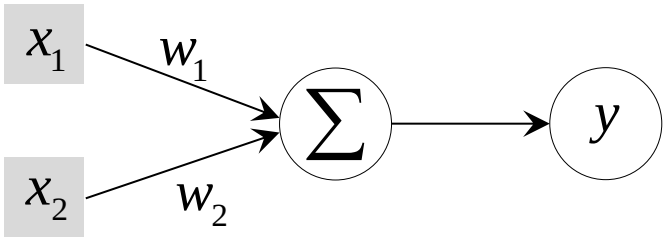
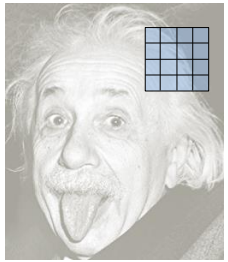

Convolutional Neural Network
Hyun Soo Park

RECALL: MULTIPLICATION LAYER

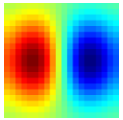


Convolutional Layer



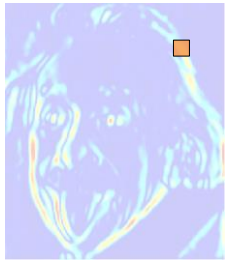
$H \times W$

$*$



$F \times F$

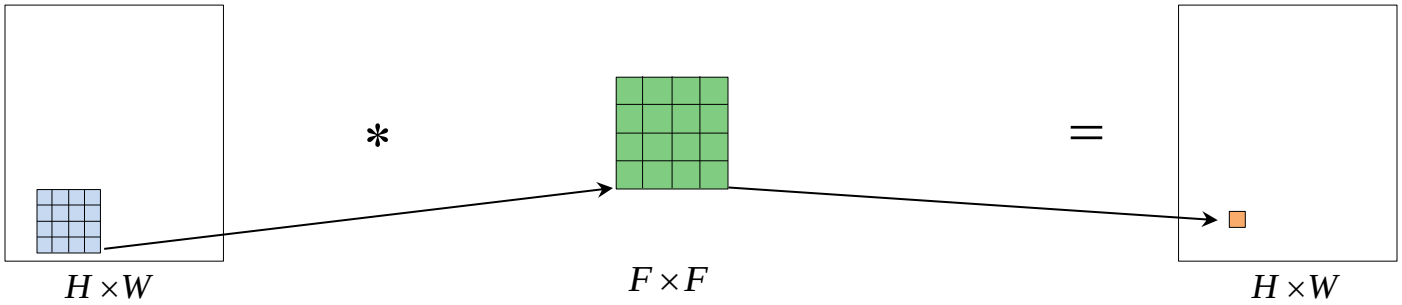
$=$



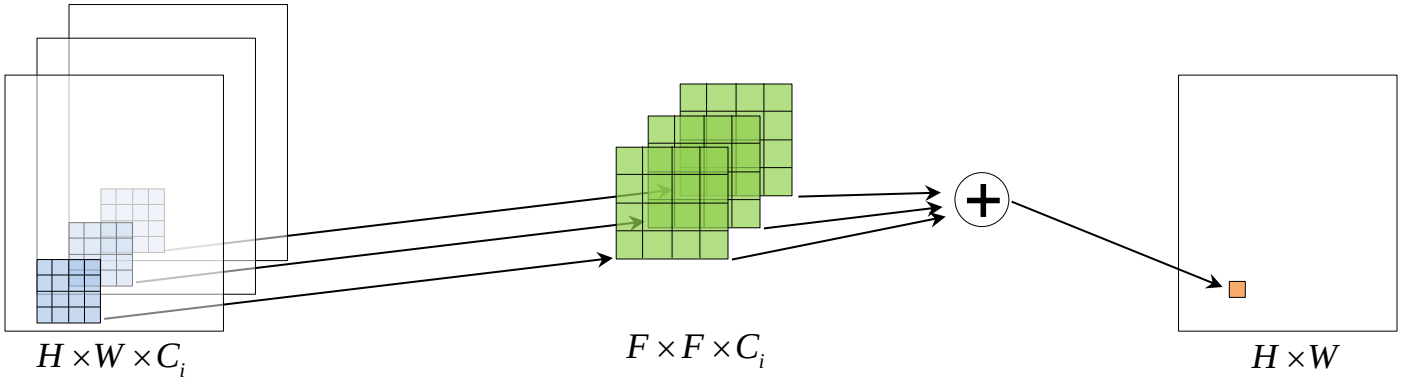
$H \times W$

Feature map

Convolutional Layer

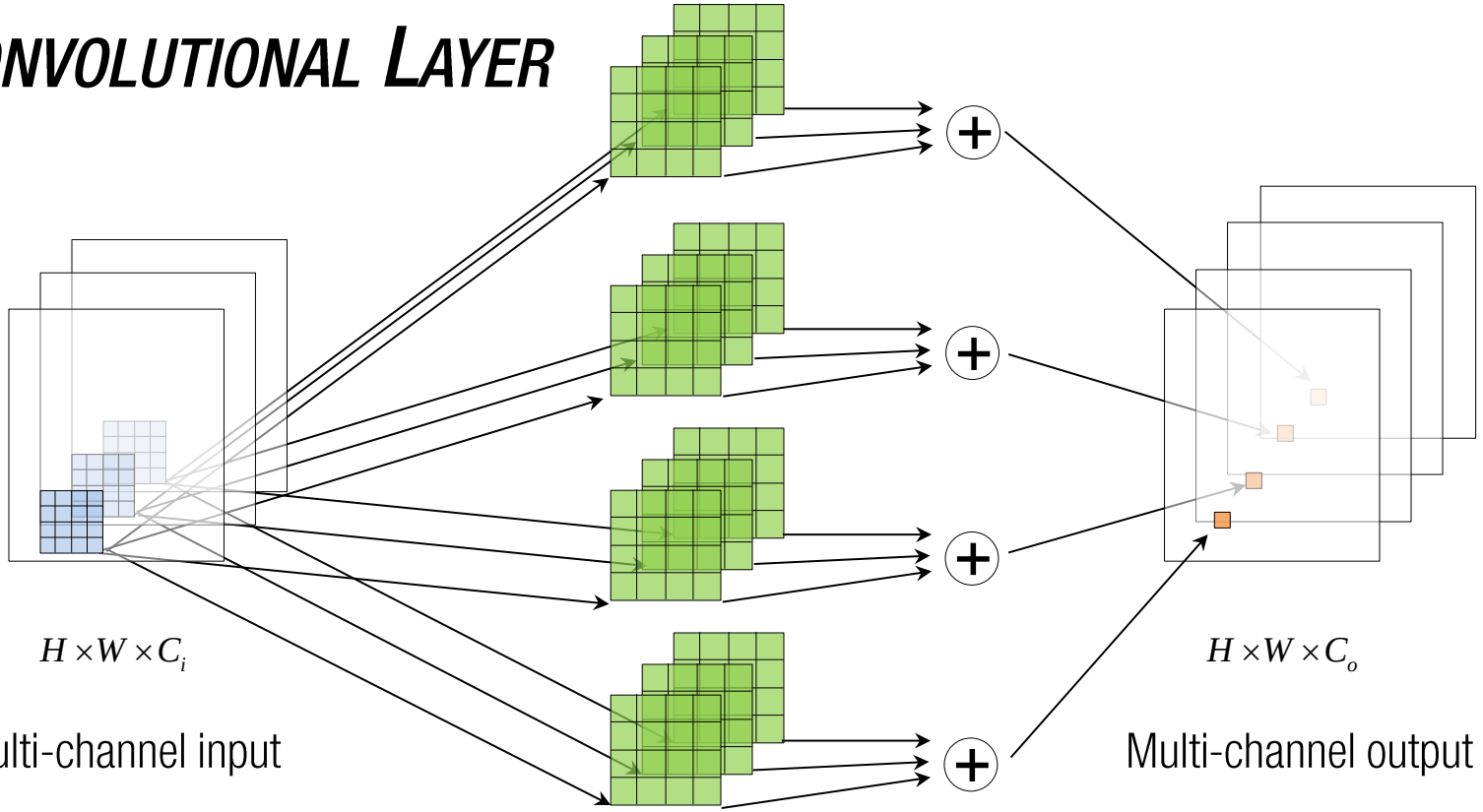


CONVOLUTIONAL LAYER

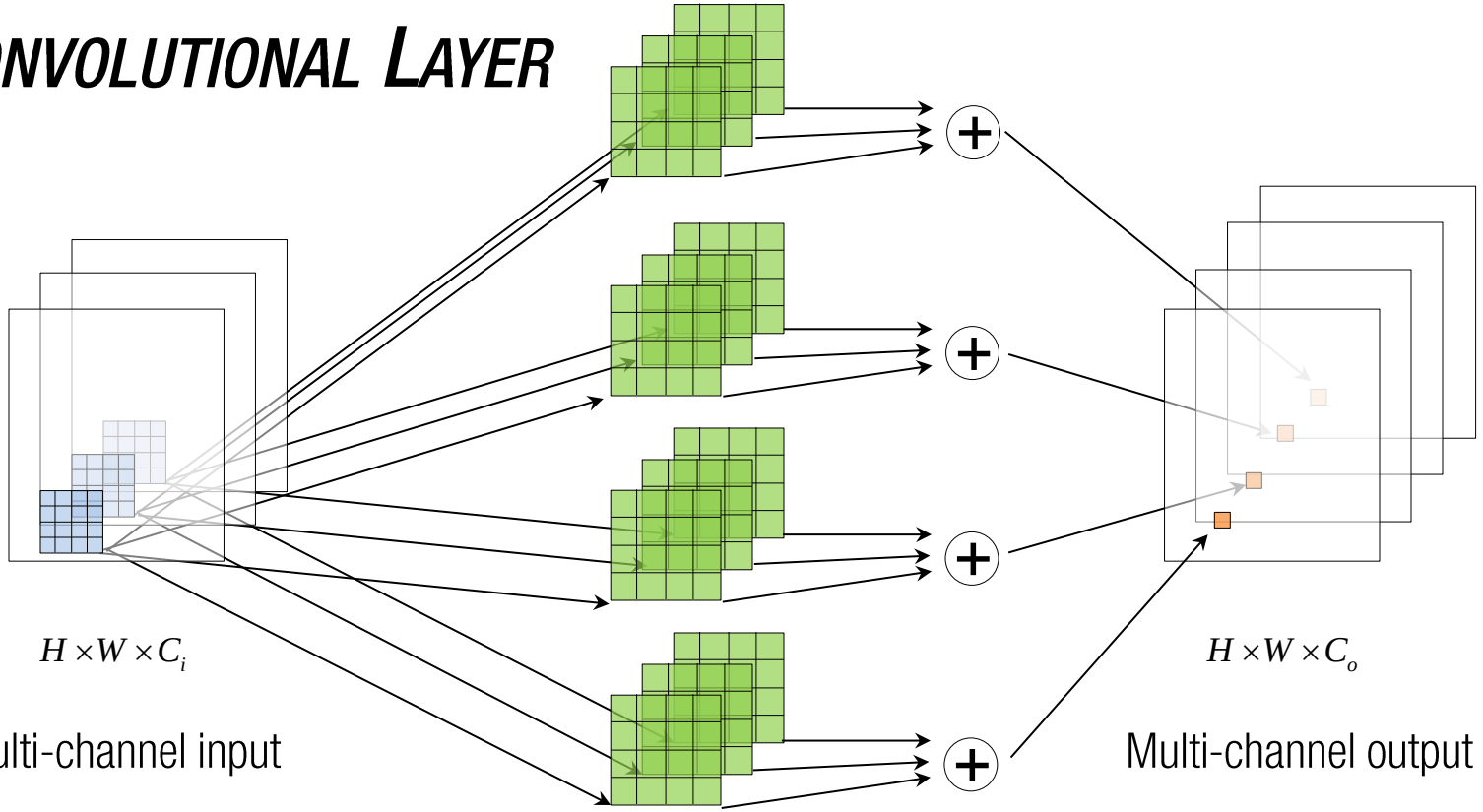


Multi-channel input

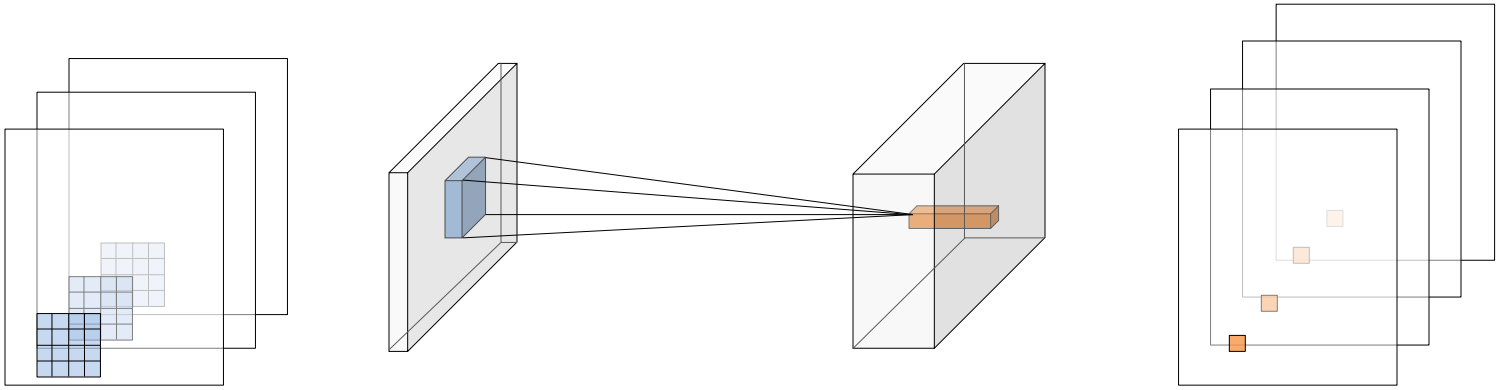
CONVOLUTIONAL LAYER



CONVOLUTIONAL LAYER



Convolutional Layer



$H \times W \times C_i$

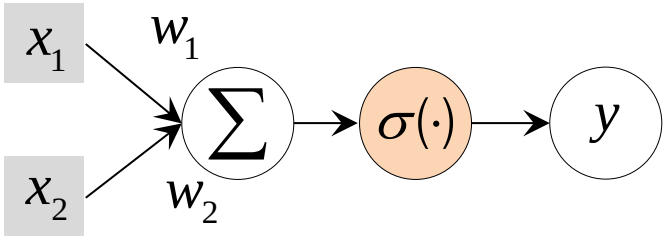
$H \times W \times C_o$

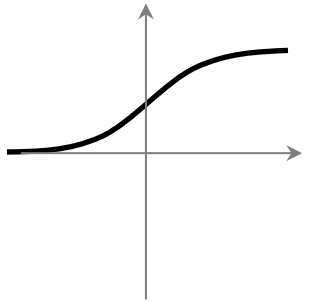
Multi-channel input

Multi-channel output

Feature map

RECALL: ACTIVATION LAYER

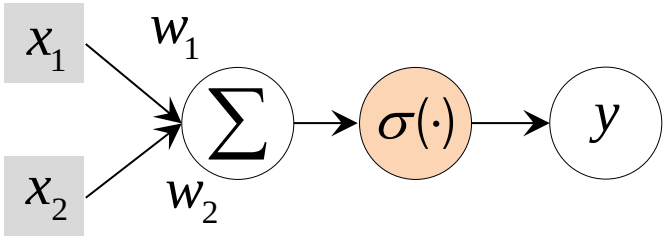


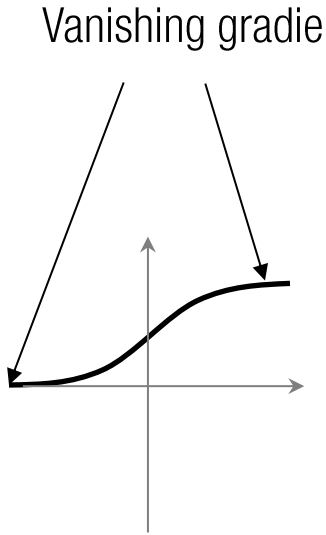


Sigmoid

$$\sigma(x) = \frac{1}{1+e^{-x}}$$

RECALL: ACTIVATION LAYER



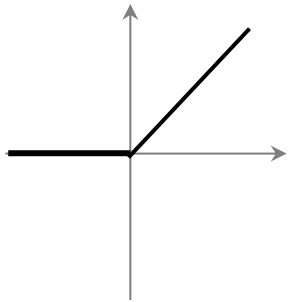


Sigmoid

$$\sigma(x) = \frac{1}{1+e^{-x}}$$

ACTIVATION: RELU

3	-2	3	-1
-1	5	3	4
1	-2	-2	2
-7	3	1	7

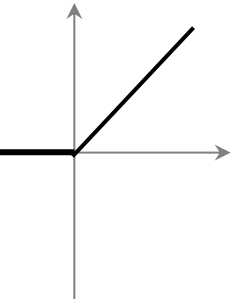


Rect. Linear (ReLu)

$\sigma(x) = \max(0, x)$

ACTIVATION: RELU

3	-2	3	-1
-1	5	3	4
1	-2	-2	2
-7	3	1	7



3	0	3	0
0	5	3	4
1	0	0	2
0	3	1	7

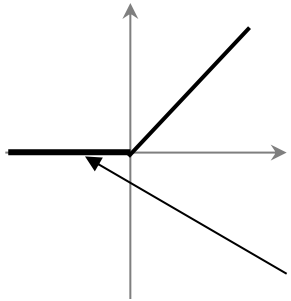
Rect. Linear (ReLu)

$\sigma(x) = \max(0, x)$

ACTIVATION: RELU

3	-2	3	-1
-1	5	3	4
1	-2	-2	2
-7	3	1	7

3	0	3	0
0	5	3	4
1	0	0	2
0	3	1	7



Rect. Linear (ReLu)

