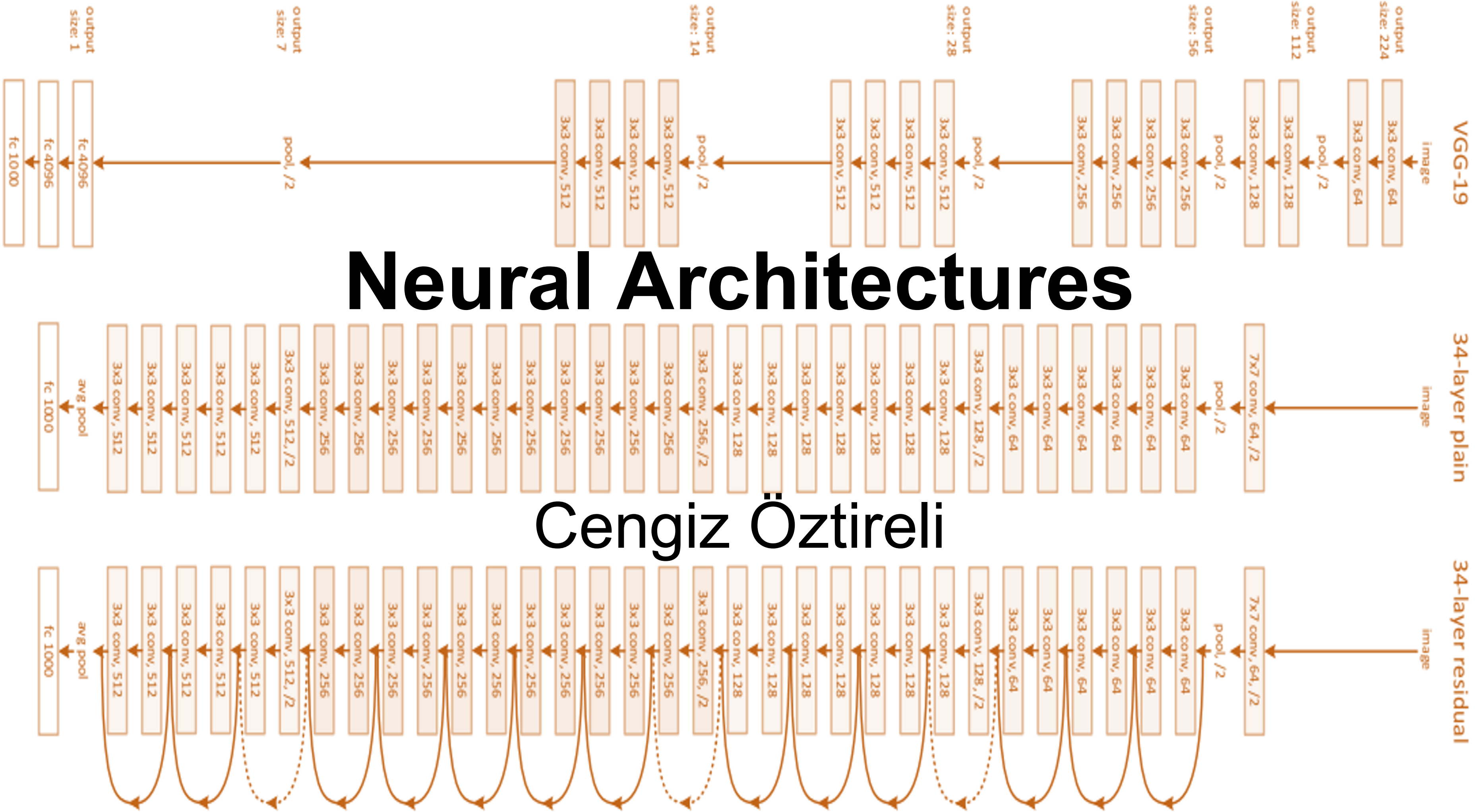
Neural Architectures

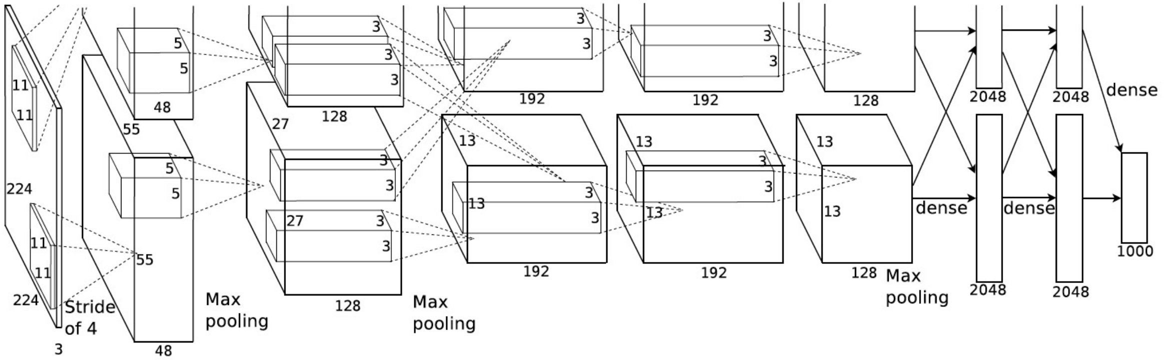
Cengiz Öztireli

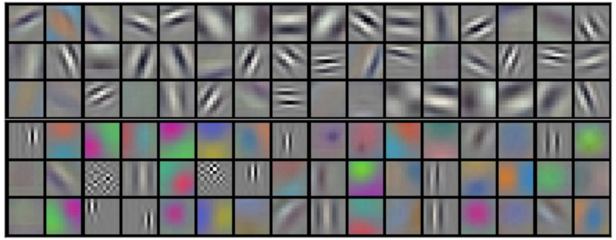


History of Neural Networks

Imagenet classification with deep convolutional neural networks

Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton, 2012





Visualization of convolutional kernels

What are convolutional neural networks?

- Neural networks: a stack of fully connected layers

If W=32, H=32,
x = 32*32*3=3072

$$\underset{10\times 1}{f(x,W)} = \overset{10\times 3072}{W}\,\underset{3072\times 1}{x} + b \;\; 10\times 1$$

Image

W * H * 3 numbers

$$f(x,W)$$

Cat	0.1
Dog	0.8
Person	0.02
Road	0.01
Tree	0.01
Airplane	0.01
Apple	0.01
Cell phone	0.03
Cup	0
Table	0.01

What are convolutional neural networks?

- Imagine the image size increase to 200 * 200 * 3

If W=200, H=200,
x = 200 * 200 * 3 = 120000!

