

# Computational Intelligence

Samaneh Hosseini

Isfahan University of Technology

# Outline

- Convolutional Neural Networks
  - Computer Vision
  - What Computers “See”
  - Edge Detection Example
  - More Edge Detection

# Convolutional Neural Networks: Computer Vision

**“To know what is  
where by looking.”**



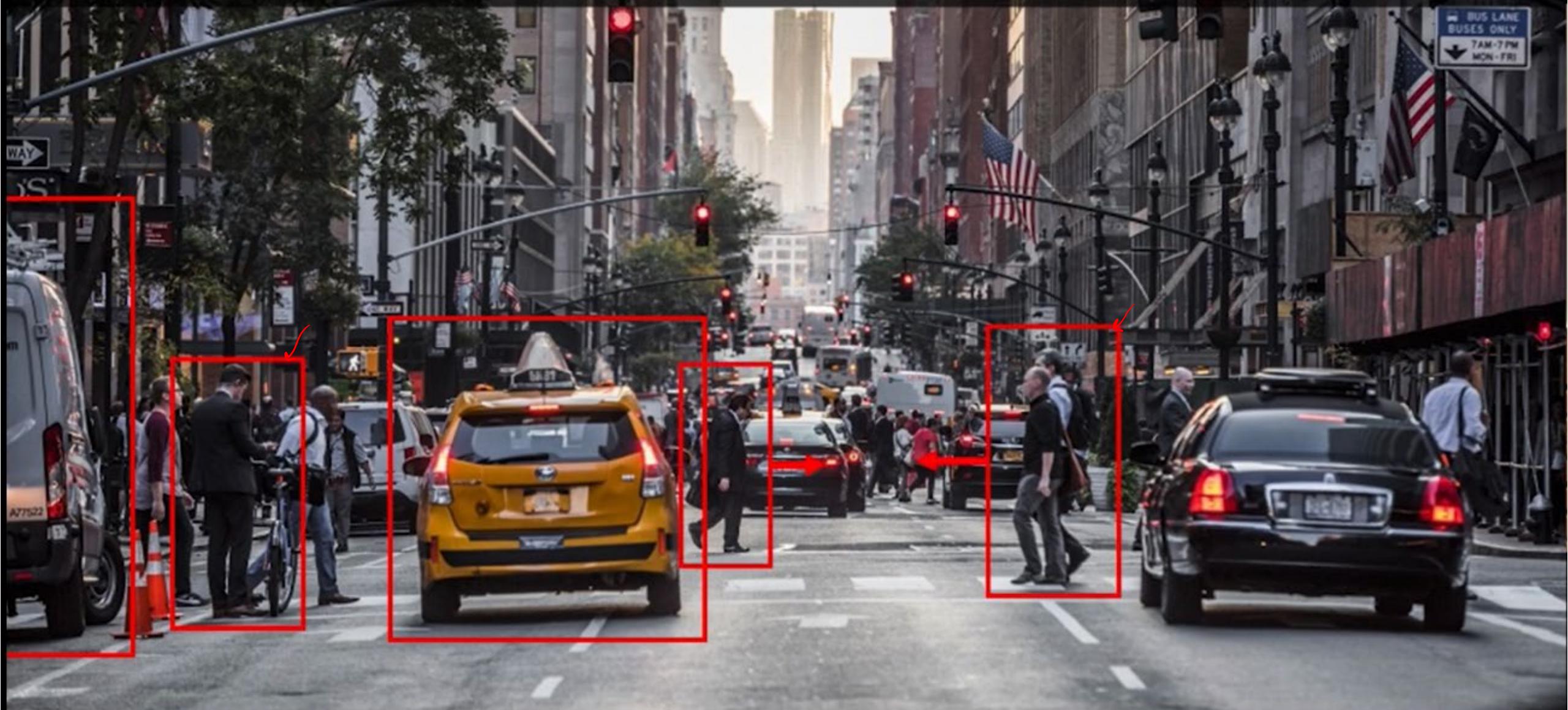
To discover from images what is present in the world, where things are, what actions are taking place, to predict and anticipate events in the world



To discover from images what is present in the world, where things are, what actions are taking place, to predict and anticipate events in the world



To discover from images what is present in the world, where things are, what actions are taking place, to predict and anticipate events in the world

