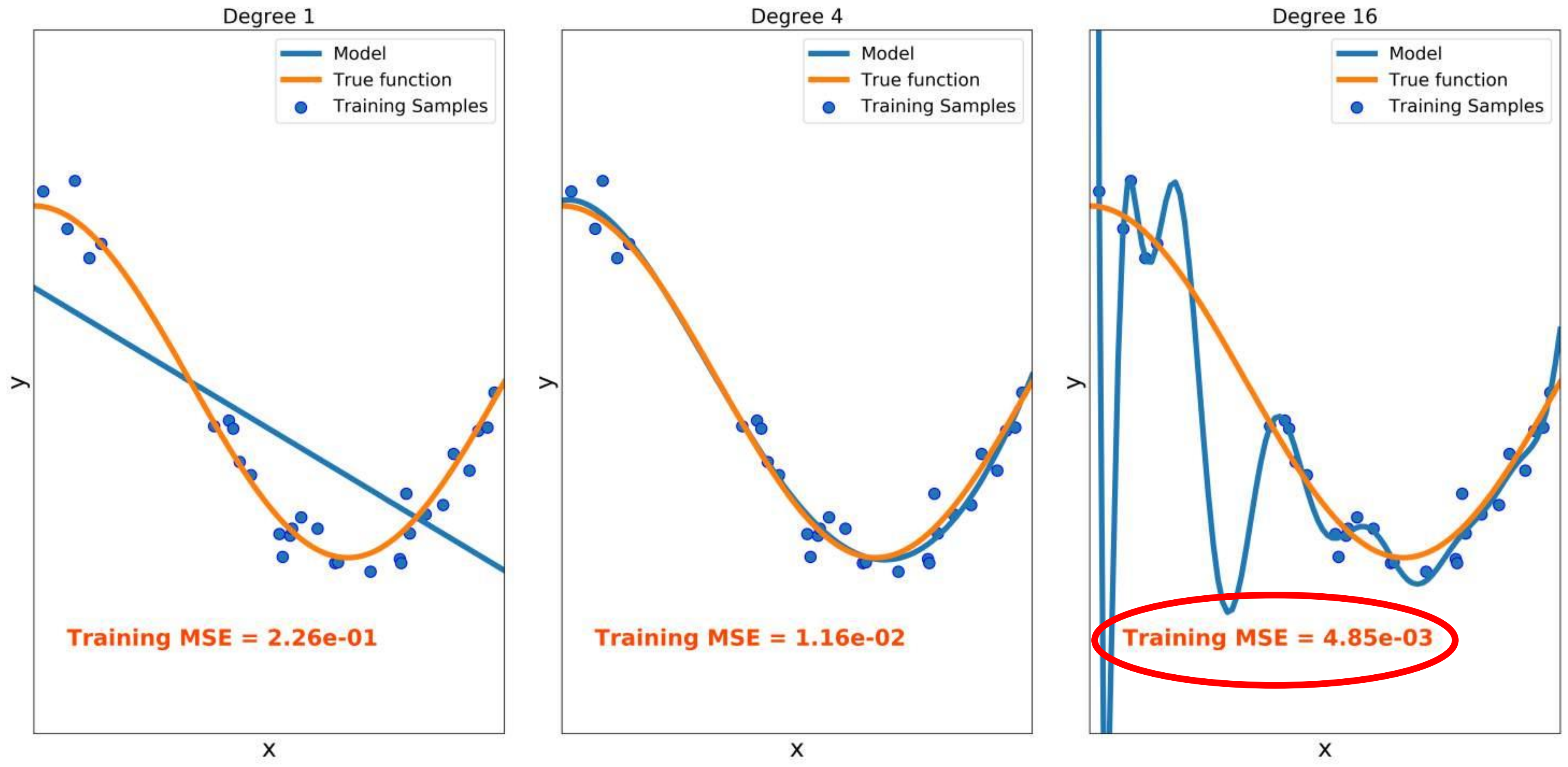
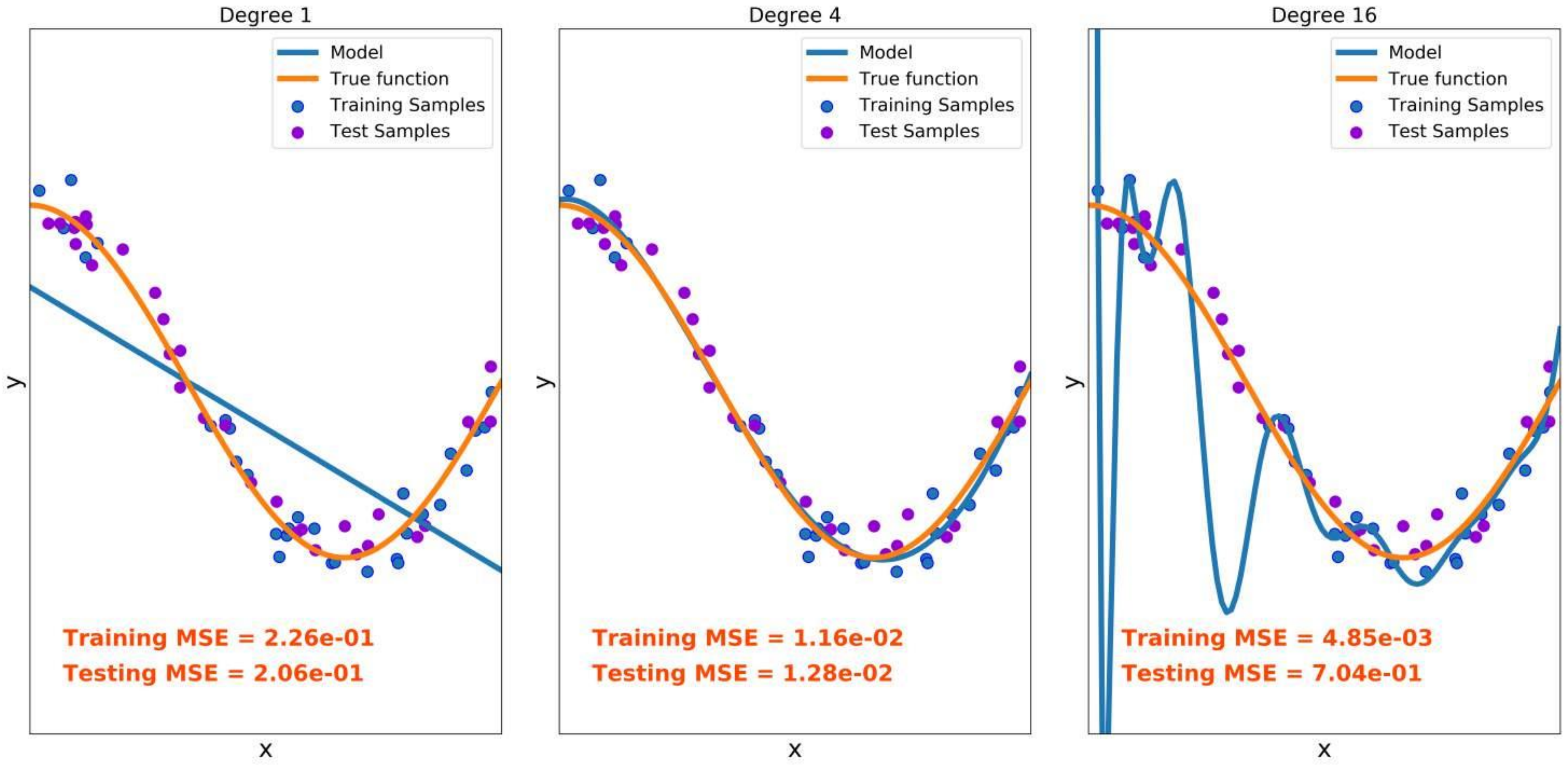
Regularisation

Overfitting/Underfitting



Overfitting/Underfitting



High Bias

High Variance

Bias-Variance Tradeoff

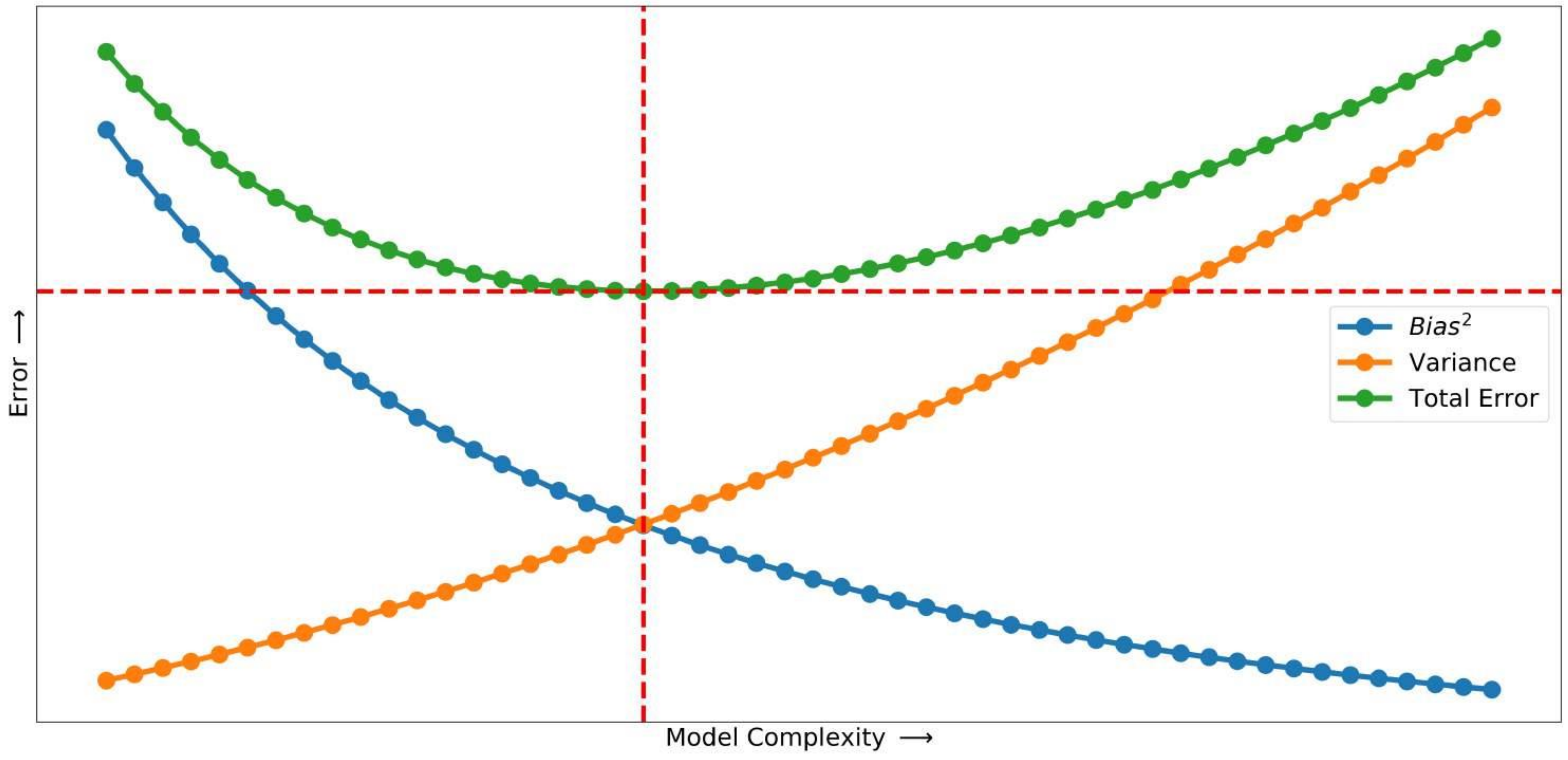
Assuming the true function is y and the learned function approximation is $\hat{y}$ the error $\mathcal{L}$ for a given input x is:

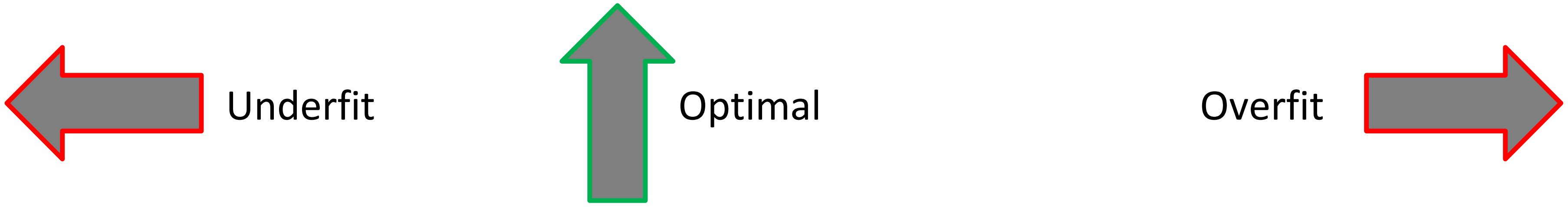
$$\mathcal{L}(x) = E[(\hat{y} - y)^2]$$

The error can be decomposed to:

