

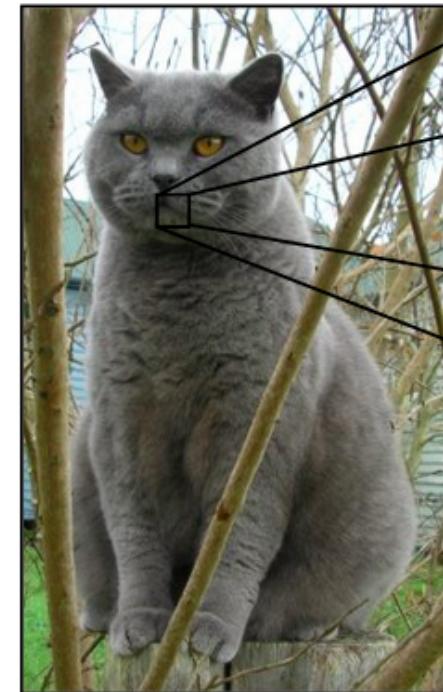
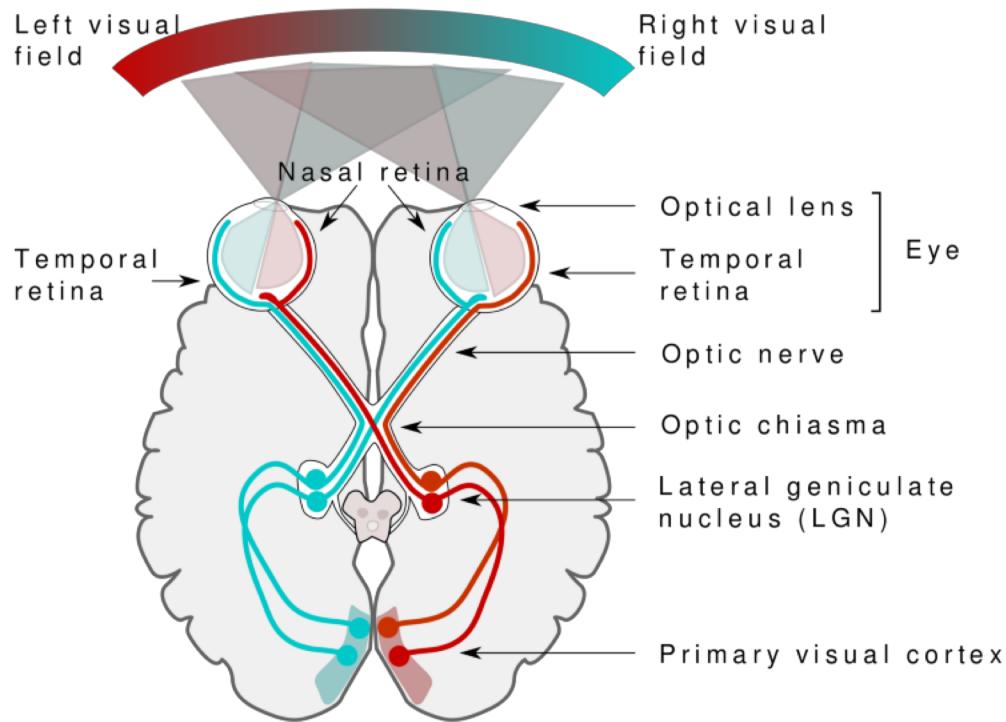


Introduction to Computer Vision

Dominique Theodore

What is computer vision?

- The science of endowing computers or other machines with vision, or the ability to see



