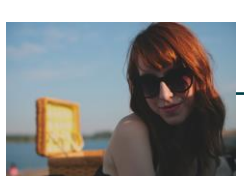
Introduction to Computer Vision with Convolutional Neural Networks

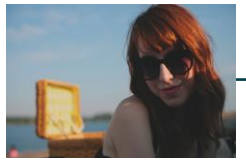
Mohammad Haghighat
Senior AI Product Manager
Intel Corp.

Outline

- High level introduction to AI
  - Conventional vs. deep learning
- Neural networks and deep learning
  - Fully connected networks
  - Elements of a neural network
  - Neural network training
- Convolutional neural networks (CNNs)
  - Building blocks of CNNs
  - Applications of CNNs
  - Popular CNN architectures

High-level introduction to AI


ML Model → person


ML Model → 

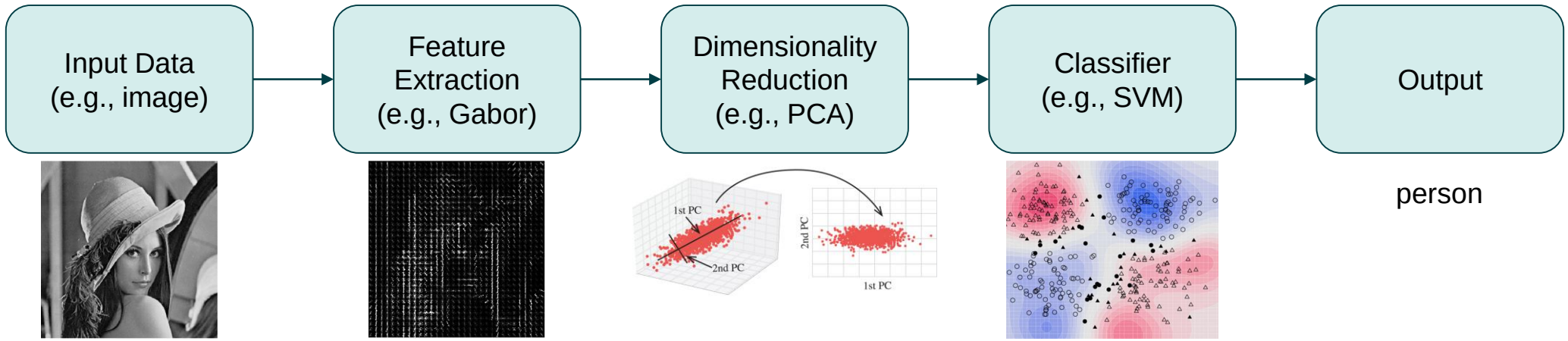

ML Model → person dancing on the beach

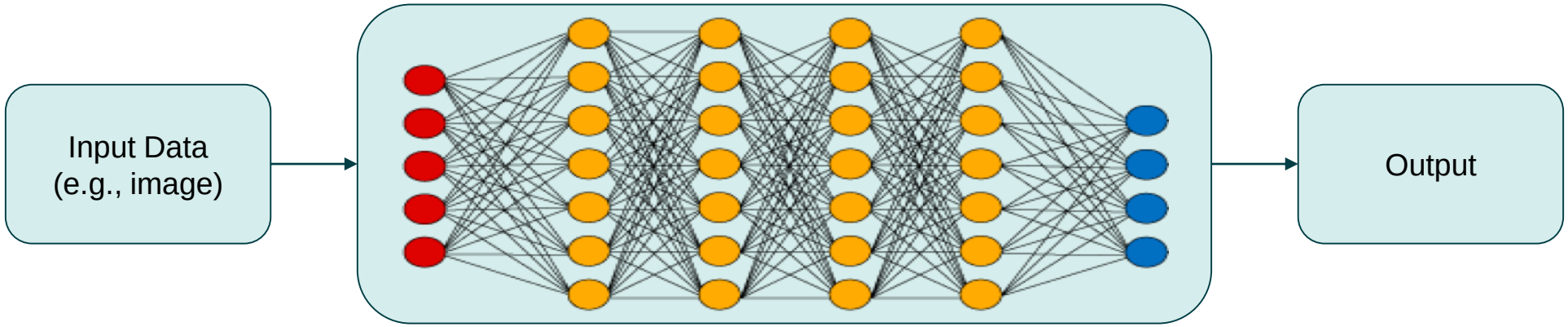
"Nothing to love about this presentation." → ML Model → negative feedback

begining → ML Model → beginning

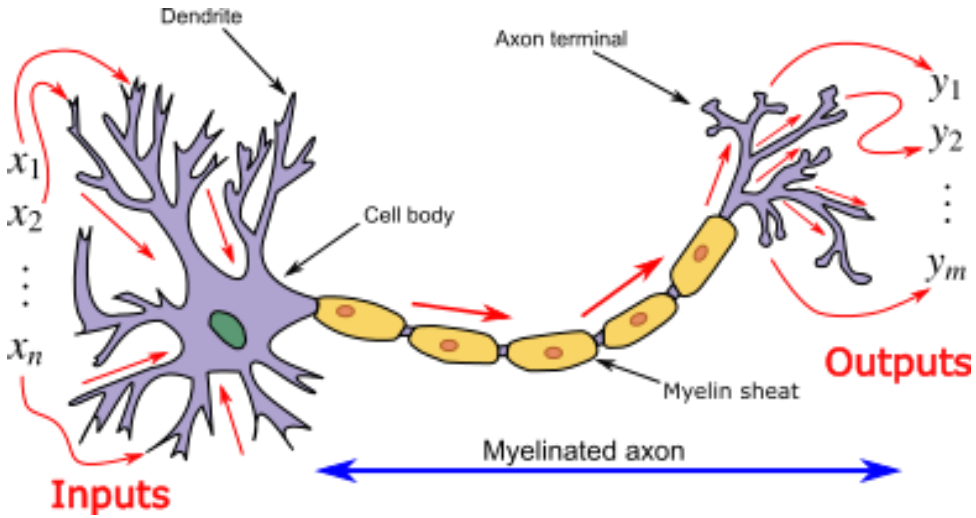
ML Model → "Let's go for lunch"