$\sigma(x) = \max(0, x)$

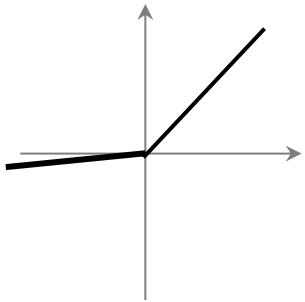
Vanishing gradient

ACTIVATION: LEAKY RELU

3	-2	3	-1
-1	5	3	4
1	-2	-2	2
-7	3	1	7

3	-0.02	3	-0.01
-0.01	5	3	4
1	-0.02	-0.02	2
-0.07	3	1	7

$\varepsilon = 0.01$



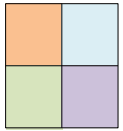
Rect. Linear (ReLu)

$\sigma(x) = \max(\varepsilon x, x)$

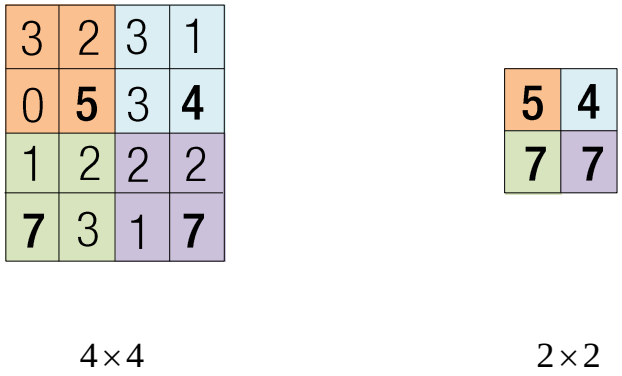
Vanishing gradient

Spatial Pooling (Sampling~Dimensional Reduction)

3	2	3	1
0	5	3	4
1	2	2	2
7	3	1	7

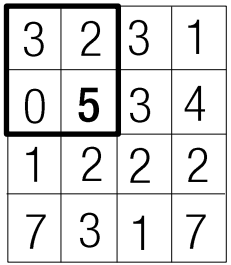


SPATIAL POOLING (SAMPLING~DIMENSIONAL REDUCTION)

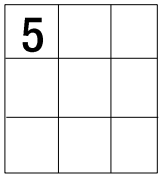


Max-pooling (window size 2x2, stride 2)

SPATIAL POOLING (SAMPLING~DIMENSIONAL REDUCTION)



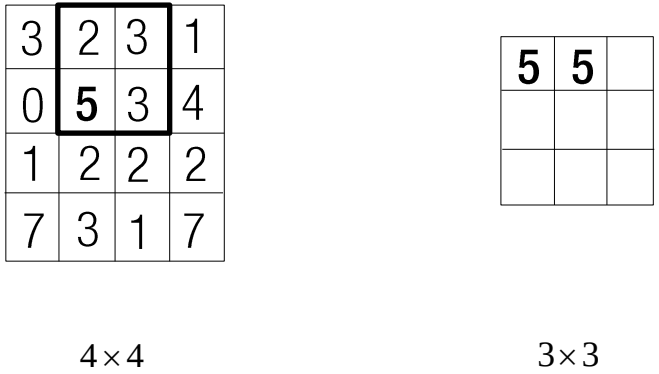
4×4



3×3

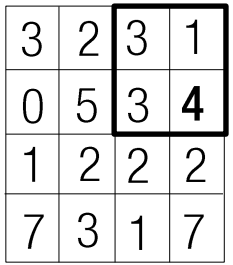
Max-pooling (window size 2x2, stride 1)

SPATIAL POOLING (SAMPLING~DIMENSIONAL REDUCTION)

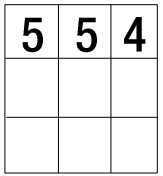


Max-pooling (window size 2x2, stride 1)

SPATIAL POOLING (SAMPLING~DIMENSIONAL REDUCTION)



4×4



3×3

Max-pooling (window size 2x2, stride 1)

Spatial Pooling (Sampling~Dimensional Reduction)

3	2	3	1
0	5	3	4
1	2	2	2
7	3	1	7

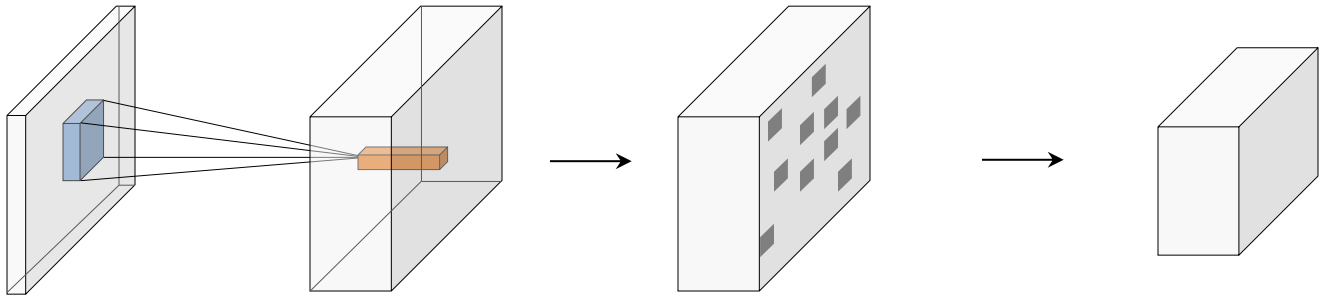
4×4

5	**5**	**4**
5	**5**	**4**
7	**3**	**7**

3×3

Max-pooling (window size 2x2, stride 1)

Conv+ReLu+Pool

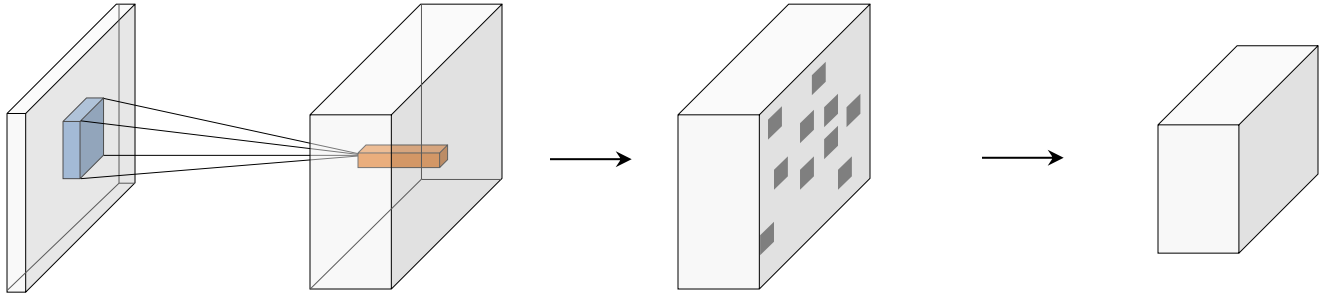


Operations:	Conv	ReLu	Max-pool
# of units:	$H \times W \times C_1$	$H \times W \times C_2$	$H \times W \times C_2$	$H_1 \times W_1 \times C_2$

# of weights:

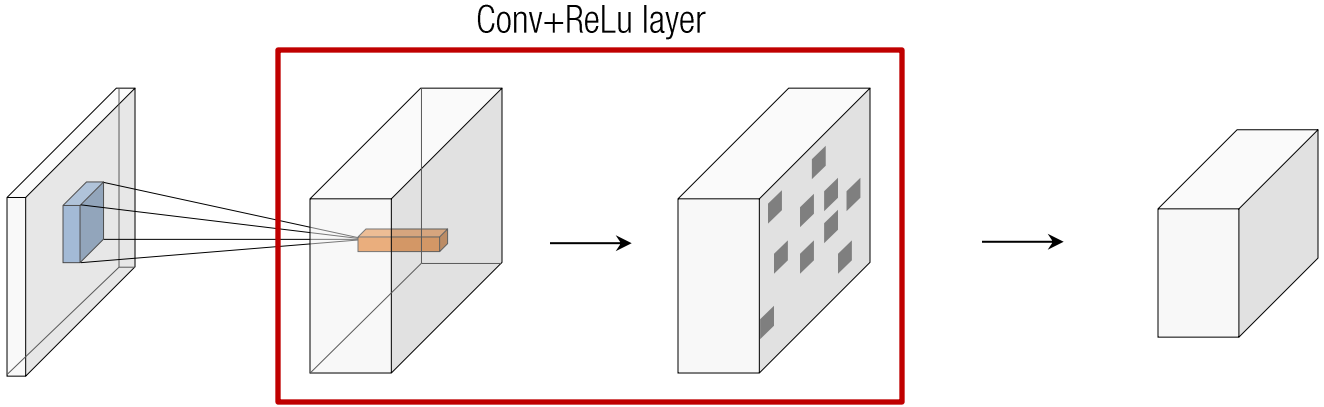
# of biases:

Conv+ReLu+Pool



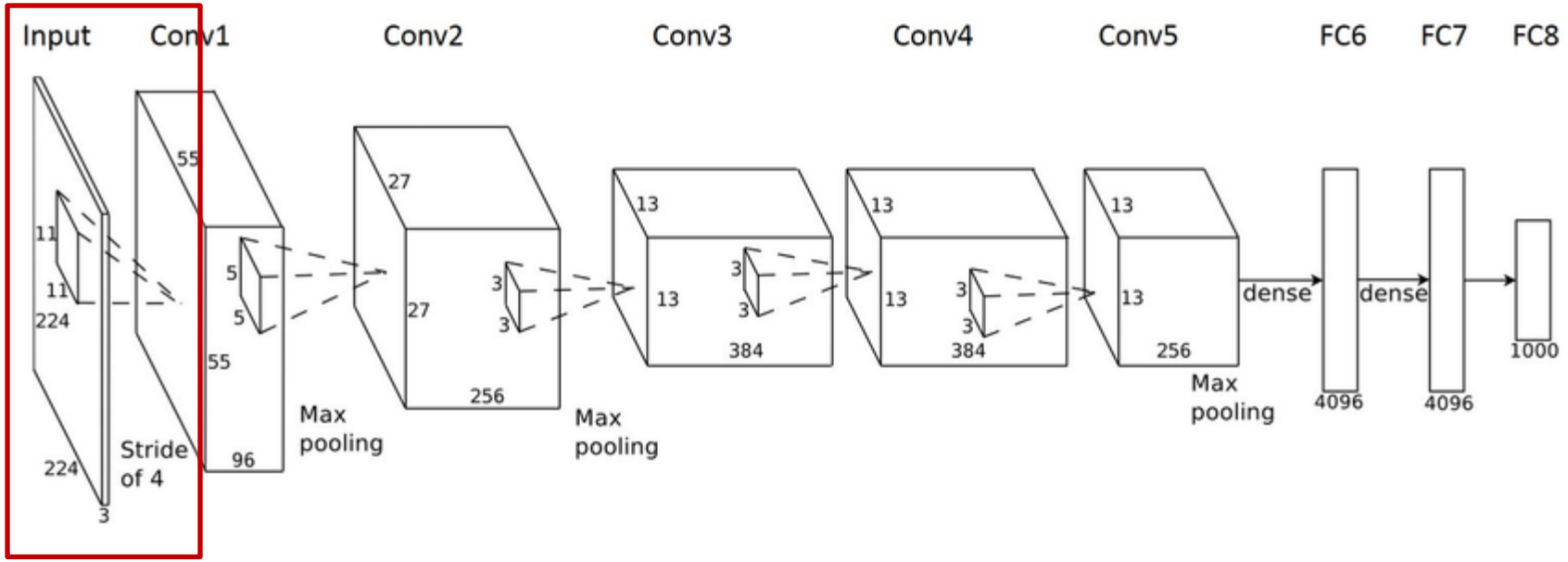
Operations:	Conv	ReLu	Max-pool
# of units:	$H \times W \times C_1$	$H \times W \times C_2$	$H \times W \times C_2$	$H_1 \times W_1 \times C_2$
# of weights:	$F \times F \times C_1 \times C_2$	0	0
# of biases:	$1 \times C_2$	0	0

CONV+RELU+POOL



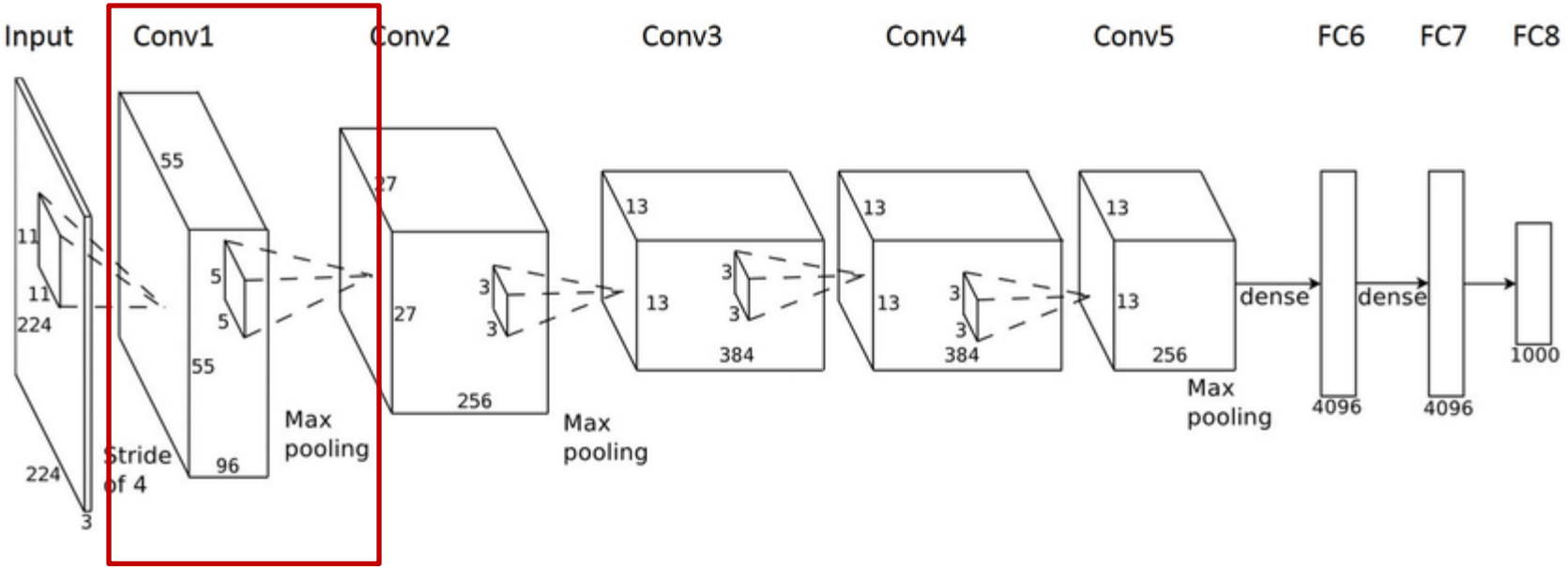
Operations:	Conv	ReLu	Max-pool
# of units:	$H \times W \times C_1$	$H \times W \times C_2$	$H \times W \times C_2$	$H_1 \times W_1 \times C_2$
# of weights:	$F \times F \times C_1 \times C_2$	0	0
# of biases:	$1 \times C_2$	0	0