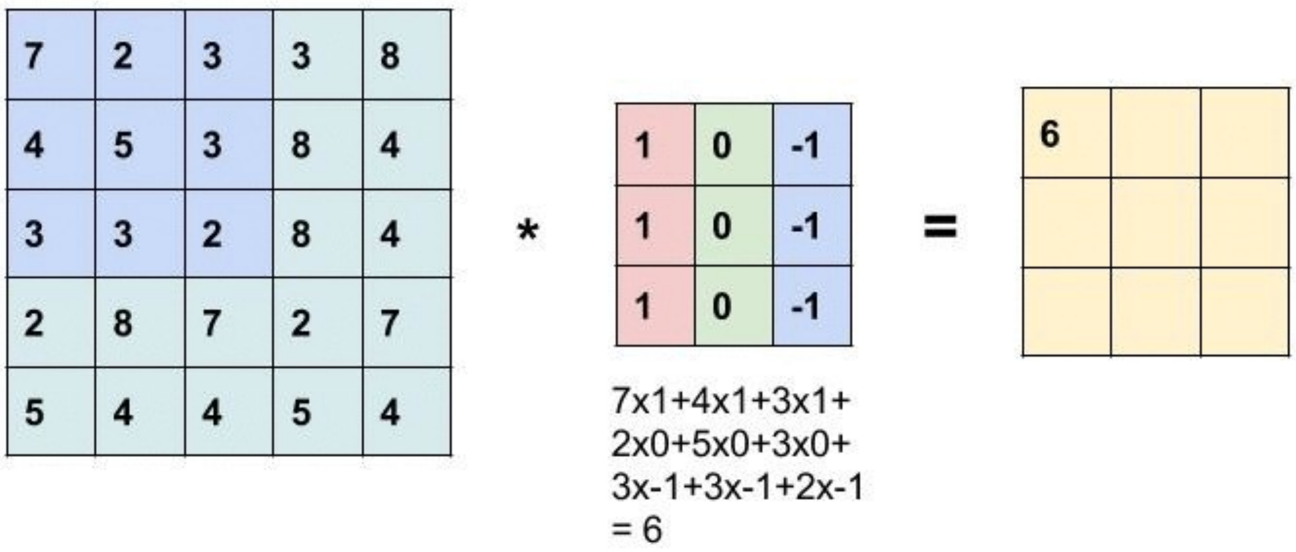
Image

W * H * 3 numbers

$$f(x, W) \longrightarrow$$

Cat	0.1
Dog	0.8
Person	0.02
Road	0.01
Tree	0.01
Airplane	0.01
Apple	0.01
Cell phone	0.03
Cup	0
Table	0.01

Convolution



Convolution

- Kernel size 3 x 3

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

Convolution

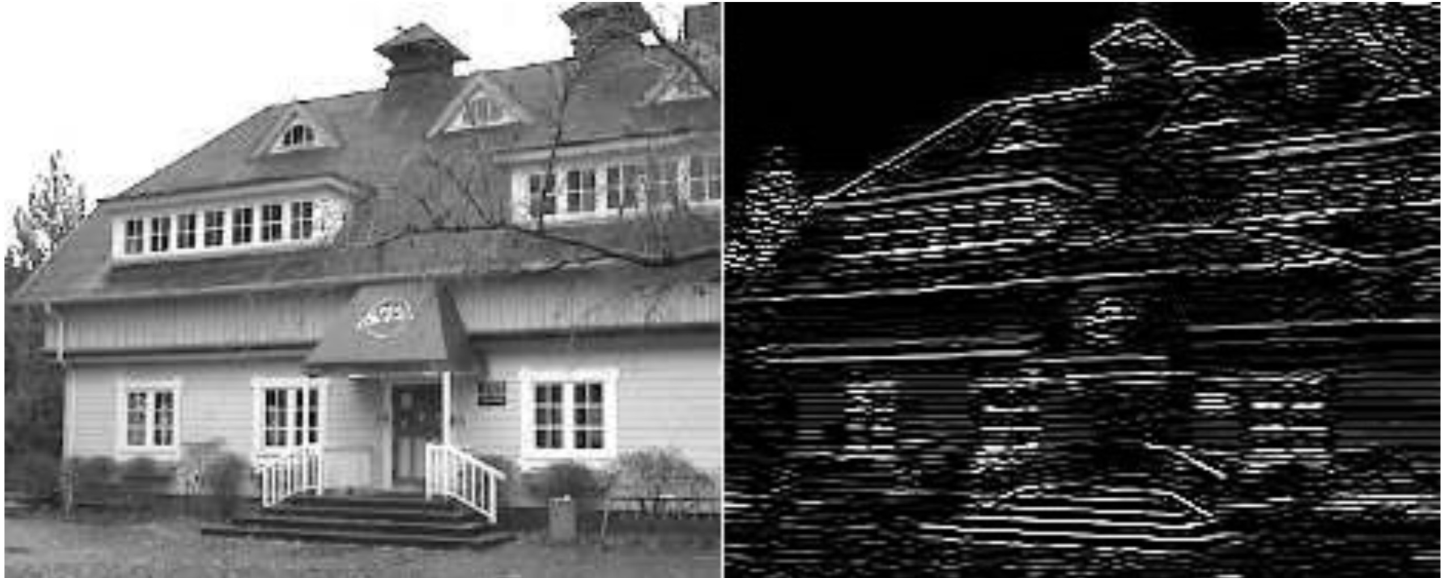
- Meaning of the kernel weights…filters

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

Convolution

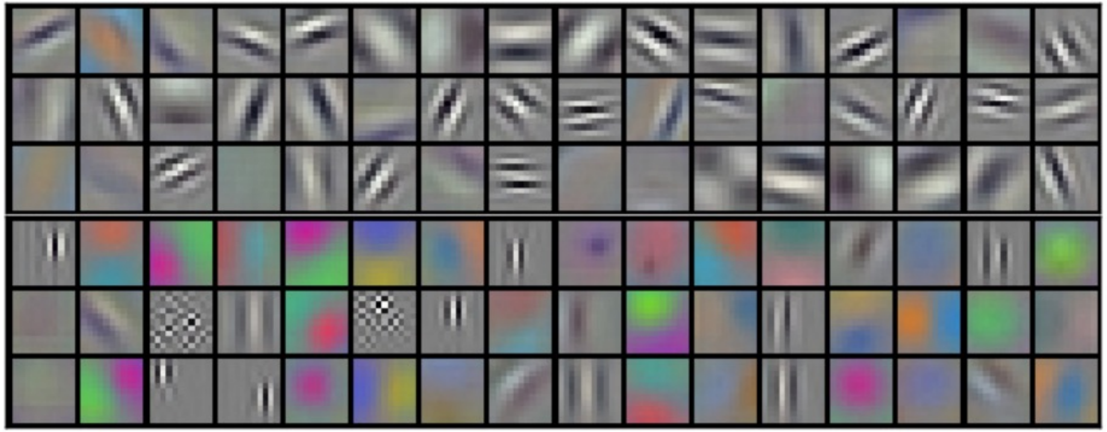
- Meaning of the kernel weights …filters