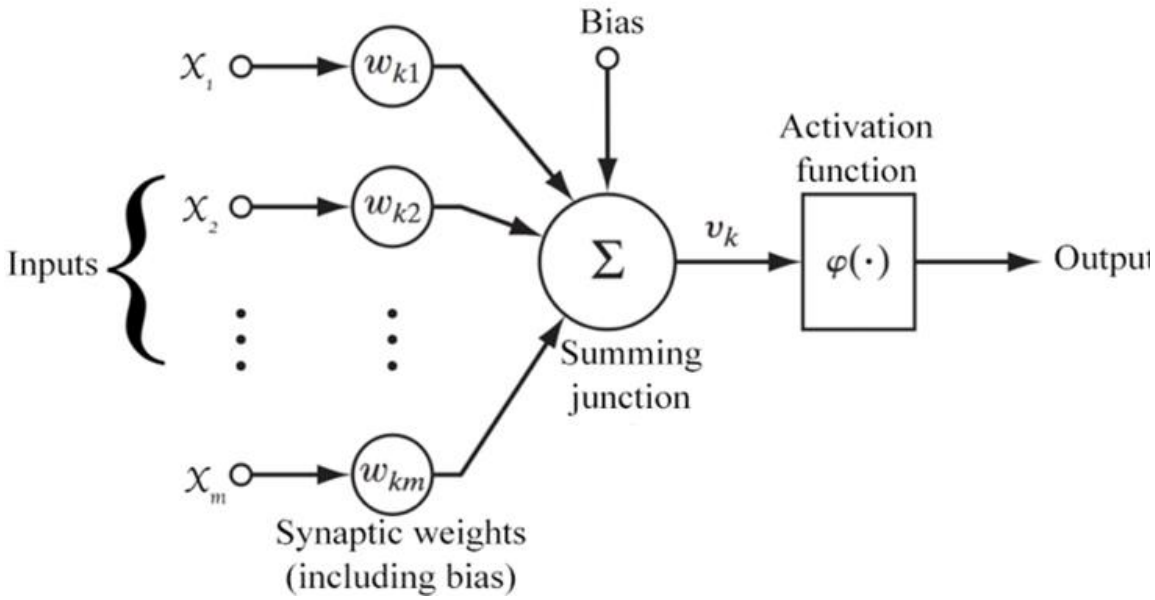
Classical learning vs deep learning

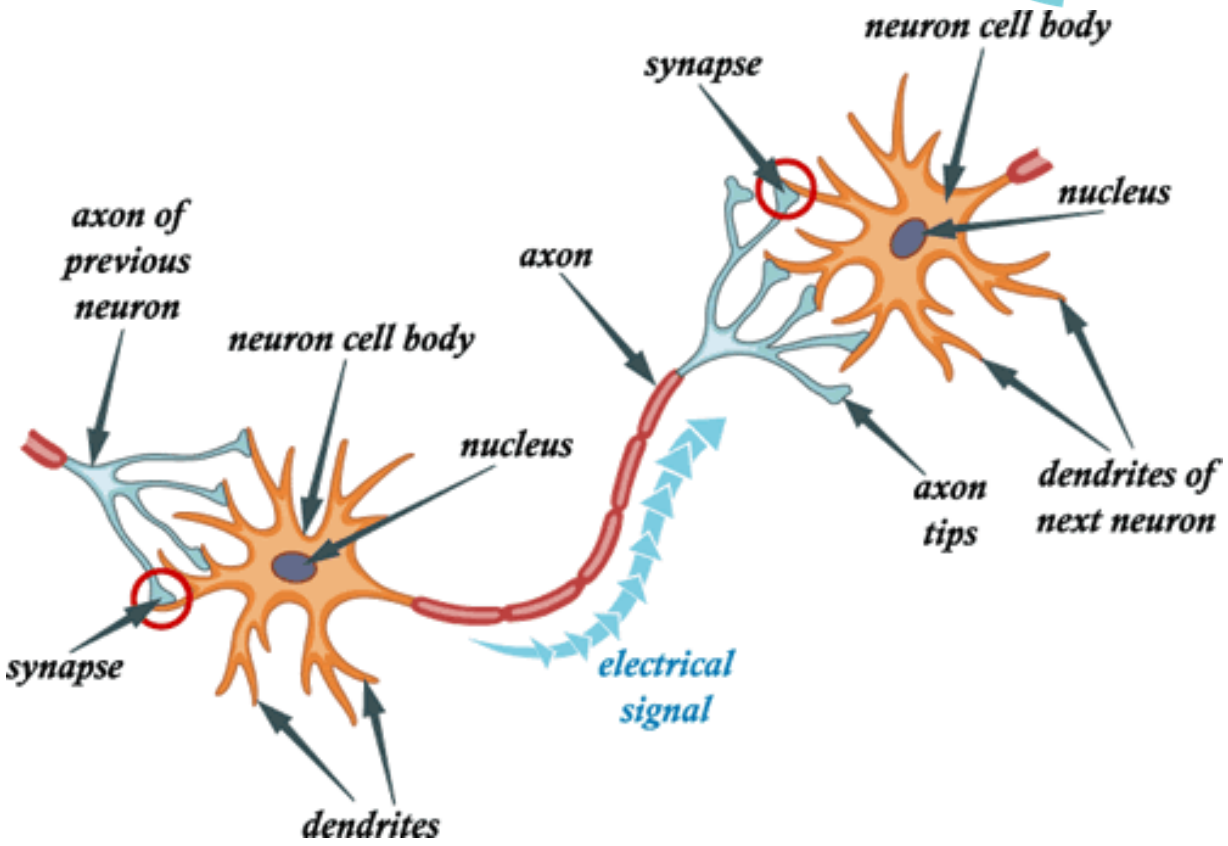




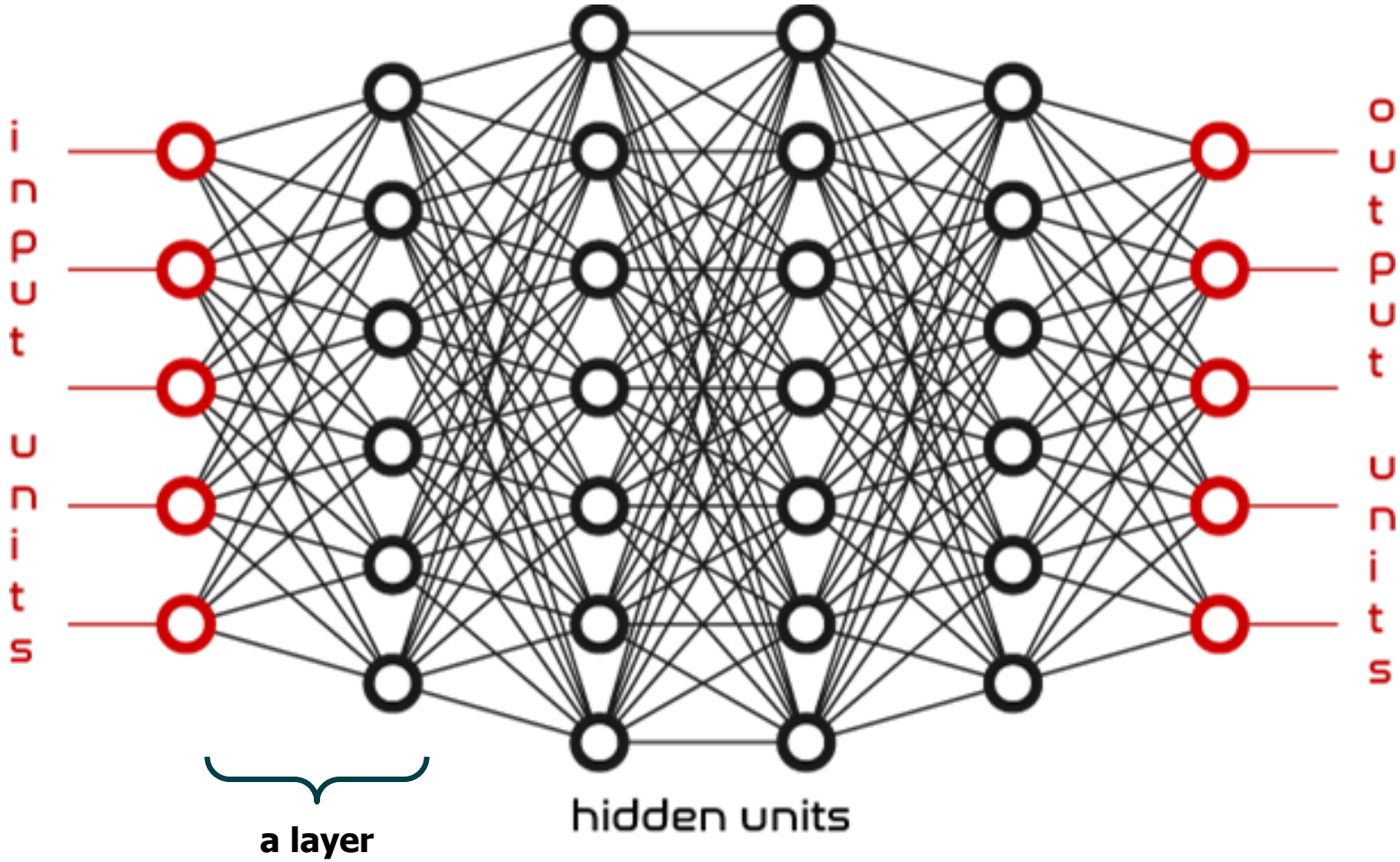
What are neurons?







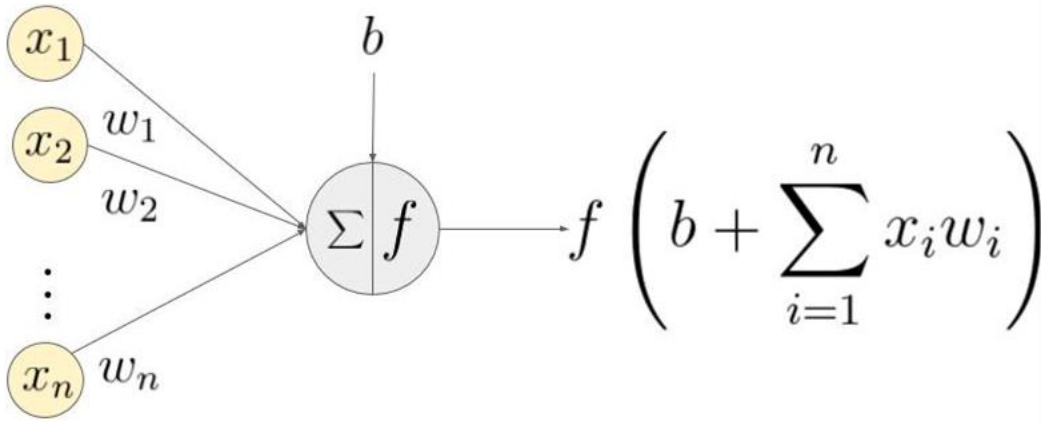
… and what are neural networks?

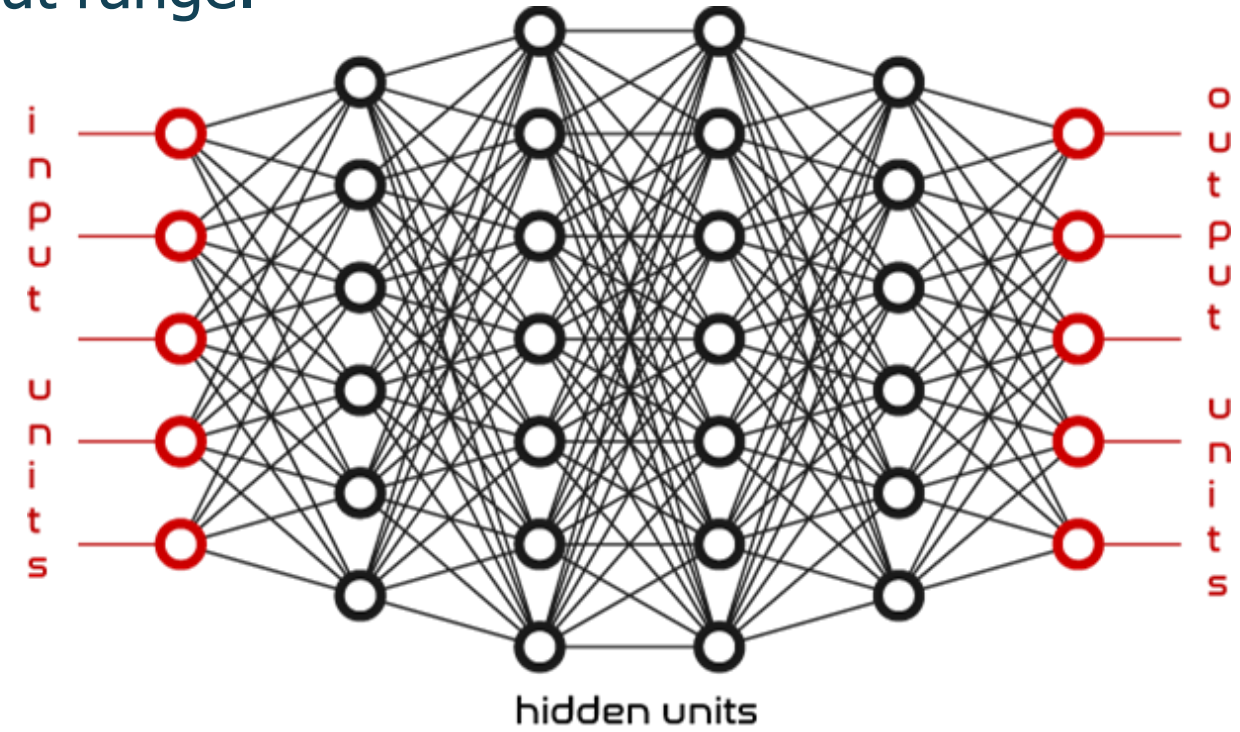


Neural networks as a vehicle for deep learning

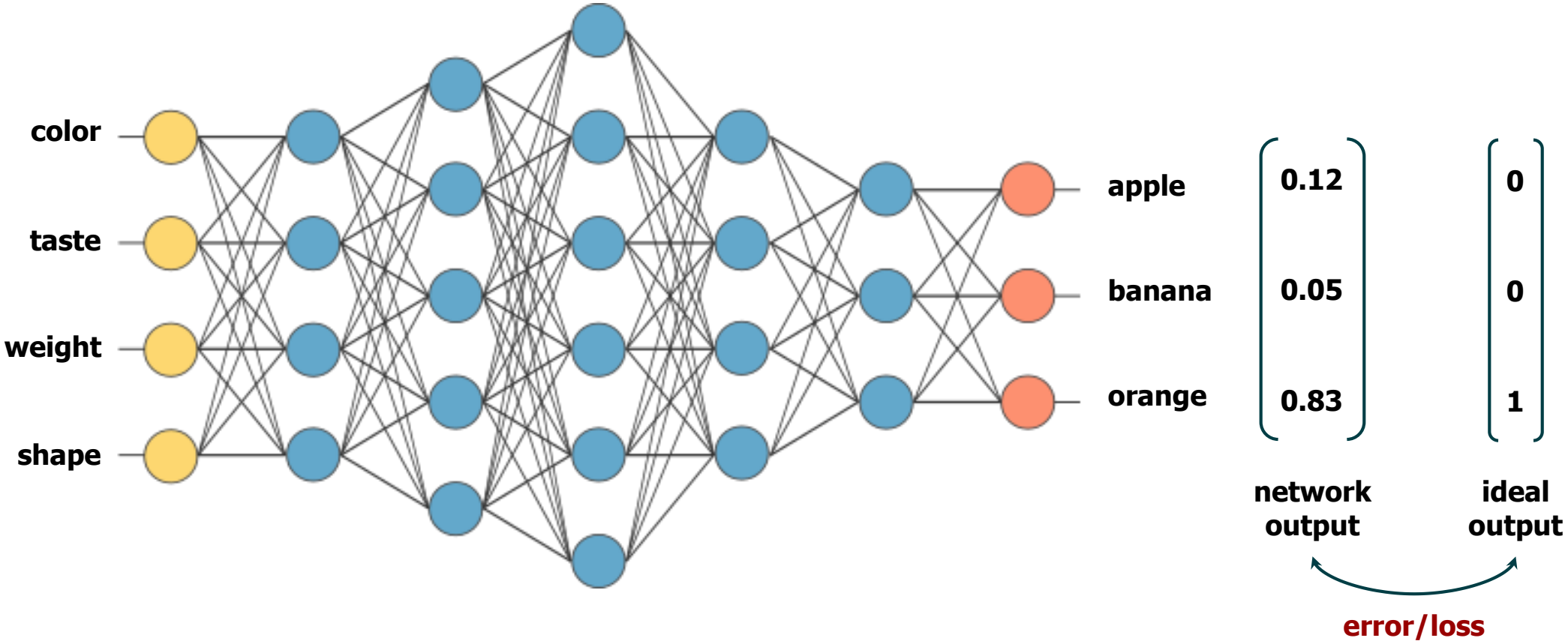
Universal Approximation Theorem

A one-hidden-layer neural network with enough neurons can approximate ***any*** continuous function within the given input range.