ALEX NET



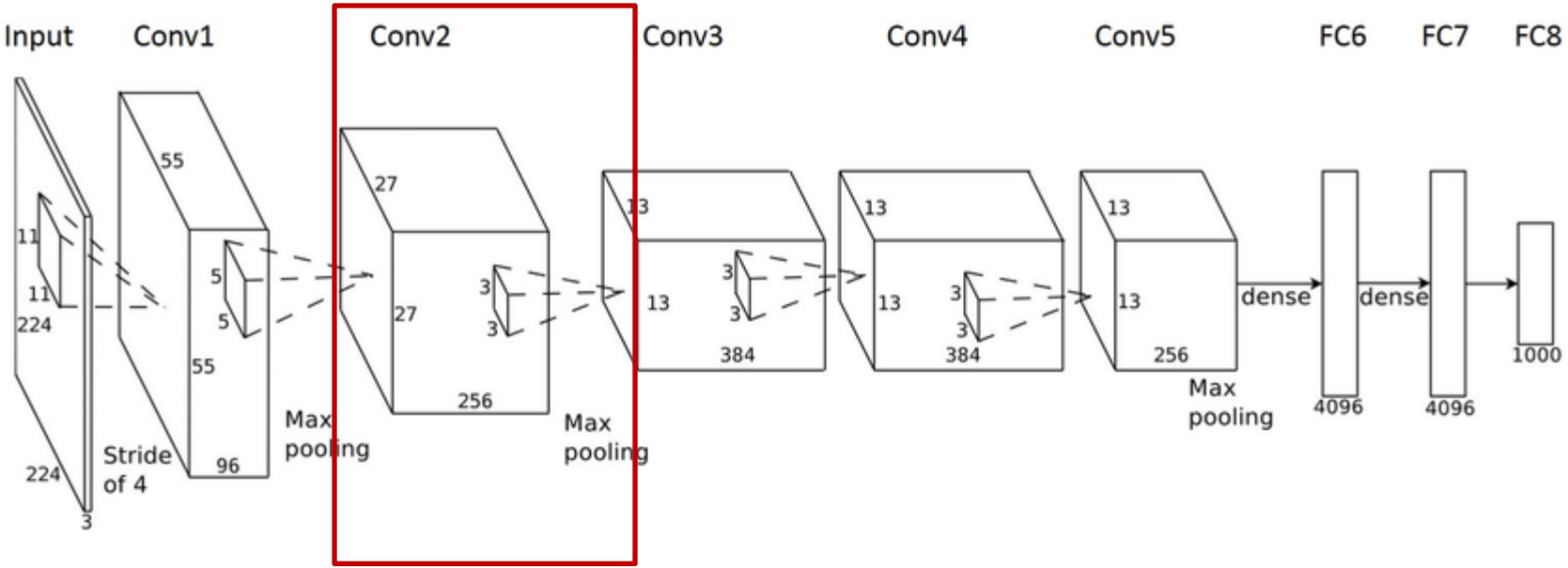
Conv+ReLu

ALEX NET



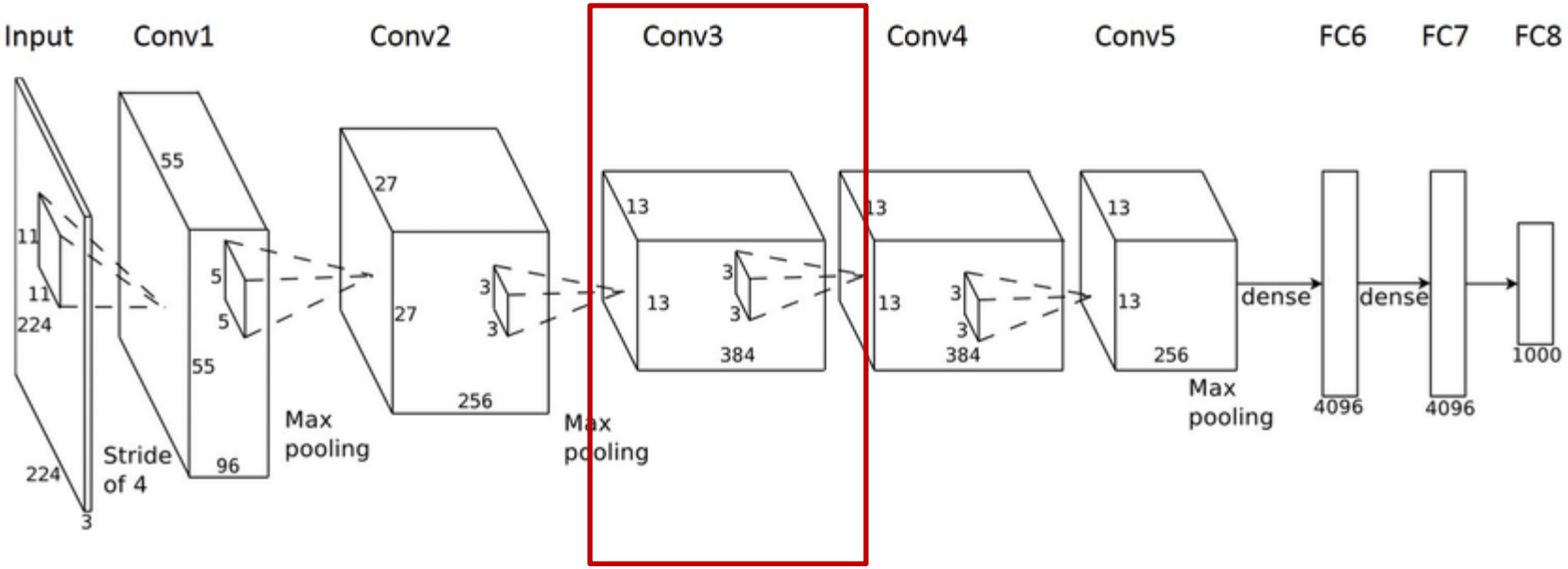
Conv+ReLu+Pool

ALEX NET



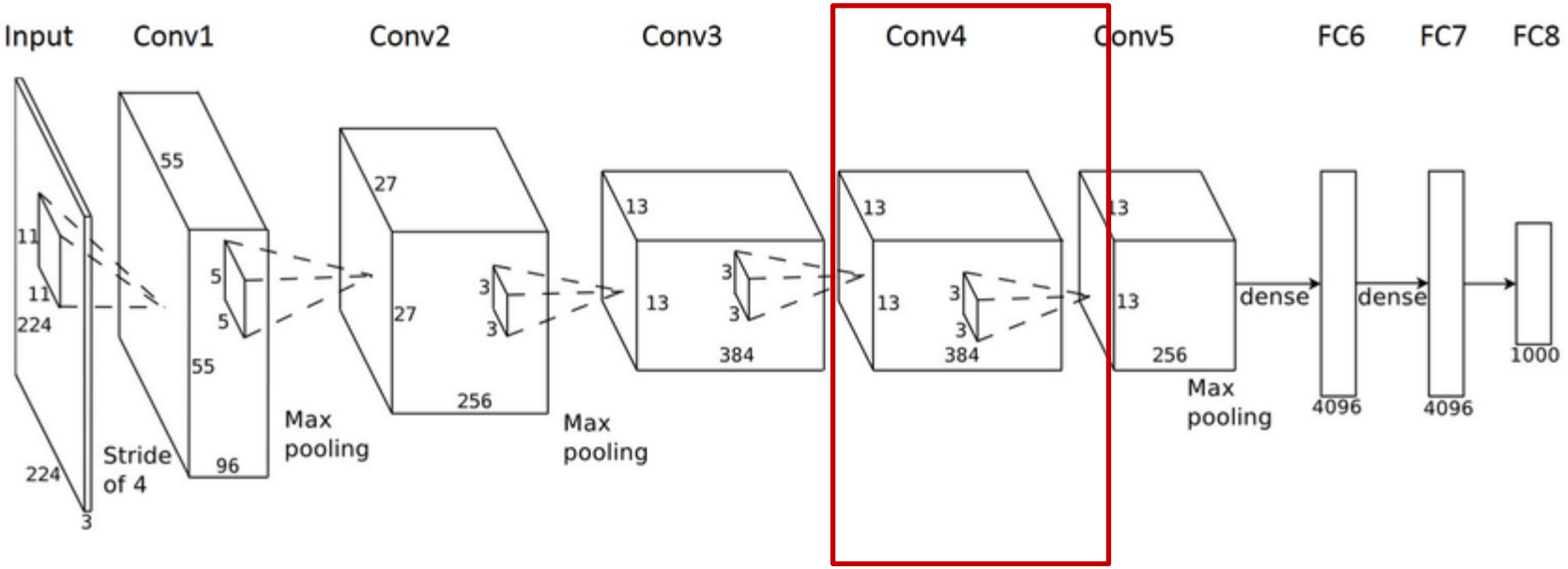
Conv+ReLu+Pool

ALEX NET



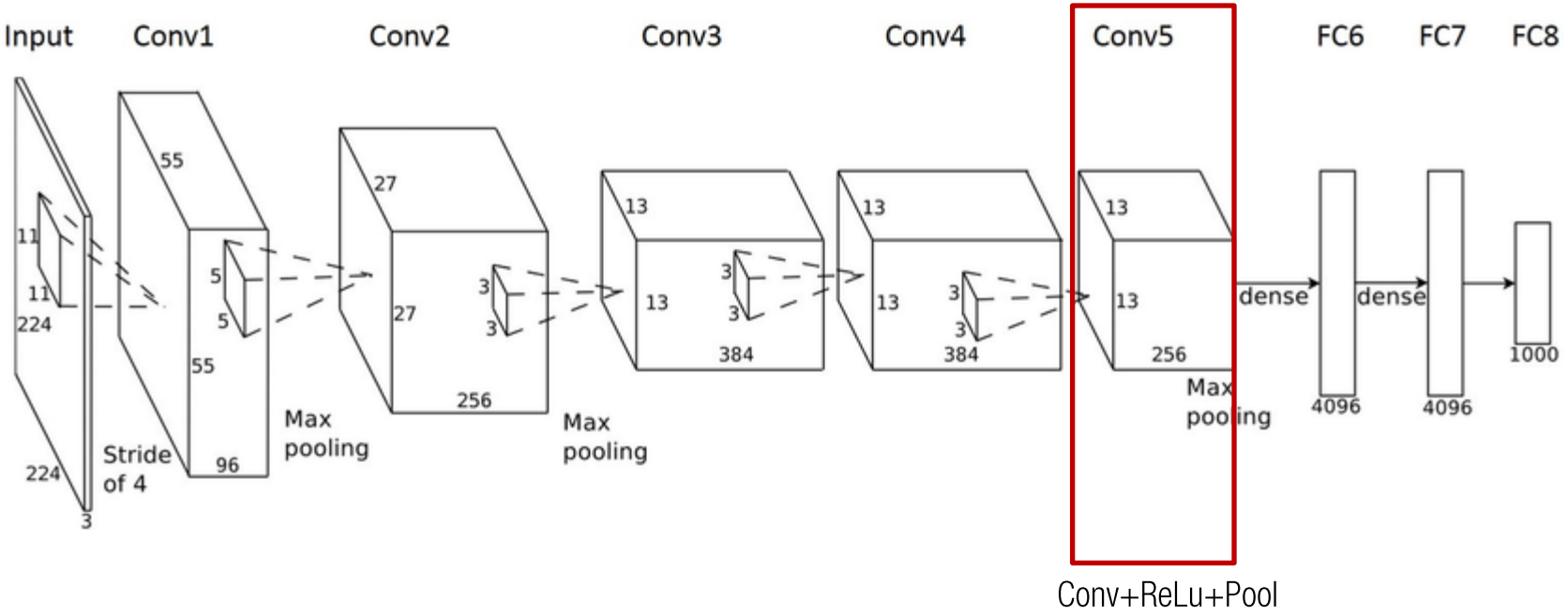
Conv+ReLu+Pool

ALEX NET



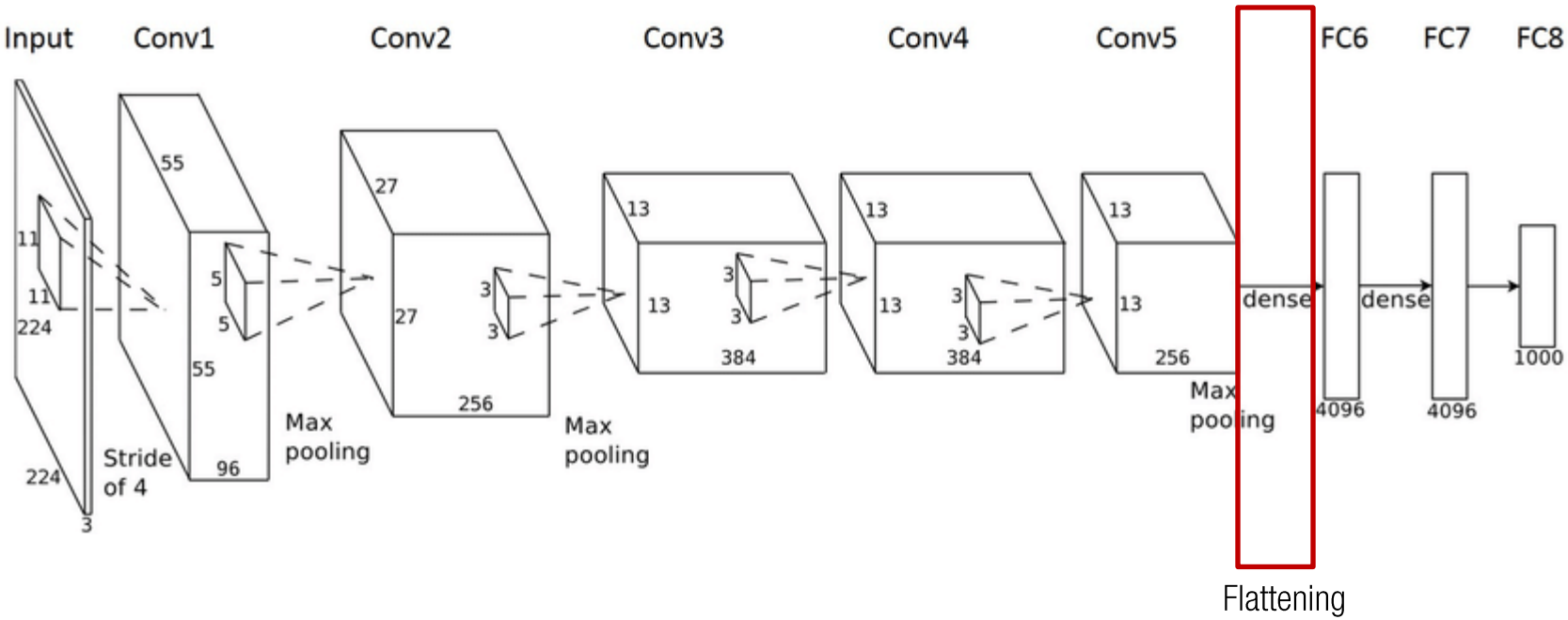
Conv+ReLu+Pool

ALEX NET

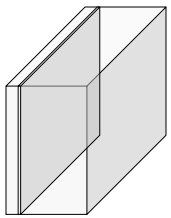


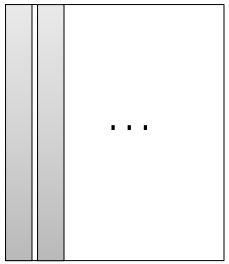
Conv+ReLu+Pool

TENSOR FLATTENING

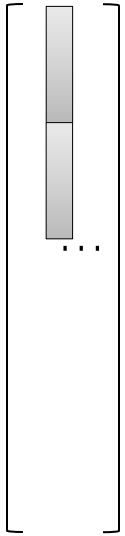


TENSOR FLATTENING

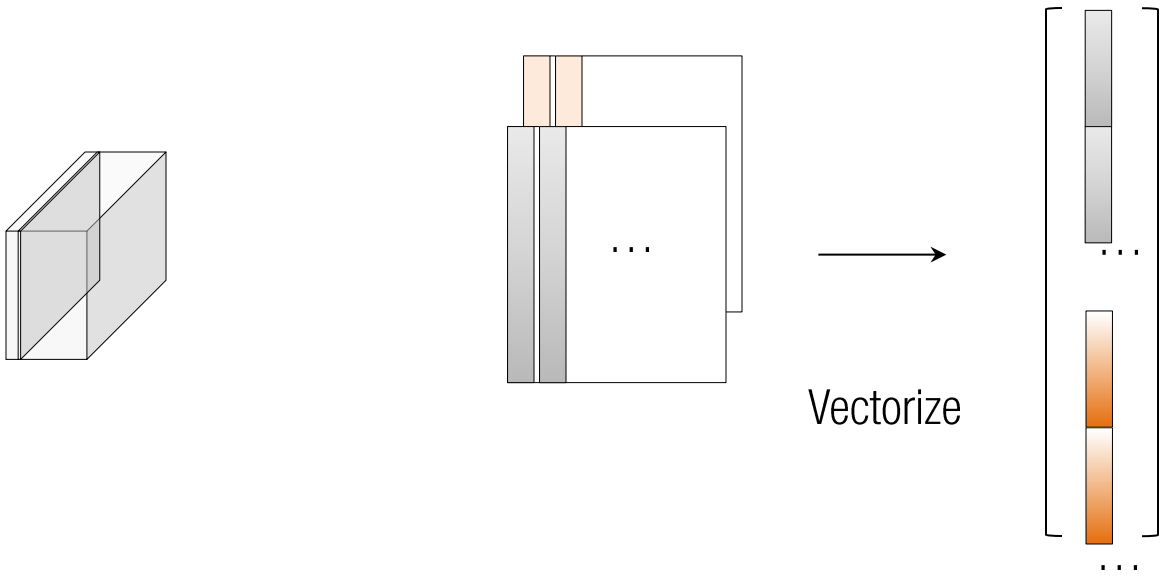




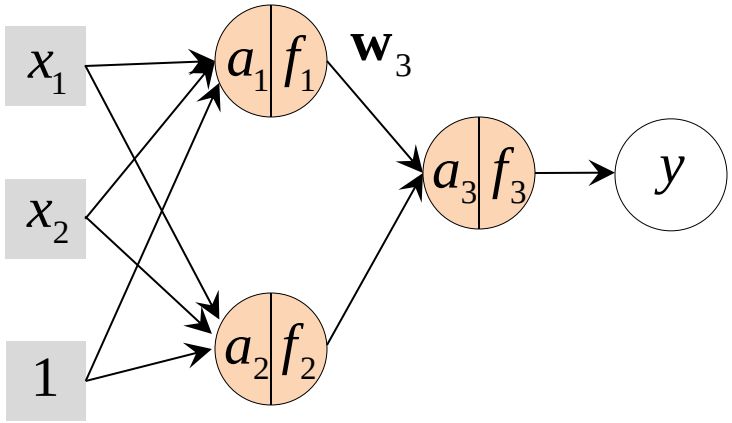
Vectorize



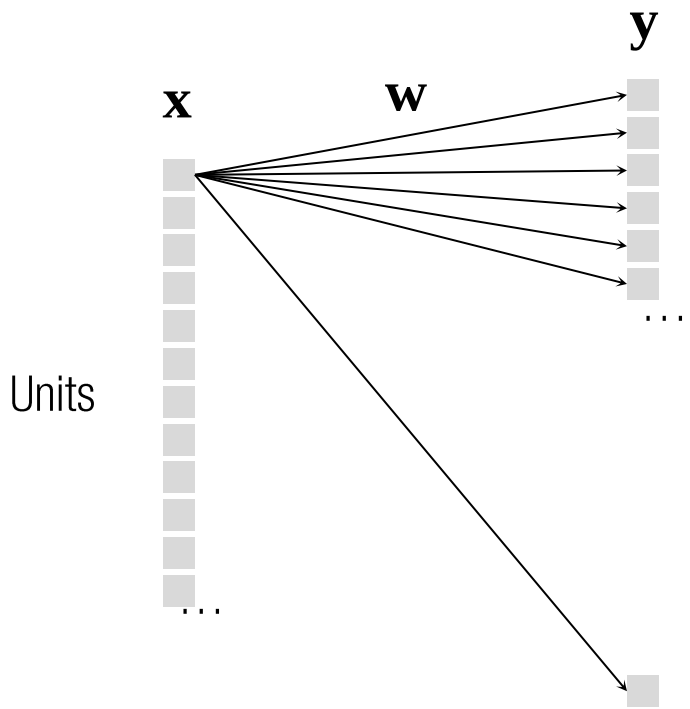
TENSOR FLATTENING



RECALL: HIDDEN LAYER

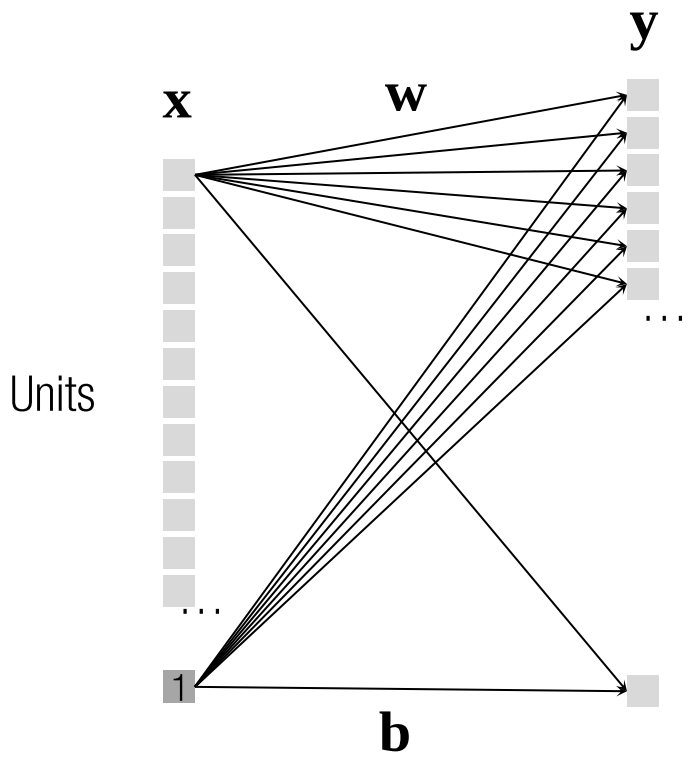


FULLY CONNECTED LAYER (FC)