-1	-1	-1
2	2	2
-1	-1	-1



Convolution

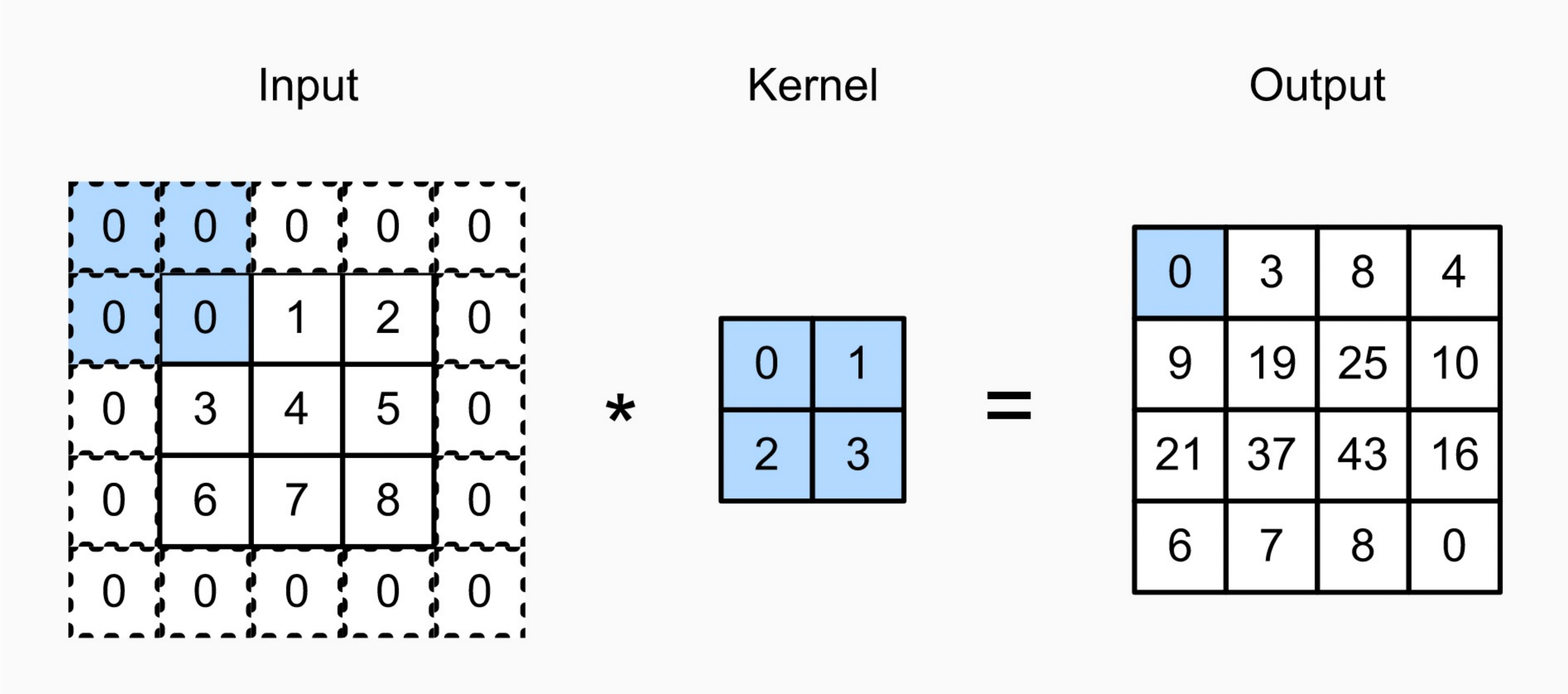
- Trained convolutional kernels
  - Human-designed features: SIFT, HOG
  - Data-driven features: CNN features



Input size:
7
Kernel size:
3
Padding:
0
Dilation:
1
Stride:
1
Input (7 × 7):
Weight (3 × 3):
Output (5 × 5):

Input size:
7
Kernel size:
3
Padding:
0
Dilation:
1
Stride:
2
Input (7 × 7):
Weight (3 × 3):
Output (3 × 3):

Padding



Input
Kernel
Output
0 0 0 0 0
0 0 1 2 0
0 3 4 5 0
0 6 7 8 0
0 0 0 0 0
*
0 1
2 3
=
0 3 8 4
9 19 25 10
21 37 43 16
6 7 8 0