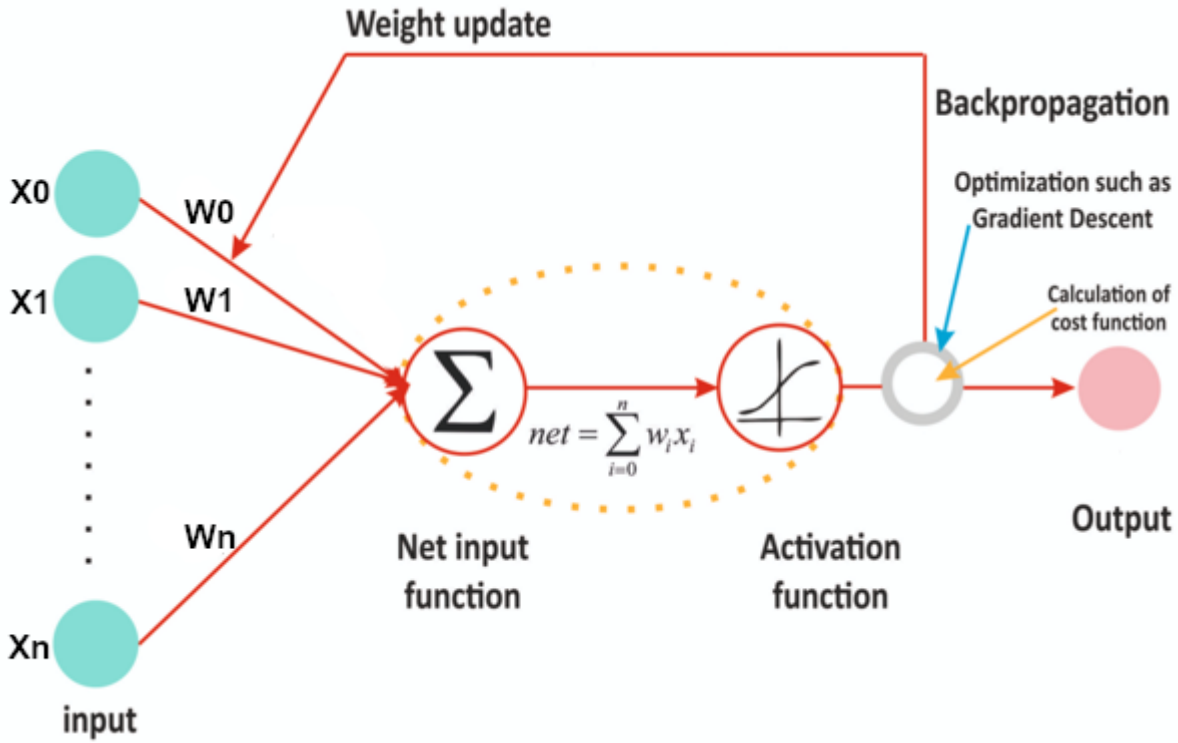


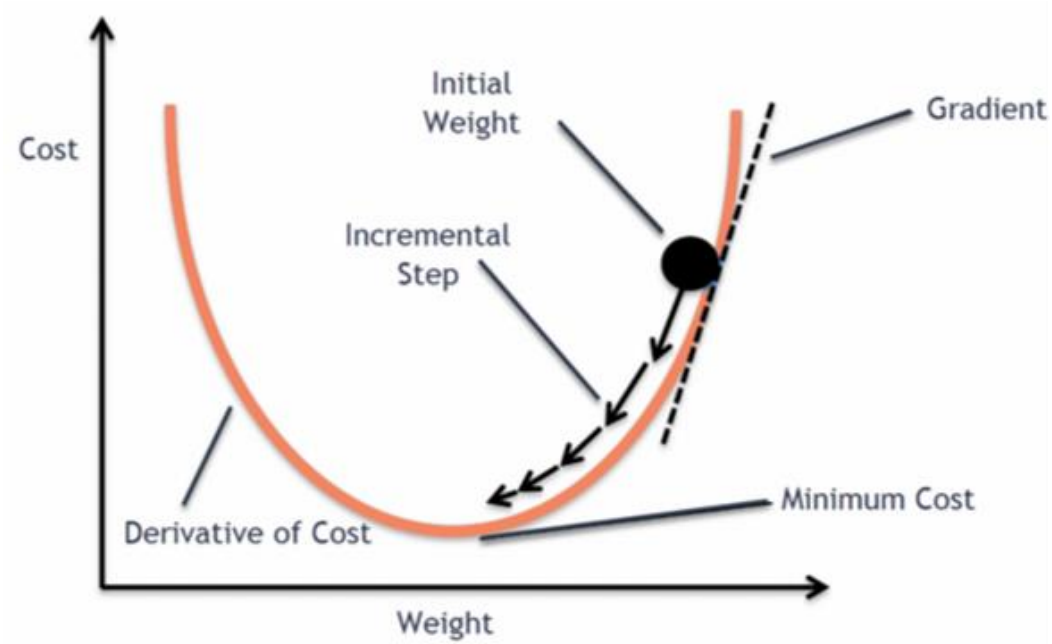
Neural network-based classifier



Neural network training

Loss and gradient descent algorithm

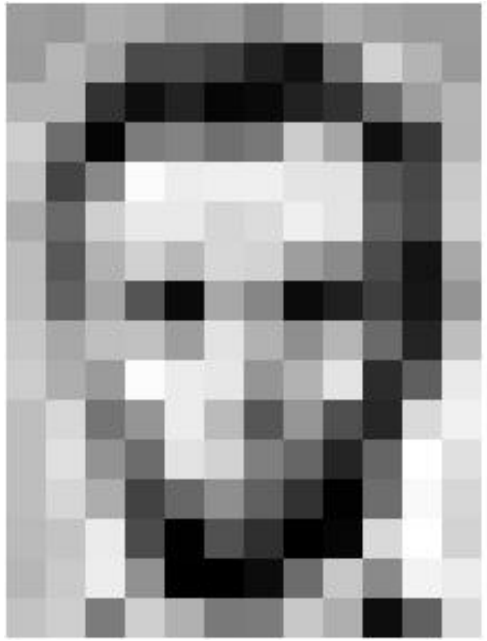


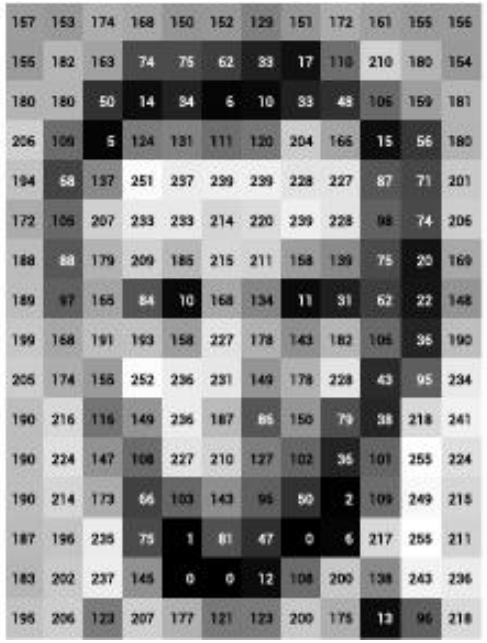


Reference

How does a computer see an image?

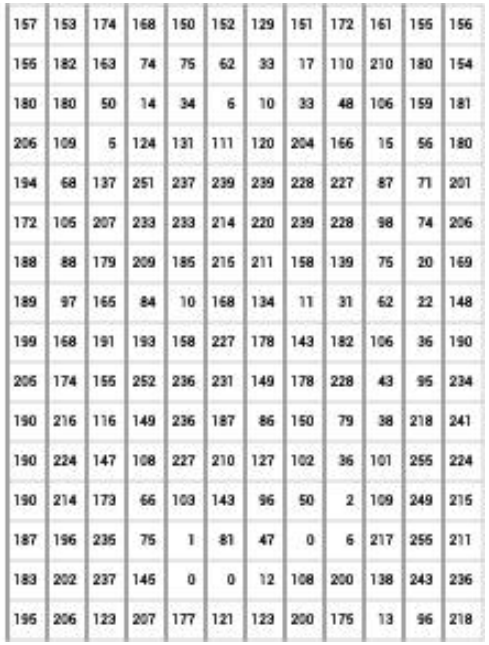
- Gray-level pixel: 8 bit ranging between 0 and 255
- Color pixel: Three 8-bit channels for Red, Green & Blue (RGB)





157	153	174	168	150	152	129
155	182	163	74	75	62	33
180	180	50	14	34	6	10
206	109	5	124	131	111	120
194	68	137	251	237	239	239
172	105	207	233	233	214	220
188	88	179	209	185	215	211
189	97	165	84	10	168	134
199	168	191	193	158	227	178
205	174	155	252	236	231	149
190	216	116	149	236	187	86
190	224	147	108	227	210	127
190	214	173	66	103	143	96
187	196	235	75	1	81	47
183	202	237	145	0	0	12
195	206	123	207	177	121	123

151	172	161	155	156		
17	110	210	180	154		
33	48	106	159	181		
204	166	15	56	180		
228	227	87	71	201		
239	228	98	74	206		
158	139	75	20	169		
11	31	62	22	148		
143	182	106	36	190		
178	228	43	95	234		
150	79	38	218	241		
102	36	101	255	224		
50	2	109	249	215		
0	6	217	255	211		
12	108	200	138	243	236	
200	175	13	96	218		

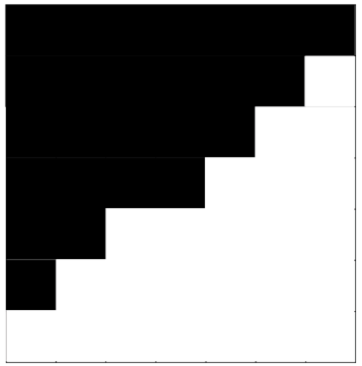


157	153	174	168	150	152	129
155	182	163	74	75	62	33
180	180	50	14	34	6	10
206	109	5	124	131	111	120
194	68	137	251	237	239	239
172	105	207	233	233	214	220
188	88	179	209	185	215	211
189	97	165	84	10	168	134
199	168	191	193	158	227	178
205	174	155	252	236	231	149
190	216	116	149	236	187	86
190	224	147	108	227	210	127
190	214	173	66	103	143	96
187	196	235	75	1	81	47
183	202	237	145	0	0	12
195	206	123	207	177	121	123

151	172	161	155	156		
17	110	210	180	154		
33	48	106	159	181		
204	166	15	56	180		
228	227	87	71	201		
239	228	98	74	206		
158	139	75	20	169		
11	31	62	22	148		
143	182	106	36	190		
178	228	43	95	234		
150	79	38	218	241		
102	36	101	255	224		
50	2	109	249	215		
0	6	217	255	211		
12	108	200	138	243	236	
200	175	13	96	218		