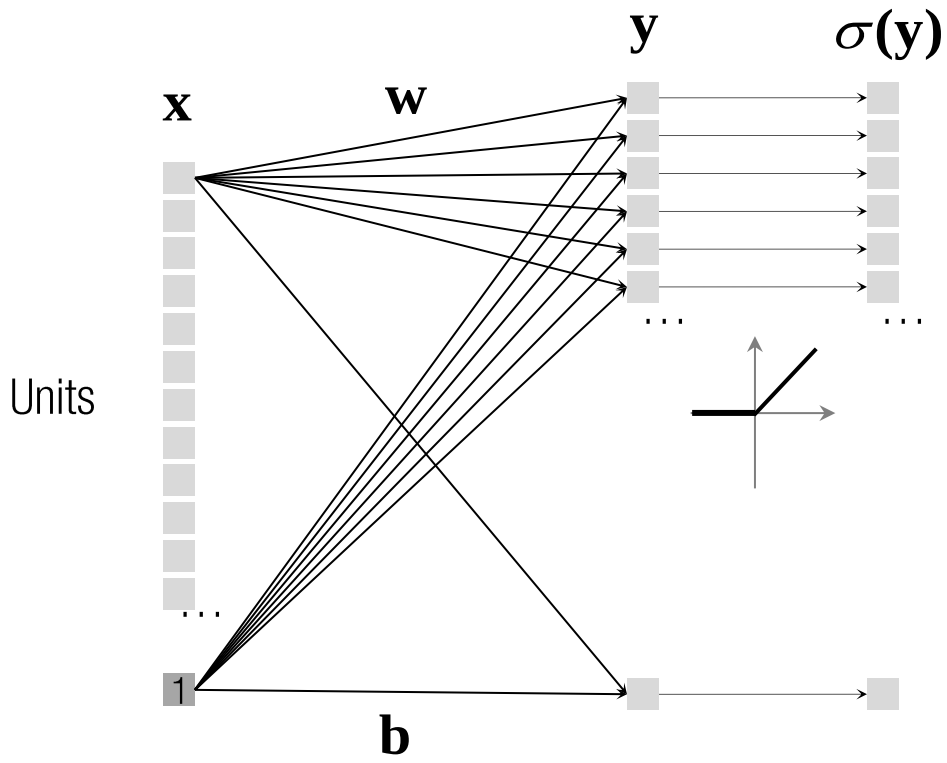
$$\mathbf{y} = \mathbf{W}\begin{bmatrix}\mathbf{x}\\1\end{bmatrix}$$

FULLY CONNECTED LAYER (FC)

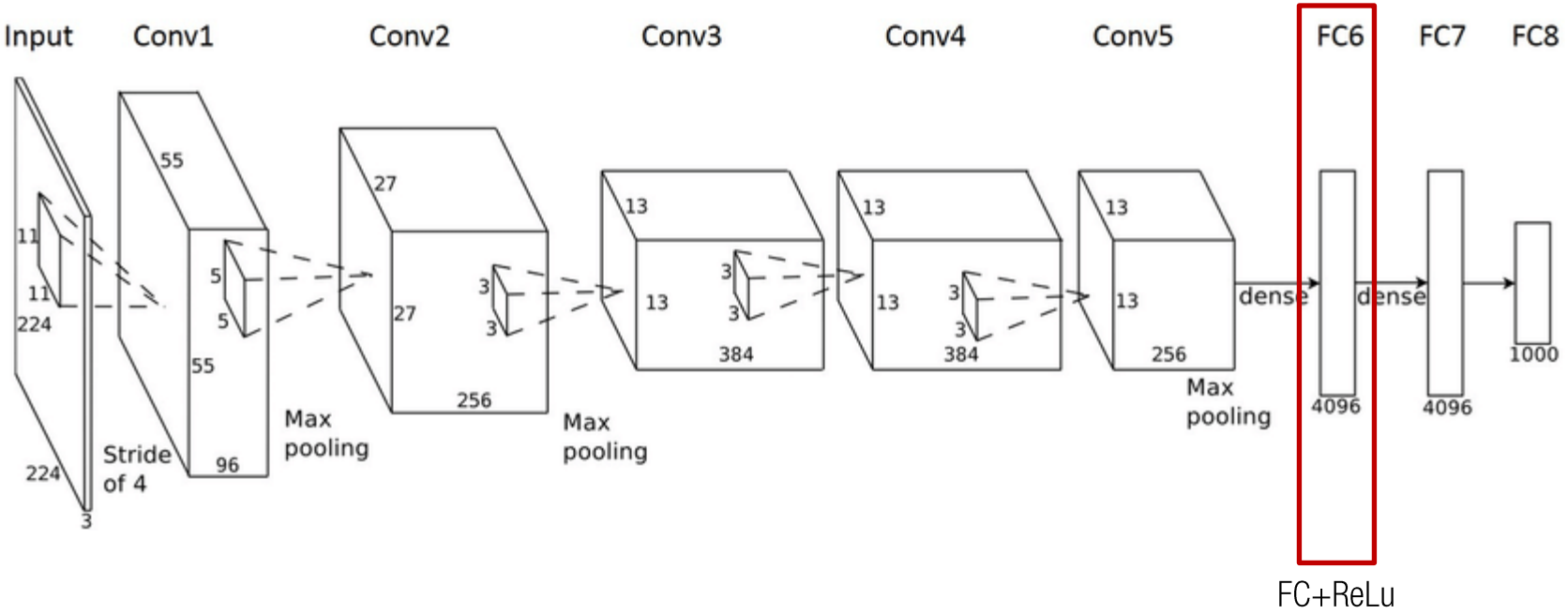


$$\mathbf{y} = \mathbf{W}\begin{bmatrix}\mathbf{x}\\1\end{bmatrix} + \mathbf{b}$$

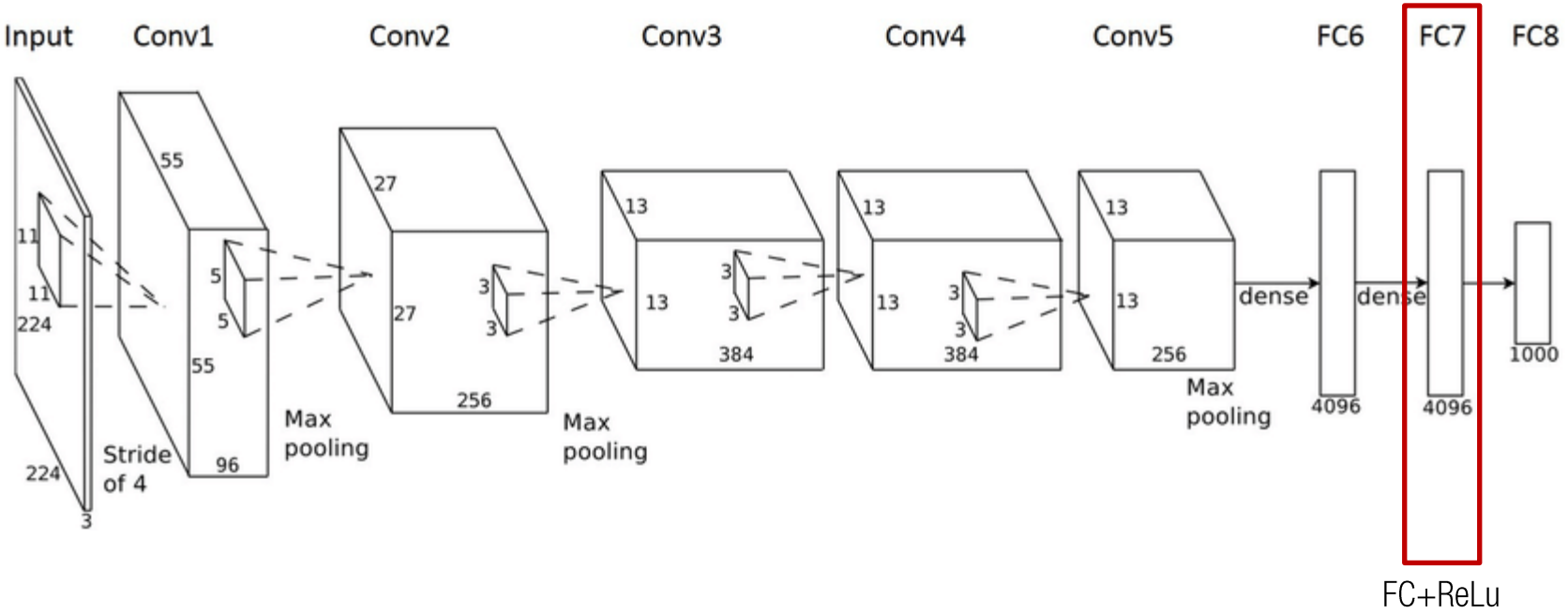
FC+ReLu



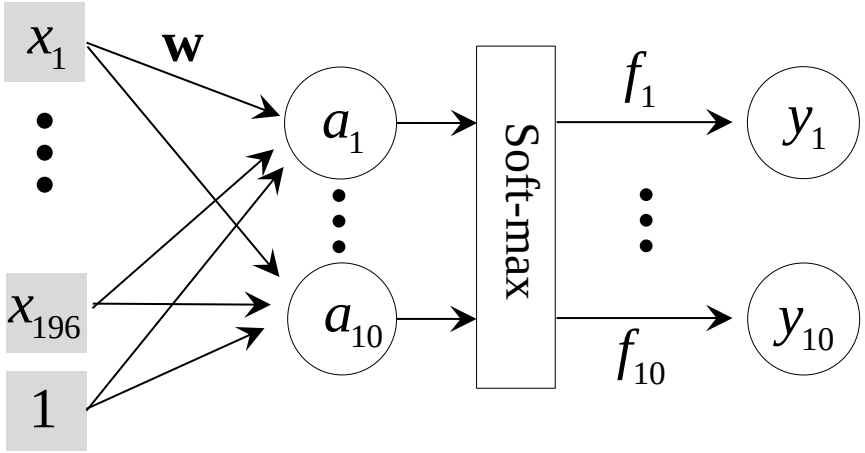
TENSOR FLATTENING



TENSOR FLATTENING

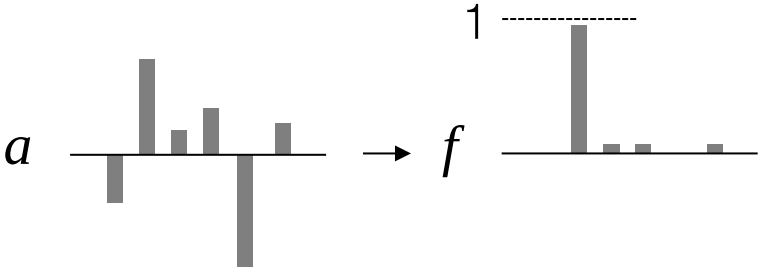


RECALL: SOFT-MAX



Soft-max (winners-take-all):

$$f_i = \frac{e^{a_i}}{\sum_i e^{a_i}}$$

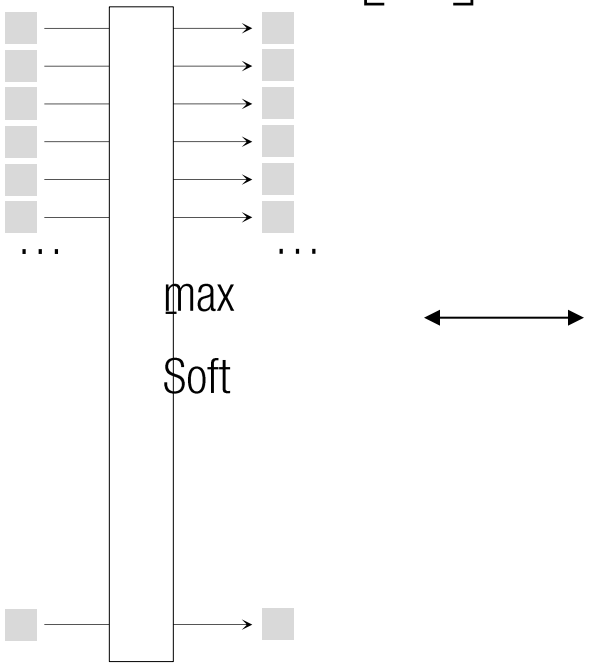


$$\frac{\partial f_i}{\partial a_j} = \begin{cases} f_i(1-f_i) & i = j \\ -f_i f_j & i \neq j \end{cases}$$

ERROR MEASURE (LOSS)

$\mathbf{x} \in \mathbb{R}^L$ $\tilde{\mathbf{y}} \in [0,1]^L$ $\tilde{\mathbf{y}} \in \{0,1\}^L$