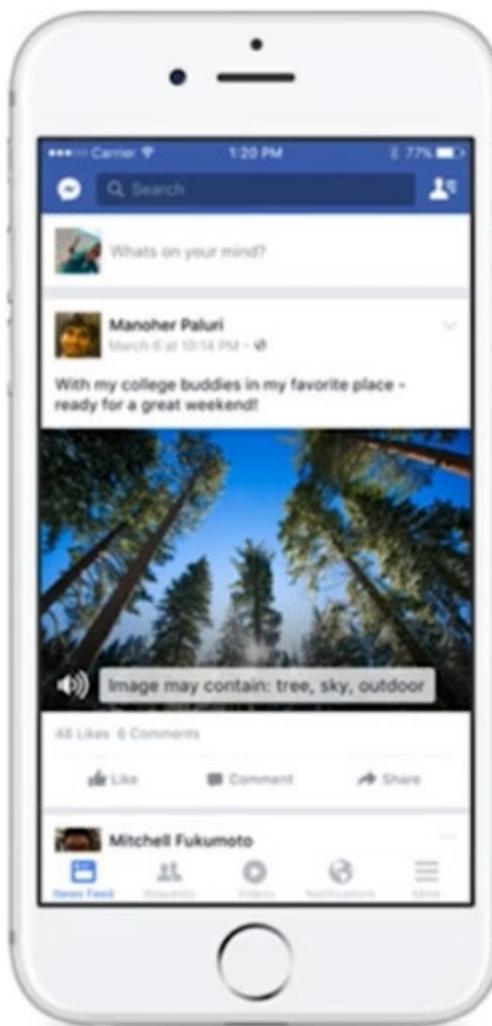


# The rise and impact of computer vision

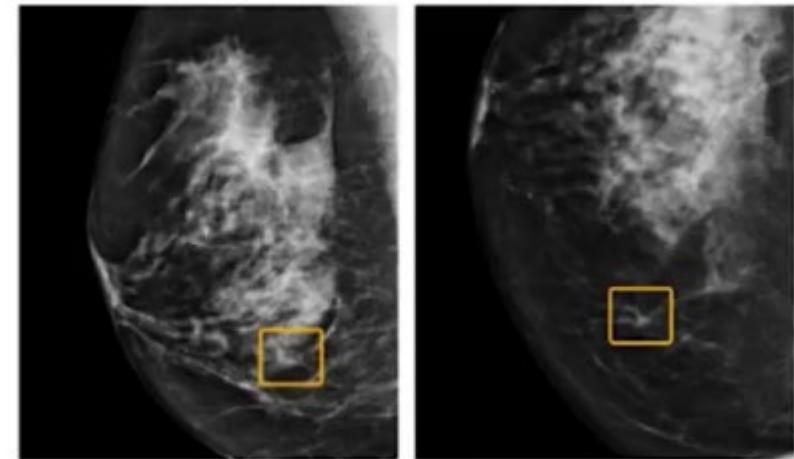
**Robotics**



**Accessibility**



**Biology & Medicine**



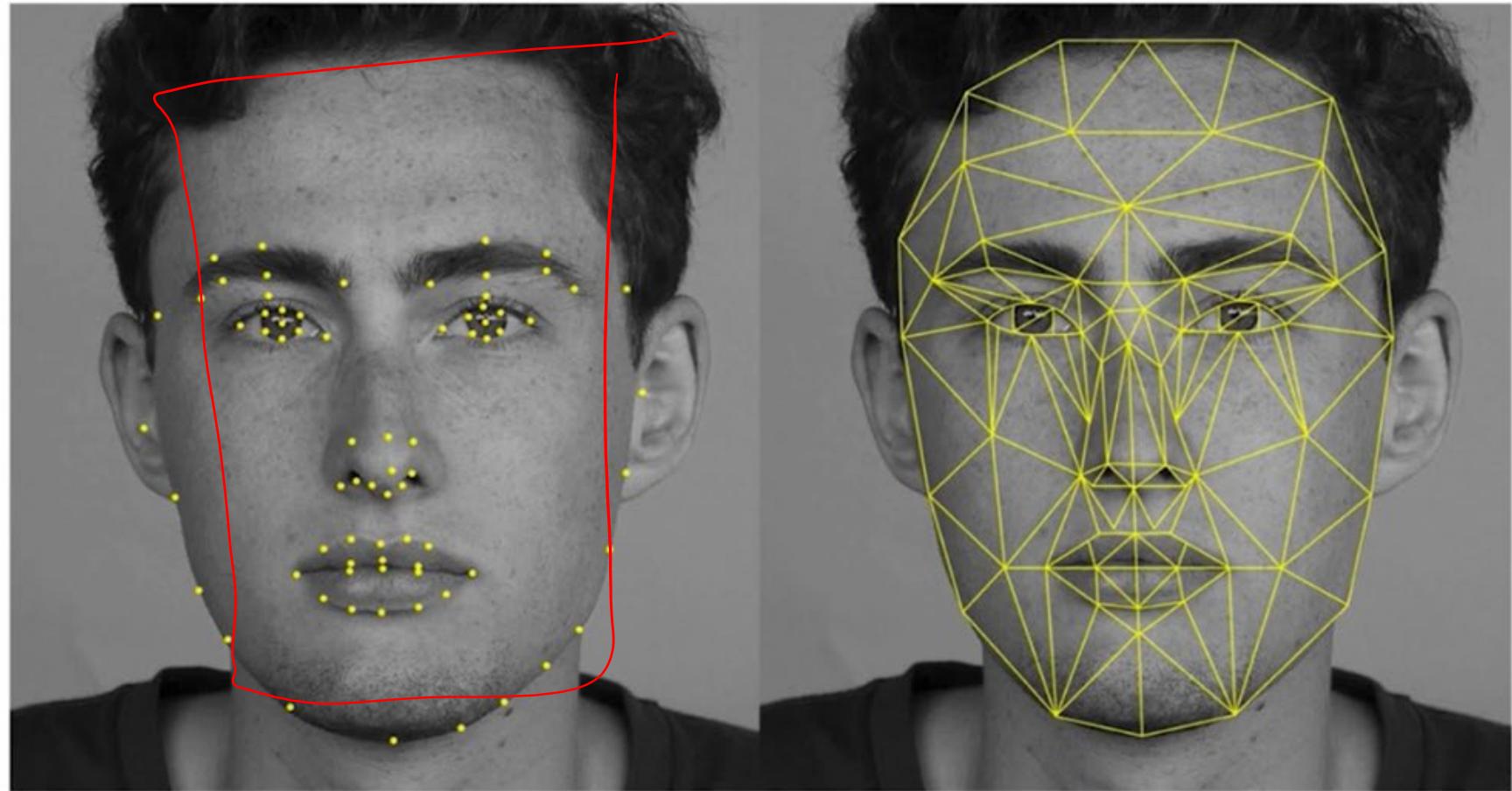
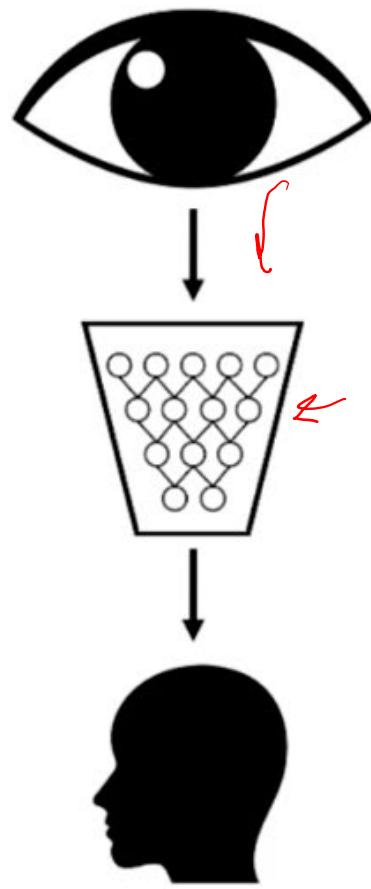
**Mobile computing**