08 02 22 97 38 15 00 40 00 75 04 05 07 78 52 12 50 77 01 06
49 49 99 40 17 81 18 57 60 87 17 40 98 43 69 45 05 56 62 00
81 49 31 73 55 79 14 29 93 71 40 67 50 88 30 03 49 13 36 65
52 70 95 23 04 60 13 42 68 11 65 56 01 32 56 71 37 02 36 91
22 31 16 71 51 67 05 89 41 92 36 54 22 40 40 28 66 33 13 80
24 47 33 60 99 03 45 02 44 75 33 53 78 36 84 20 35 17 12 50
32 98 81 28 64 23 67 10 26 38 40 67 59 54 70 66 18 38 64 70
67 26 20 68 02 62 12 20 95 63 94 39 63 08 40 91 66 49 94 21
24 55 58 05 66 73 99 26 97 17 78 78 96 83 14 88 34 89 63 72
21 36 23 09 75 00 76 44 20 45 35 14 00 62 33 97 34 51 33 95
78 17 53 28 22 75 31 67 15 94 03 80 04 62 16 14 09 53 56 92
16 39 05 42 96 35 31 47 55 58 88 24 00 17 54 24 36 29 85 57
86 56 00 48 35 71 89 07 05 44 44 37 44 60 21 58 51 54 17 58
19 80 81 68 05 94 47 69 28 73 92 13 86 52 17 77 04 89 55 40
04 52 08 83 97 35 99 16 07 97 57 32 16 26 79 33 27 98 66
01 36 68 87 57 62 20 72 03 46 33 67 46 55 12 32 63 93 53 69
04 42 16 73 55 54 39 11 24 94 72 18 08 46 29 32 40 62 76 36
20 69 36 41 72 30 23 88 34 64 89 69 82 67 59 85 74 04 36 16
20 73 35 29 78 31 90 01 74 31 49 71 48 04 01 16 23 57 05 54
01 70 54 71 83 51 54 69 16 92 33 46 61 43 52 01 89 17 47 48

What the computer sees

→ 82% cat
15% dog
2% hat
1% mug

image classification

Applications

- Retail and Retail Security
- Automotive
- Healthcare
- Agriculture
- Banking
- Industrial



Convolution

- a mathematical way of combining two signals to form a third signal
- The process of transforming an image by applying a **kernel** over each pixel and its **local neighbors** across the entire image

The diagram shows the convolution operation between an input image I and a kernel K . The input image I is a 7x7 matrix with values ranging from 0 to 1. A 3x3 red box highlights a local neighborhood around the center pixel (1,1). The kernel K is a 3x3 matrix with values 1, 0, 1; 0, 1, 0; 1, 0, 1. Dotted blue lines connect the highlighted pixels in I to the corresponding weights in K . The result of the multiplication is shown in a green box. The final output of the convolution step $I * K$ is a 5x5 matrix.

0	1	1	1	0	0	0	0	0
0	0	1	1	1	0	0	0	0
0	0	0	1	1	1	1	0	0
0	0	0	1	1	0	0	0	0
0	0	1	1	0	0	0	0	0
0	1	1	0	0	0	0	0	0
1	1	0	0	0	0	0	0	0

I

1	0	1	0	1	0	1	0	1
0	1	0	1	0	1	0	1	0
1	0	1	0	1	0	1	0	1

K

1	4	3	4	1				
1	2	4	3	3				
1	2	3	4	1				
1	3	3	1	1				
3	3	1	1	0				

$I * K$

The diagram shows the resulting convolved feature map and the final output. The convolved feature map is a 5x5 matrix where each cell contains the result of the convolution of a 3x3 neighborhood in the input with the kernel. The final output is a 3x3 matrix.

1	1	1	0	0				
0	1	1	1	0				
0	0	1	1	1				
0	0	1	1	0				
0	1	1	0	0				

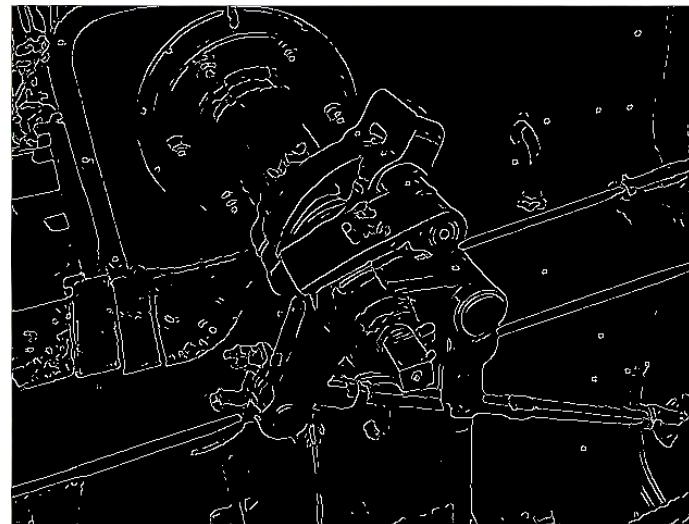
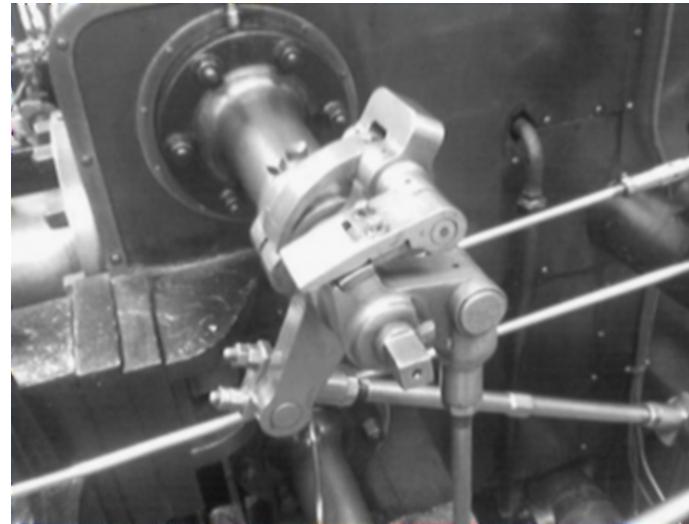
Image