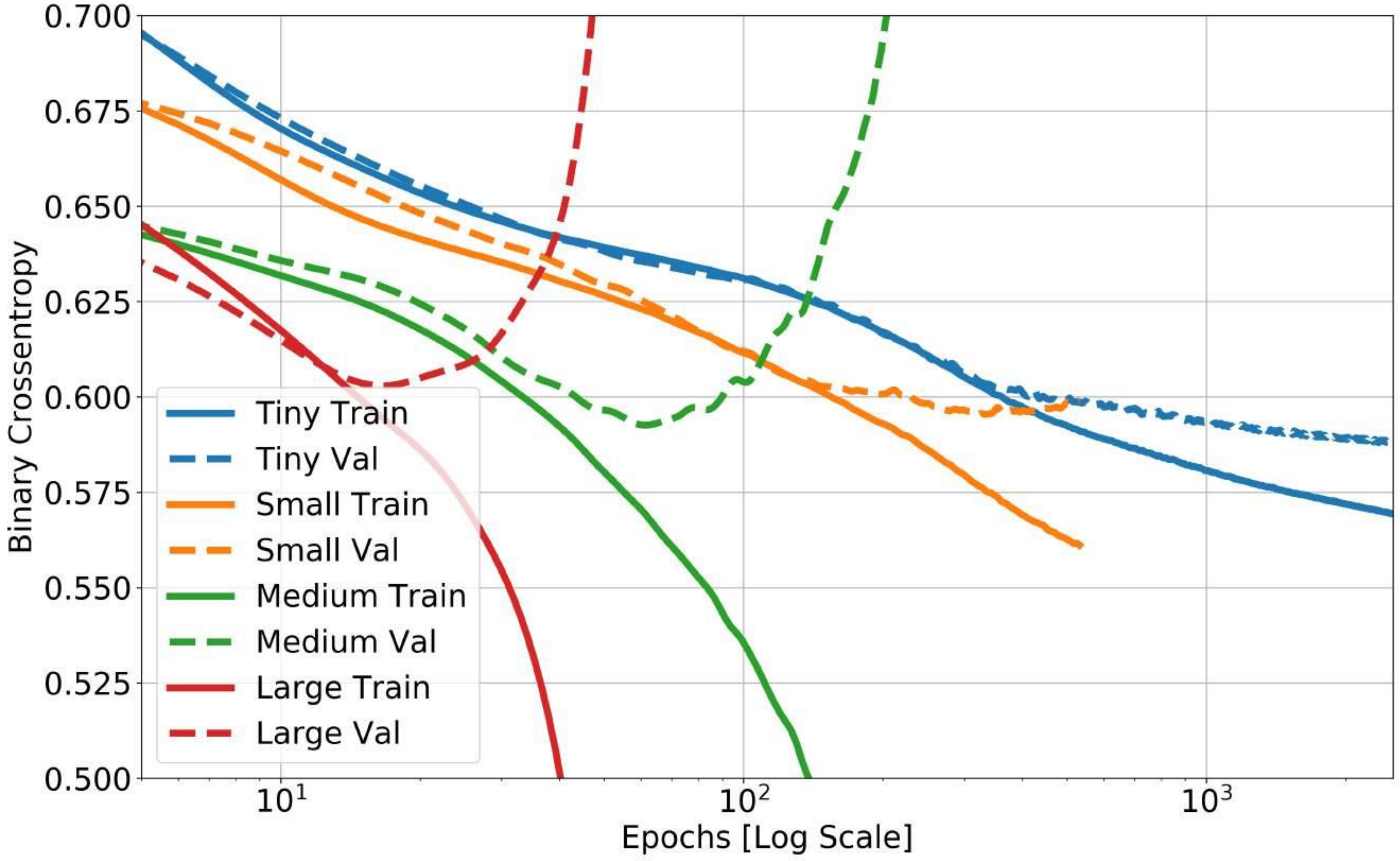
$$\mathcal{L}(x) = Bias^2 + Var + Irreducible\ Error$$
$$Bias^2 = (E[\hat{y}] - y)^2$$
$$Var = E[(\hat{y} - E[\hat{y}])^2]$$
$$\sigma_e^2 = constant$$

Bias-Variance Tradeoff





Underfitting/Overfitting



Regularisation

Modify Loss function $\mathcal{L}$ to penalise complex models

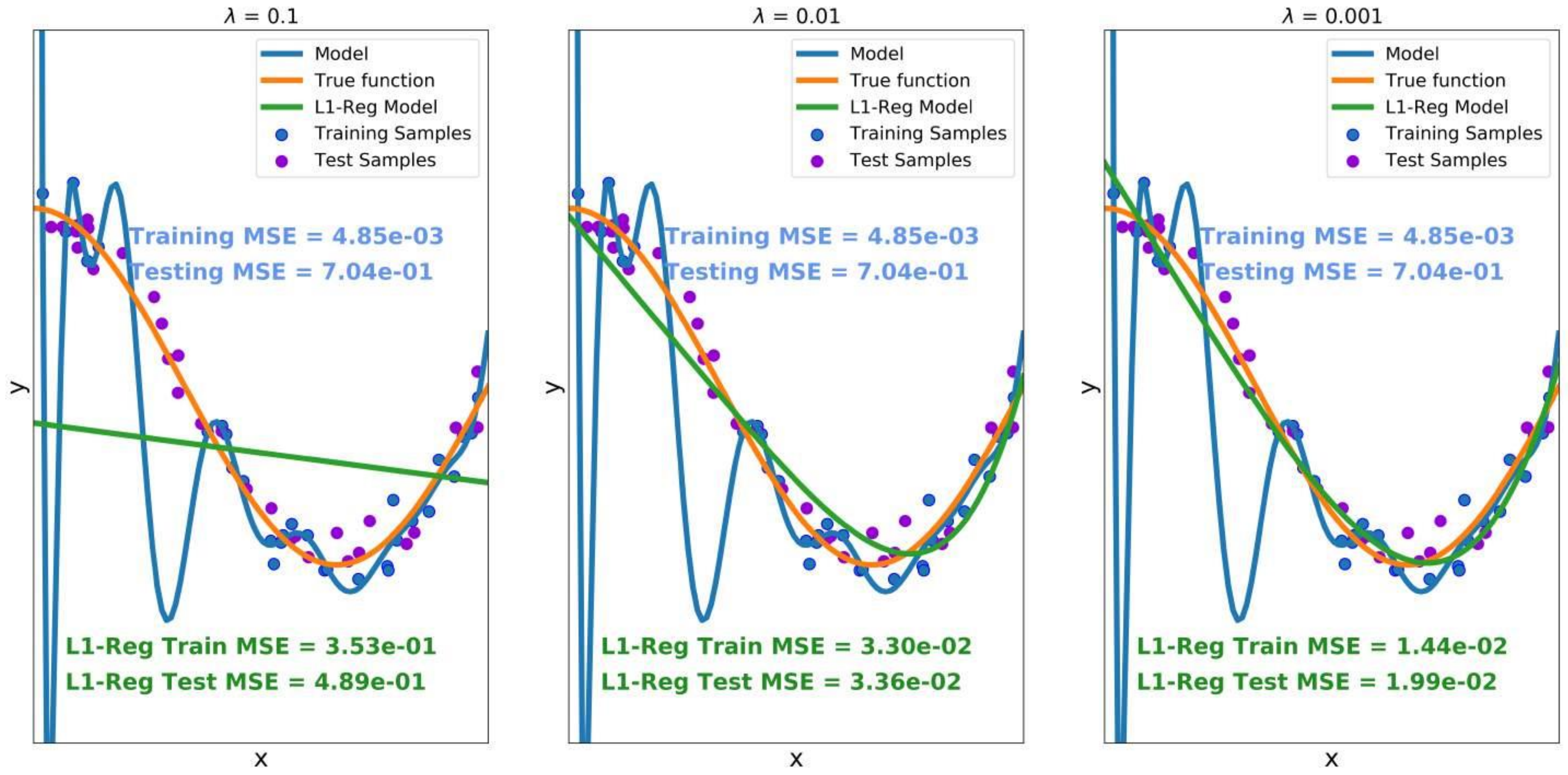
$$\mathcal{L}(x) = E\left[\left(f(x) - \hat{f}(x)\right)^2\right] + \lambda \|w\|_p^p$$

Where $\|w\|_{\ddot{Y}}$ is the p norm of the model parameters:

- $p = 1$, $L1$ norm: $\|w\|_1 = \sum_{j=1}^{p} |w_j|$
- $p = 2$, $L2$ norm: $\|w\|_2 = \sqrt{\sum_{j=1}^{p} {w_j}^2}$

and λ the regularisation coefficient for $\lambda \rightarrow 0$ the loss function reduces to no regularisation and for $\lambda \rightarrow \infty$ error becomes large and all the weights will approach zero

Effect of λ



How to select λ?

- λ is another hyperparameter
- To find the best λ run multiple iteration with random train/test splits and select the λ that minimises the variance

Weight Decay

Update rule for SGD:

$$w_{t+1} = w_t - \eta \frac{\partial L}{\partial w} - \eta \lambda w_t$$

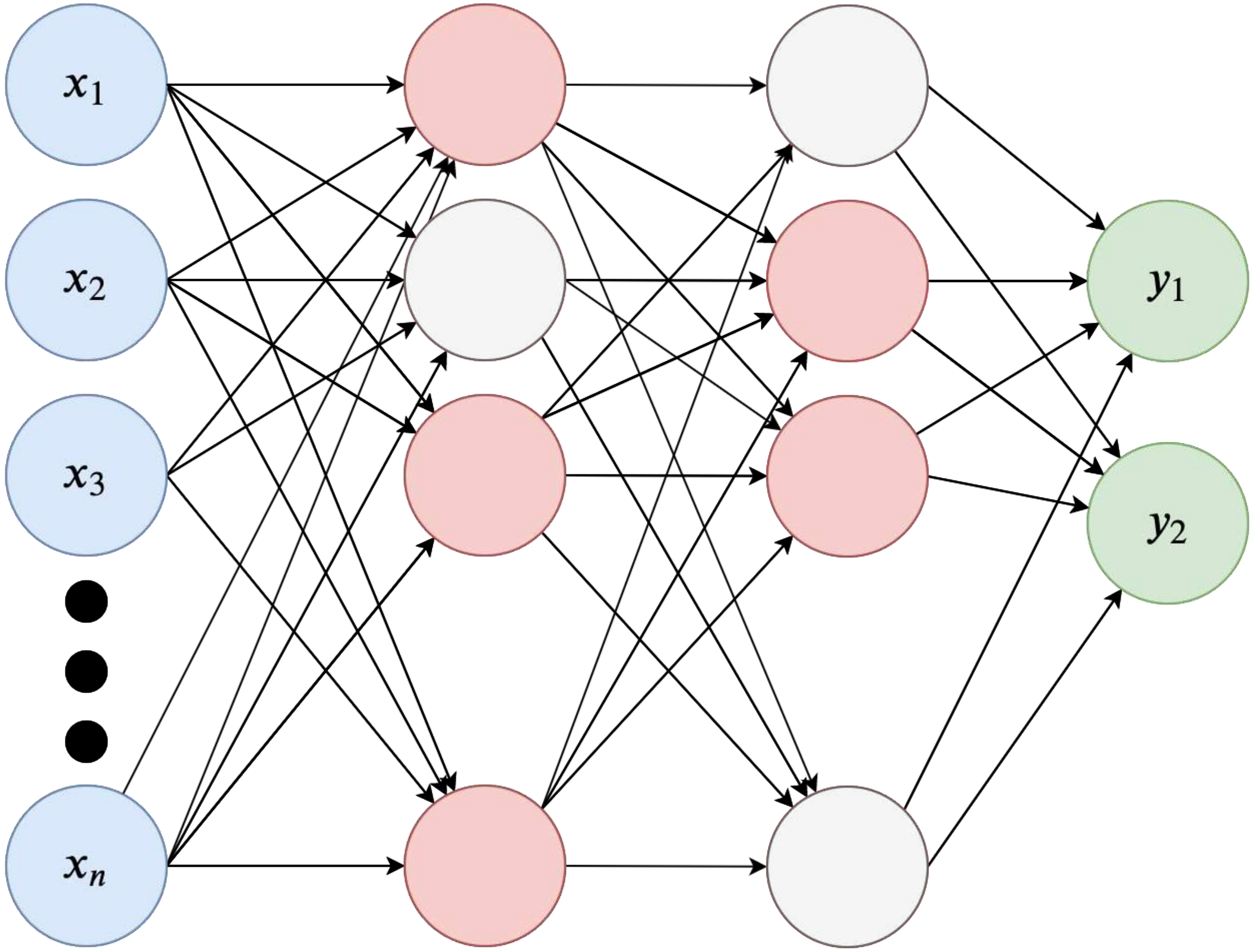
Every update we subtract $\eta \lambda w_t$ and we decay the weights

Equivalent to L2 Regularisation (for vanilla SGD)