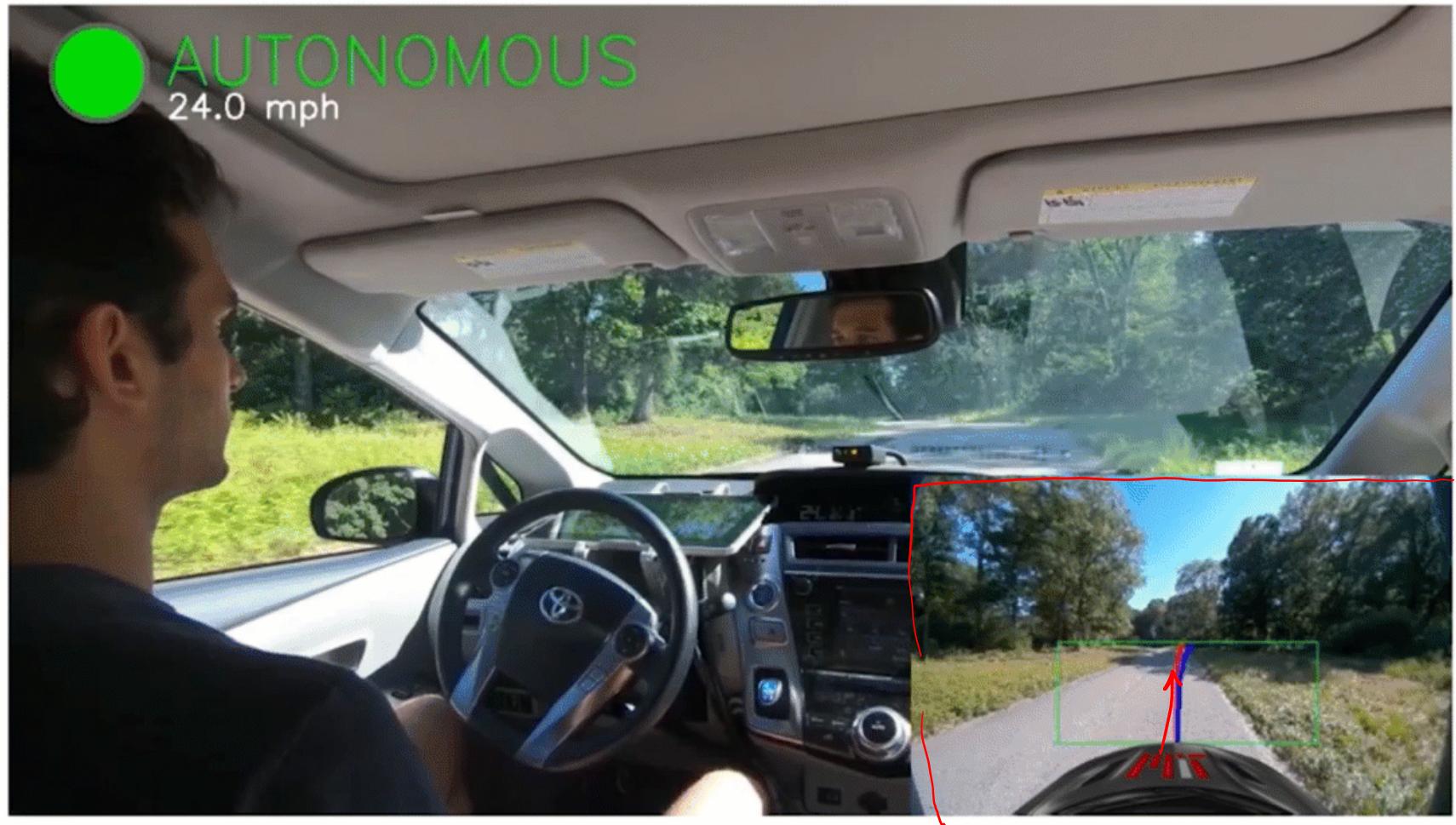
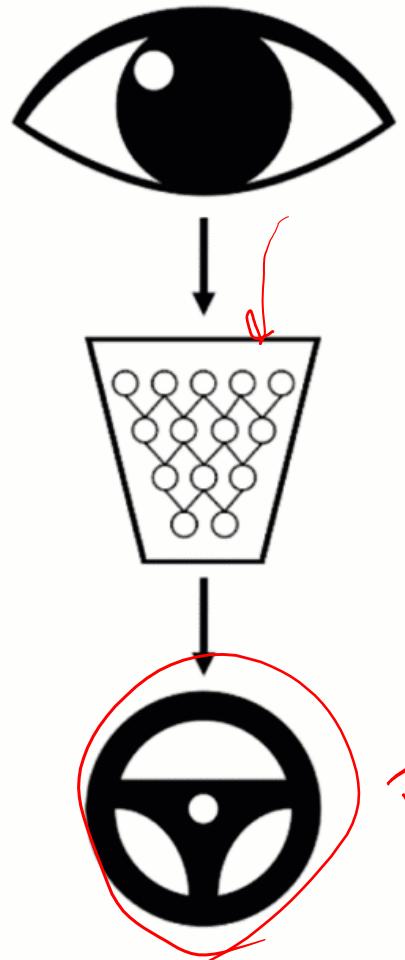
**Autonomous driving**



