1	−2	−1
0	0	0
1	2	1

−1	0	1
−2	0	2
−1	0	1

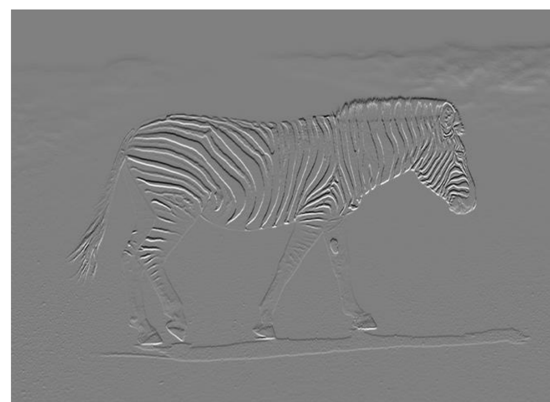
Image Derivatives

$$f$$



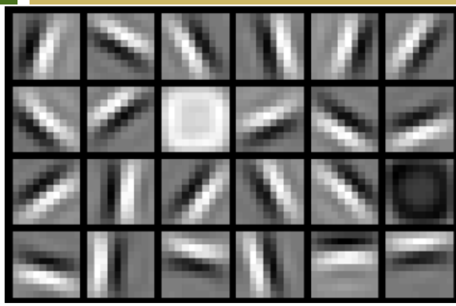
$$\frac{\partial f}{\partial x}$$

$$\|\nabla f\| = \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2}$$

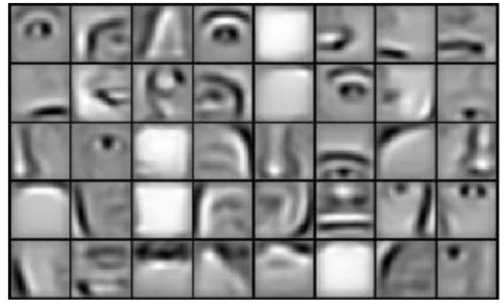


$$\frac{\partial f}{\partial y}$$

CNN … Convolution Layer

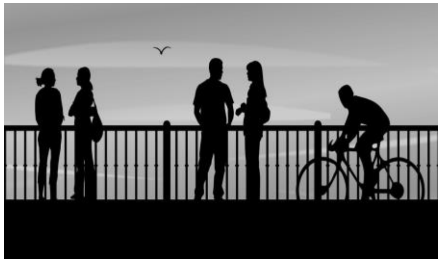


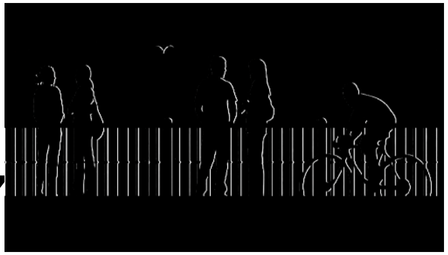
Edge detection in this layer



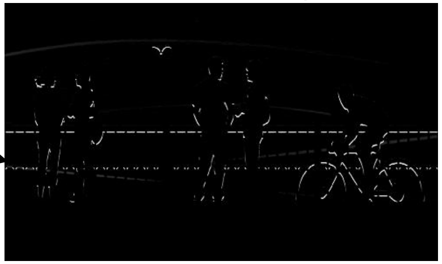
Features detection in this layer

Faces detection in this layer



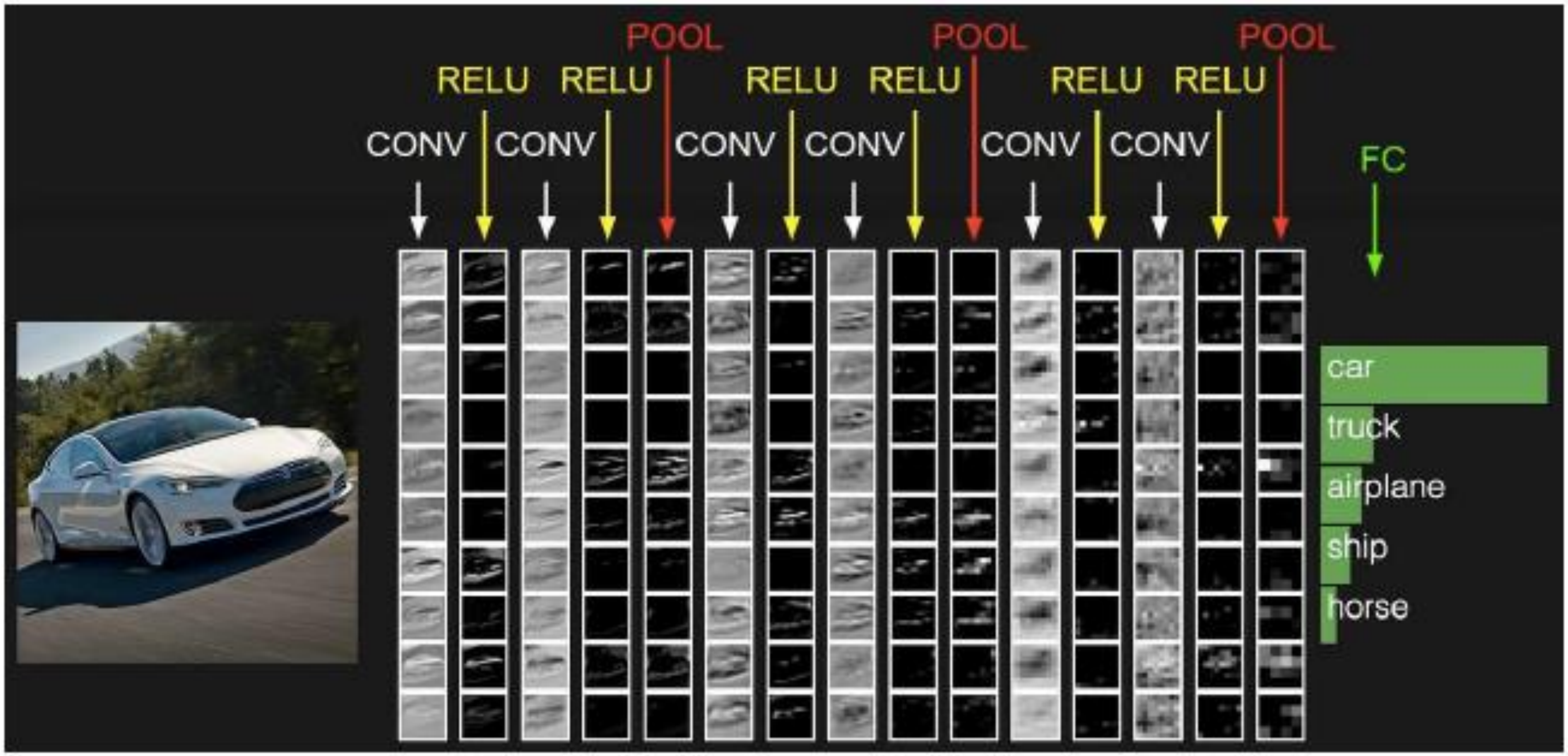


vertical edges



horizontal edges

CNN … Example


POOL
POOL
POOL
RELU
RELU
RELU
RELU
RELU
RELU
CONV
CONV
CONV
CONV
CONV
CONV
FC
car
truck
airplane
ship
horse

Acknowledgements

Stuart J. Russell and Peter Norvig, Tom M. Mitchell, Jiwon Jeong, Floydhub, Andrej Karpathy