L : # of class labels

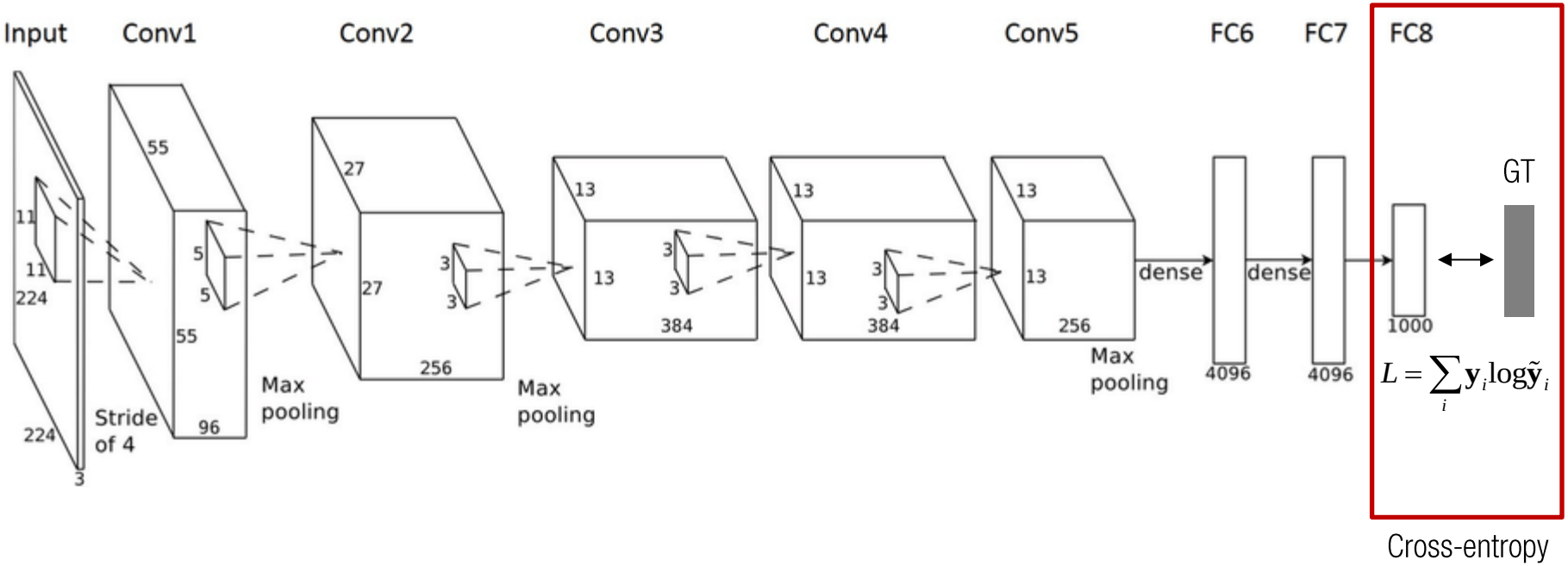


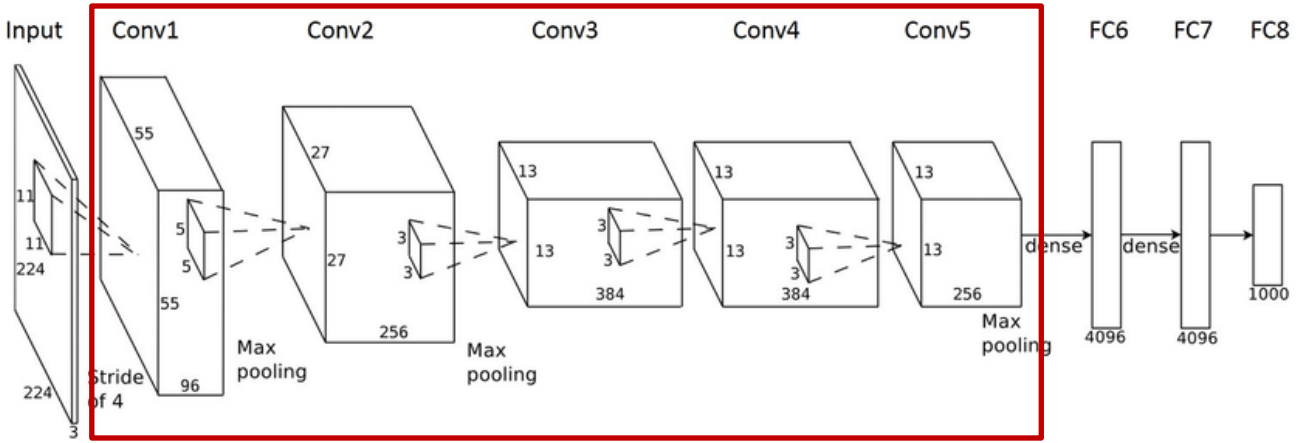
Prediction Ground truth

Cross-entropy loss (matching prob. distribution)

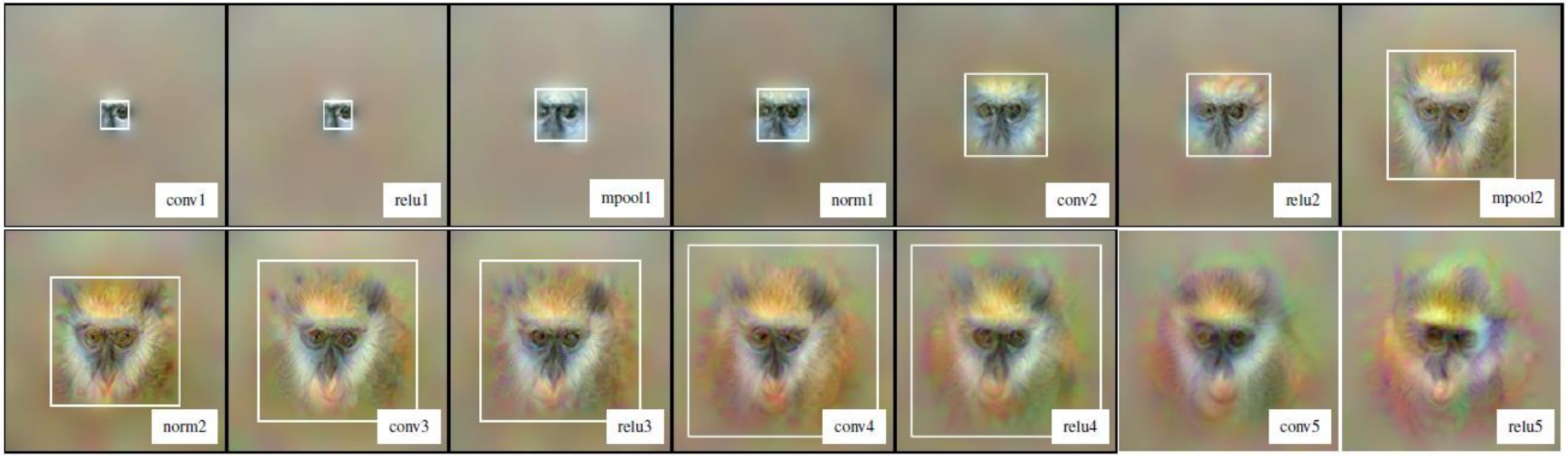
$$L = \sum_i \mathbf{y}_i \log \tilde{\mathbf{y}}_i$$