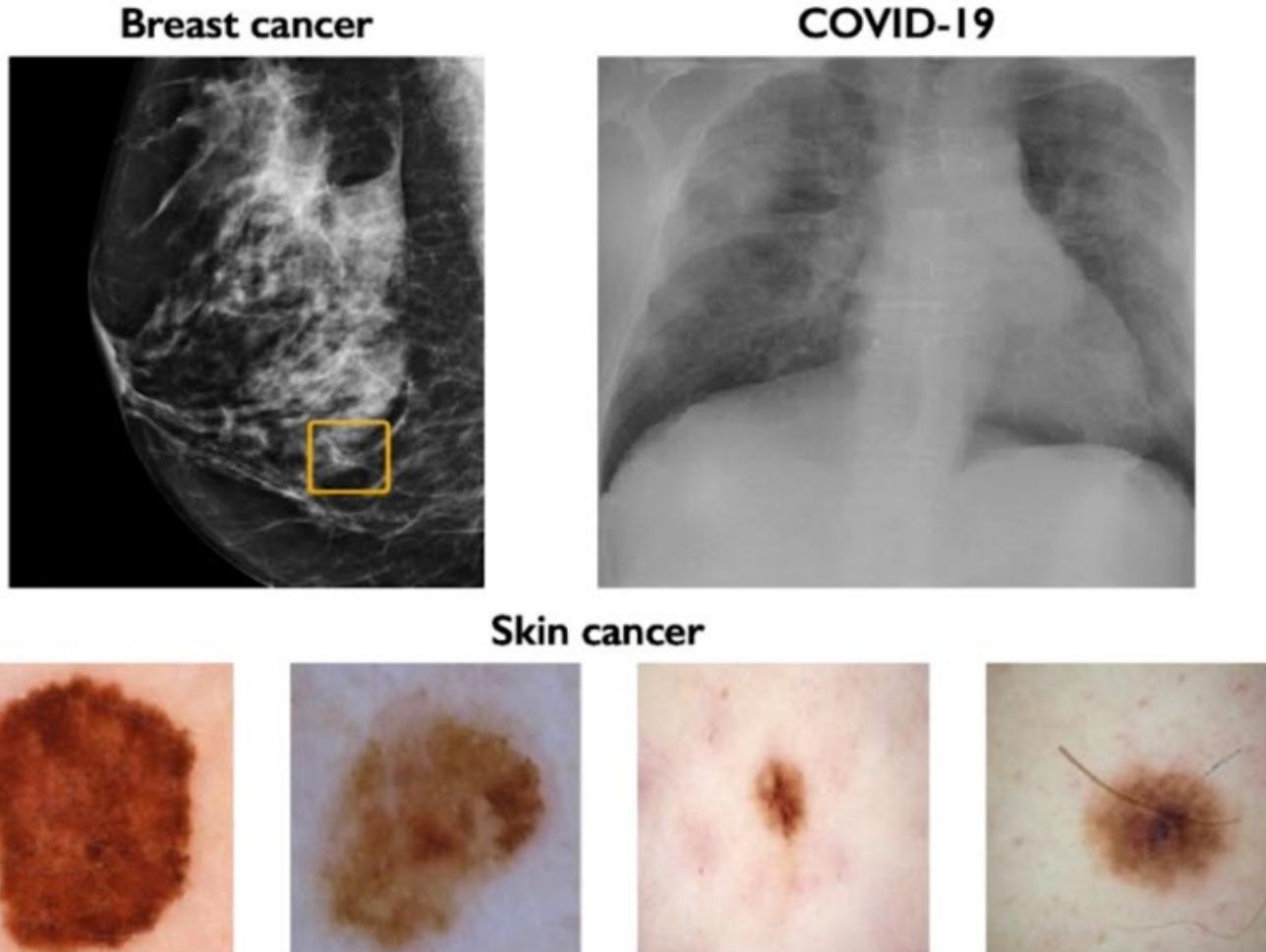
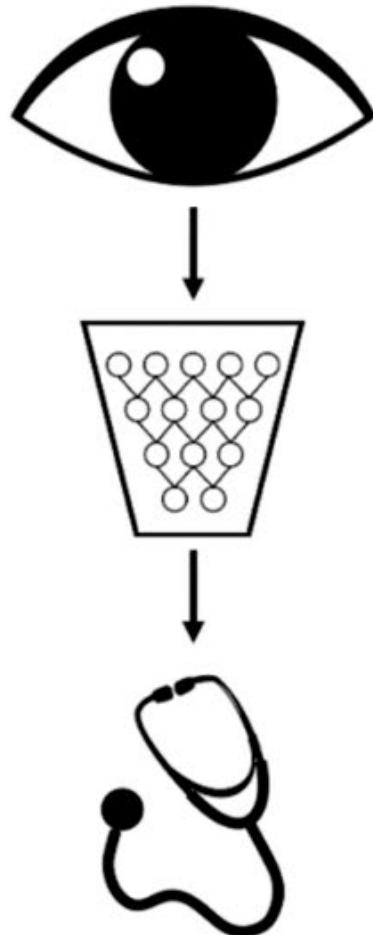
# Impact: Facial Detection and Recognition