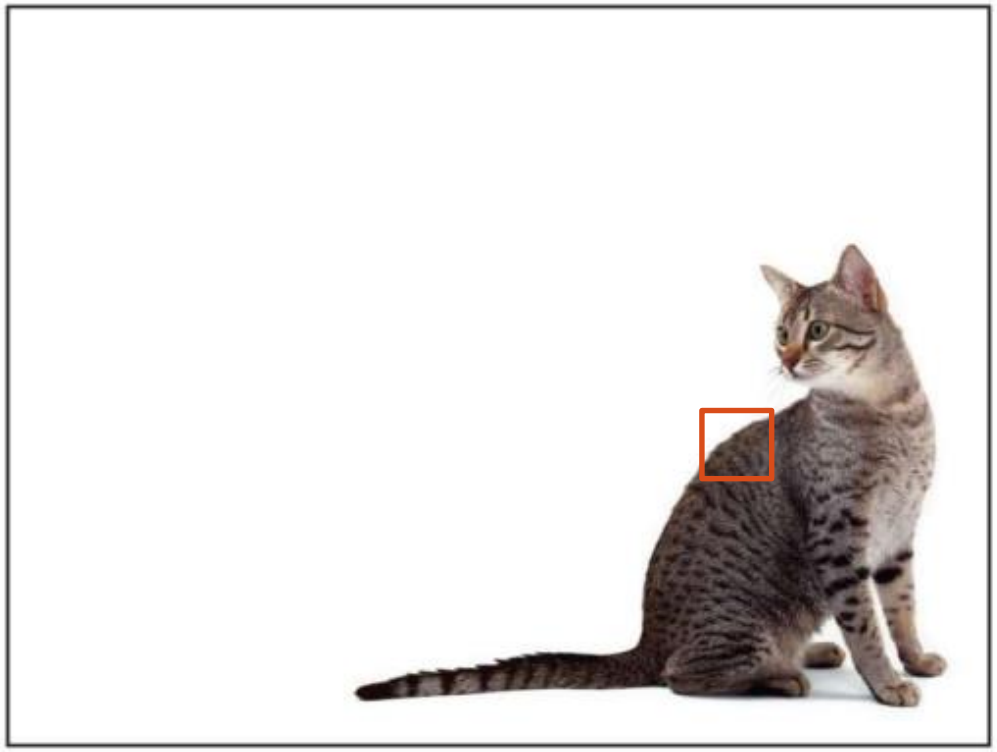
0	0	0	0	0	0	0
0	0	0	0	0	0	255
0	0	0	0	0	255	255
0	0	0	0	255	255	255
0	0	255	255	255	255	255
0	255	255	255	255	255	255
255	255	255	255	255	255	255

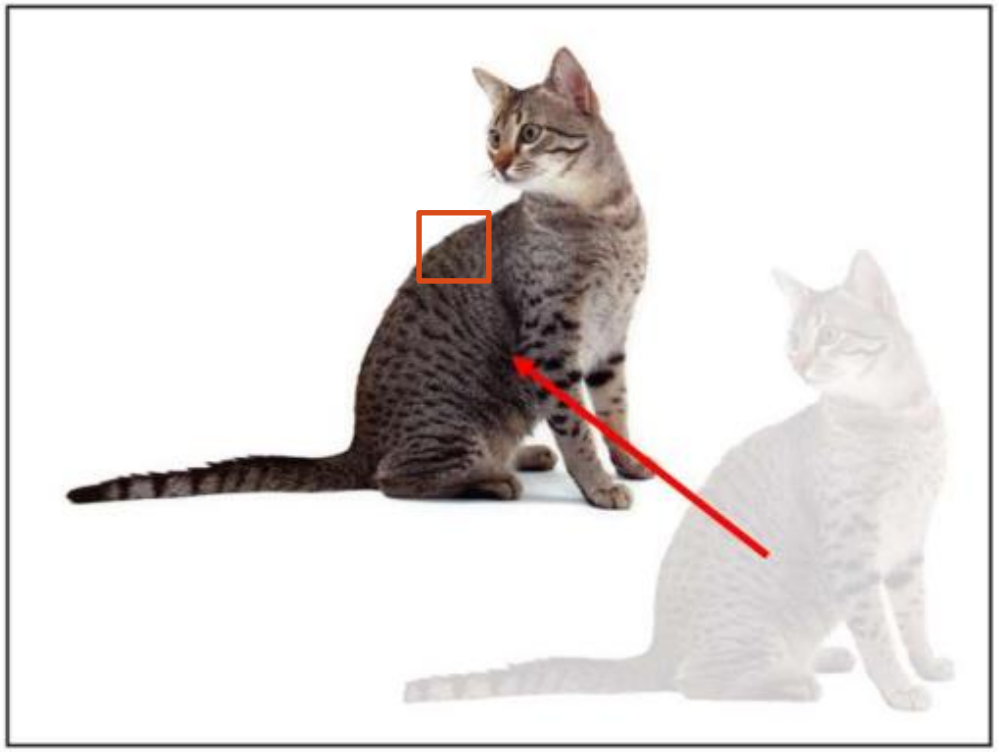
how computer sees the edge



how we sees the edge

What is different about the image data?





Introduction to CNNs

Convolutional layer

0	0	0	0	0	0	…
0	156	155	156	158	158	…
0	153	154	157	159	159	…
0	149	151	155	158	159	…
0	146	146	149	153	158	…
0	145	143	143	148	158	…
…	…	…	…	…	…	…

Input Channel #1 (Red)

0	0	0	0	0	0	…
0	167	166	167	169	169	…
0	164	165	168	170	170	…
0	160	162	166	169	170	…
0	156	156	159	163	168	…
0	155	153	153	158	168	…
…	…	…	…	…	…	…

Input Channel #2 (Green)

0	0	0	0	0	0	…
0	163	162	163	165	165	…
0	160	161	164	166	166	…
0	156	158	162	165	166	…
0	155	155	158	162	167	…
0	154	152	152	157	167	…
…	…	…	…	…	…	…

Input Channel #3 (Blue)

