

The background of the slide is a reproduction of the painting 'The Starry Night' by the Dutch Impressionist painter J.M.W. Turner. The painting depicts a night scene with a dark, swirling sky filled with numerous bright, glowing stars and a prominent, dark, vertical structure resembling a tower or a church spire in the center. The overall color palette is dominated by deep blues and greens, with bright yellow and white highlights from the stars and the tower's structure.

# *CONVOLUTIONAL NEURAL NETWORK*

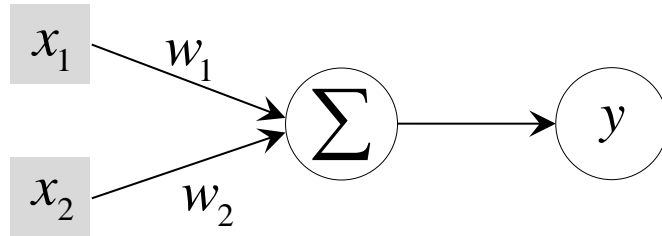
HYUN SOO PARK

# CHALLENGES OF VISUAL RECOGNITION

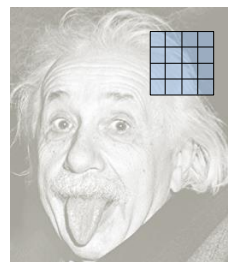
- Appearance
  - DOF: texture, illumination, material, shading, ...
- **Shape**
  - DOF: object category, geometric pose, viewpoint, ...