# Impact: Self-Driving Cars



# Impact: Medicine, Biology, Healthcare



# Computer vision problems

Image Classification



Cat? (0/1)

64x64

Neural Style Transfer



Object detection



# Convolutional Neural Networks: What Computers “See”

# Images are numbers

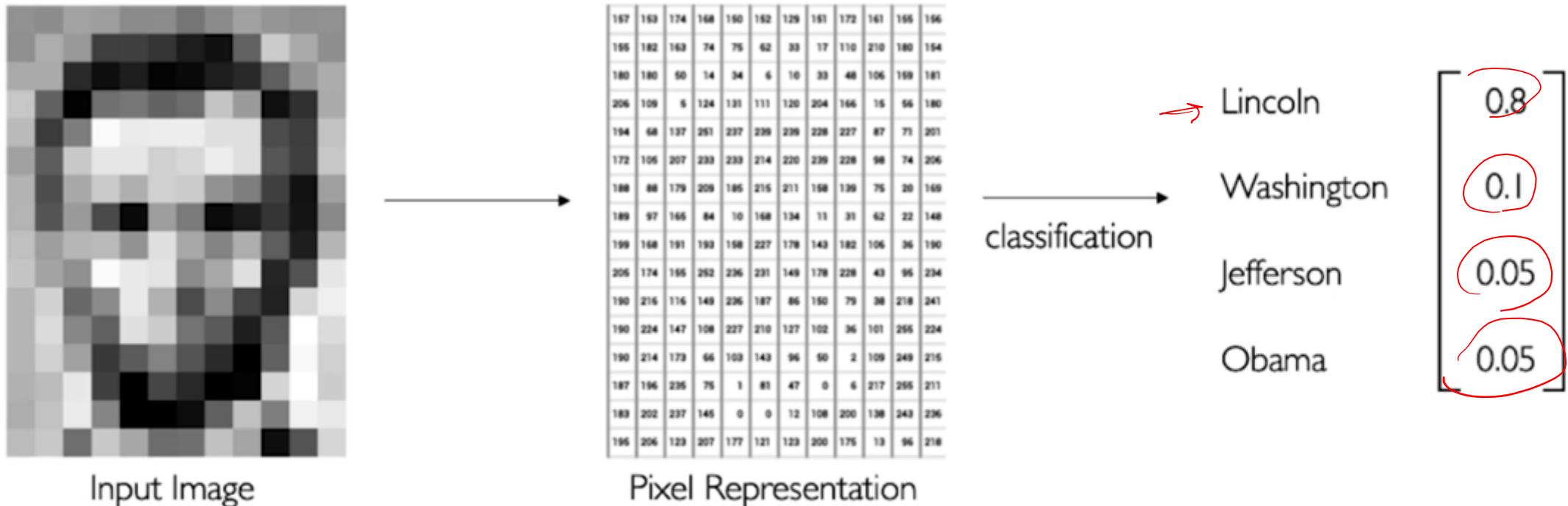


157	153	174	168	150	152	129	151	172	161	155	156
155	182	163	74	75	62	33	17	110	210	180	154
180	180	50	14	34	6	10	33	48	106	159	181
206	109	5	124	131	111	120	204	166	15	56	180
194	68	137	251	237	239	239	228	227	87	71	201
172	105	207	233	233	214	220	239	228	98	74	206
188	88	179	209	185	215	211	158	139	75	20	169
189	97	165	84	10	168	134	11	31	62	22	148
199	168	191	193	158	227	178	143	182	106	36	190
205	174	155	252	236	231	149	178	228	43	95	234
190	216	116	149	236	187	85	150	79	38	218	241
190	224	147	108	227	210	127	102	36	101	255	224
190	214	173	66	103	143	96	50	2	109	249	215
187	196	235	75	1	81	47	0	6	217	255	211
183	202	237	145	0	0	12	108	200	138	243	236
196	206	123	207	177	121	123	200	175	13	96	218