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

Kernel Channel #1

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

Kernel Channel #2

0	1	1
0	1	0
1	-1	1

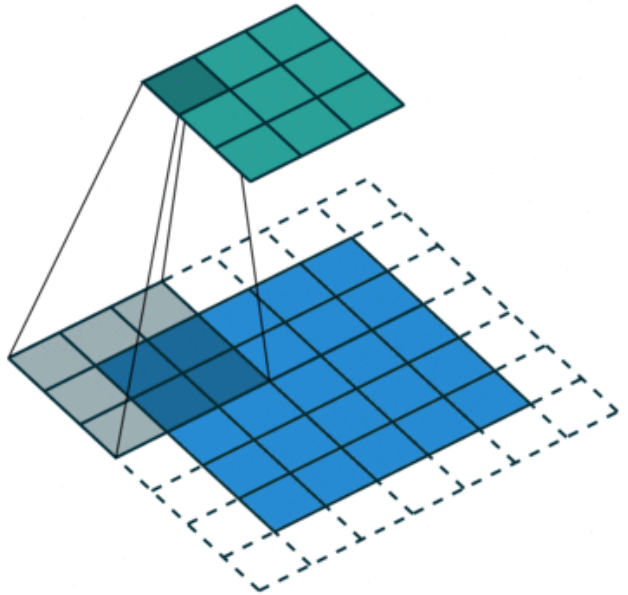
Kernel Channel #3

$308 + -498 + 164 + 1 = -25$

Bias = 1

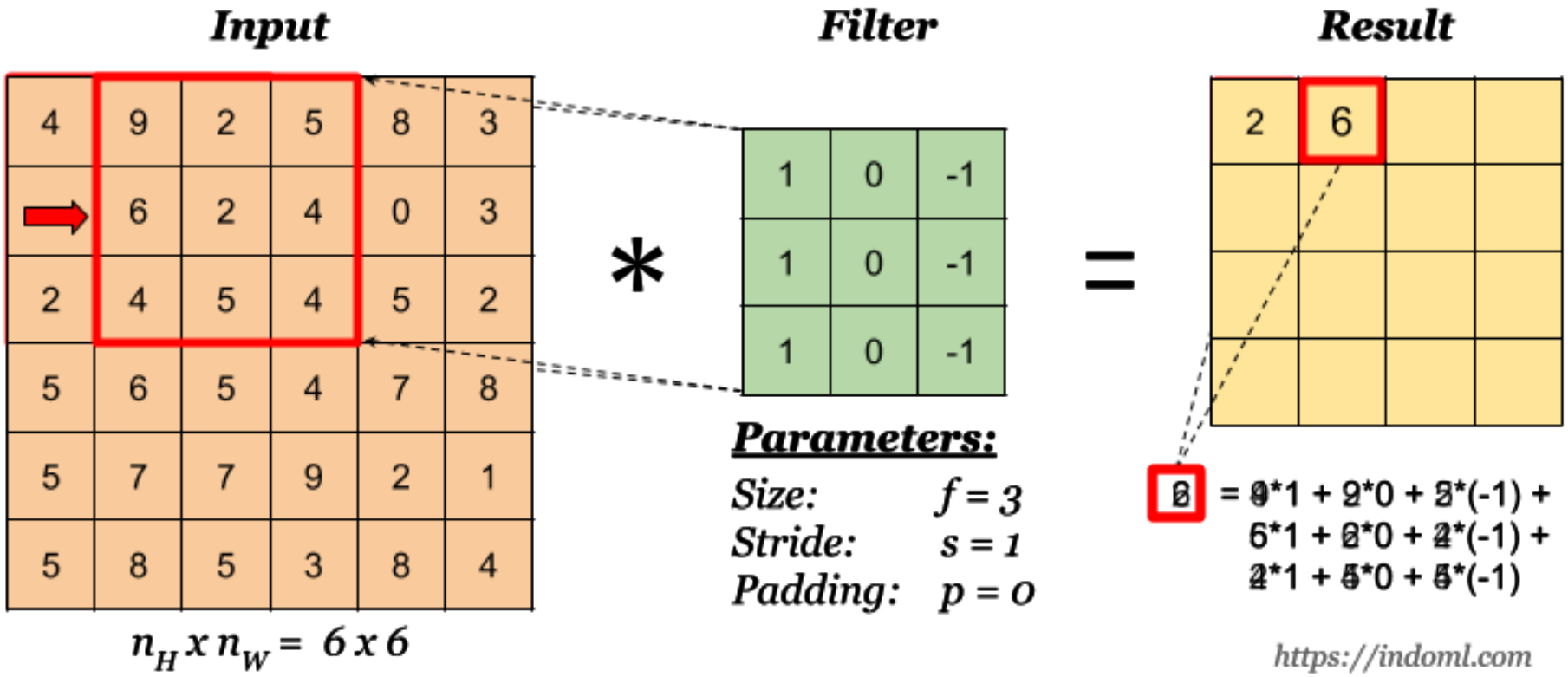
Output

-25				…
				…
				…
				…
…	…	…	…	…



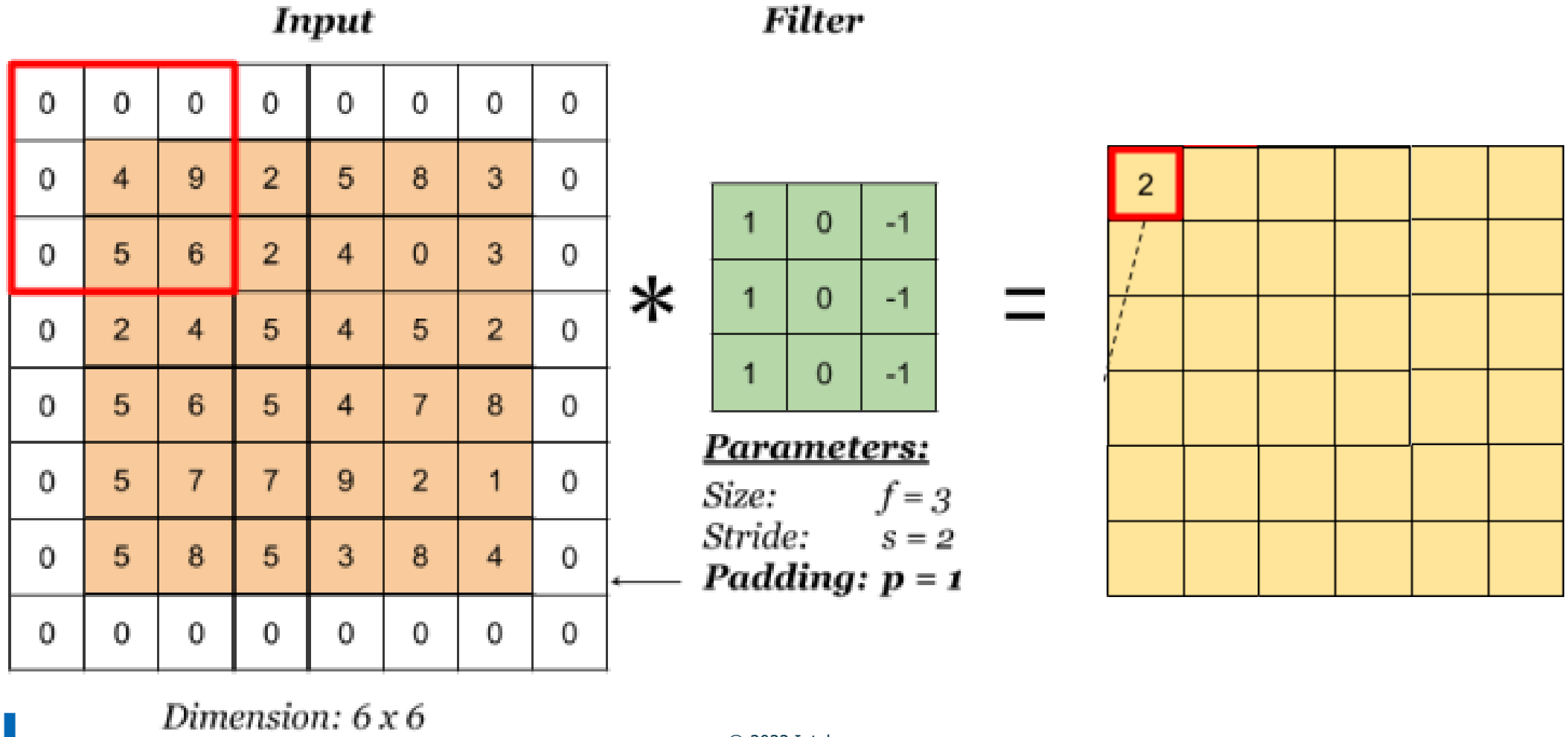
Building blocks of CNNs

Convolution operation



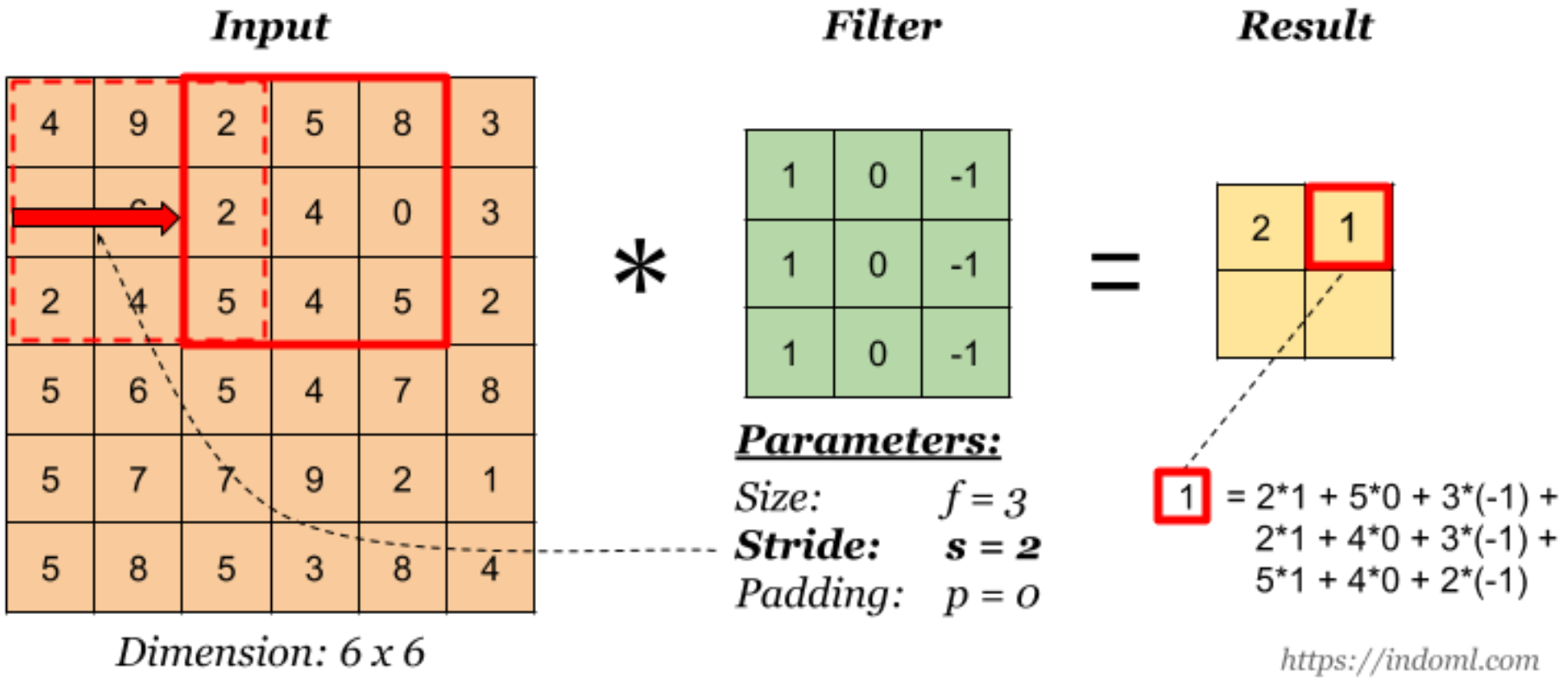
Building blocks of CNNs

Padding: same vs. valid