Output Size

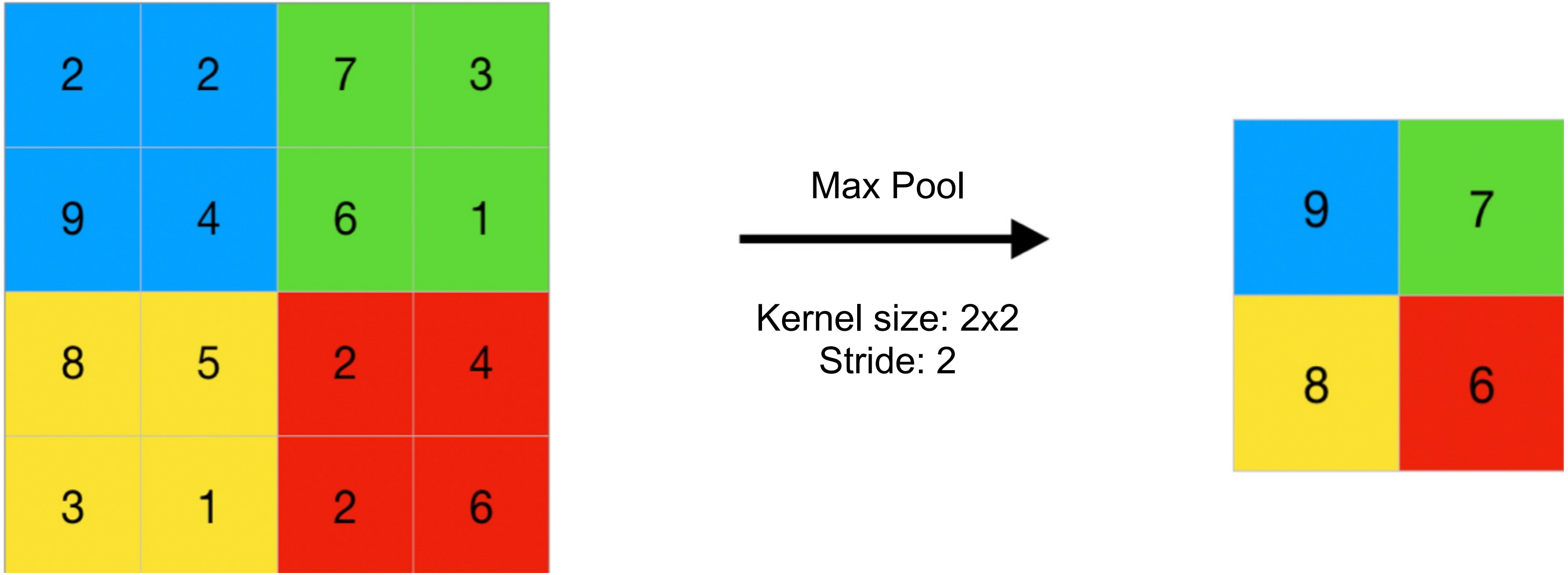
Input size Kernel size Padding size

$$Out = \frac{In - K + 2P}{S} + 1$$

Stride

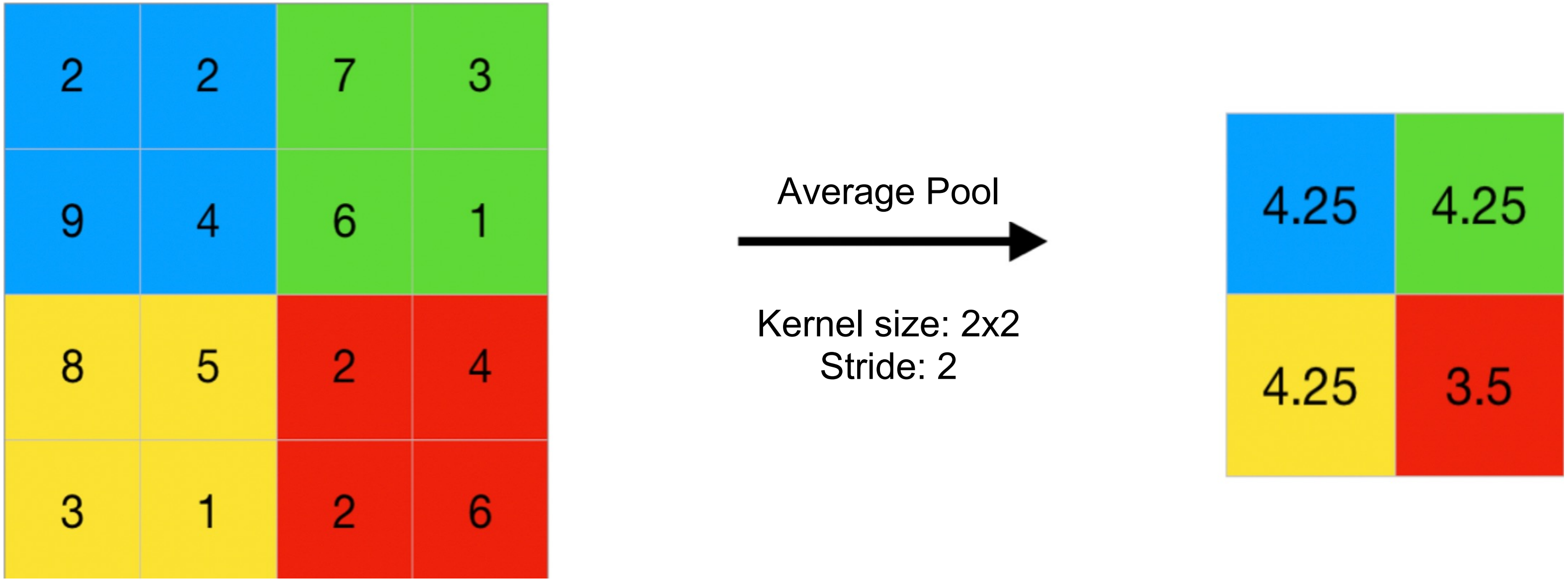
Pooling Layer

- Reduce the size of the feature map



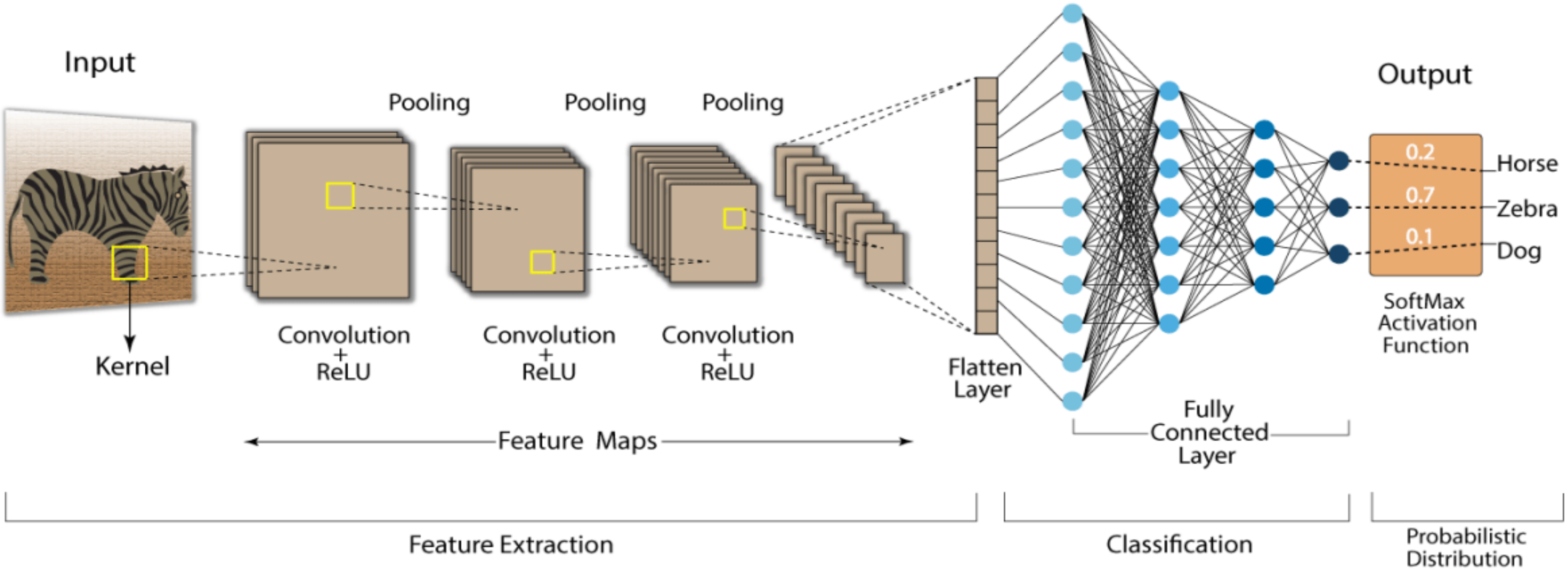
Pooling Layer

- Reduce the size of the feature map

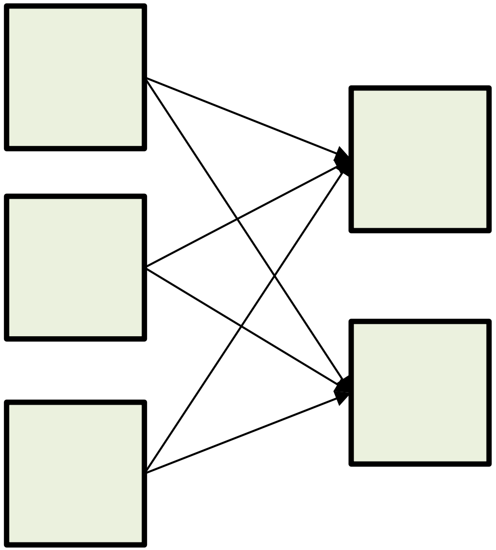


Fully Connected Layer

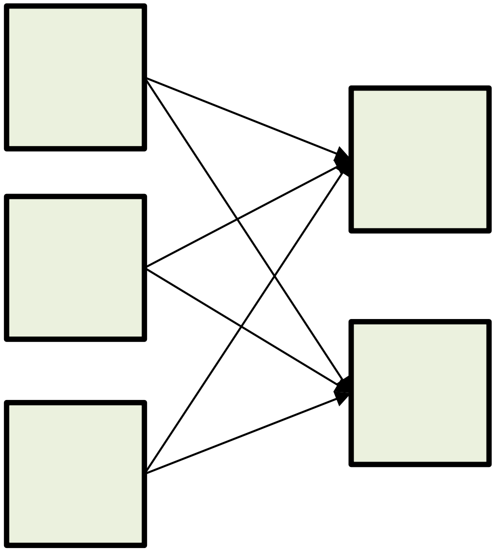
Convolutional neural network (CNN)