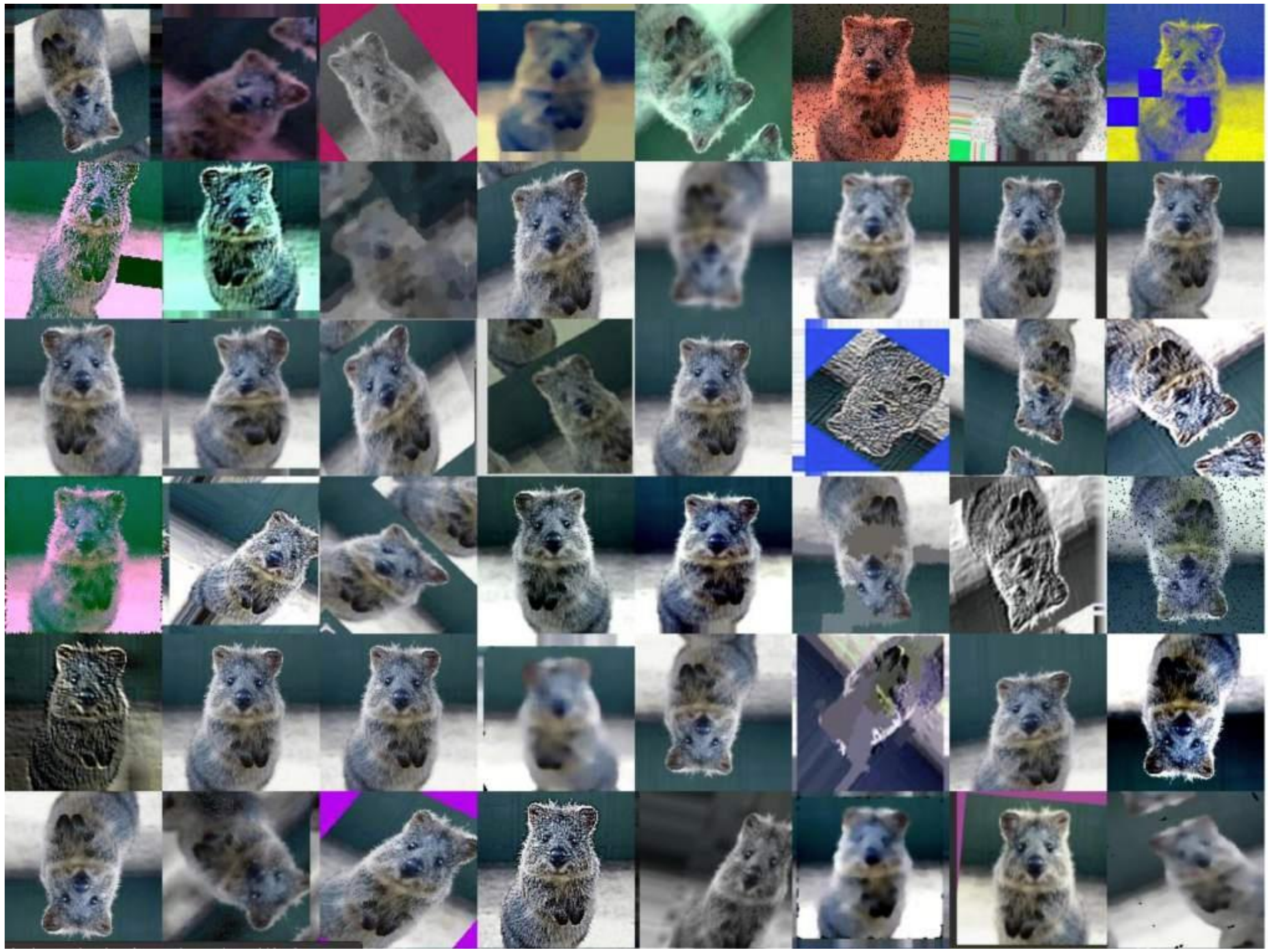
$$\mathcal{L}' = \mathcal{L} + \frac{\lambda}{2} \|w\|_2^2$$

(Note instead of λ we use $\frac{\lambda}{2}$ to make math easier)

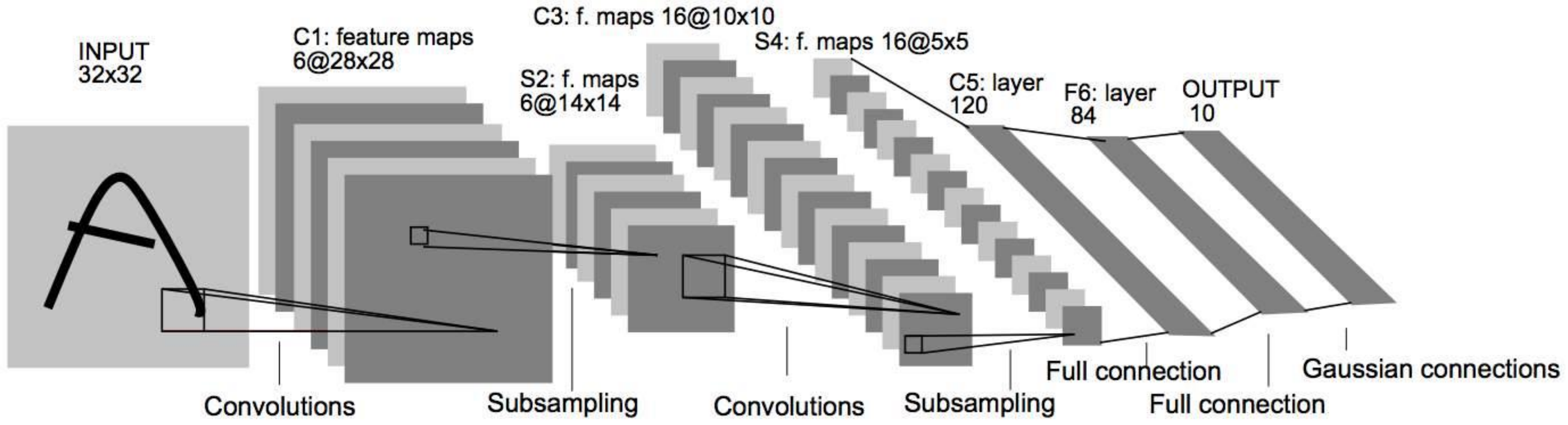
Dropout



Data Augmentation



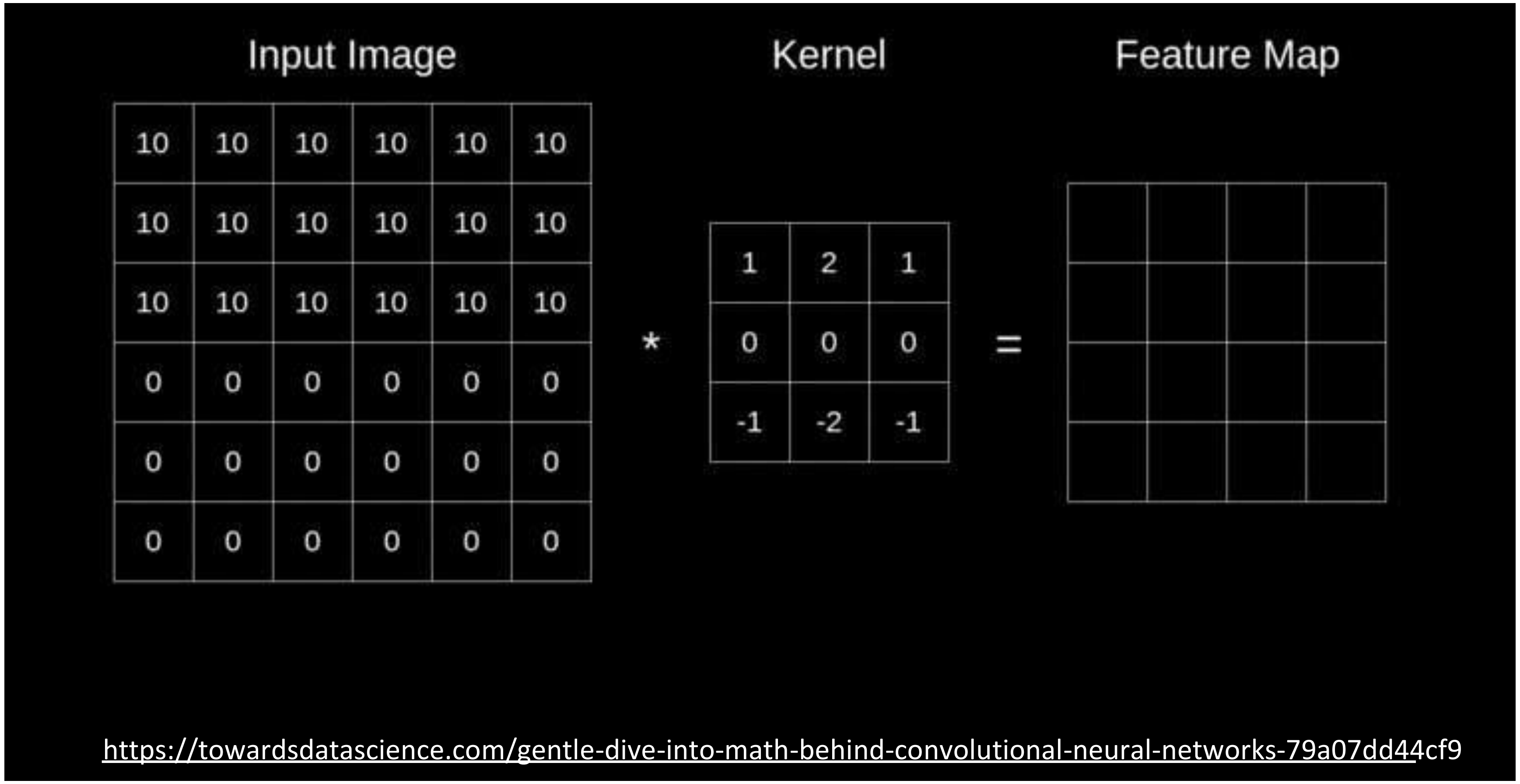
LeNet-5



INPUT 32x32
C1: feature maps 6@28x28
S2: f. maps 6@14x14
C3: f. maps 16@10x10
S4: f. maps 16@5x5
C5: layer 120
F6: layer 84
OUTPUT 10
Convolutions
Subsampling
Convolutions
Subsampling
Full connection
Full connection
Gaussian connections

2D Convolution

$$G[m,n] = (f * g)[m,n] = \sum_{j}\sum_{k} f[m-j, n-k]\,g[j,k]$$



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

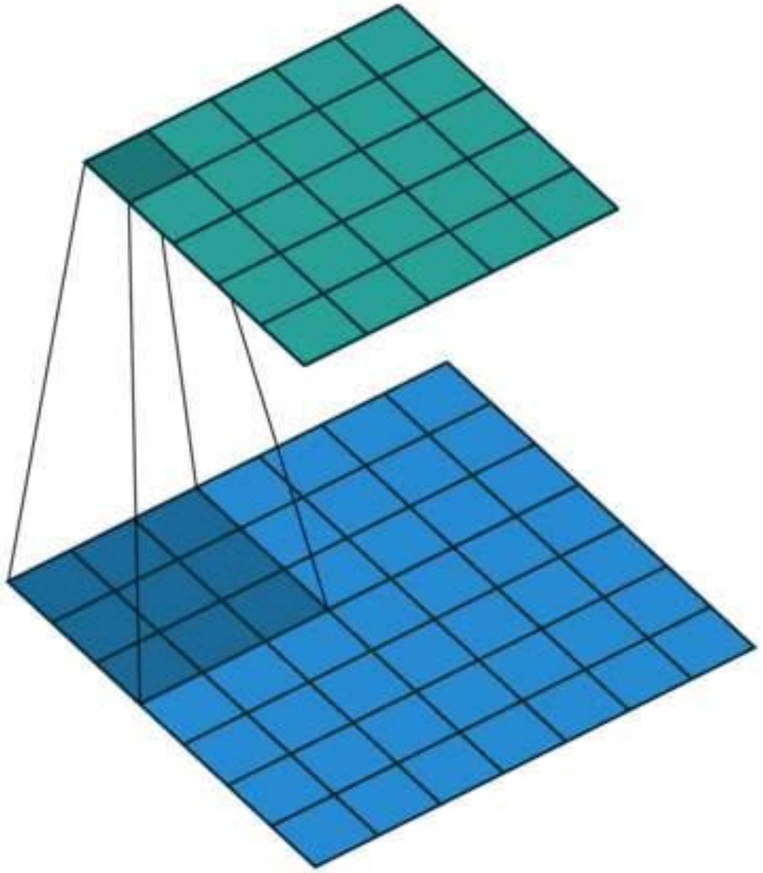
1	2	1
0	0	0
-1	-2	-1

https://towardsdatascience.com/gentle-dive-into-math-behind-convolutional-neural-networks-79a07dd44cf9