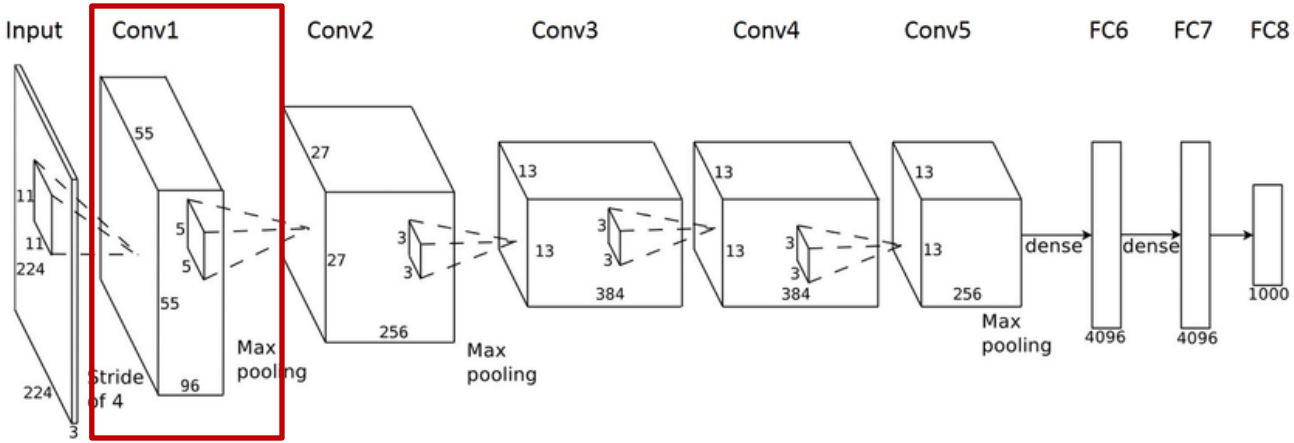
TENSOR FLATTENING



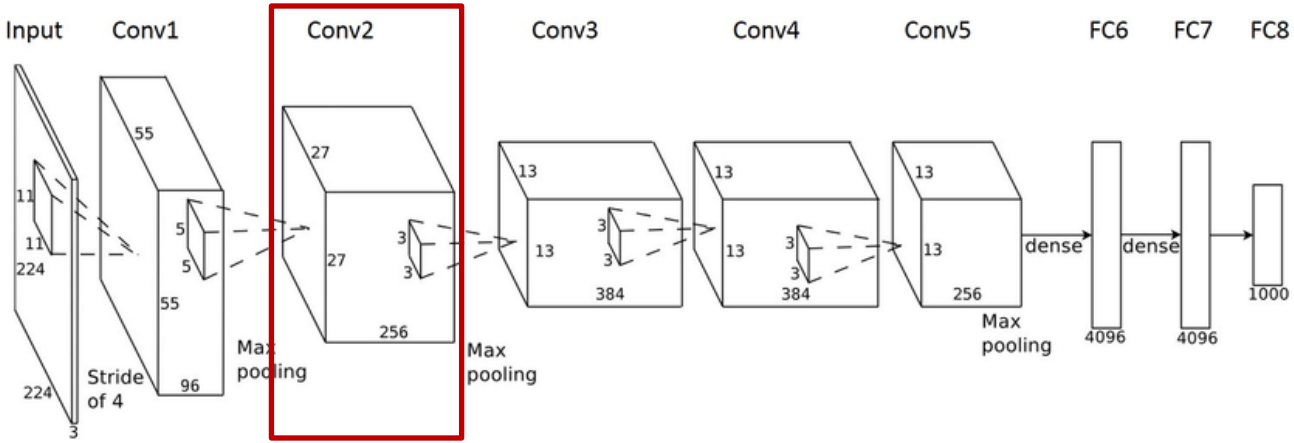


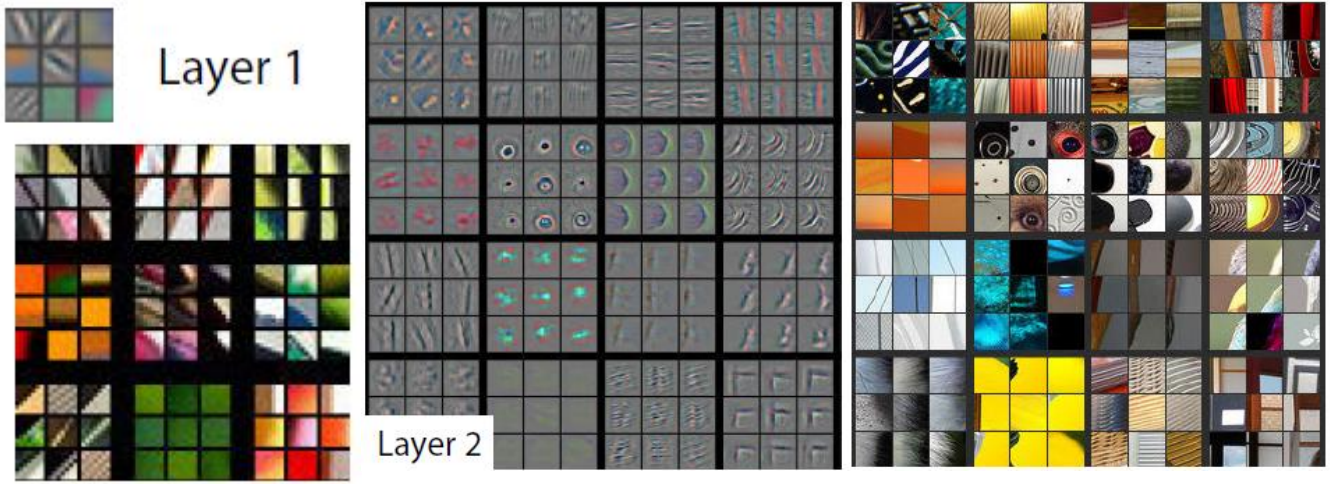
Local feature extraction

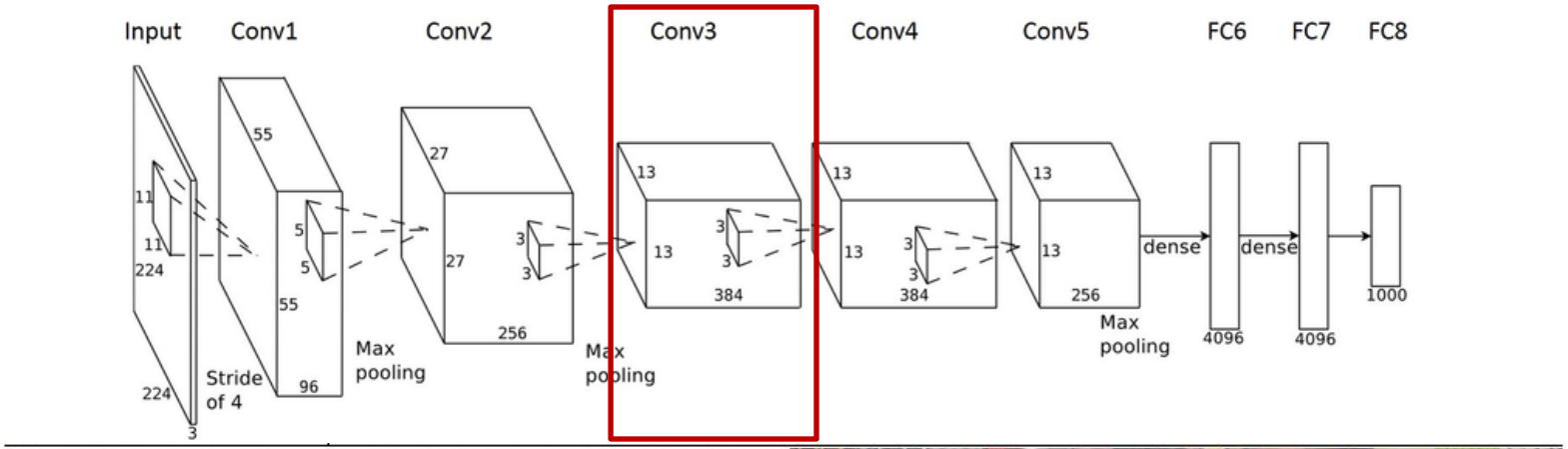


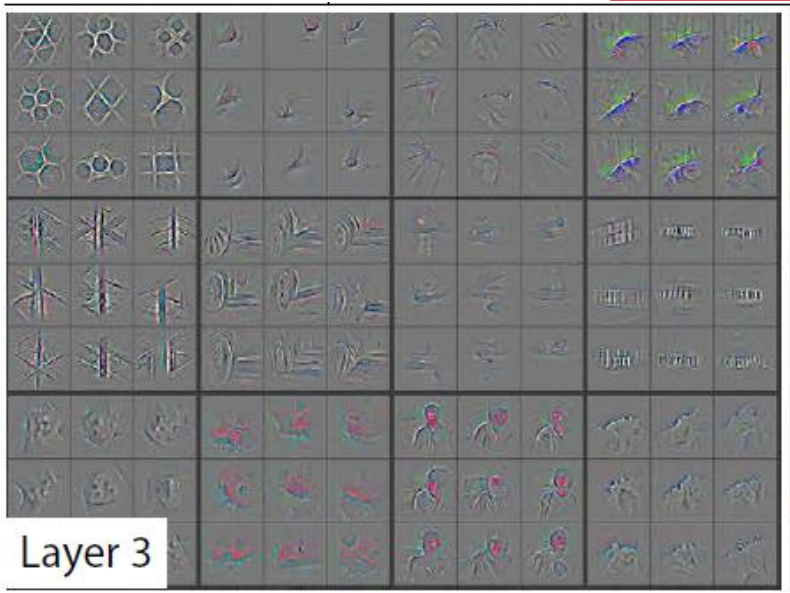

Input
Conv1
Conv2
Conv3
Conv4
Conv5
FC6
FC7
FC8
224
224
3
11
11
55
55
96
5
5
Stride of 4
Max pooling
27
27
256
3
3
Max pooling
13
13
384
3
3
13
13
384
3
3
13
13
256
Max pooling
dense
4096
dense
4096
1000

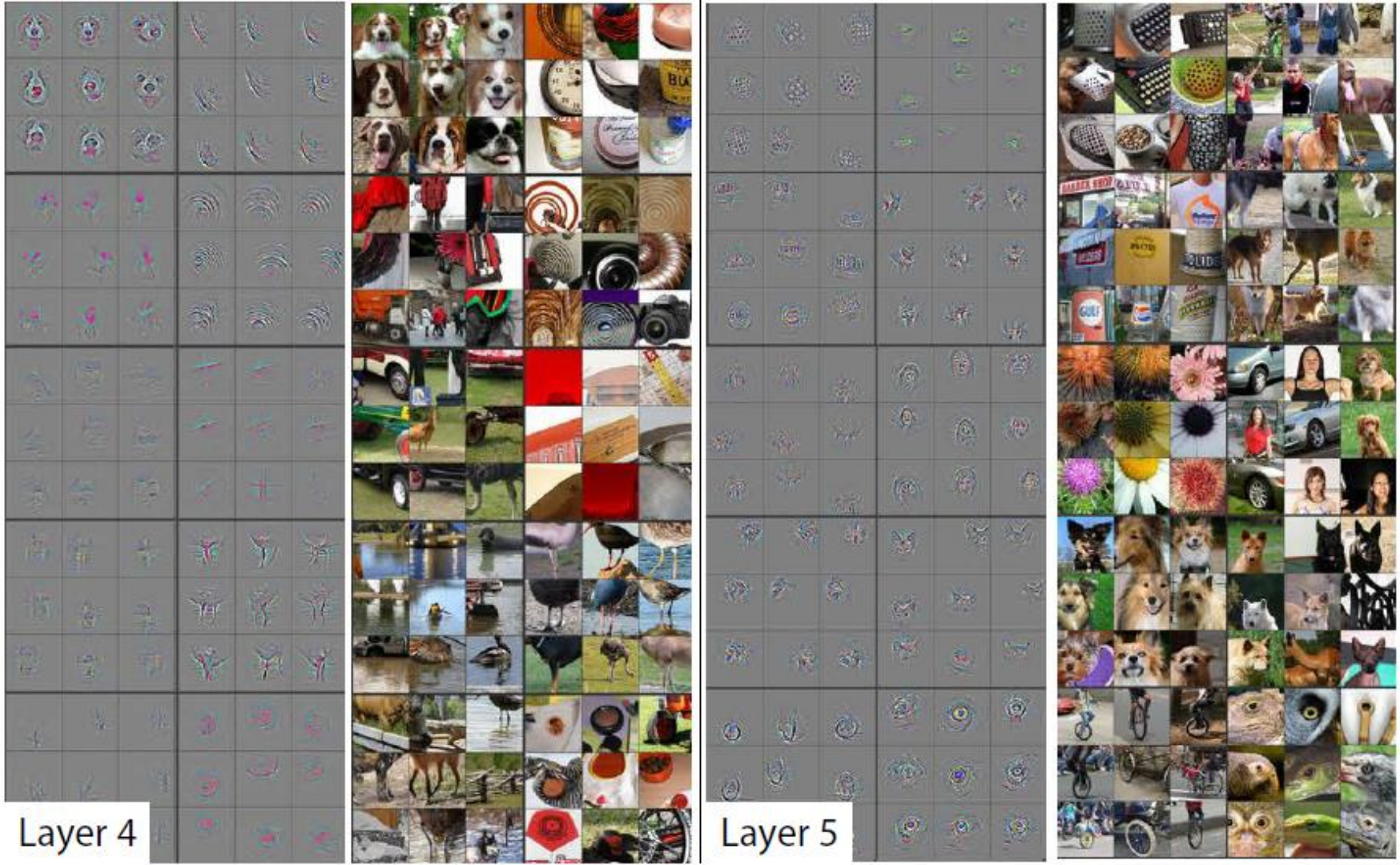
Layer 1

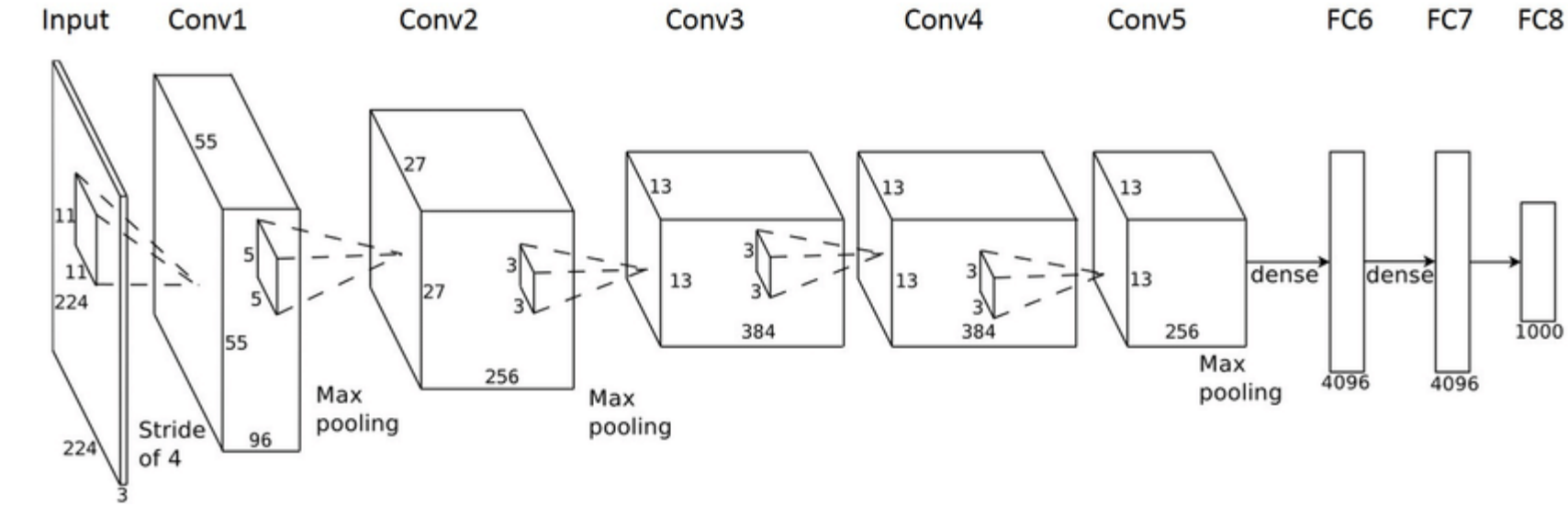

Input
Conv1
Conv2
Conv3
Conv4
Conv5
FC6
FC7
FC8
224
224
3
11
11
55
55
96
5
5
Stride of 4
27
27
256
3
3
Max pooling
13
13
384
Max pooling
13
13
384
13
13
256
Max pooling
dense
4096
dense
4096
1000

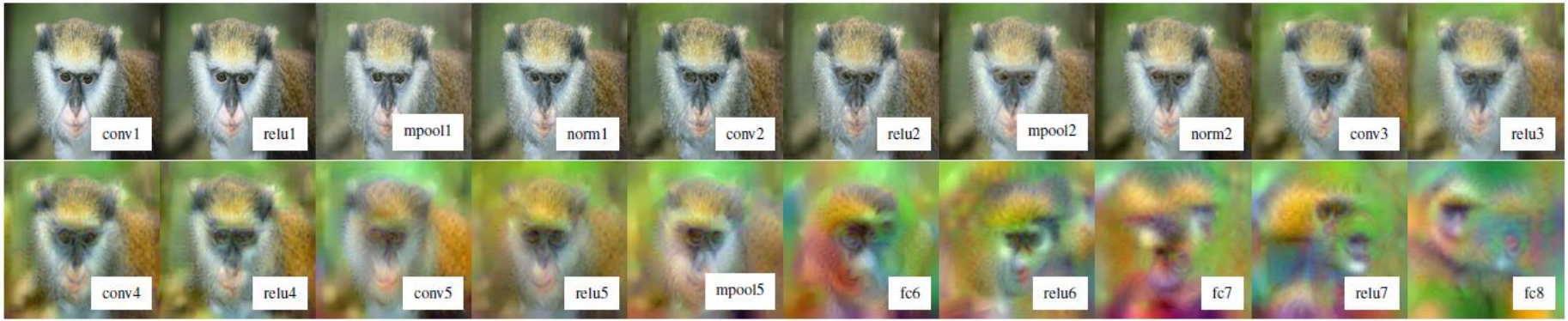

Layer 1
Layer 2


Input
Conv1
Conv2
Conv3
Conv4
Conv5
FC6
FC7
FC8
11
11
224
224
3
55
55
5
5
96
Stride
of 4
27
27
5
256
Max
pooling
13
13
3
3
384
Max
pooling
13
13
3
3
384
13
13
3
3
256
Max
pooling
dense
4096
dense
4096
1000