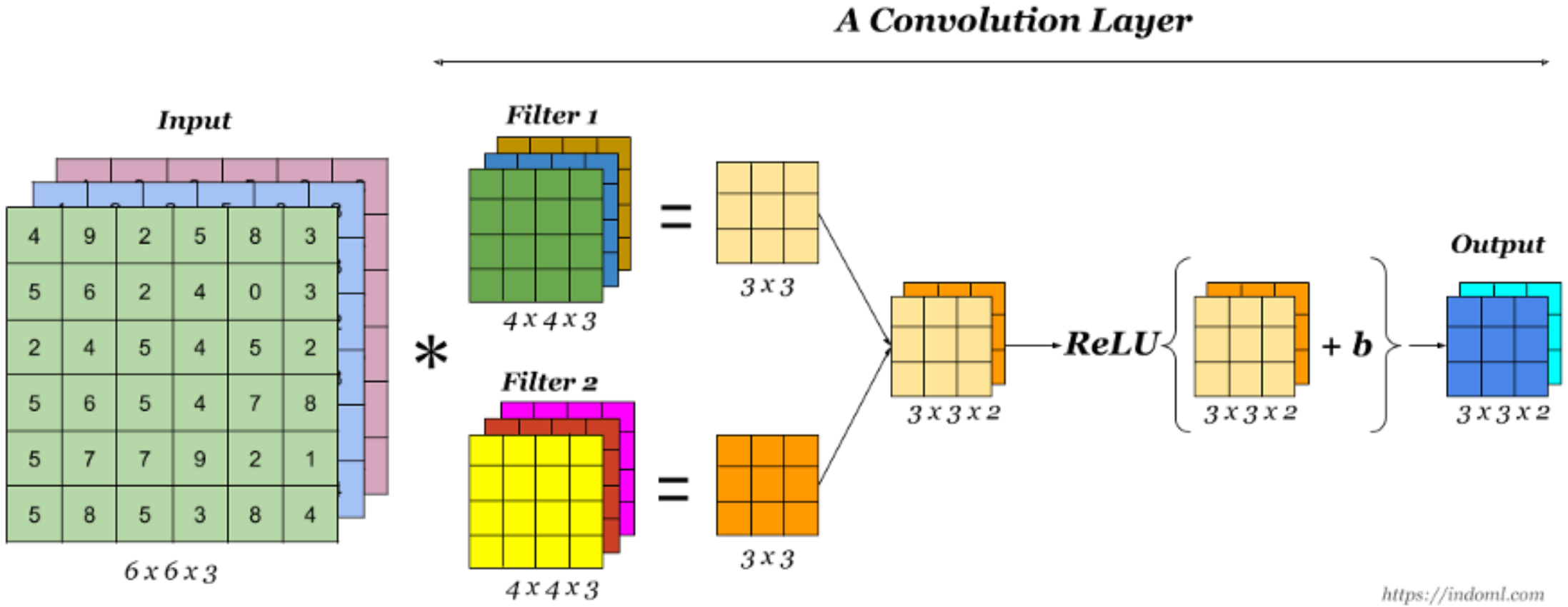


Building blocks of CNNs

Stride

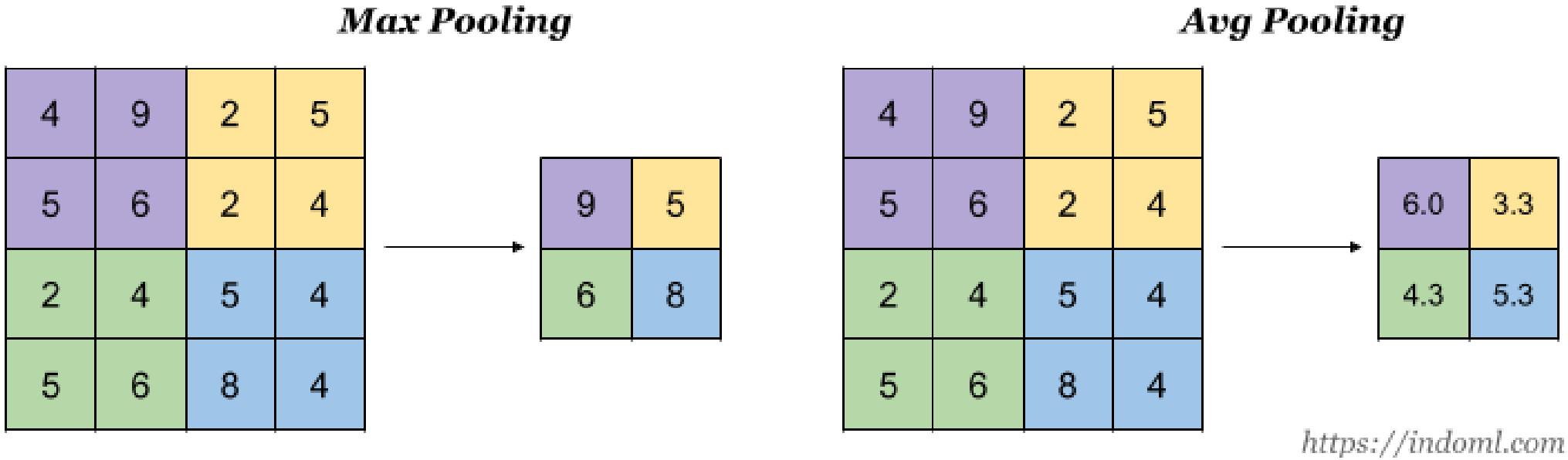


Building blocks of CNNs



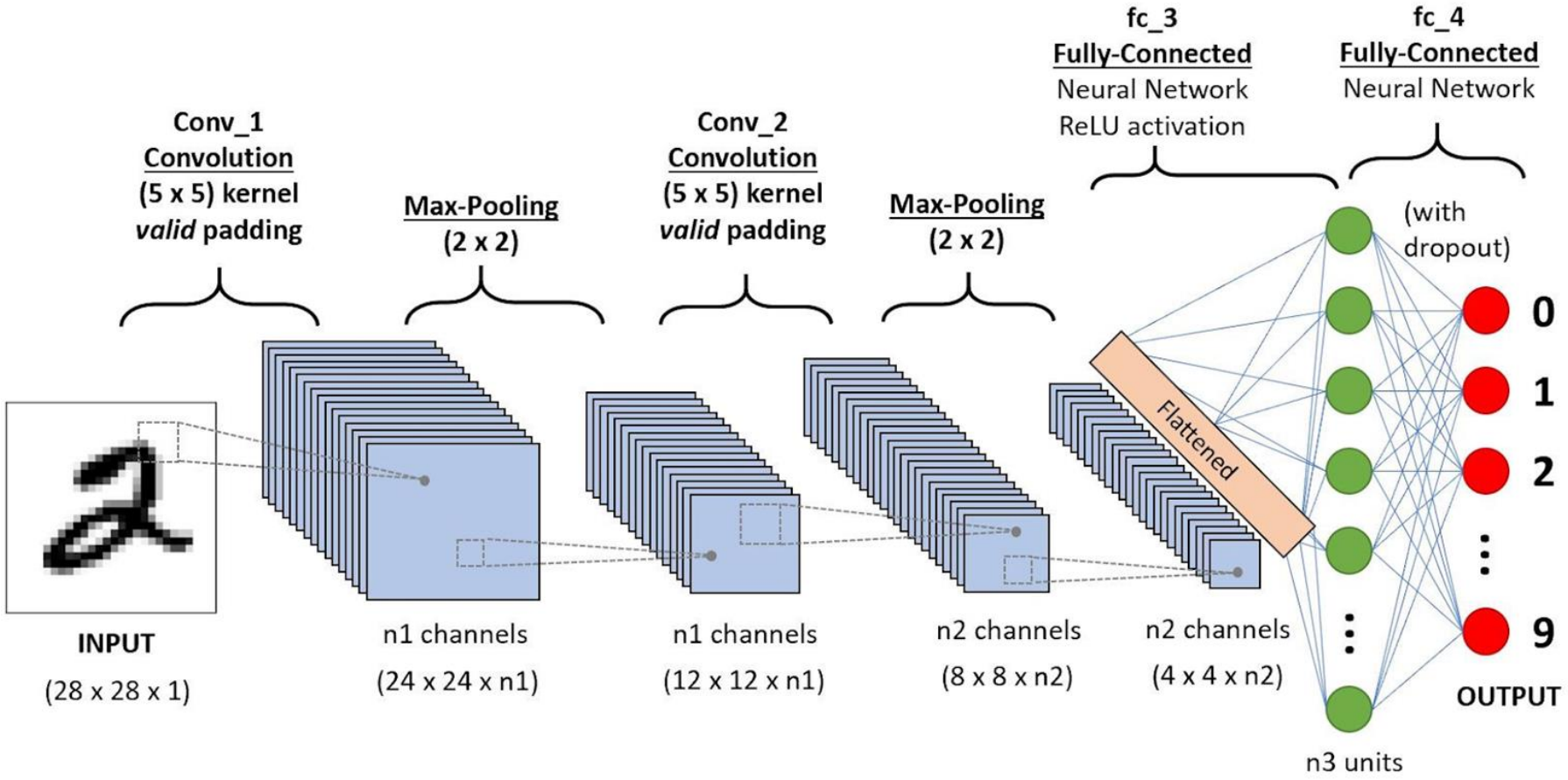
Building blocks of CNNs

Pooling layer



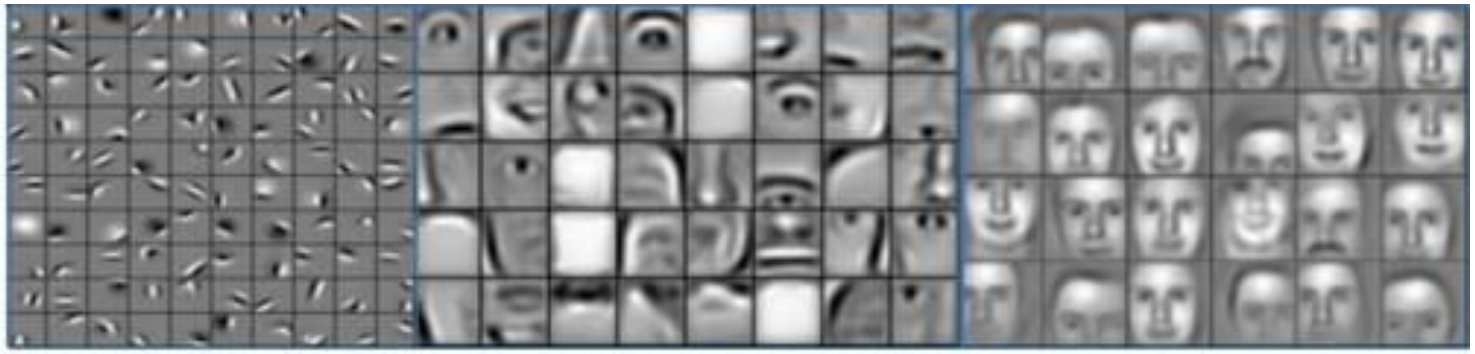
Building blocks of CNNs

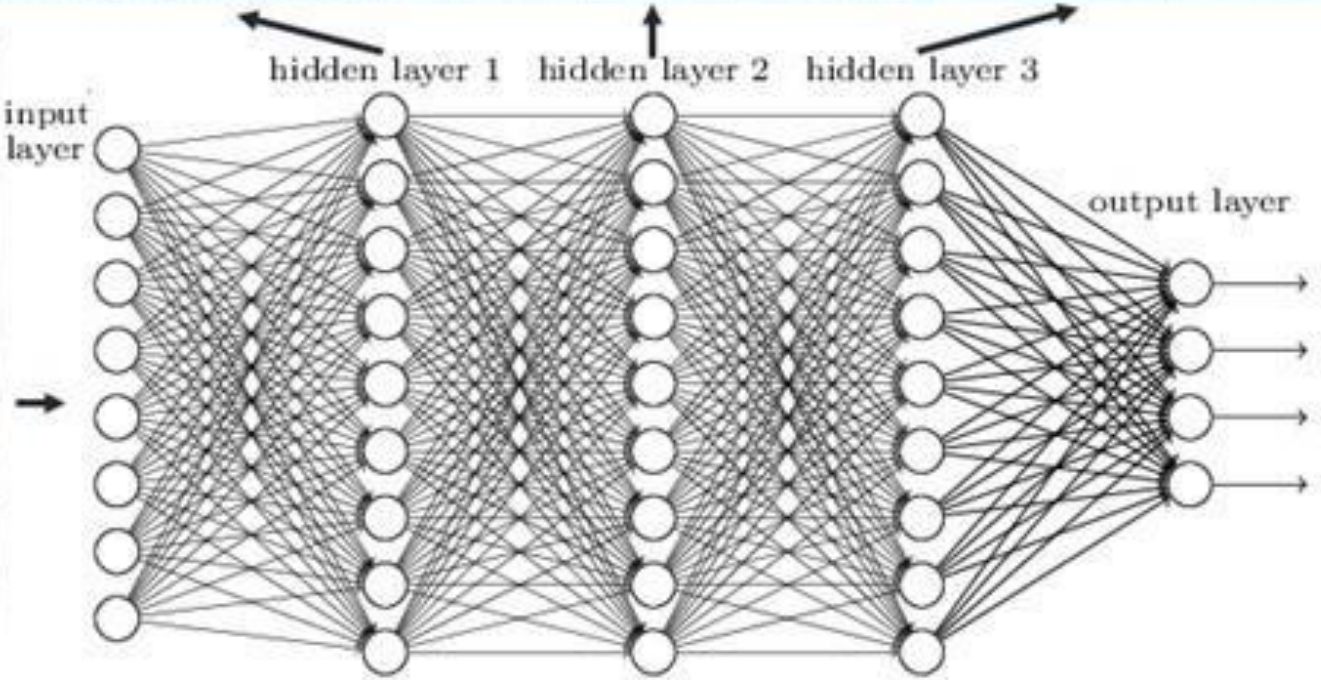
A multi-layer CNN



Deep learning is representation learning (a.k.a. feature learning)

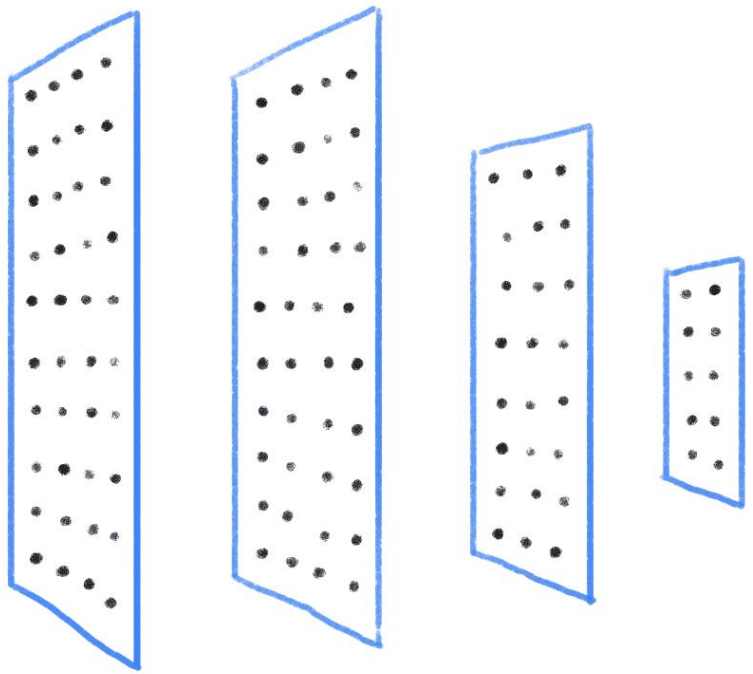
Deep neural networks learn hierarchical feature representations