2D Convolution Examples for different Kernels

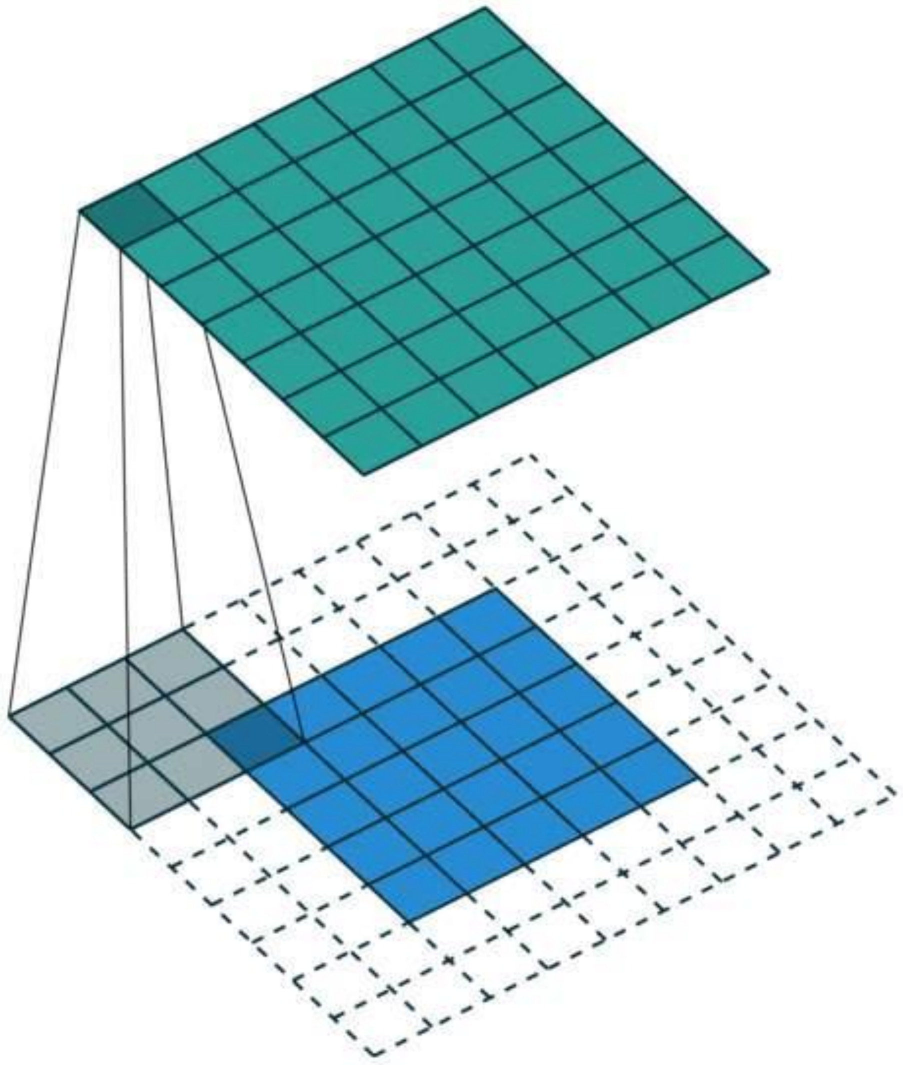
https://towardsdatascience.com/gentle-dive-into-math-behind-convolutional-neural-networks-79a07dd44cf9

2D Convolution

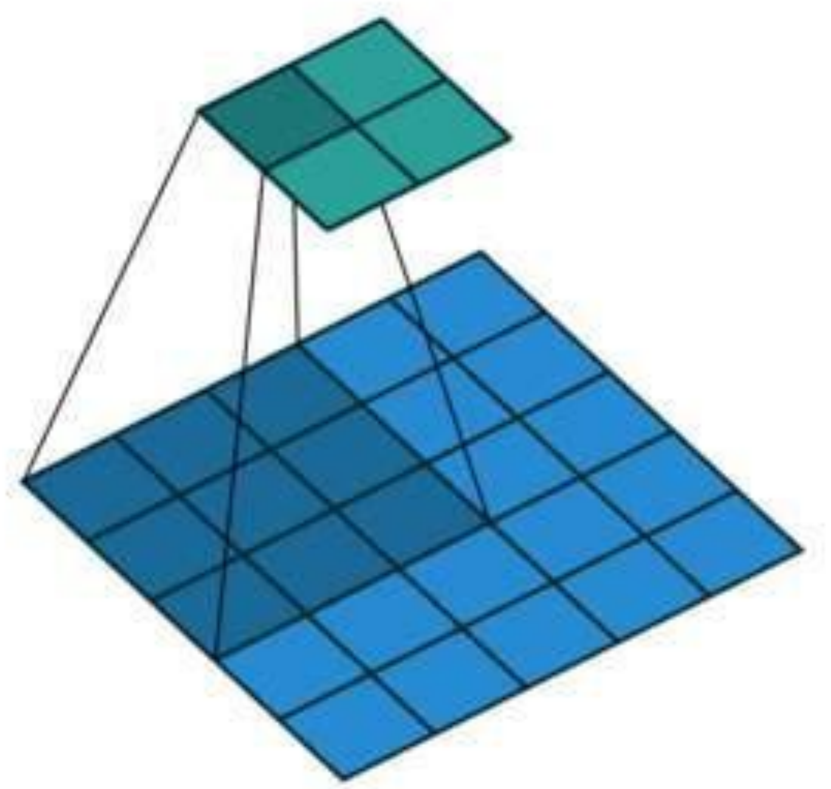


https://github.com/vdumoulin/conv_arithmetic

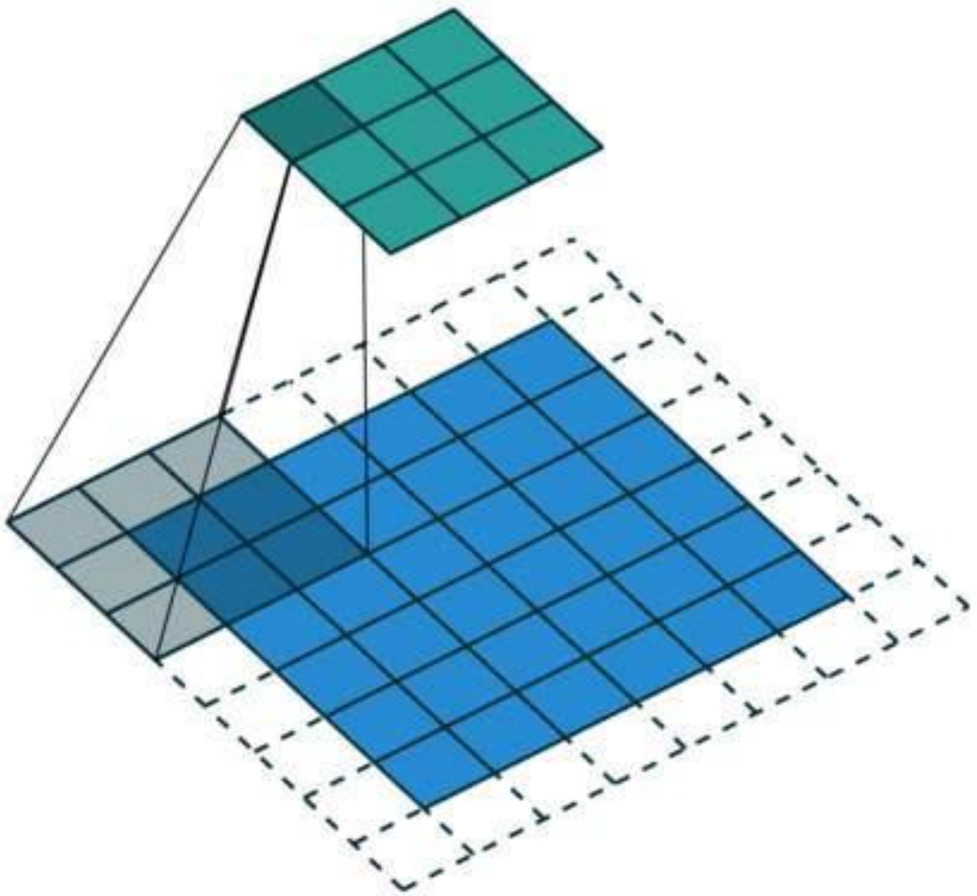
2D Convolution Padding



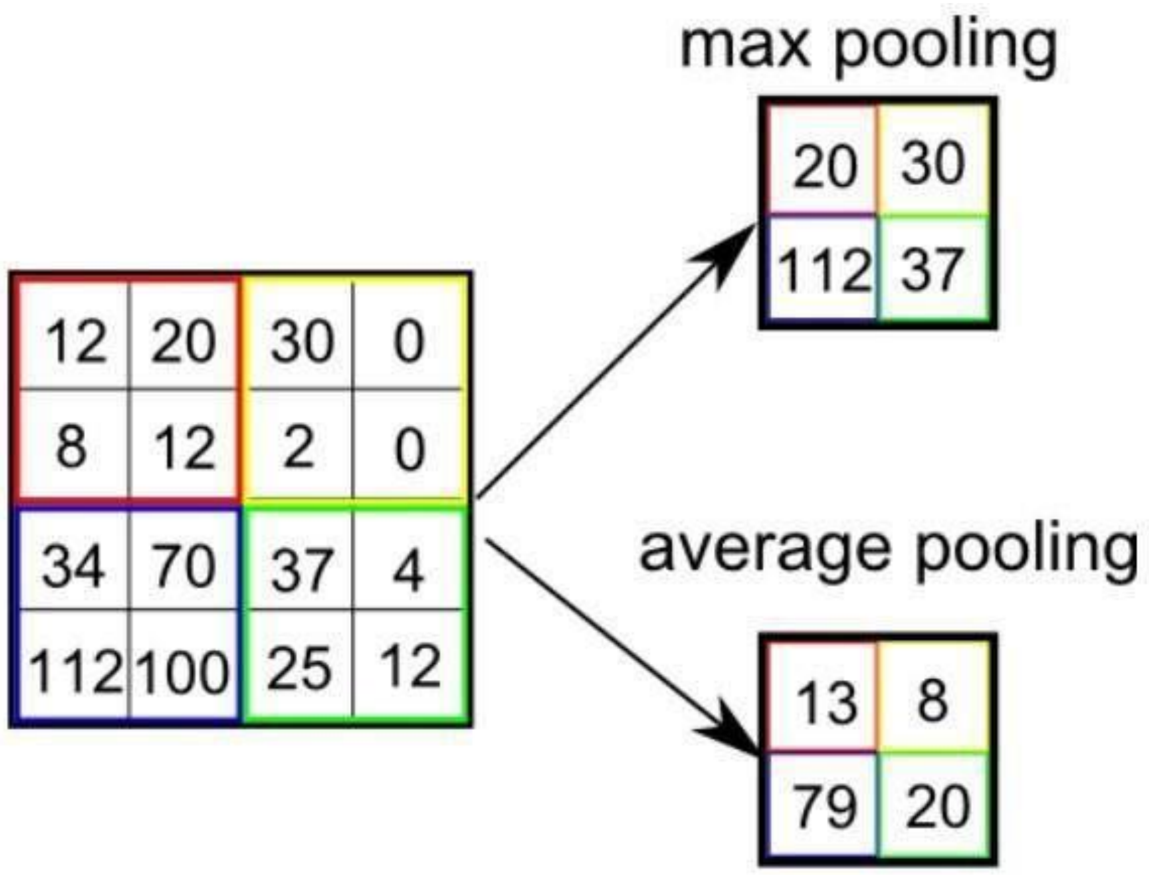
2D Convolution with Stride



Combination of Strides and Padding

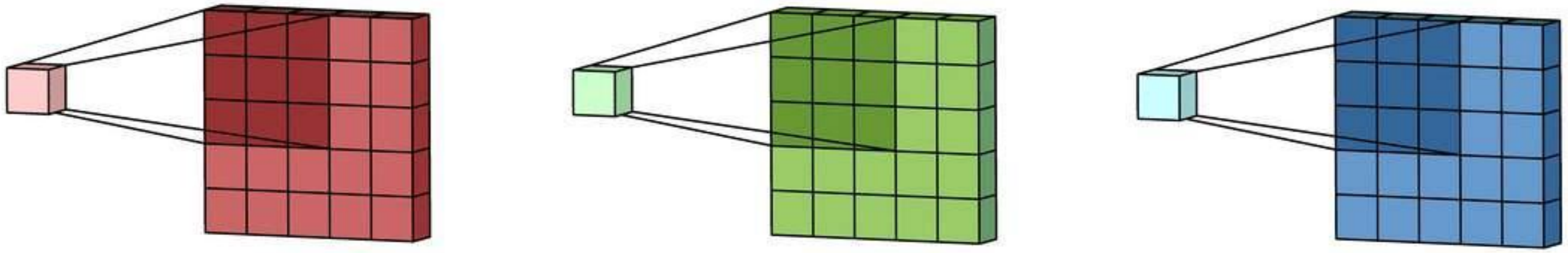


Pooling

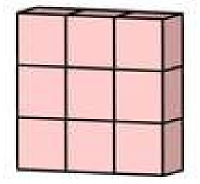


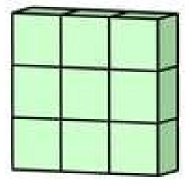
max pooling
average pooling
12
20
30
0
8
12
2
0
34
70
37
4
112
100
25
12
20
30
112
37
13
8
79
20