What the computer sees

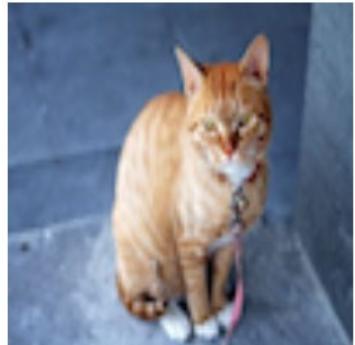
157	153	174	168	150	152	129	151	172	161	155	156
155	182	163	74	75	62	33	17	110	210	180	154
180	180	50	14	34	6	10	33	48	106	159	181
206	109	5	124	131	111	120	204	166	15	56	180
194	68	137	251	237	239	239	228	227	87	71	201
172	105	207	233	233	214	220	239	228	98	74	206
188	88	179	209	185	215	211	158	139	75	20	169
189	97	165	84	10	168	134	11	31	62	22	148
199	168	191	193	158	227	178	143	182	106	36	190
205	174	155	252	236	231	149	178	228	43	95	234
190	216	116	149	236	187	85	150	79	38	218	241
190	224	147	108	227	210	127	102	36	101	255	224
190	214	173	66	103	143	96	50	2	109	249	215
187	196	235	75	1	81	47	0	6	217	255	211
183	202	237	145	0	0	12	108	200	138	243	236
196	206	123	207	177	121	123	200	175	13	96	218

# Take in computer vision



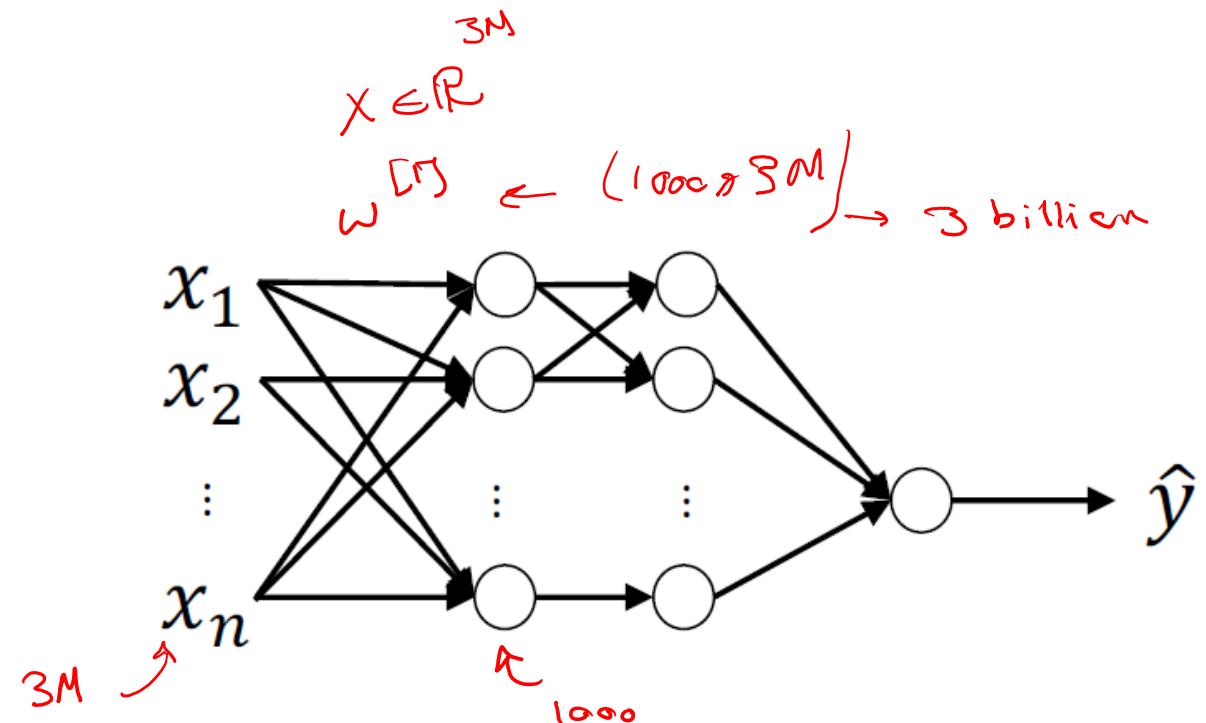
- **Regression:** output variable takes continuous value
- **Classification:** output variable takes class label. Can produce probability of belonging to a particular class