Applications of CNNs

Image Classification

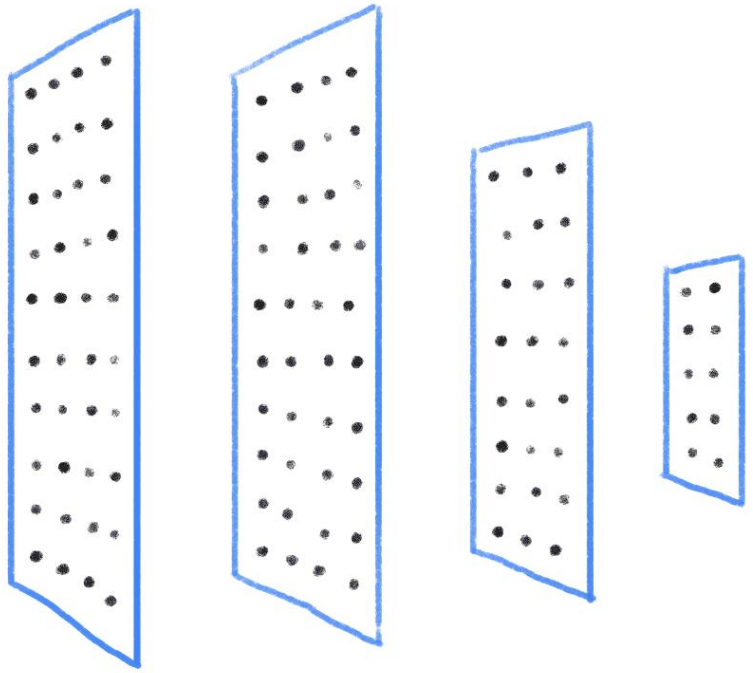


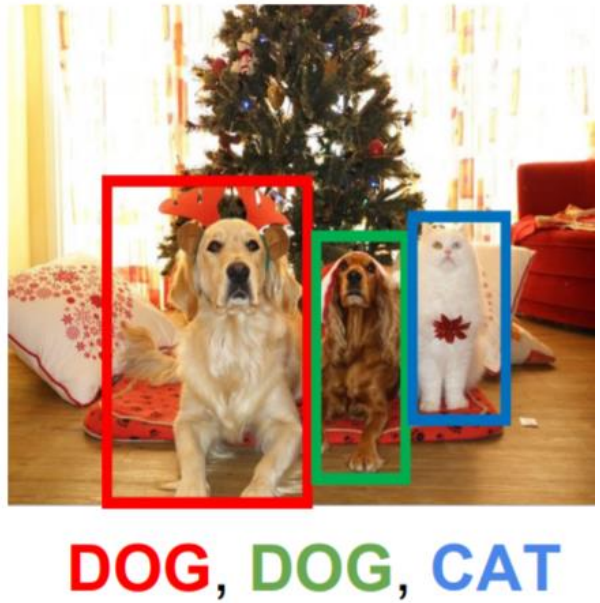
$P_{dog} = 0.9$

$P_{cat} = 0.1$

Applications of CNNs

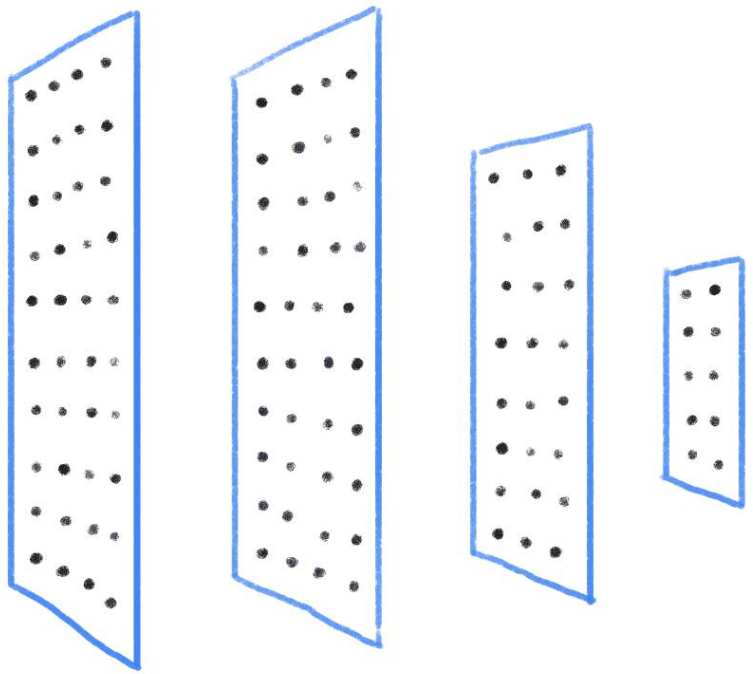
Object Detection





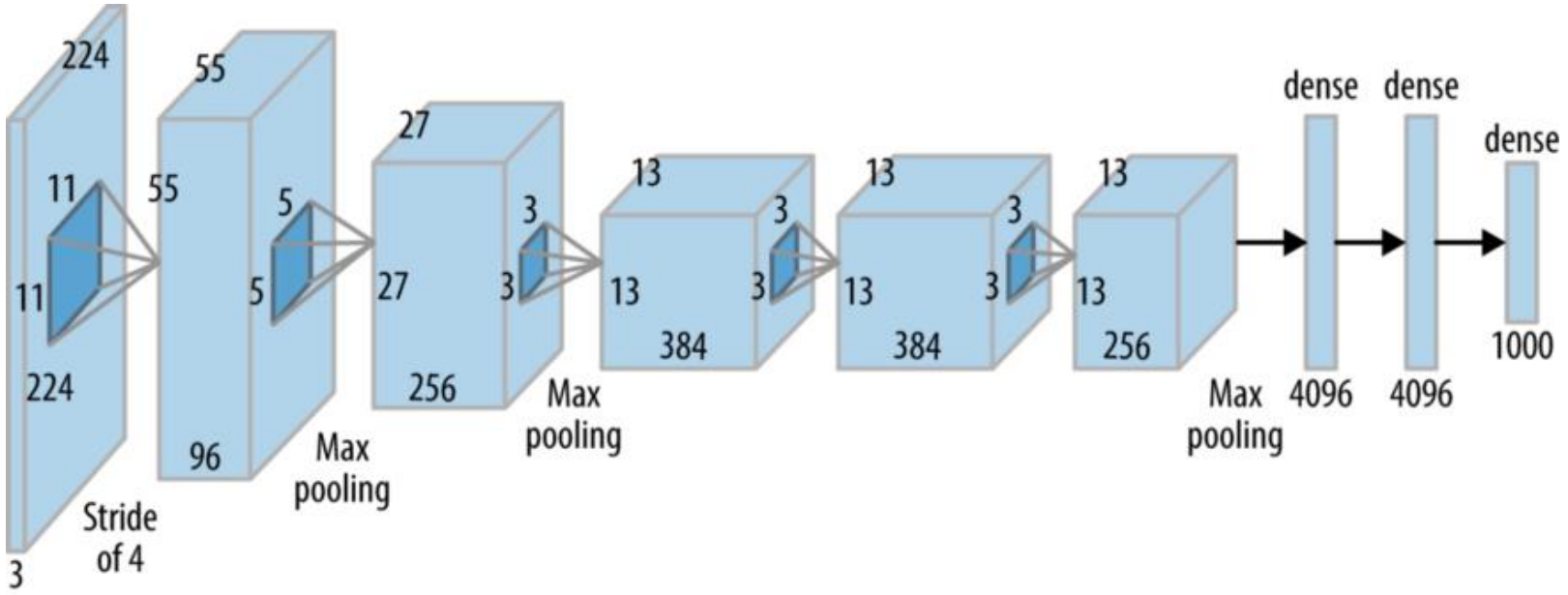
Applications of CNNs

Instance Segmentation



Popular CNN architectures

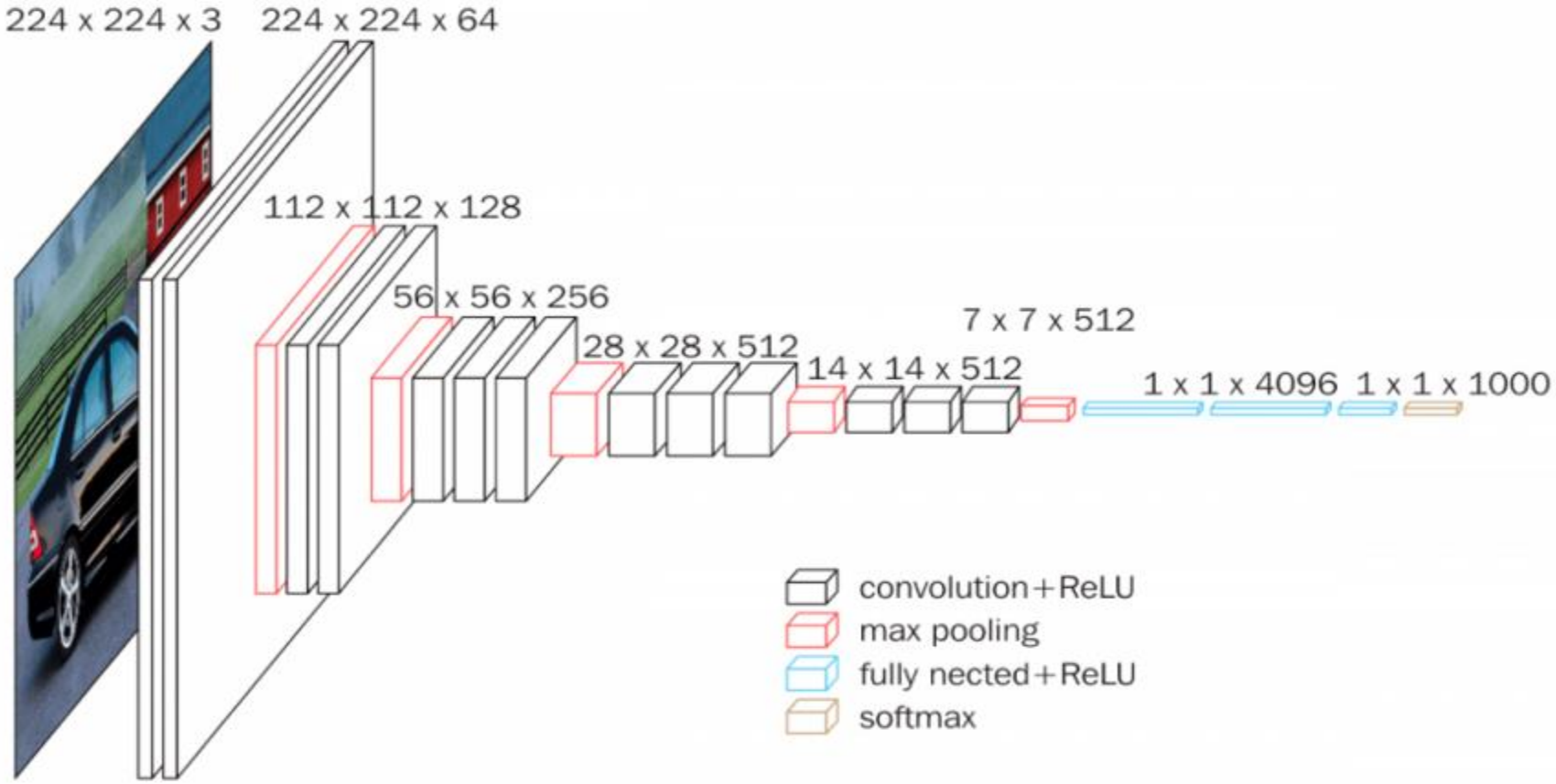
AlexNet (2012) – 62.5% accuracy on ImageNet



"The neural network, which has **60 million parameters** and 500,000 neurons, consists of **five convolutional layers**, some of which are followed by **max-pooling layers**, and **two globally connected layers** with a final **1000-way softmax**."