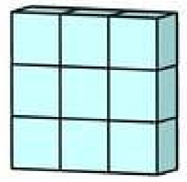
Convolutions on RGB Image

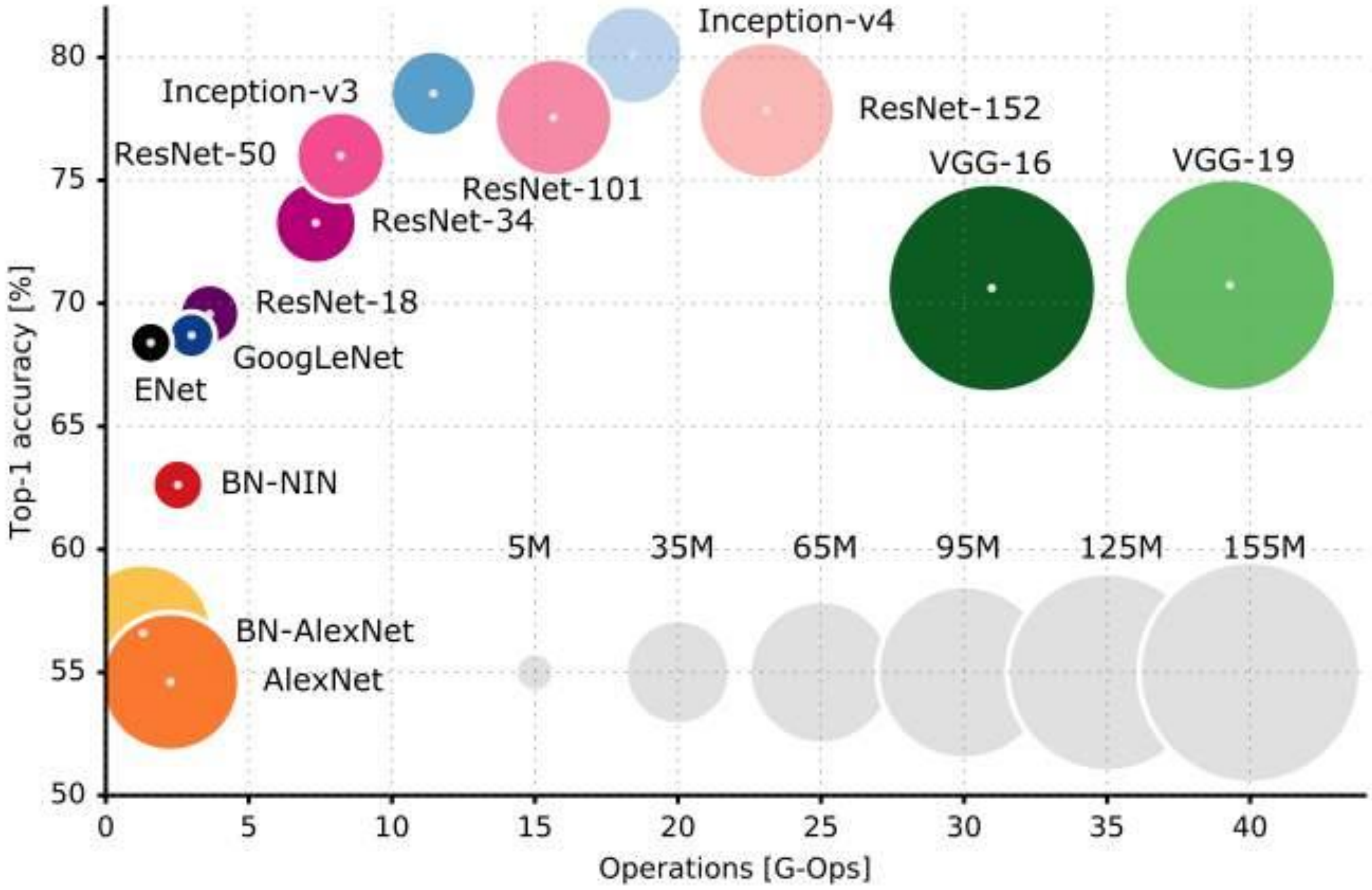


Convolutions on RGB Image

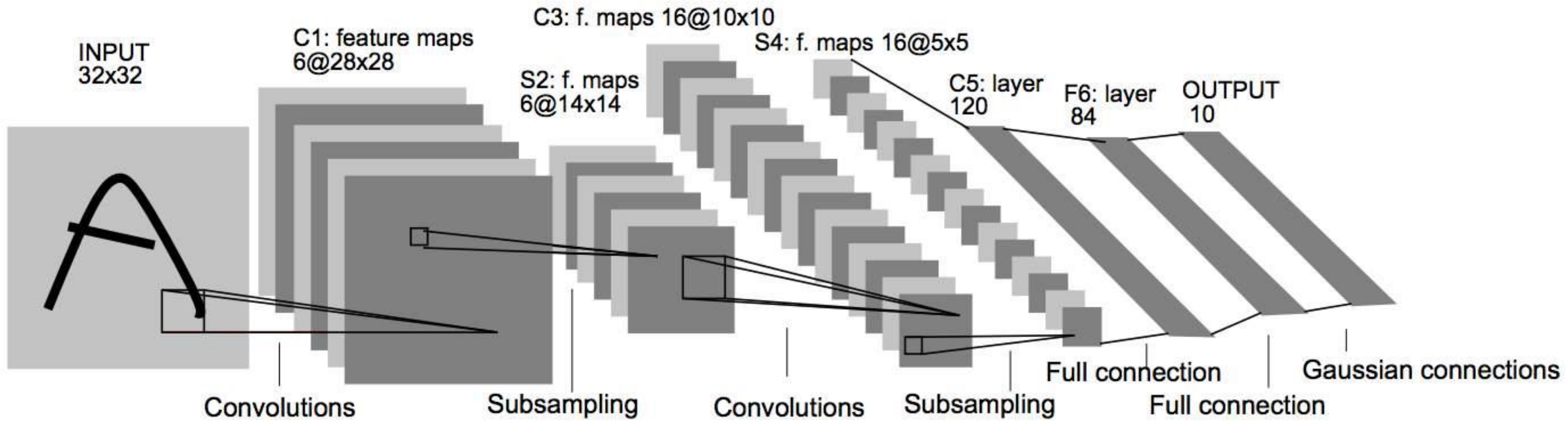




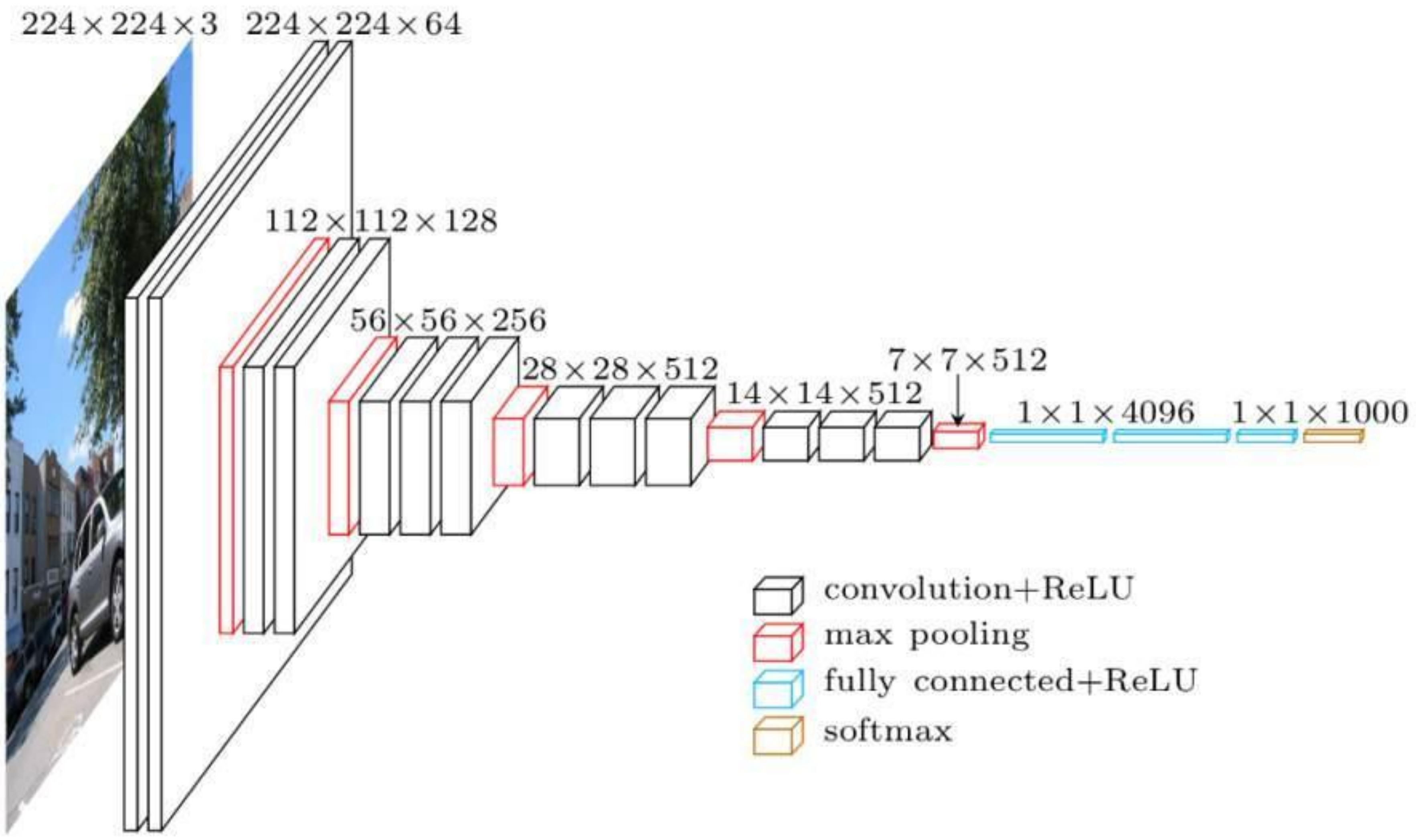



Inception-v4
Inception-v3
ResNet-152
ResNet-50
ResNet-101
VGG-16
VGG-19
ResNet-34
ResNet-18
GoogLeNet
ENet
BN-NIN
BN-AlexNet
AlexNet
5M
35M
65M
95M
125M
155M
Top-1 accuracy [%]
Operations [G-Ops]
50
55
60
65
70
75
80
0
5
10
15
20
25
30
35
40

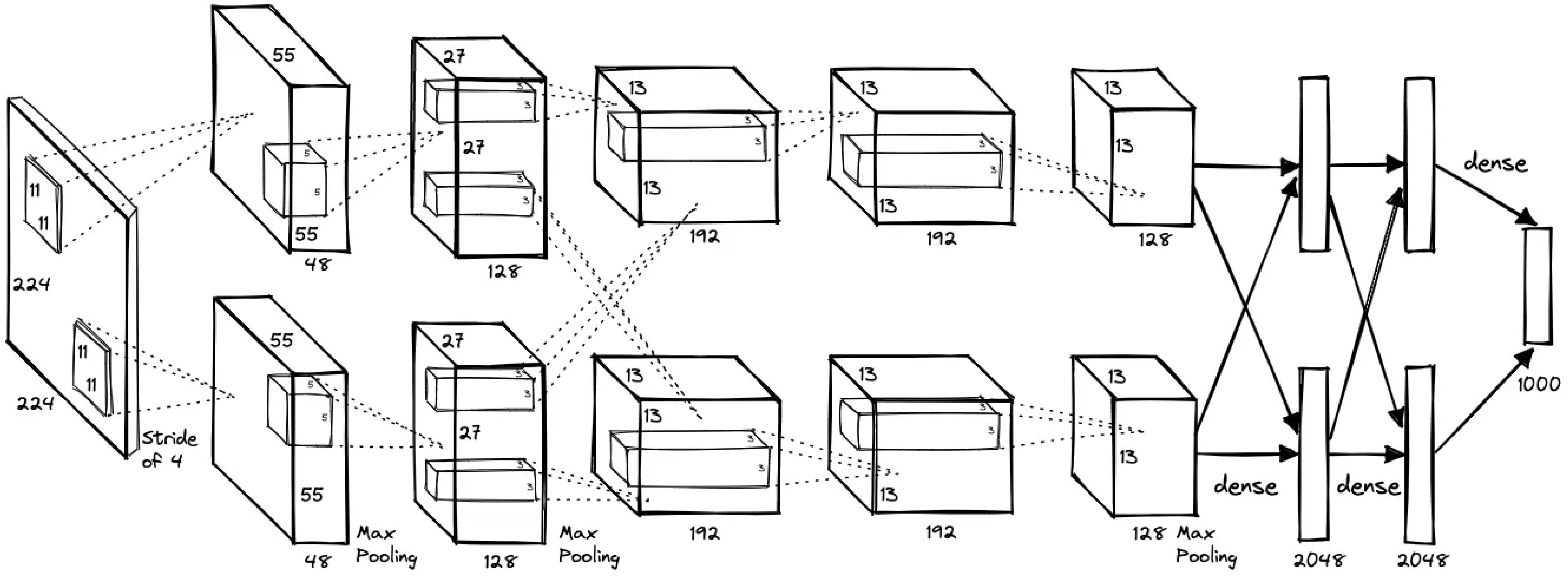
Convolutional Neural Networks (LeNet-5)