# *RECALL: MULTIPLICATION LAYER*

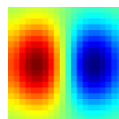


# *CONVOLUTIONAL LAYER*



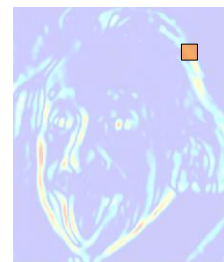
$H \times W$

\*



$F \times F$

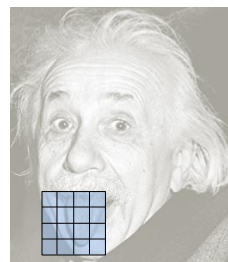
=



$H \times W$

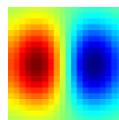
**Feature map**

# *CONVOLUTIONAL LAYER*



$H \times W$

\*



$F \times F$

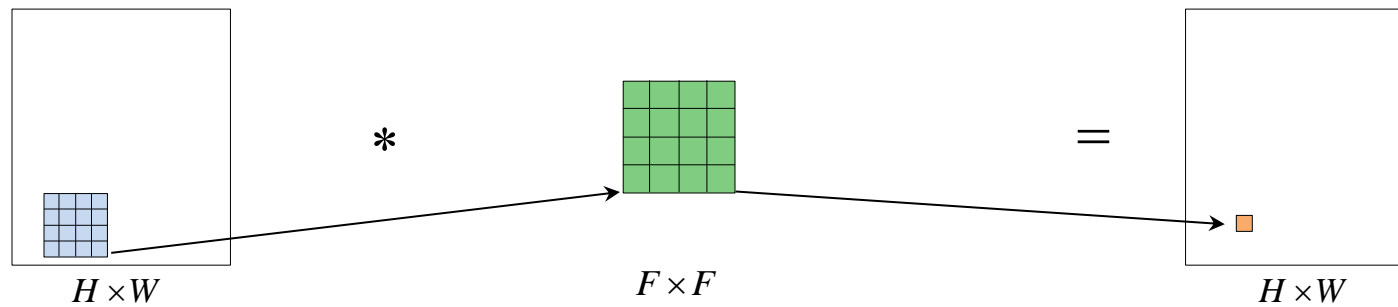
=



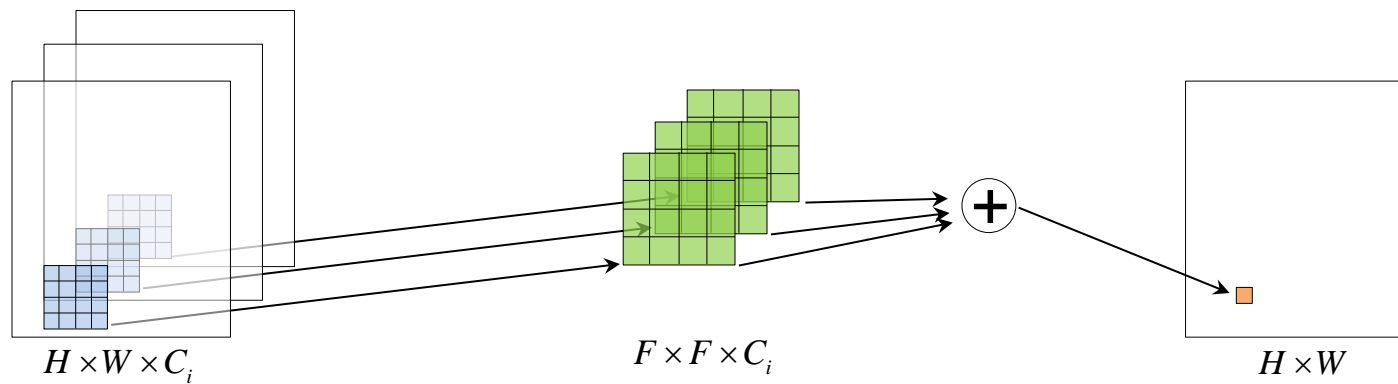
$H \times W$

**Feature map**

# *CONVOLUTIONAL LAYER*



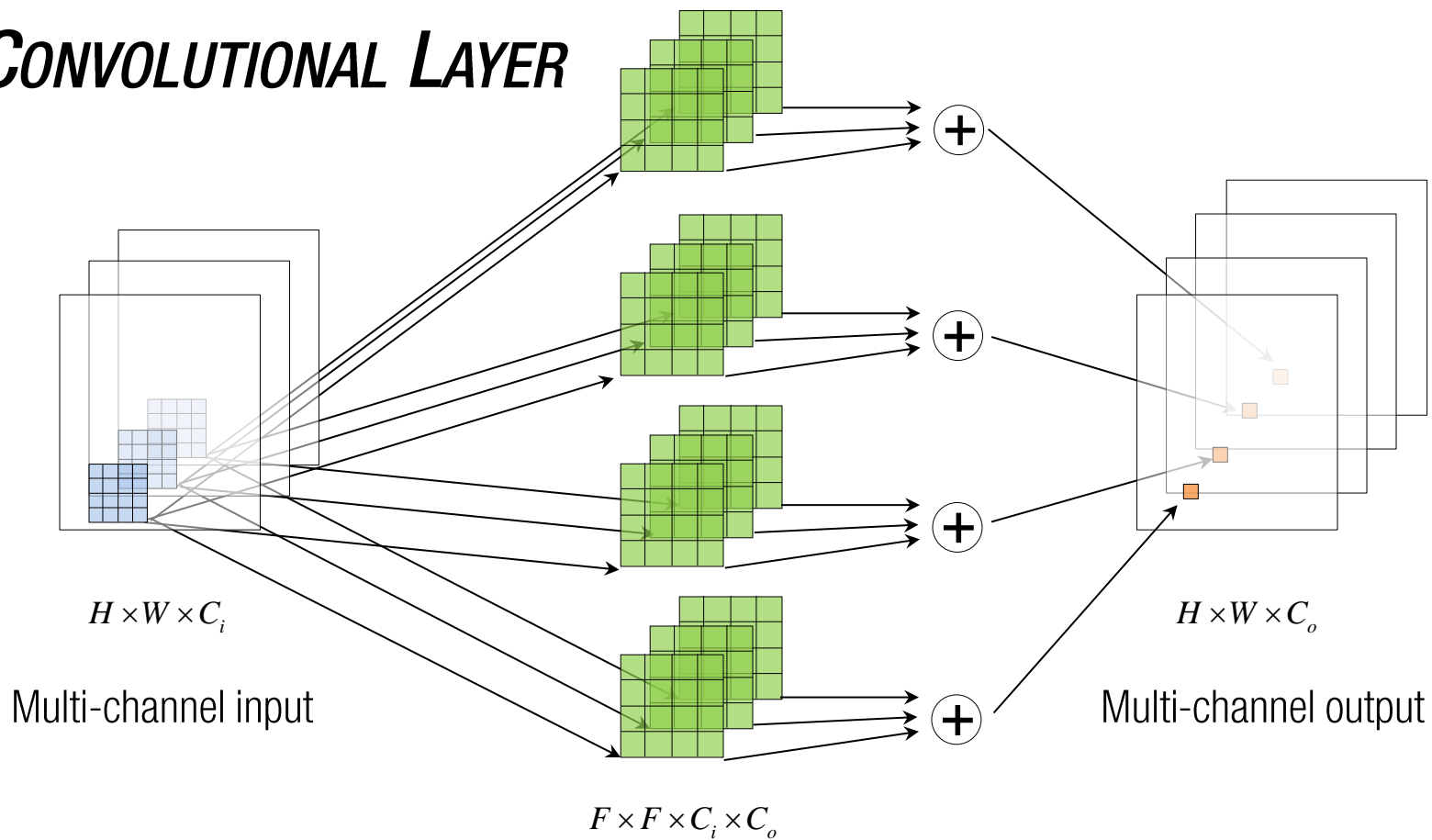
# CONVOLUTIONAL LAYER



Multi-channel input

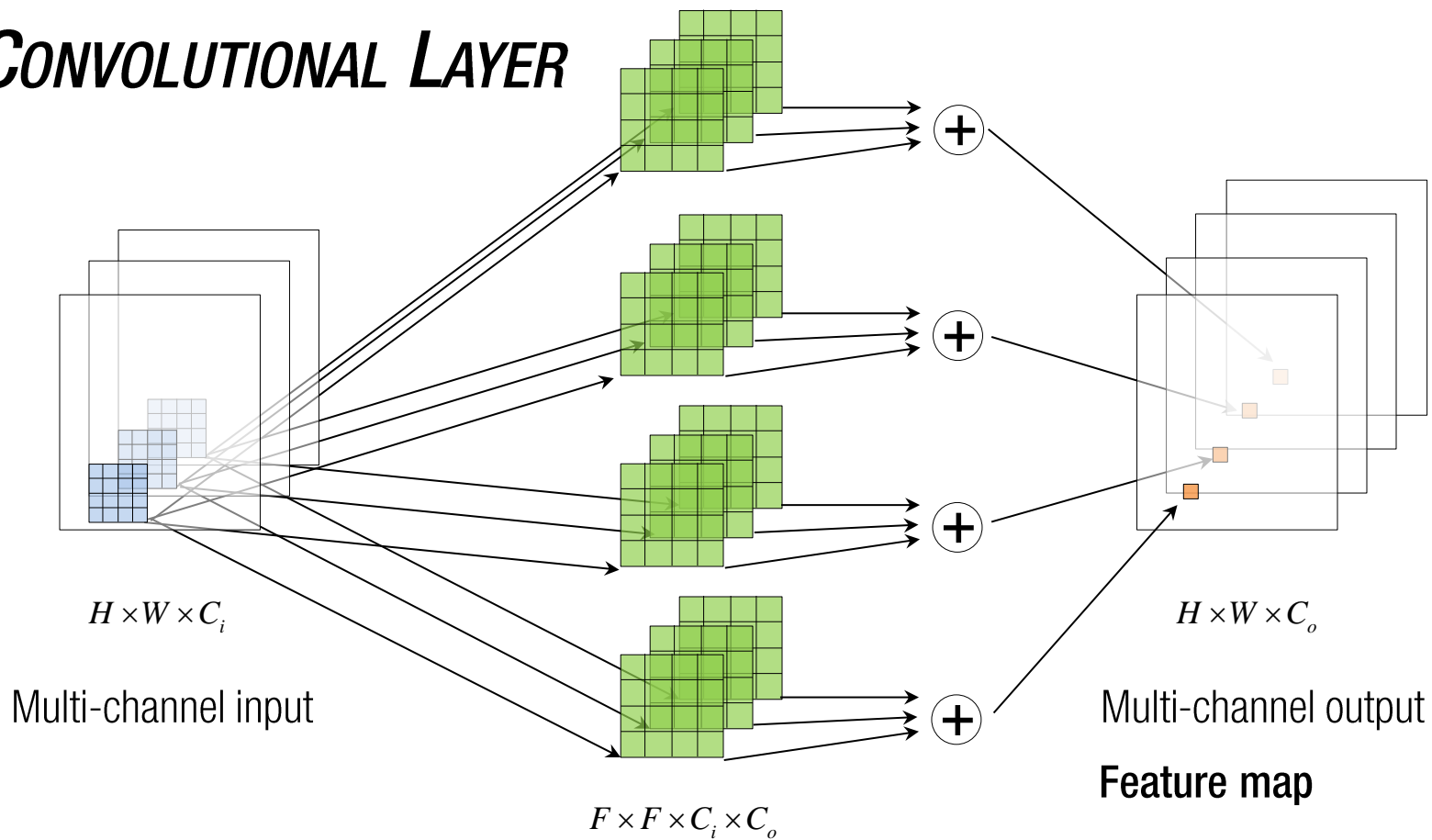