# Deep Learning on Large Images



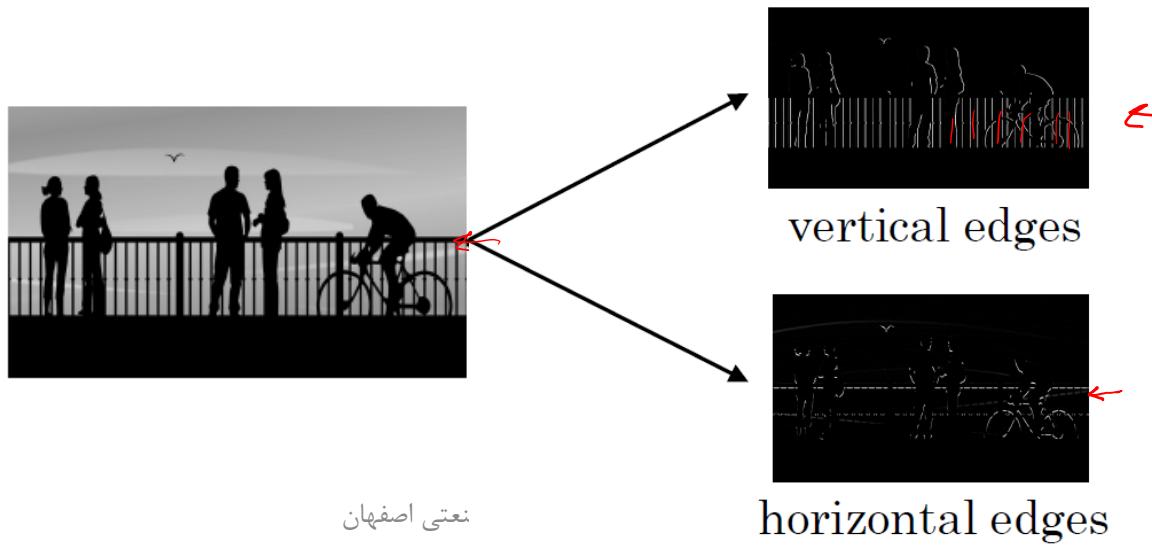
→ Cat? (0/1)

12288



# Convolutional Neural Networks: Edge detection example

# Edge detection example



# Vertical edge detection

$3 \times 1 + (x1 + 2 \times 1 + \cancel{0 \times 0} + \cancel{5 \times 0} + \cancel{7 \times 0} + 1 \times (-1) + 8 \times (-1) + 2 \times (-1) = -5)$

The diagram shows a 6x6 input image matrix and a 3x3 filter kernel. The input matrix has values ranging from 0 to 9. The filter kernel is labeled as a "3x3 filter kernel". A red arrow points to the calculation of the output value at position (3,3), which is highlighted in yellow. The formula for this calculation is shown above:  $3 \times 1 + (x1 + 2 \times 1 + \cancel{0 \times 0} + \cancel{5 \times 0} + \cancel{7 \times 0} + 1 \times (-1) + 8 \times (-1) + 2 \times (-1) = -5)$ . The result of the convolution is a 4x4 output matrix.

3	0	1	2	7	4
1	5	8	9	3	1
2	7	2	5	1	3
0	1	3	1	7	8
4	2	1	6	2	8
2	4	5	2	3	9

6x6

\*      "Convolution"

3x3  
filter  
kernel

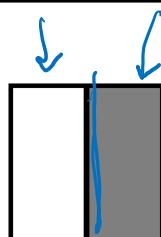
=

-5	-4	0	8
-10	-2	2	3
0	-2	-4	-7
-3	-2	-3	-6

4x4

# Vertical edge detection

10	10	10	0	0	0
10	10	10	0	0	0
10	10	10	0	0	0
10	10	10	0	0	0
10	10	10	0	0	0
10	10	10	0	0	0



$6 \times 6$

\*

1	0	-1
1	0	-1
1	0	-1

$3 \times 3$

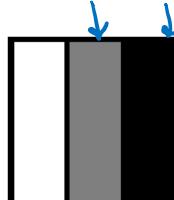
=

0	30	30	0
0	30	30	0
0	30	30	0
0	30	30	0

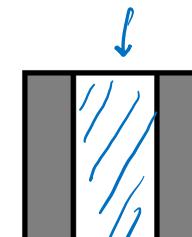
$4 \times 4$

نعمي اصفهانی

\*



سمانه حسينی



20

# Convolutional Neural Networks: More edge detection

# Vertical edge detection examples

10	10	10	0	0	0
10	10	10	0	0	0
10	10	10	0	0	0
10	10	10	0	0	0
10	10	10	0	0	0
10	10	10	0	0	0

↑ سیم ← سیم

\*

1	0	-1
1	0	-1
1	0	-1

↓ II

=

0	30	30	0
0	30	30	0
0	30	30	0
0	30	30	0

↓ II

سیم → سیم



0	0	0	10	10	10
0	0	0	10	10	10
0	0	0	10	10	10
0	0	0	10	10	10
0	0	0	10	10	10
0	0	0	10	10	10