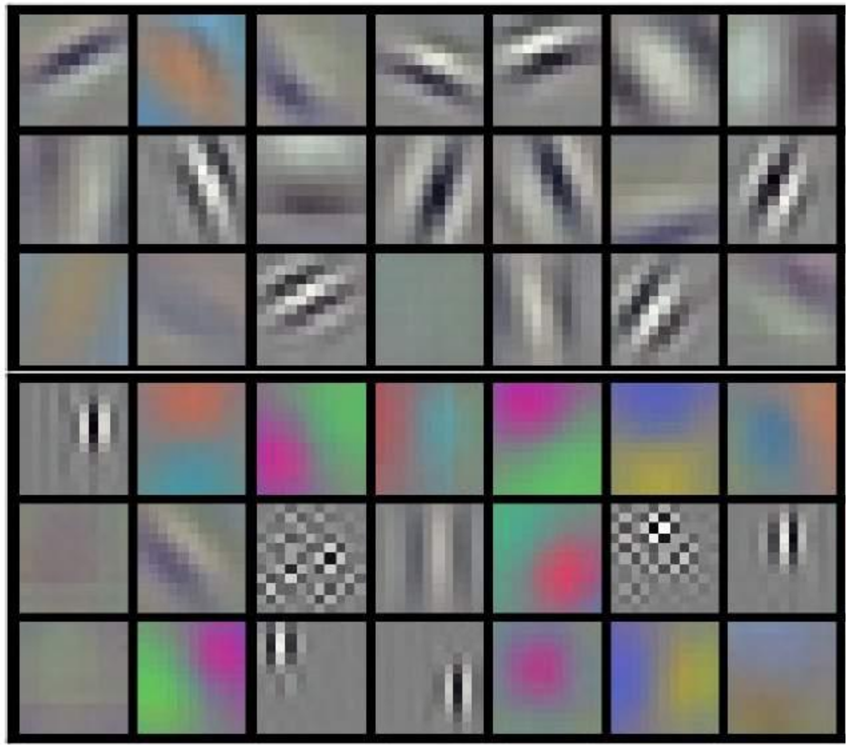
VGG-16

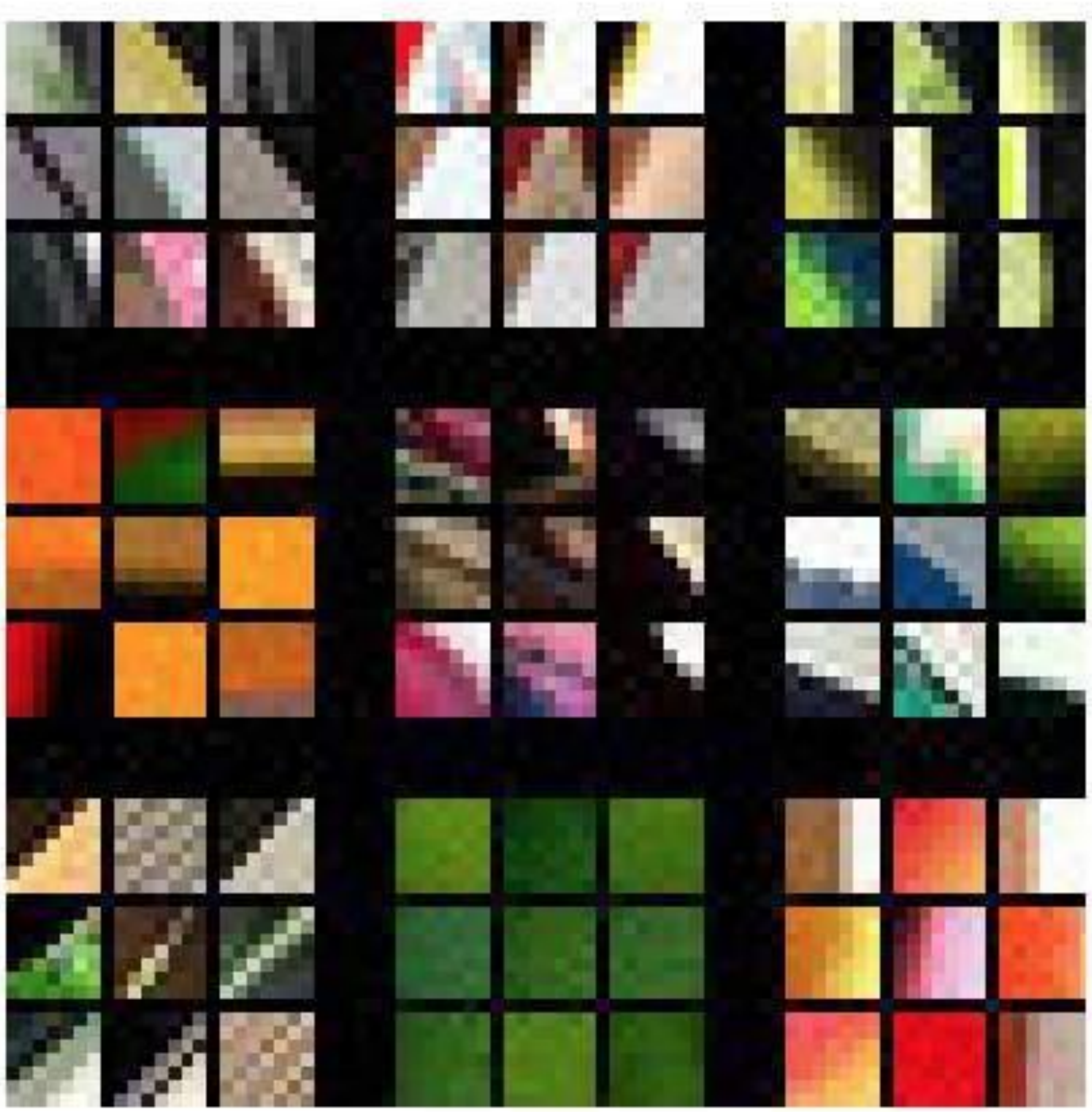

224 × 224 × 3
224 × 224 × 64
112 × 112 × 128
56 × 56 × 256
28 × 28 × 512
14 × 14 × 512
7 × 7 × 512
1 × 1 × 4096
1 × 1 × 1000
convolution+ReLU
max pooling
fully connected+ReLU
softmax

AlexNet

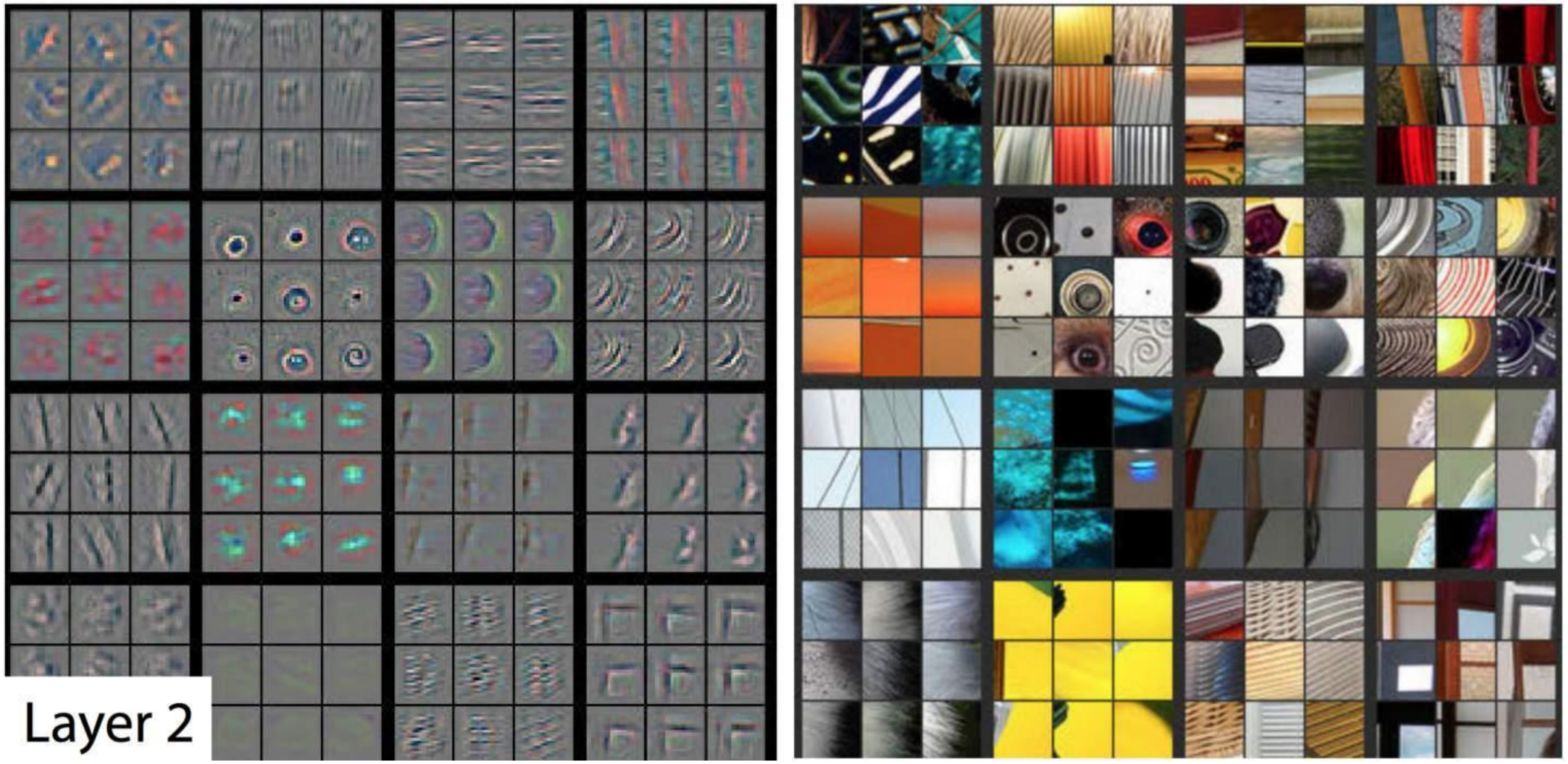

224
224
11
11
Stride of 4
55
55
48
5
27
27
128
3
Max Pooling
13
13
192
128
dense
2048
1000