# CONVOLUTIONAL LAYER

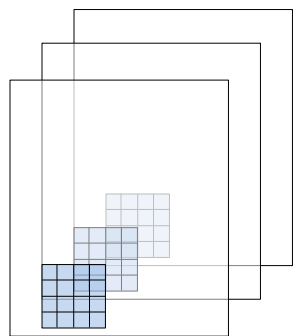




# CONVOLUTIONAL LAYER

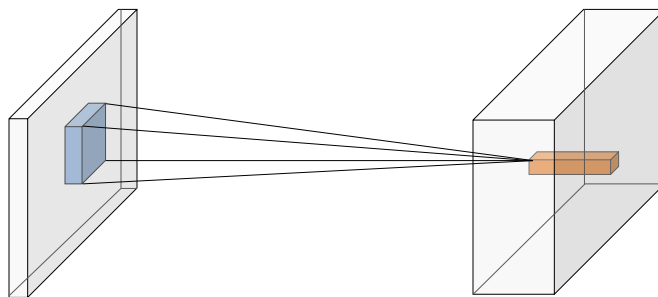


# *CONVOLUTIONAL LAYER*



$$H \times W \times C_i$$

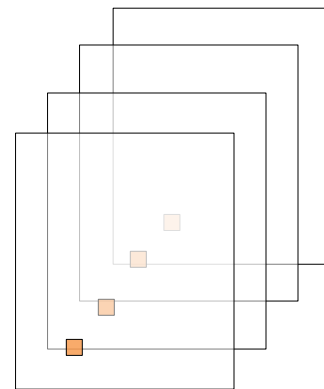
Multi-channel input



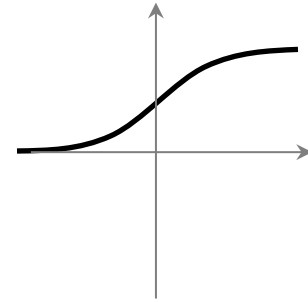
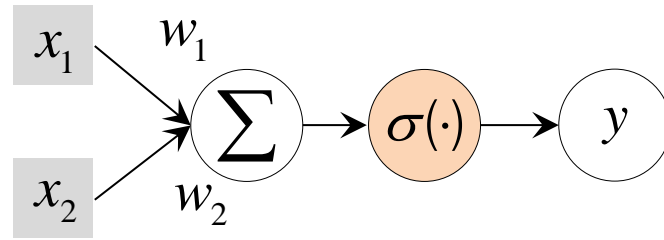
$$H \times W \times C_o$$