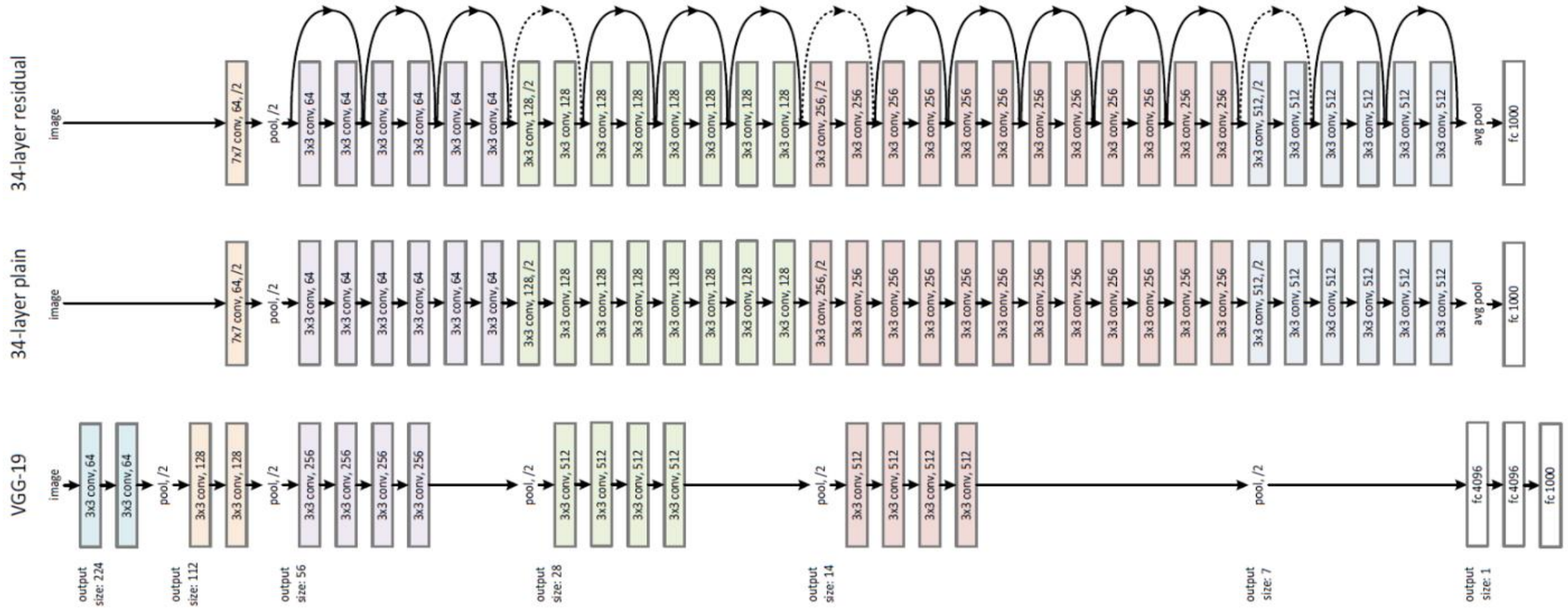
Popular CNN architectures

VGG16 (2014) – 71.5% accuracy on ImageNet



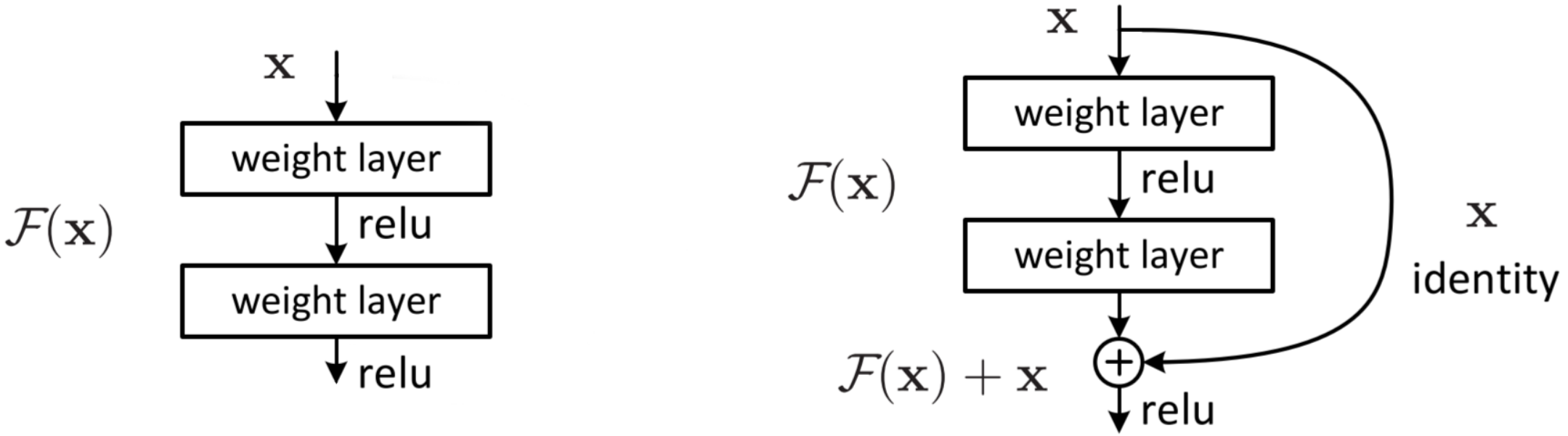
Popular CNN architectures

ResNet (2015) – >80% accuracy on ImageNet

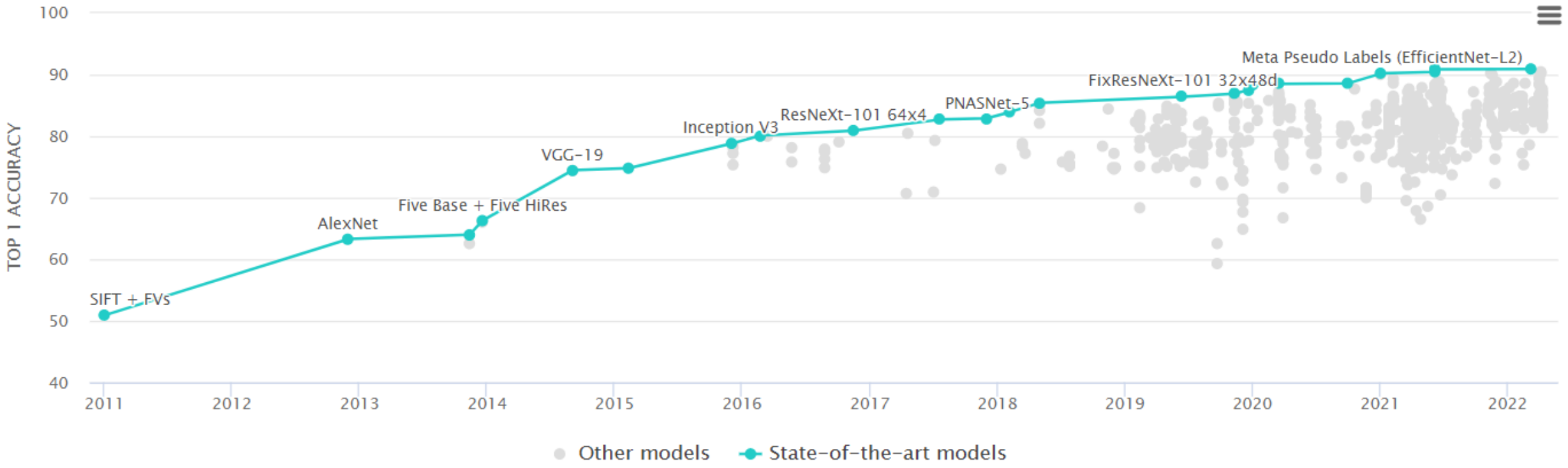


Popular CNN architectures

Residual block with a skip connection

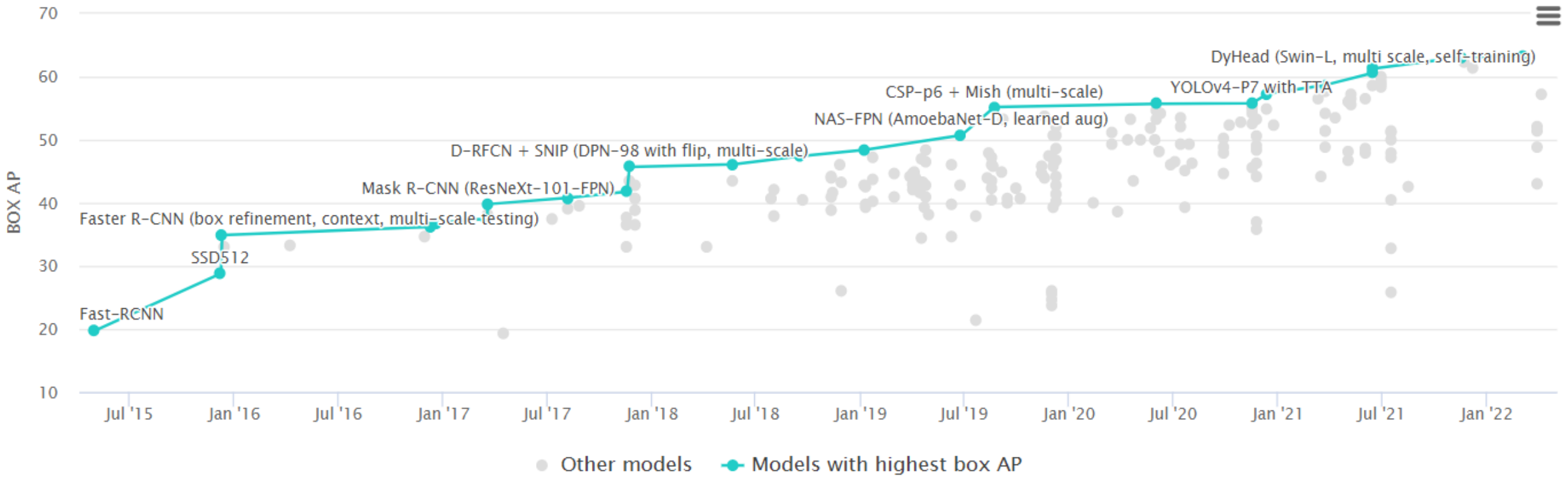


Trend of CNN-based classifiers



The data is collected from https://paperswithcode.com.

Trend of CNN-based object detectors



The data is collected from https://paperswithcode.com.

Questions?

References

- https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53
- https://indoml.com/2018/03/07/student-notes-convolutional-neural-networks-cnn-introduction/
- https://towardsdatascience.com/beginners-guide-to-understanding-convolutional-neural-networks-ae9ed58bb17d
- https://towardsdatascience.com/an-overview-of-resnet-and-its-variants-5281e2f56035
- https://www.doc.ic.ac.uk/~bkainz/teaching/DL/notes/equivariance.pdf
- https://paperswithcode.com