Layer 1

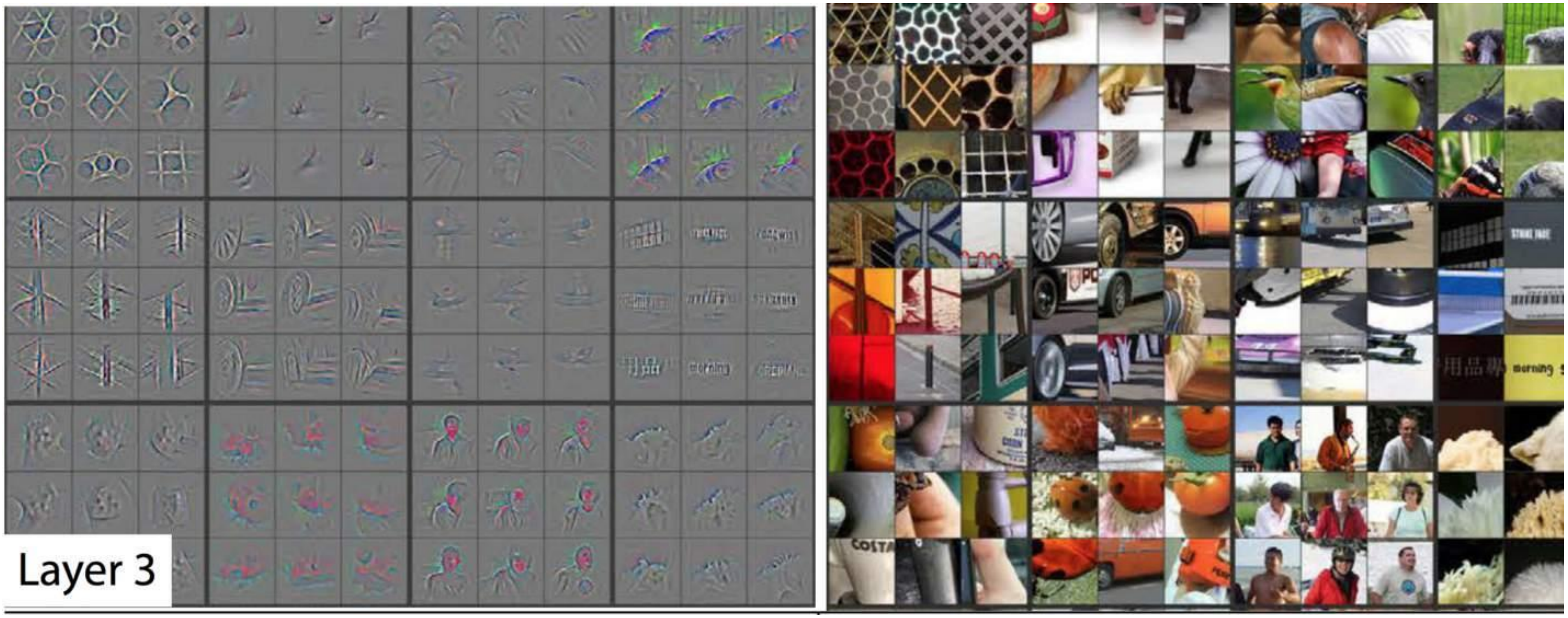




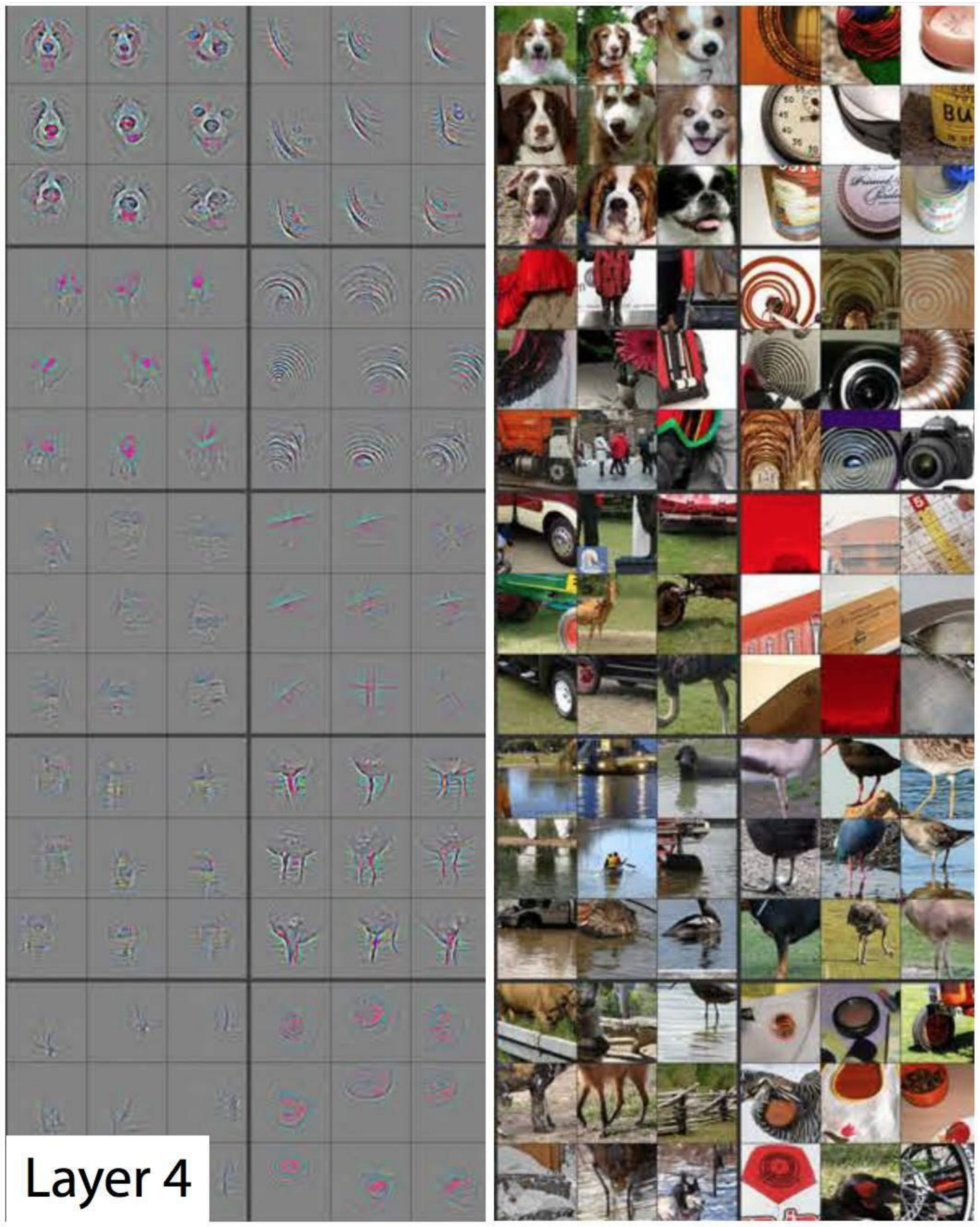
Layer 2



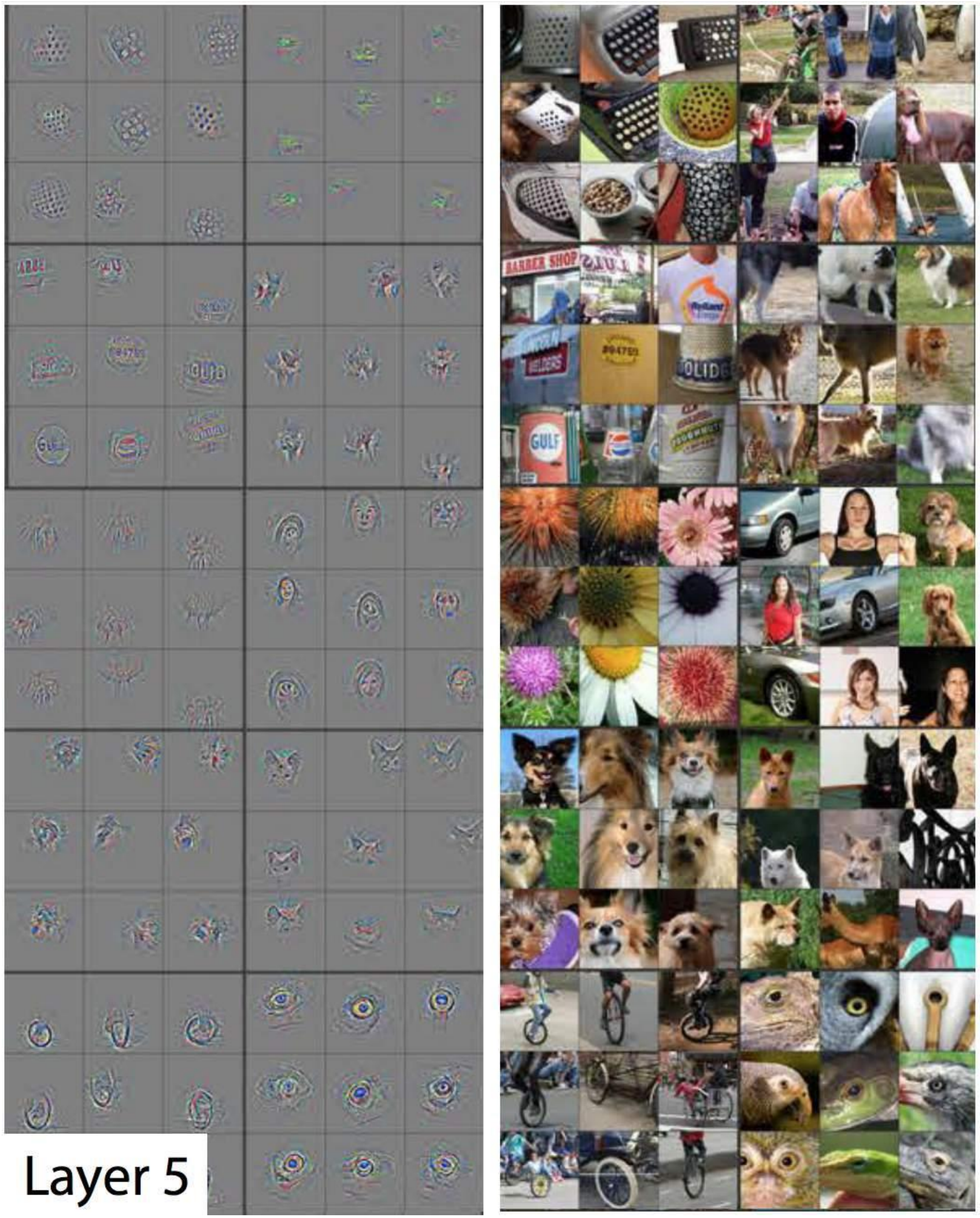
Layer 3



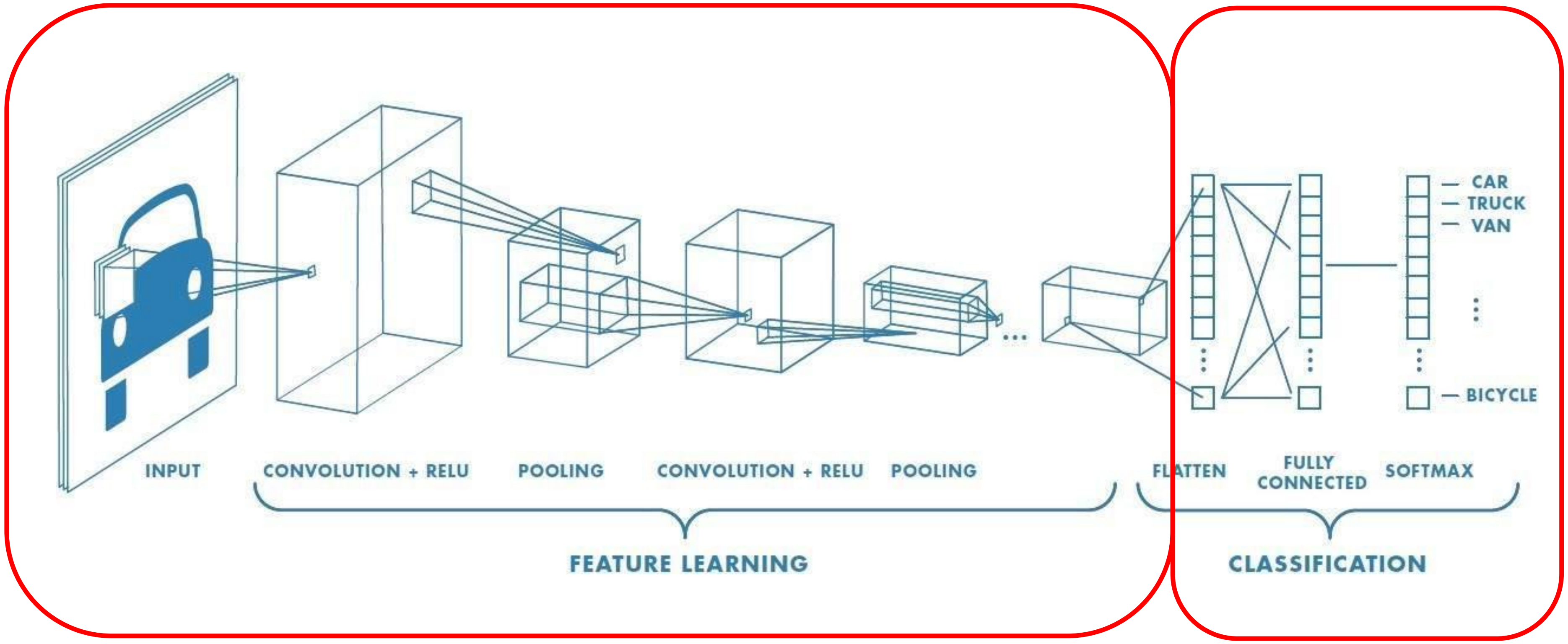
Layer 4



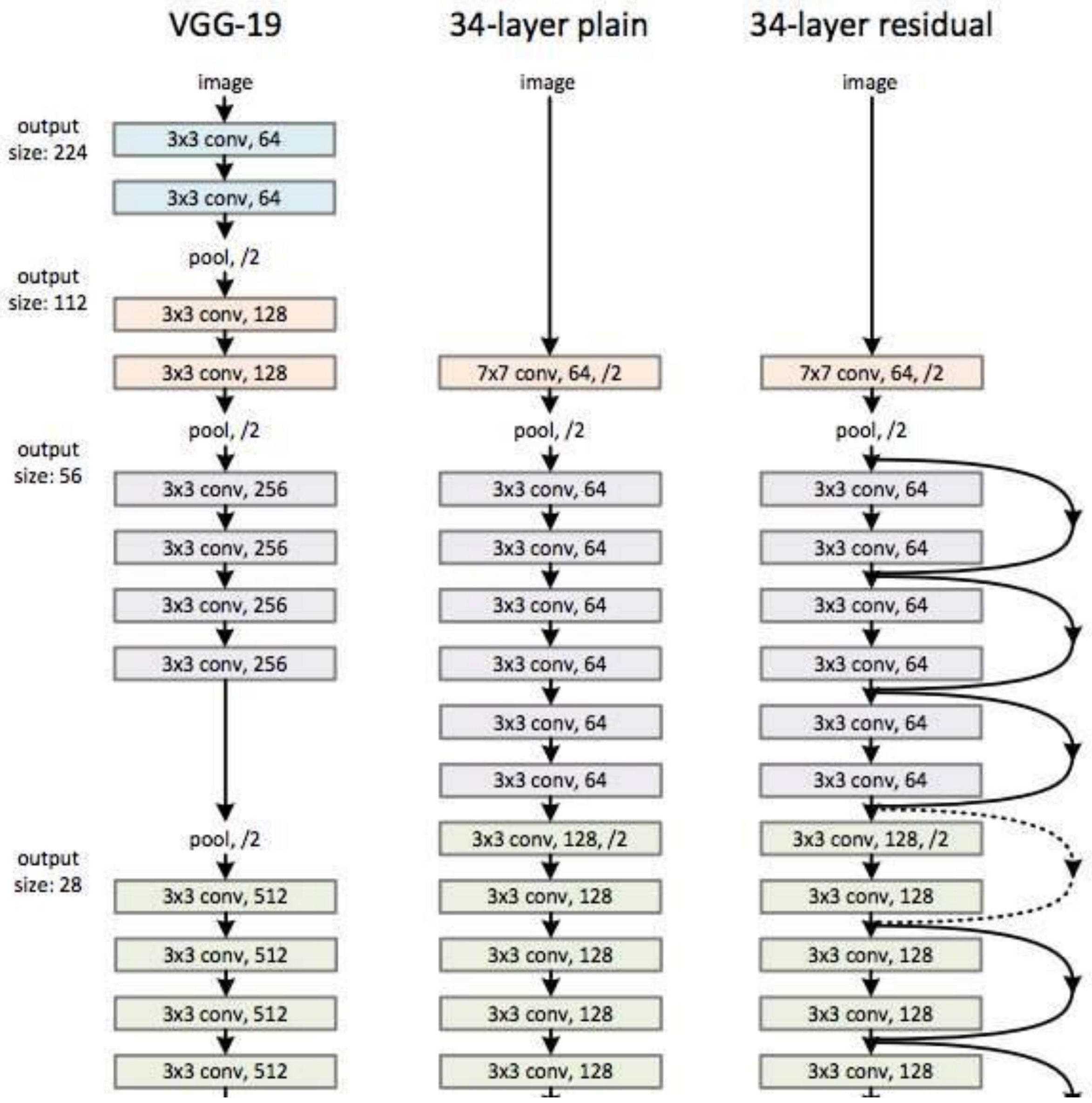
Layer 5



Transfer Learning



ResNet



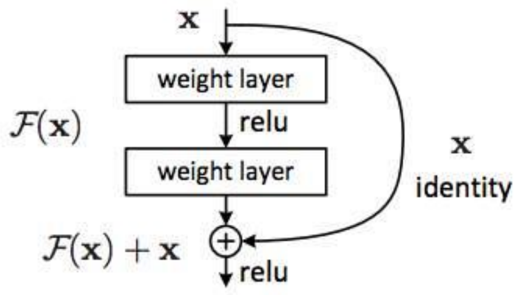
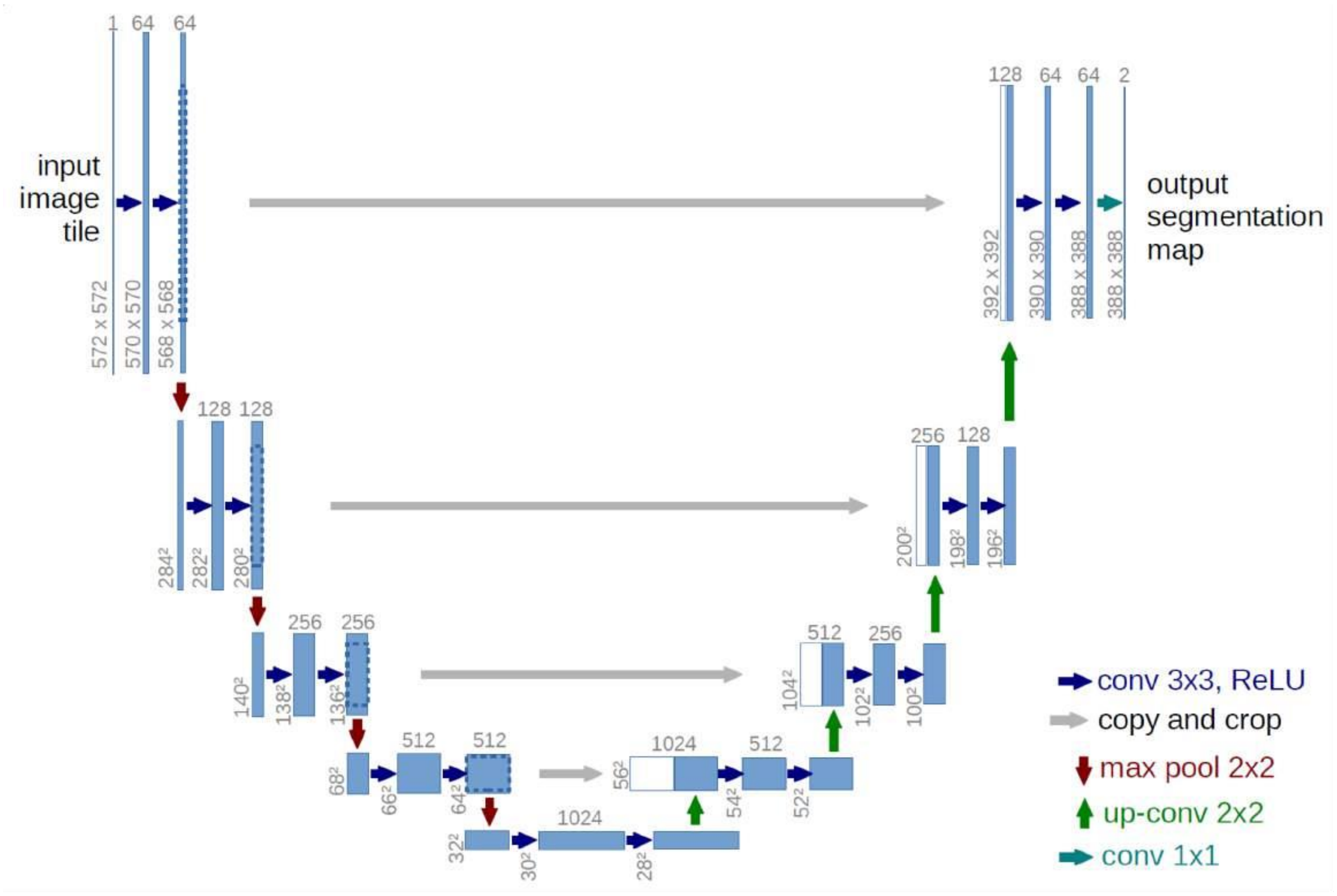


Figure 2. Residual learning: a building block.

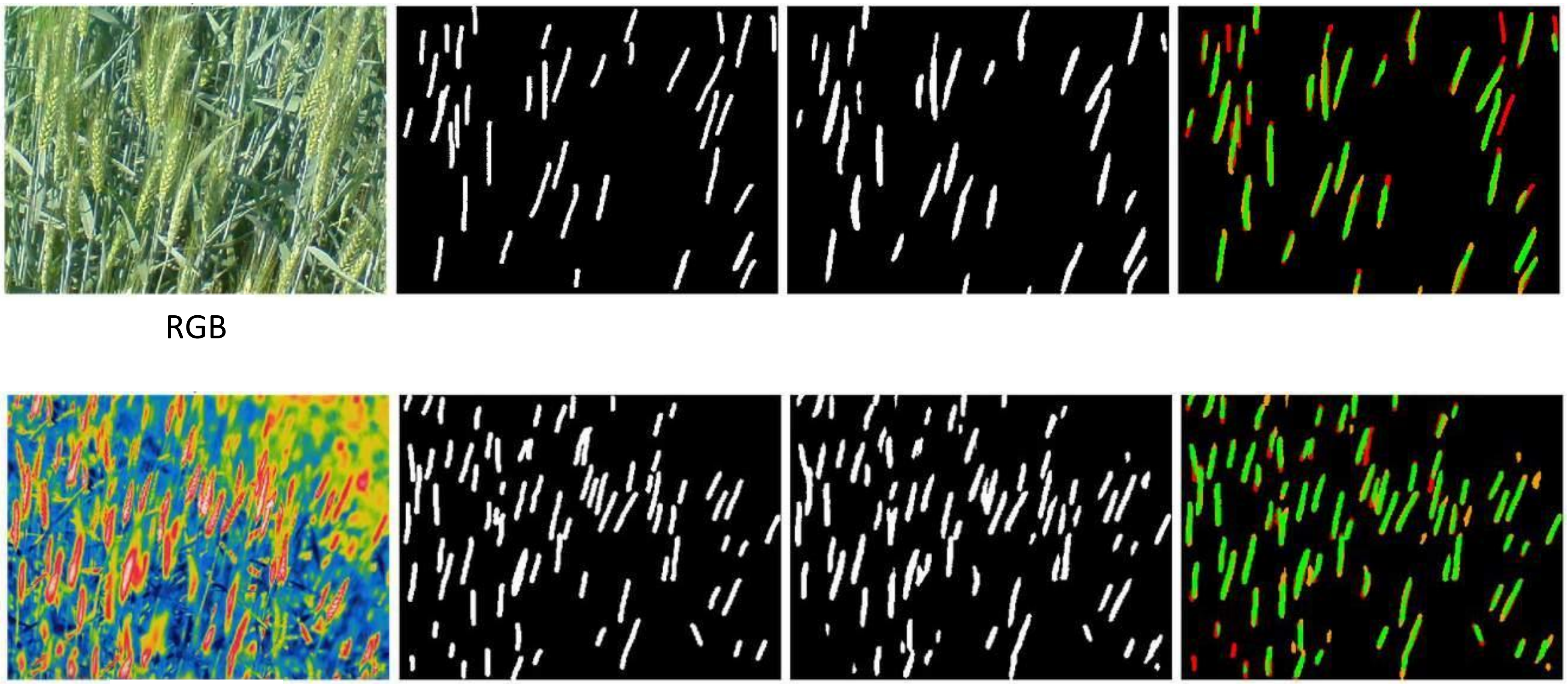
UNet



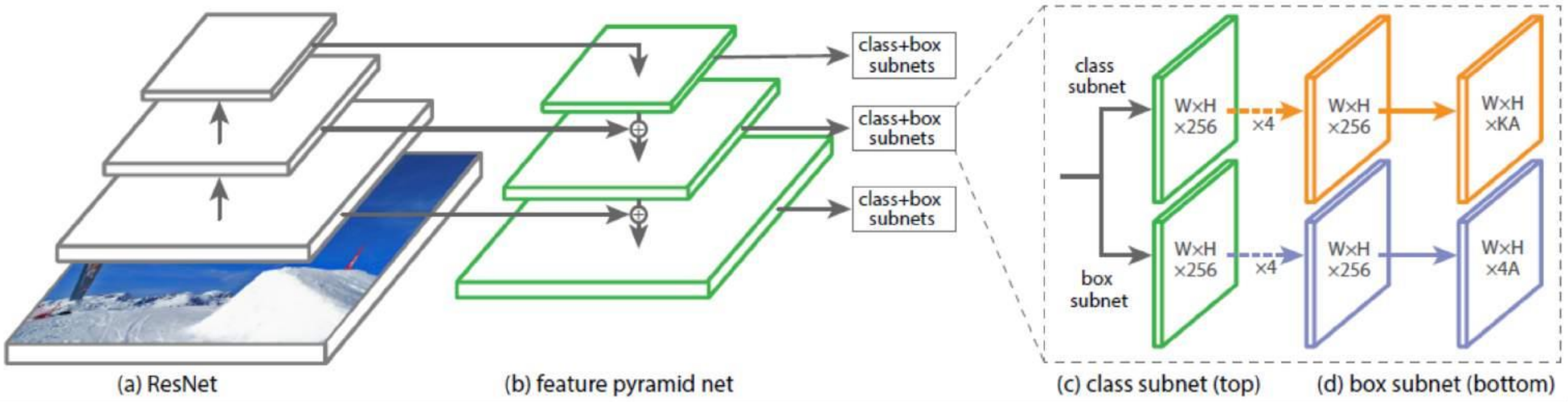
1
64
64
input image tile
572 x 572
570 x 570
568 x 568
128
128
284²
282²
280²
256
256
140²
138²
136²
512
512
68²
66²
64²
1024
32²
30²
28²
1024
512
56²
54²
52²