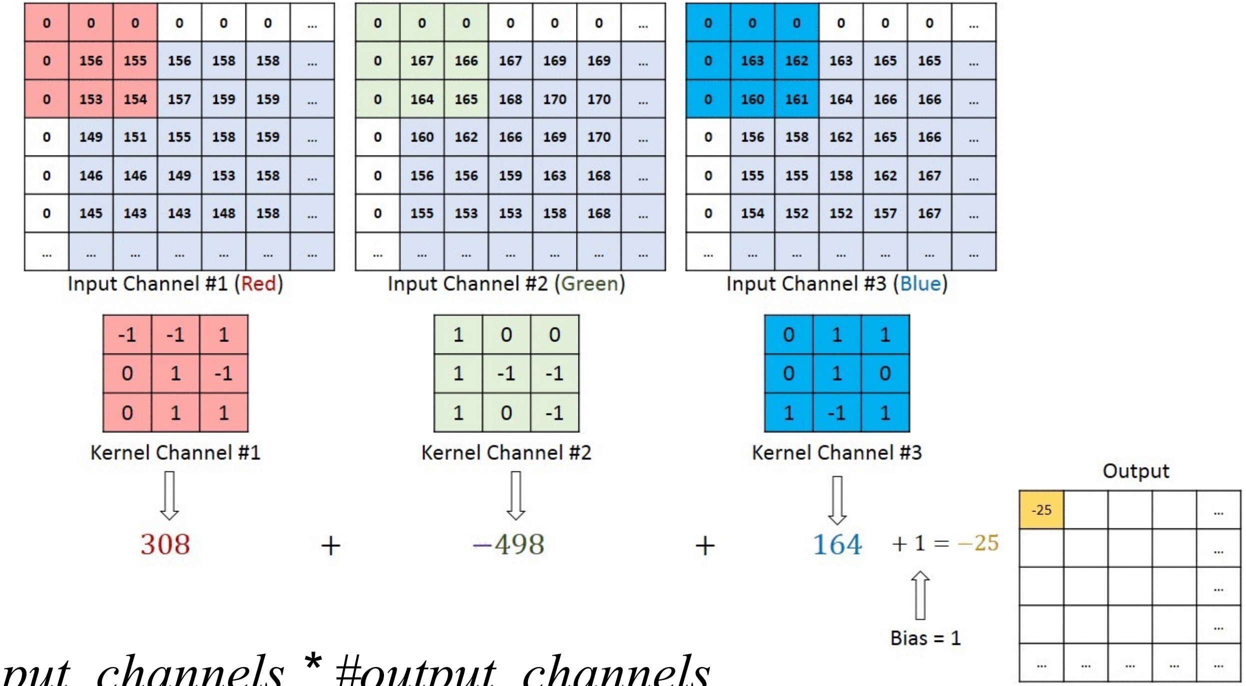


Parameters of Convolution



Parameters of Convolution

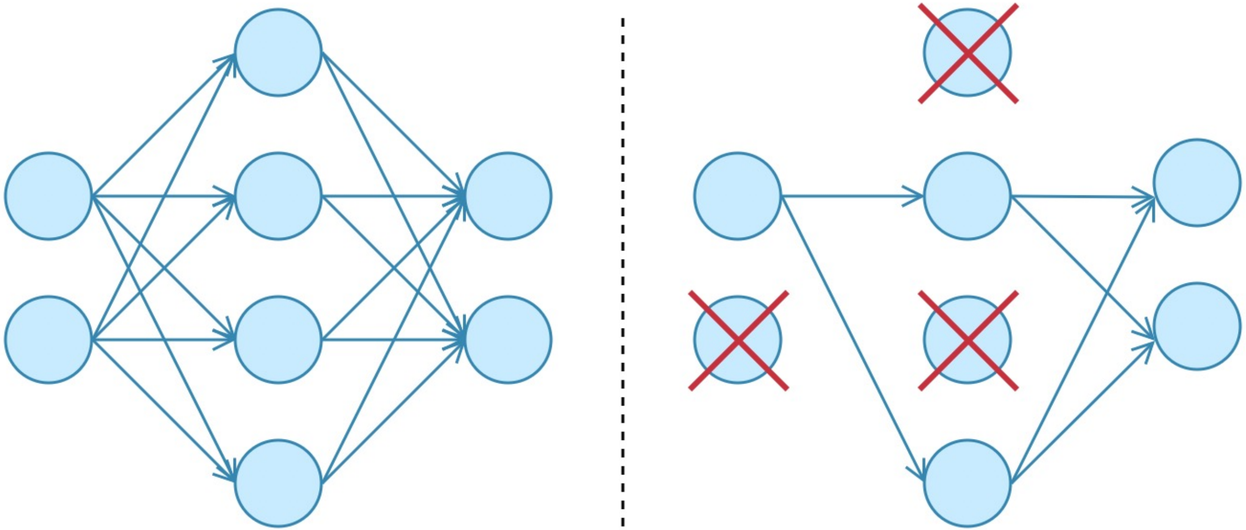




$$\#parameters = K * K * \#input\_channels * \#output\_channels$$

UNIVERSITY OF CAMBRIDGE

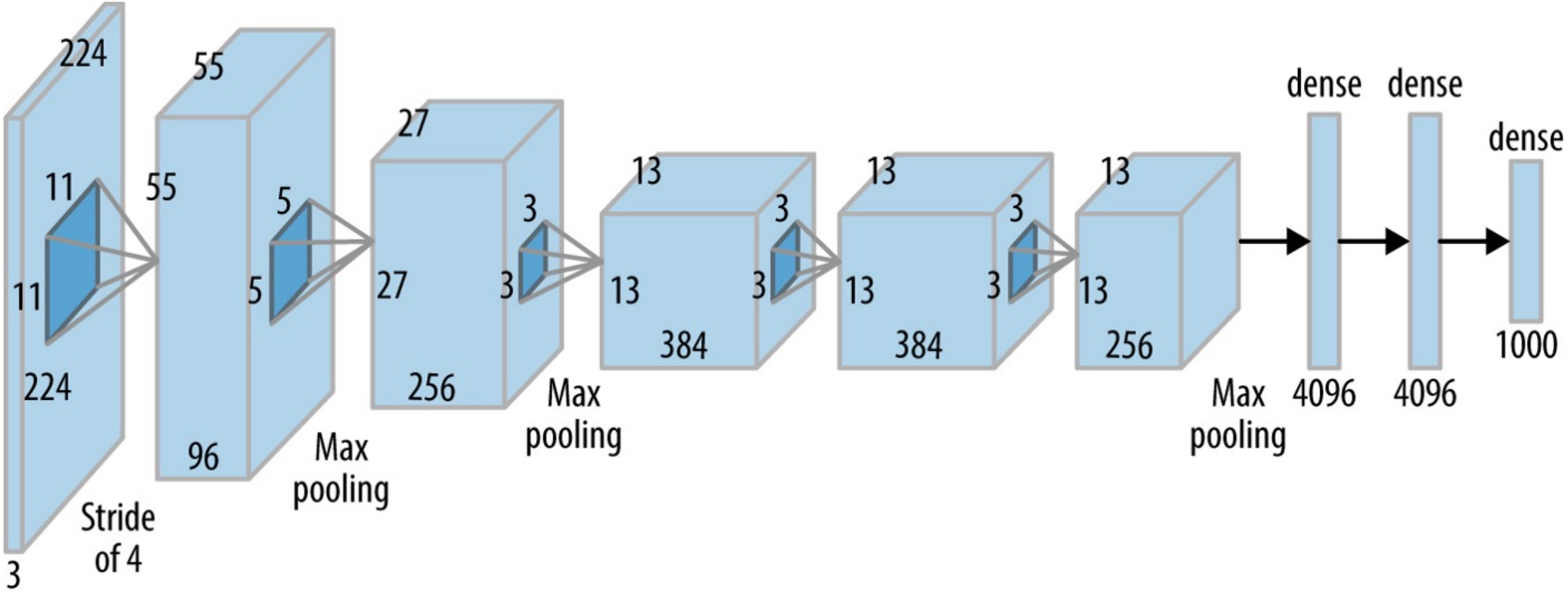
Dropout Layer



No Dropout

With Dropout

AlexNet

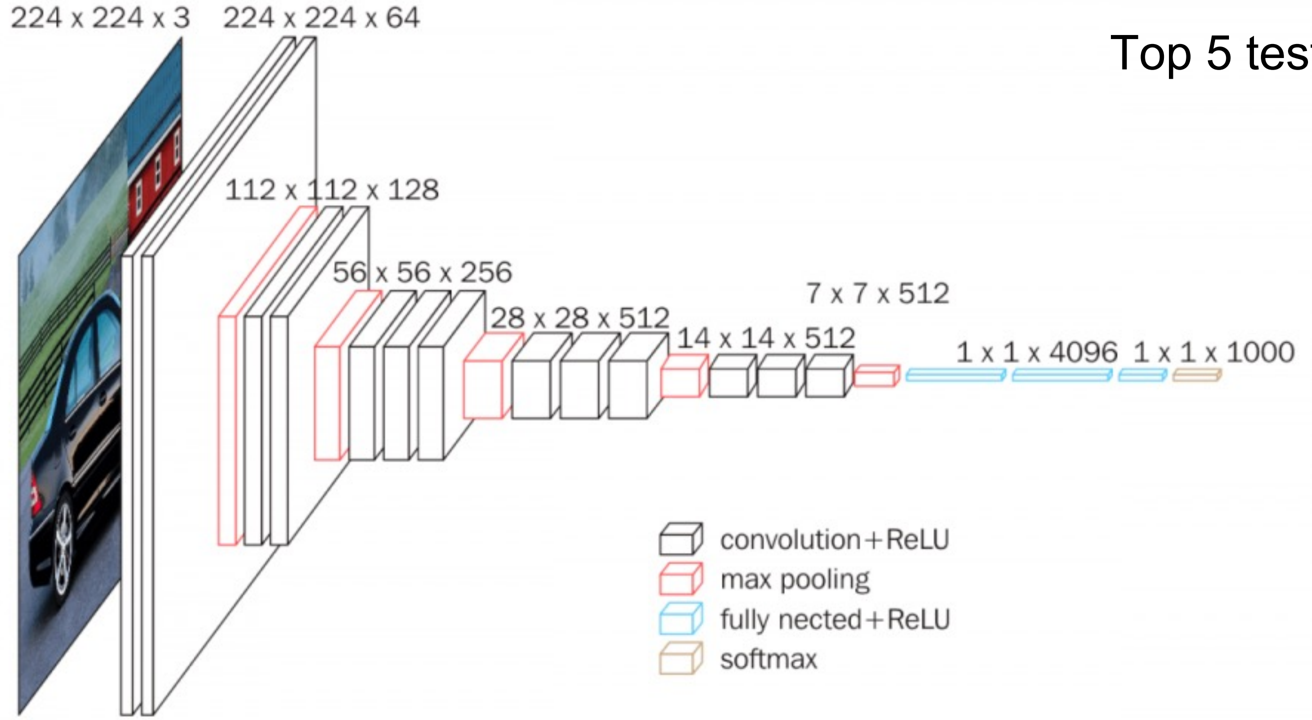


UNIVERSITY OF CAMBRIDGE

Parameters

AlexNet Network - Structural Details													
Input			Output			Layer	Stride	Pad	Kernel size		in	out	# of Param
227	227	3	55	55	96	conv1	4	0	11	11	3	96	34944
55	55	96	27	27	96	maxpool1	2	0	3	3	96	96	0
27	27	96	27	27	256	conv2	1	2	5	5	96	256	614656
27	27	256	13	13	256	maxpool2	2	0	3	3	256	256	0
13	13	256	13	13	384	conv3	1	1	3	3	256	384	885120
13	13	384	13	13	384	conv4	1	1	3	3	384	384	1327488
13	13	384	13	13	256	conv5	1	1	3	3	384	256	884992
13	13	256	6	6	256	maxpool5	2	0	3	3	256	256	0
						fc6			1	1	9216	4096	37752832
						fc7			1	1	4096	4096	16781312
						fc8			1	1	4096	1000	4097000
						Total							**62,378,344**