Layer 3

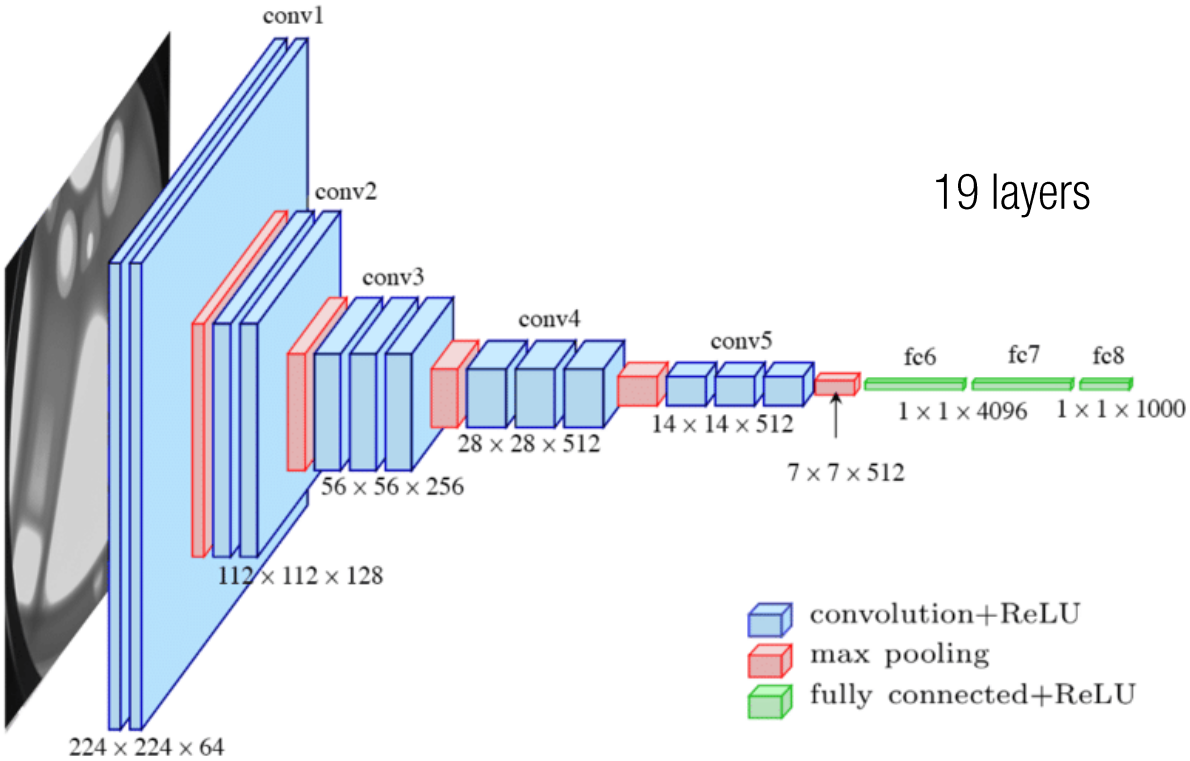

Layer 4
Layer 5


Input
Conv1
Conv2
Conv3
Conv4
Conv5
FC6
FC7
FC8
224
224
3
11
11
55
55
96
5
5
Stride of 4
Max pooling
27
27
256
3
3
Max pooling
13
13
384
3
3
13
13
384
3
3
13
13
256
3
3
Max pooling
dense
4096
dense
4096
1000


conv1
relu1
mpool1
norm1
conv2
relu2
mpool2
norm2
conv3
relu3
conv4
relu4
conv5
relu5
mpool5
fc6
relu6
fc7
relu7
fc8

https://www.youtube.com/watch?v=AgkfIQ4IGaM

VGG-16 (Depth Matters)



19 layers

Table 7: **Comparison with the state of the art in ILSVRC classification**. Our method is denoted as "VGG". Only the results obtained without outside training data are reported.

Method	top-1 val. error (%)	top-5 val. error (%)	top-5 test error (%)
VGG (2 nets, multi-crop & dense eval.)	**23.7**	**6.8**	**6.8**
VGG (1 net, multi-crop & dense eval.)	24.4	7.1	7.0
VGG (ILSVRC submission, 7 nets, dense eval.)	24.7	7.5	7.3
GoogLeNet (Szegedy et al., 2014) (1 net)	-	7.9	
GoogLeNet (Szegedy et al., 2014) (7 nets)	-	**6.7**	
MSRA (He et al., 2014) (11 nets)	-	-	8.1
MSRA (He et al., 2014) (1 net)	27.9	9.1	9.1
Clarifai (Russakovsky et al., 2014) (multiple nets)	-	-	11.7
Clarifai (Russakovsky et al., 2014) (1 net)	-	-	12.5
Zeiler & Fergus (Zeiler & Fergus, 2013) (6 nets)	36.0	14.7	14.8
Zeiler & Fergus (Zeiler & Fergus, 2013) (1 net)	37.5	16.0	16.1
OverFeat (Sermanet et al., 2014) (7 nets)	34.0	13.2	13.6
OverFeat (Sermanet et al., 2014) (1 net)	35.7	14.2	-
Krizhevsky et al. (Krizhevsky et al., 2012) (5 nets)	38.1	16.4	16.4
Krizhevsky et al. (Krizhevsky et al., 2012) (1 net)	40.7	18.2	-

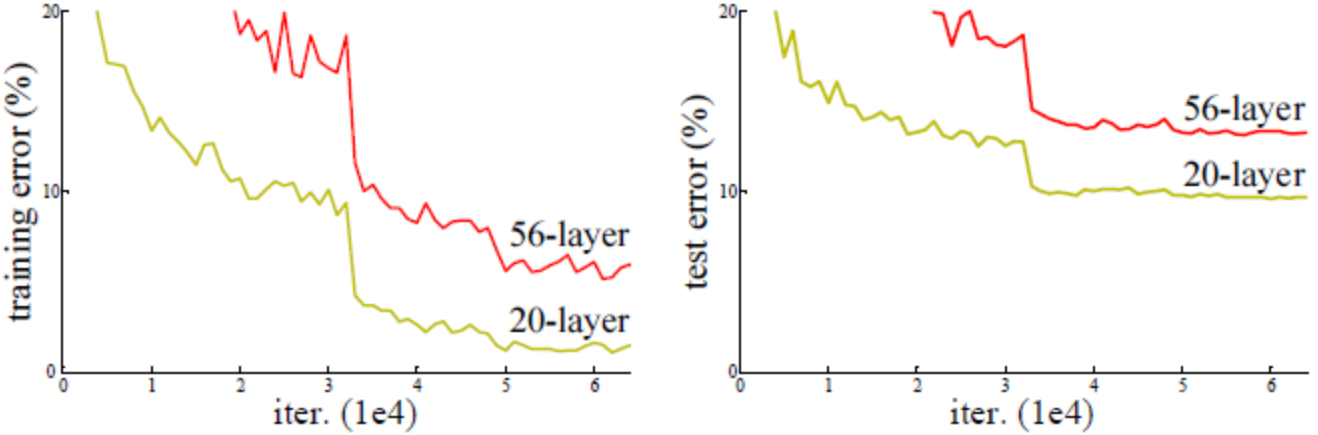
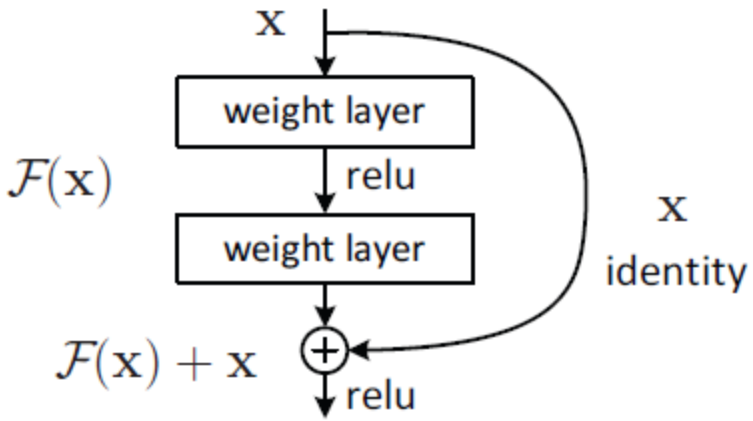
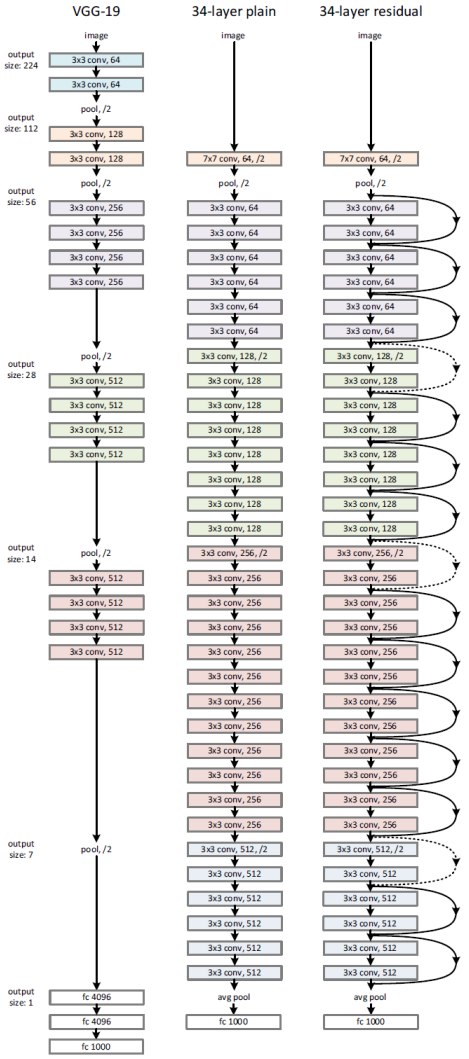


Figure 1. Training error (left) and test error (right) on CIFAR-10 with 20-layer and 56-layer "plain" networks. The deeper network has higher training error, and thus test error. Similar phenomena on ImageNet is presented in Fig. 4.

ResNet


x
weight layer
$\mathcal{F}(x)$
relu
weight layer
x
identity
$\mathcal{F}(x) + x$
relu


VGG-19
34-layer plain
34-layer residual
image
output size: 224
3x3 conv, 64
3x3 conv, 64
pool, /2
output size: 112
3x3 conv, 128
3x3 conv, 128
pool, /2
output size: 56
3x3 conv, 256
3x3 conv, 256
3x3 conv, 256
3x3 conv, 256
pool, /2
output size: 28
3x3 conv, 512
3x3 conv, 512
3x3 conv, 512
3x3 conv, 512
pool, /2
output size: 14
3x3 conv, 512
3x3 conv, 512
3x3 conv, 512
3x3 conv, 512
pool, /2
output size: 7
fc 4096
fc 4096
fc 1000
output size: 1
7x7 conv, 64, /2
pool, /2
3x3 conv, 64
3x3 conv, 64
3x3 conv, 64
3x3 conv, 64
3x3 conv, 64
3x3 conv, 64
3x3 conv, 128, /2
3x3 conv, 128
3x3 conv, 128
3x3 conv, 128
3x3 conv, 128
3x3 conv, 128
3x3 conv, 128
3x3 conv, 128
3x3 conv, 256, /2
3x3 conv, 256
3x3 conv, 256
3x3 conv, 256
3x3 conv, 256
3x3 conv, 256
3x3 conv, 256
3x3 conv, 256
3x3 conv, 512, /2
3x3 conv, 512
3x3 conv, 512
3x3 conv, 512
3x3 conv, 512
3x3 conv, 512
avg pool
fc 1000
7x7 conv, 64, /2
pool, /2
3x3 conv, 64
3x3 conv, 64
3x3 conv, 64
3x3 conv, 64
3x3 conv, 64
3x3 conv, 64
3x3 conv, 128, /2
3x3 conv, 128
3x3 conv, 128
3x3 conv, 128
3x3 conv, 128
3x3 conv, 128
3x3 conv, 128
3x3 conv, 128
3x3 conv, 256, /2
3x3 conv, 256
3x3 conv, 256
3x3 conv, 256
3x3 conv, 256
3x3 conv, 256
3x3 conv, 256
3x3 conv, 256
3x3 conv, 512, /2
3x3 conv, 512
3x3 conv, 512
3x3 conv, 512
3x3 conv, 512
3x3 conv, 512
avg pool
fc 1000

method	top-5 err. (**test**)
VGG [41] (ILSVRC'14)	7.32
GoogLeNet [44] (ILSVRC'14)	6.66
VGG [41] (v5)	6.8
PReLU-net [13]	4.94
BN-inception [16]	4.82
ResNet (ILSVRC'15)	**3.57**

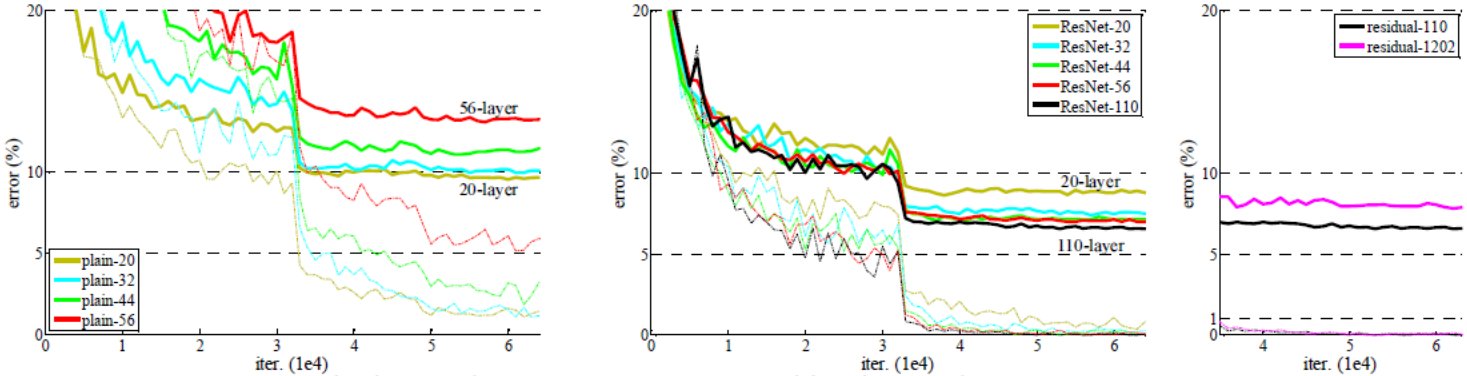


Figure 6. Training on **CIFAR-10**. Dashed lines denote training error, and bold lines denote testing error. **Left**: plain networks. The error of plain-110 is higher than 60% and not displayed. **Middle**: ResNets. **Right**: ResNets with 110 and 1202 layers.