4	3	4						

Convolved Feature

Edge detection

- Algorithms used to detect edges in an image
- The most common methods of edge detection
 - Sobel
 - Laplacian
 - Prewitt
 - Canny algorithm



The Sobel Edge Detector

- Gradient-based method
- Based on first order derivatives, calculates the gradients of the image for X and Y axes

$$\mathbf{G} = \sqrt{\mathbf{G}_x^2 + \mathbf{G}_y^2}$$

Magnitude

$$\Theta = \text{atan}\left(\frac{\mathbf{G}_y}{\mathbf{G}_x}\right)$$

Direction

$$G_x = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}$$

$$G_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

Laplacian edge detector

- uses only one kernel.
- calculates second order derivatives in a single pass.

$$\nabla^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$$

0	-1	0
-1	4	-1
0	-1	0

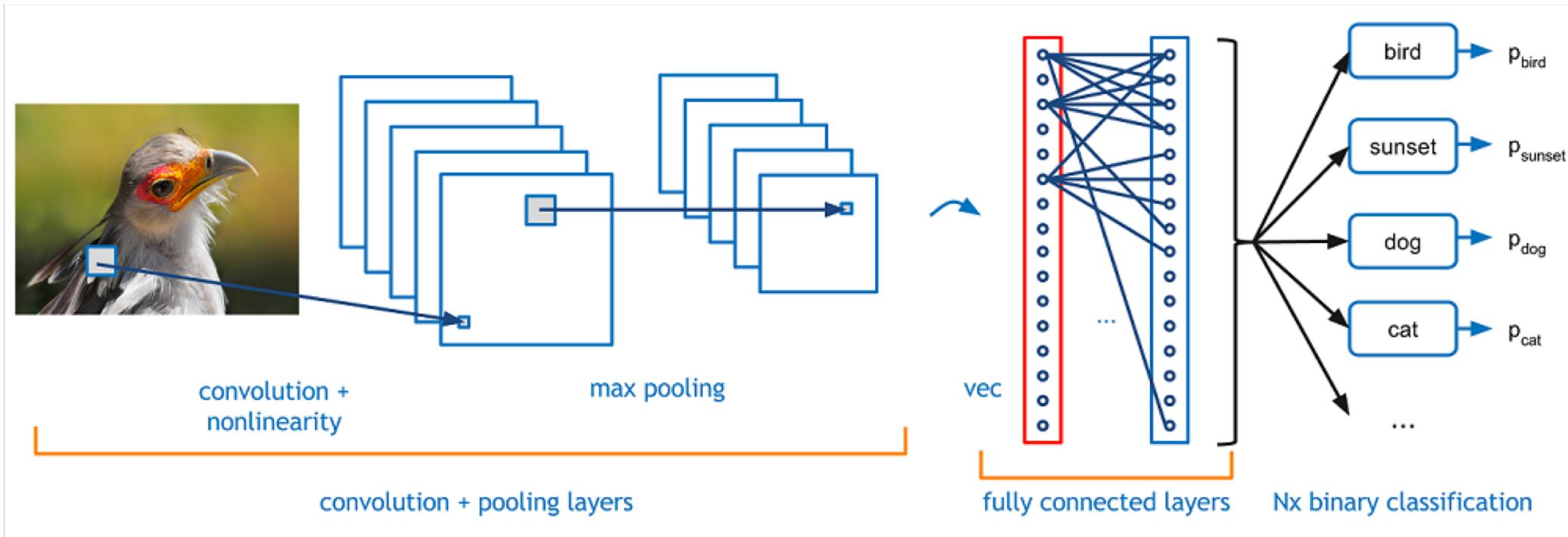
The laplacian operator

-1	-1	-1
-1	8	-1
-1	-1	-1

The laplacian operator
(include diagonals)

What is a Convolutional Neural Network?