512
256
104²
102²
100²
256
128
200²
198²
196²
128
64
64
2
392 x 392
390 x 390
388 x 388
388 x 388
output segmentation map
conv 3x3, ReLU
copy and crop
max pool 2x2
up-conv 2x2
conv 1x1

HPC-Enabled Precision Agriculture

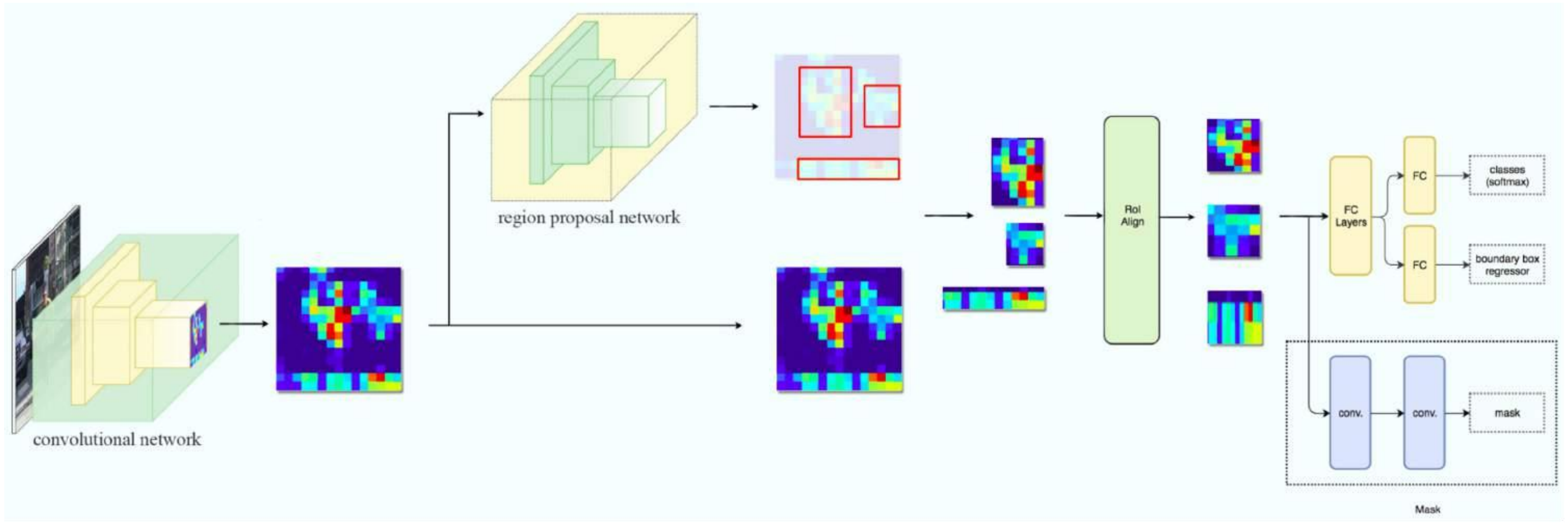
Automatic counting of wheat ears



RetinaNet


class+box subnets
class+box subnets
class+box subnets
class subnet
box subnet
W×H ×256
×4
W×H ×256
W×H ×KA
W×H ×256
×4
W×H ×256
W×H ×4A
(a) ResNet
(b) feature pyramid net
(c) class subnet (top)
(d) box subnet (bottom)

Mask-RCNN


region proposal network
convolutional network
RoI Align
FC Layers
FC
FC
classes (softmax)
boundary box regressor
conv.
conv.
mask
Mask