VGG16

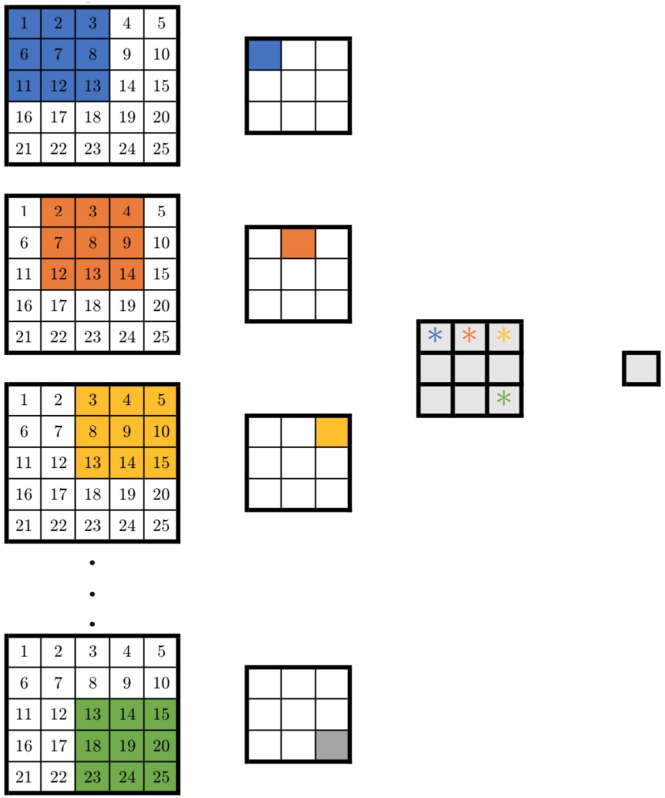


Top 5 test: 15.4% → 7.3%

Parameters

VGG16 - Structural Details														
#	Input Image			output			Layer	Stride	Kernel		in	out	Param
1	224	224	3	224	224	64	conv3-64	1	3	3	3	64	1792
2	224	224	64	224	224	64	conv3064	1	3	3	64	64	36928
	224	224	64	112	112	64	maxpool	2	2	2	64	64	0
3	112	112	64	112	112	128	conv3-128	1	3	3	64	128	73856
4	112	112	128	112	112	128	conv3-128	1	3	3	128	128	147584
	112	112	128	56	56	128	maxpool	2	2	2	128	128	65664
5	56	56	128	56	56	256	conv3-256	1	3	3	128	256	295168
6	56	56	256	56	56	256	conv3-256	1	3	3	256	256	590080
7	56	56	256	56	56	256	conv3-256	1	3	3	256	256	590080
	56	56	256	28	28	256	maxpool	2	2	2	256	256	0
8	28	28	256	28	28	512	conv3-512	1	3	3	256	512	1180160
9	28	28	512	28	28	512	conv3-512	1	3	3	512	512	2359808
10	28	28	512	28	28	512	conv3-512	1	3	3	512	512	2359808
	28	28	512	14	14	512	maxpool	2	2	2	512	512	0
11	14	14	512	14	14	512	conv3-512	1	3	3	512	512	2359808
12	14	14	512	14	14	512	conv3-512	1	3	3	512	512	2359808
13	14	14	512	14	14	512	conv3-512	1	3	3	512	512	2359808
	14	14	512	7	7	512	maxpool	2	2	2	512	512	0
14	1	1	25088	1	1	4096	fc		1	1	25088	4096	102764544
15	1	1	4096	1	1	4096	fc		1	1	4096	4096	16781312
16	1	1	4096	1	1	1000	fc		1	1	4096	1000	4097000
Total													**138,423,208**

Receptive Field

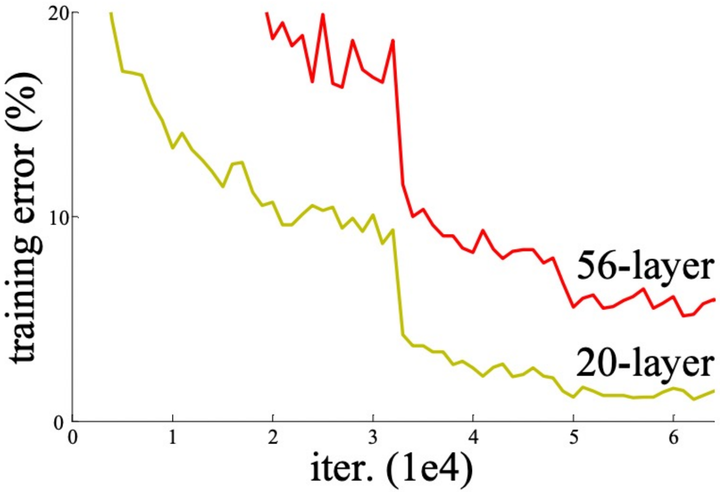


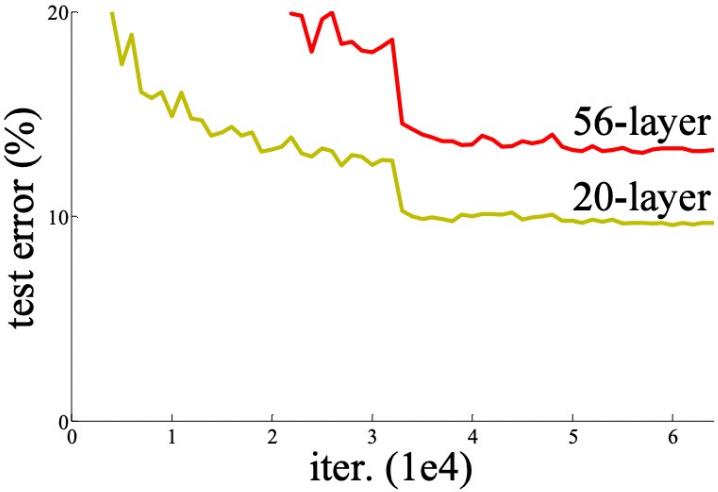
Receptive field: the size of the region in the input that produces the feature.

Logarithmic relationship between classification accuracy and receptive field size.

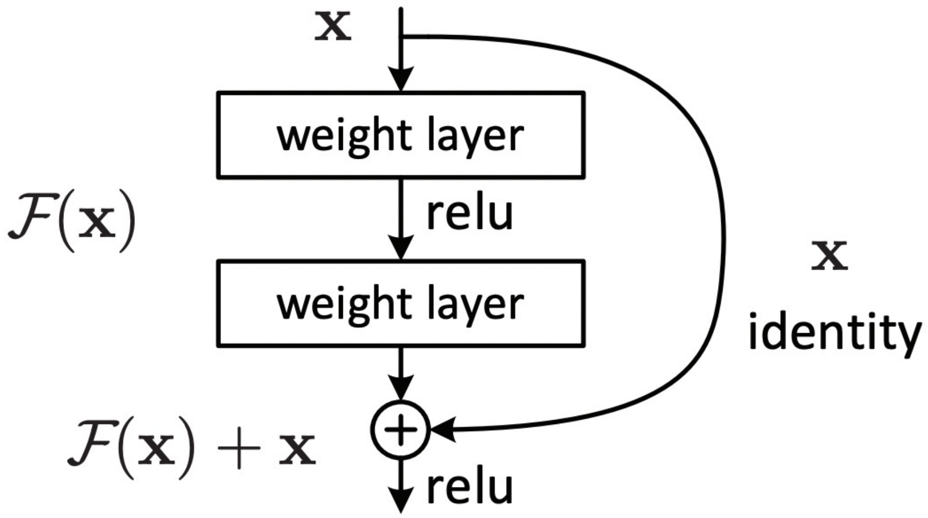
ResNet

- Deep Networks are hard to train





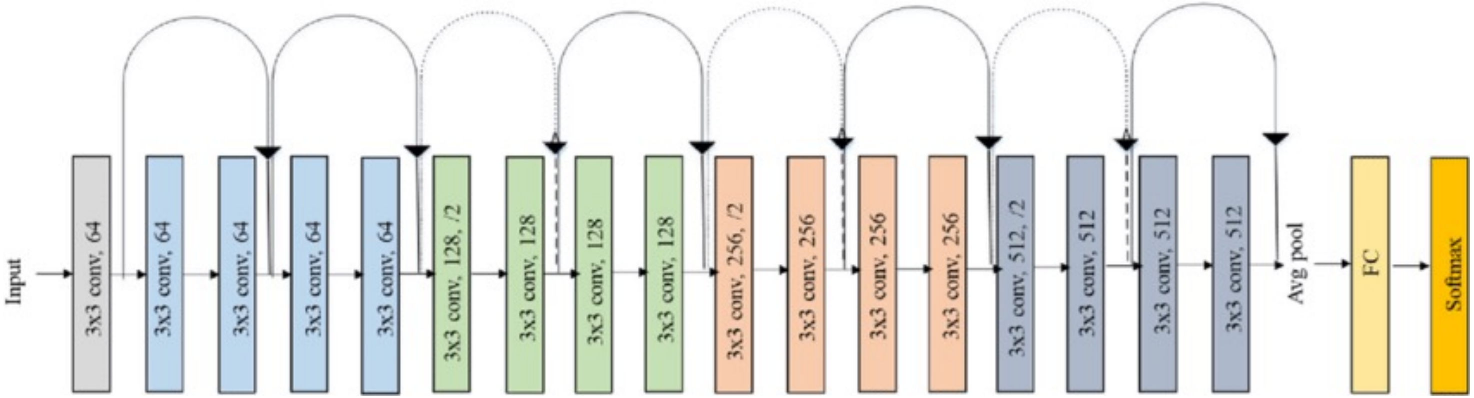
ResNet



Shortcut connections:
Solve the problem of vanishing/exploding gradients.