- A class of deep neural network used primarily in image classification



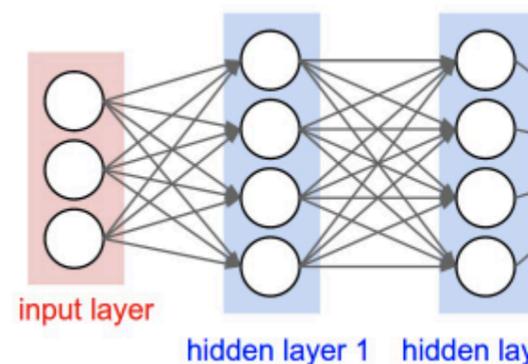
- A CNN consists of : convolution + classification

A CNN has two main parts

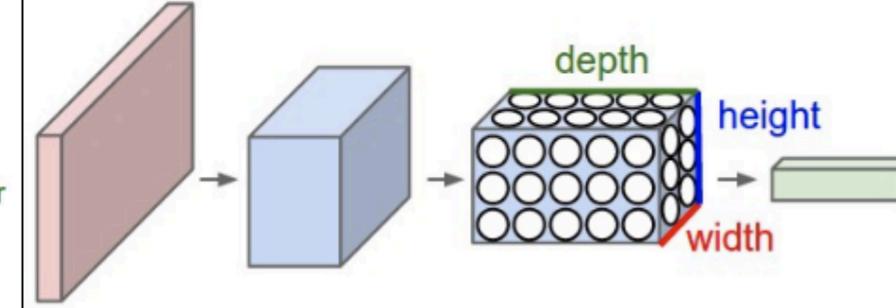
- A **convolution**/pooling mechanism to breaks up the image into features and analyzes them. This is where the features are detected.
- A fully connected **layer** that takes the output of **convolution**/pooling and predicts the best label to describe the image. This is where classification takes place.

Why do we need CNNs?

- Images are big!!
- Multi Layer Perceptrons (MLPs) are not translation invariant
- MLPs flatten out spatial data



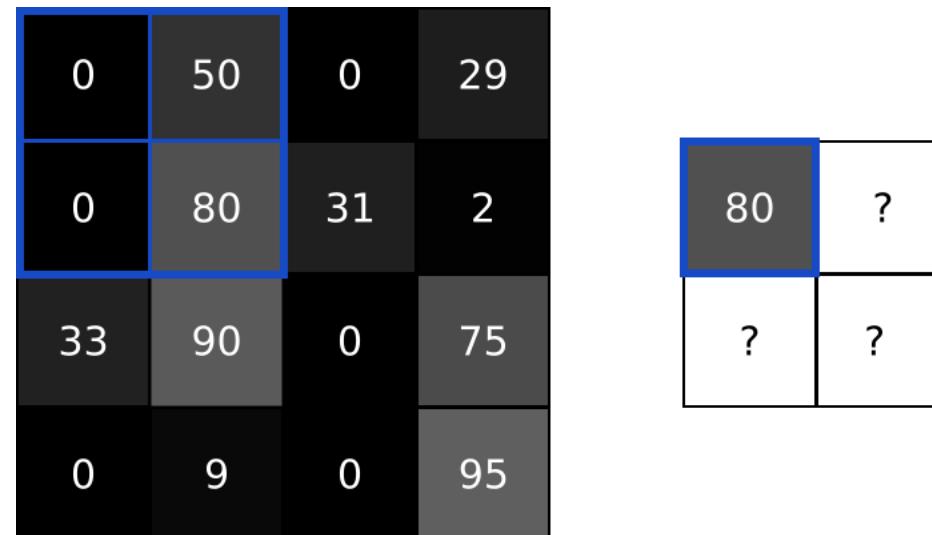
MLP



CNN

Pooling

- Neighbouring pixels in an image tend to have similar values
- Pooling helps to remove redundancy and reduce the size of the input
- Max pooling is the most common technique



A simple convolutional network