↑ سیم ← سیم

6x6

\*

1	0	-1
1	0	-1
1	0	-1

↓ II

=

0	-30	-30	0
0	-30	-30	0
0	-30	-30	0
0	-30	-30	0

↓ II

سیم → سیم



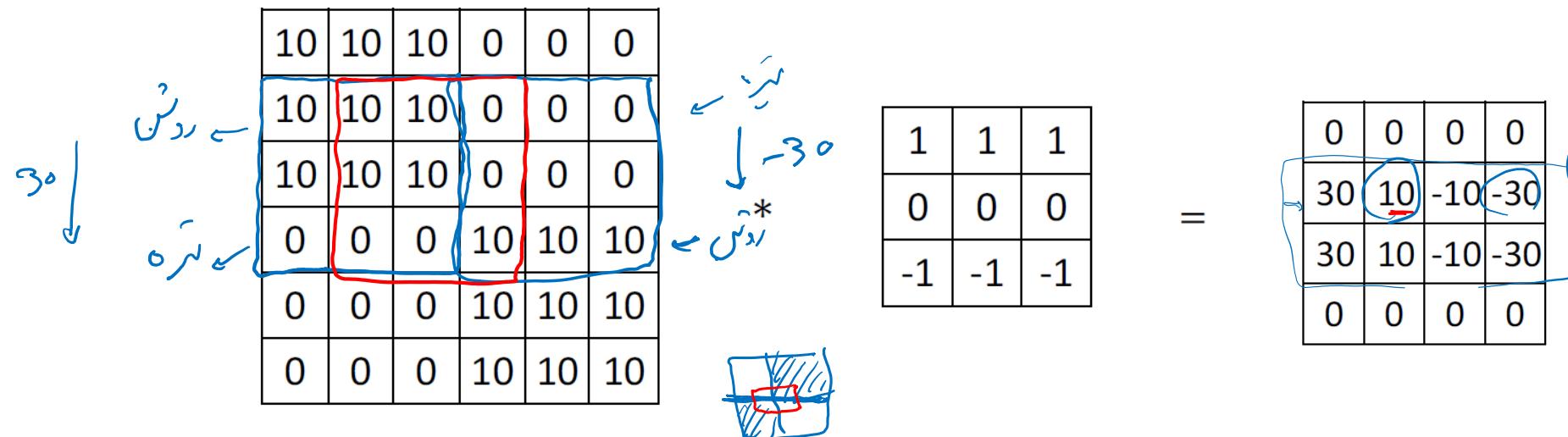
# Vertical and Horizontal Edge Detection

1	0	-1
1	0	-1
1	0	-1

Vertical

1	1	1
0	0	0
-1	-1	-1

Horizontal



# High level feature detection

Let's identify key features in each image category



Nose,  
Eyes,  
Mouth

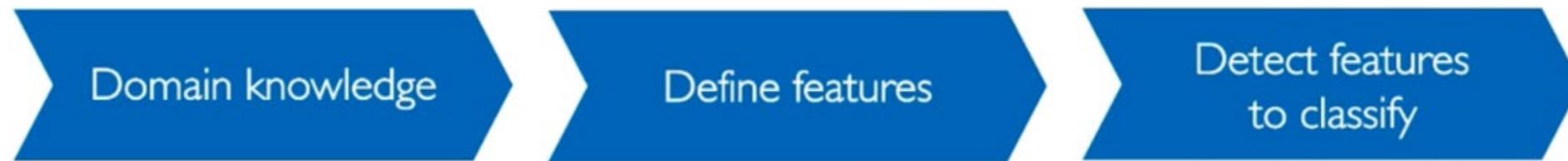


Wheels,  
License Plate,  
Headlights



Door,  
Windows,  
Steps

# Manual feature extraction



Problems?

# Manual feature extraction



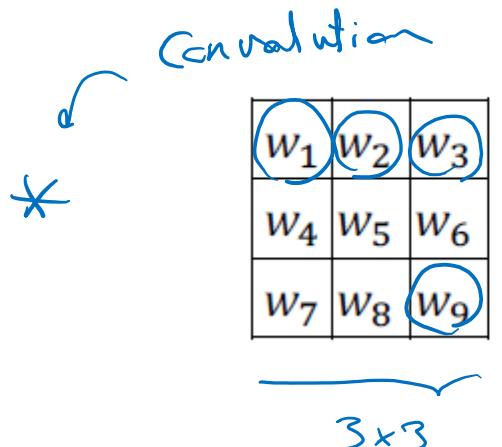
# Learning to detect edges

1	0	-1
1	0	-1
1	0	-1

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

Sobel filter

3	0	1	2	7	4
1	5	8	9	3	1
2	7	2	5	1	3
0	1	3	1	7	8
4	2	1	6	2	8
2	4	5	2	3	9



3	5	-3
10	0	-10
3	0	-3

Scharr filter

# Core Foundation Review

- Convolutional Neural Networks
  - Computer Vision
  - What Computers “See”
  - Edge Detection Example
  - More Edge Detection