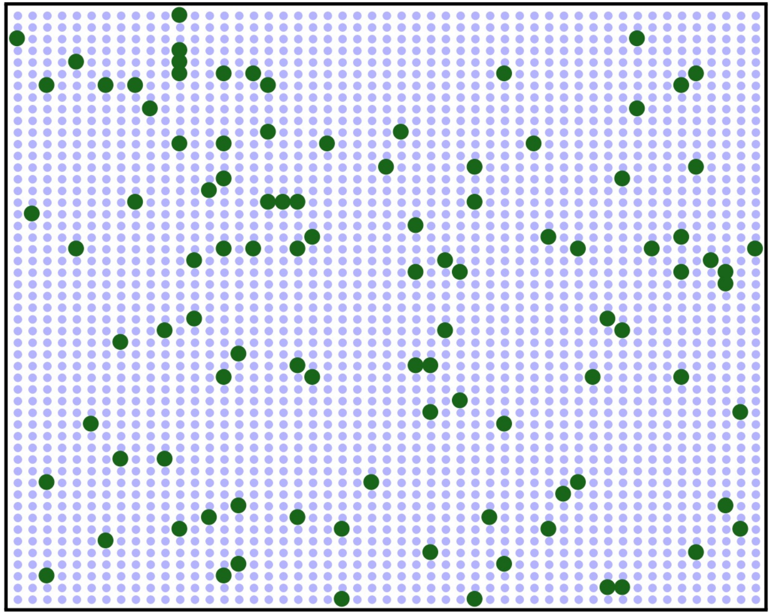
Residual block:
Repeated throughout the network

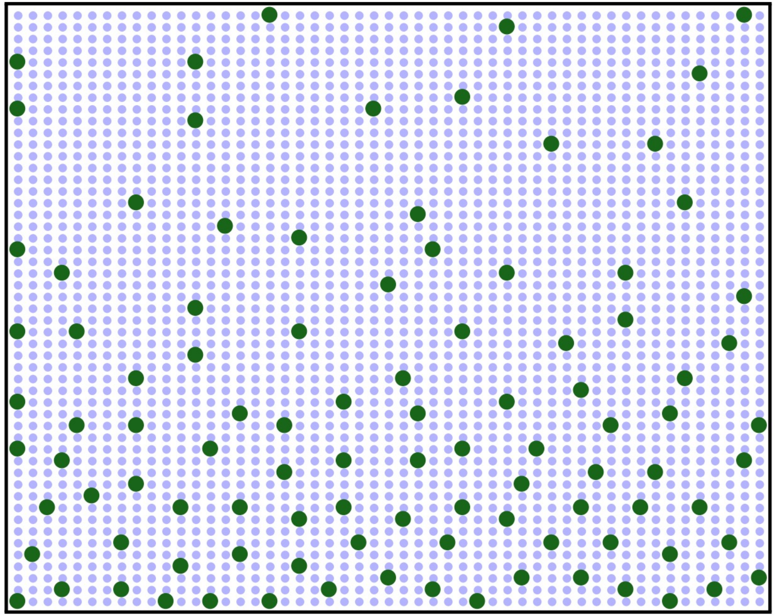
ResNet18



UNIVERSITY OF CAMBRIDGE

Batch Normalization





Batch Normalization

Input: Values of x over a mini-batch: $\mathcal{B} = \{x_{1 \ldots m}\}$;
Parameters to be learned: γ, β

Output: $\{y_i = \mathrm{BN}_{\gamma,\beta}(x_i)\}$

$$\mu_{\mathcal{B}} \leftarrow \frac{1}{m} \sum_{i=1}^{m} x_i \qquad \text{// mini-batch mean}$$

$$\sigma_{\mathcal{B}}^2 \leftarrow \frac{1}{m} \sum_{i=1}^{m} (x_i - \mu_{\mathcal{B}})^2 \qquad \text{// mini-batch variance}$$

$$\widehat{x}_i \leftarrow \frac{x_i - \mu_{\mathcal{B}}}{\sqrt{\sigma_{\mathcal{B}}^2 + \epsilon}} \qquad \text{// normalize}$$

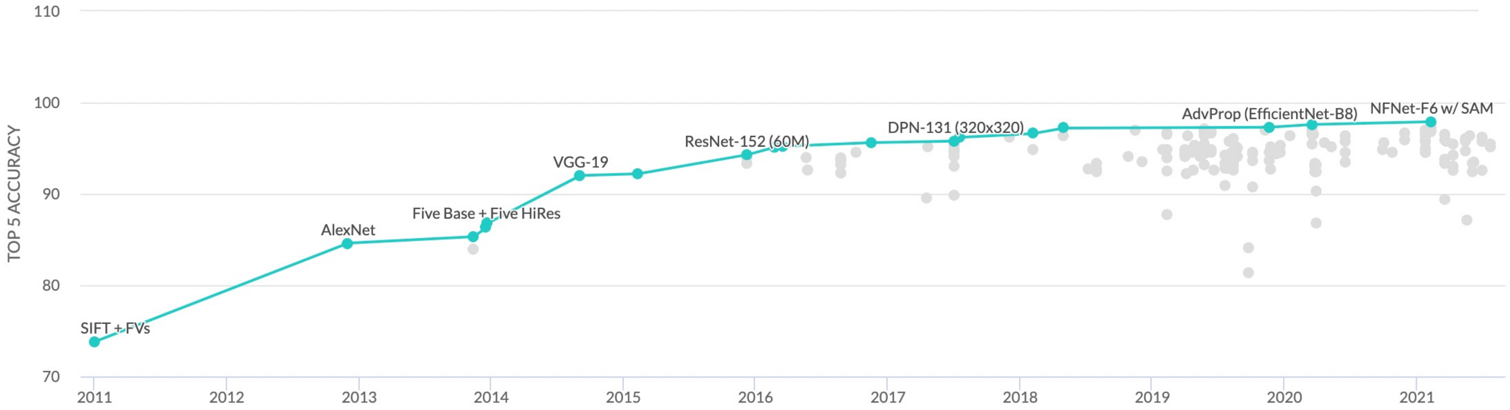
$$y_i \leftarrow \gamma \widehat{x}_i + \beta \equiv \mathrm{BN}_{\gamma,\beta}(x_i) \qquad \text{// scale and shift}$$

We update the mean and the variance during the training.

During the inference, we fix the BN layer, and use a fixed mean and variance from the training stage.

Comparison

Network	Year	Top 5 Accuracy	Parameters	Gflops
AlexNet	2012	84.70%	62M	1.5B
VGG16	2014	92.3%	138M	19.6B
ResNet152	2015	95.51%	60.3M	11B



UNIVERSITY OF CAMBRIDGE