Multi-channel output

**Feature map**



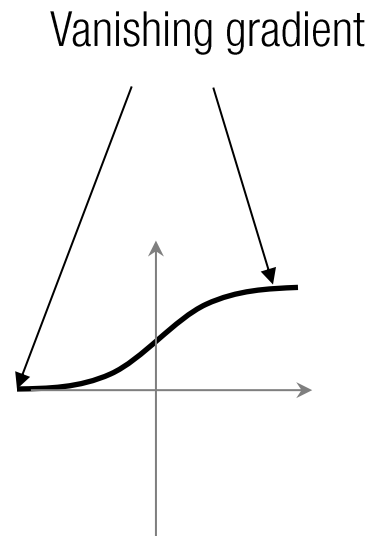
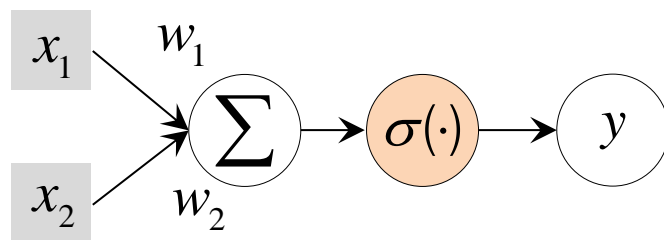
# RECALL: ACTIVATION LAYER



Sigmoid

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

# RECALL: ACTIVATION LAYER

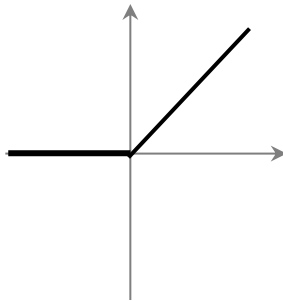


Sigmoid

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

# *ACTIVATION: ReLU*

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

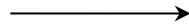


Rect. Linear (ReLU)

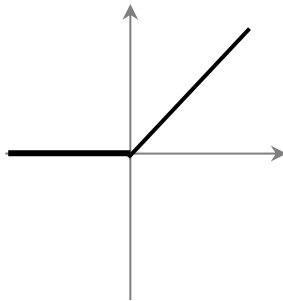
$$\sigma(x) = \max(0, x)$$

# 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

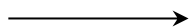


Rect. Linear (ReLU)

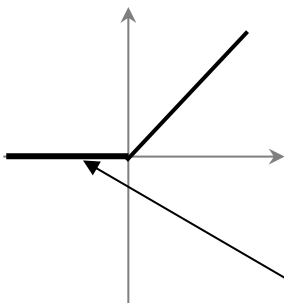
$$\sigma(x) = \max(0, x)$$

# 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)

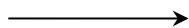
$$\sigma(x) = \max(0, x)$$

Vanishing gradient



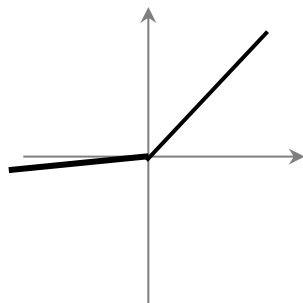
# ACTIVATION: LEAKY RELU

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)***

3	2	3	1
0	<b>5</b>	3	4
1	2	2	2
<b>7</b>	3	1	7

4×4

<b>5</b>	4
7	7

2×2

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

# ***SPATIAL POOLING (SAMPLING~DIMENSIONAL REDUCTION)***

3	2	3	1
0	<b>5</b>	3	4
1	2	2	2
7	3	1	7

4×4

<b>5</b>		

3×3

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

# ***SPATIAL POOLING (SAMPLING~DIMENSIONAL REDUCTION)***

3	2	3	1
0	<b>5</b>	3	4
1	2	2	2
7	3	1	7

4×4

<b>5</b>	<b>5</b>	

3×3

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

# ***SPATIAL POOLING (SAMPLING~DIMENSIONAL REDUCTION)***

3	2	<b>3</b>	1
0	5	<b>3</b>	<b>4</b>
1	2	2	2
7	3	1	7

4×4

<b>5</b>	<b>5</b>	<b>4</b>

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

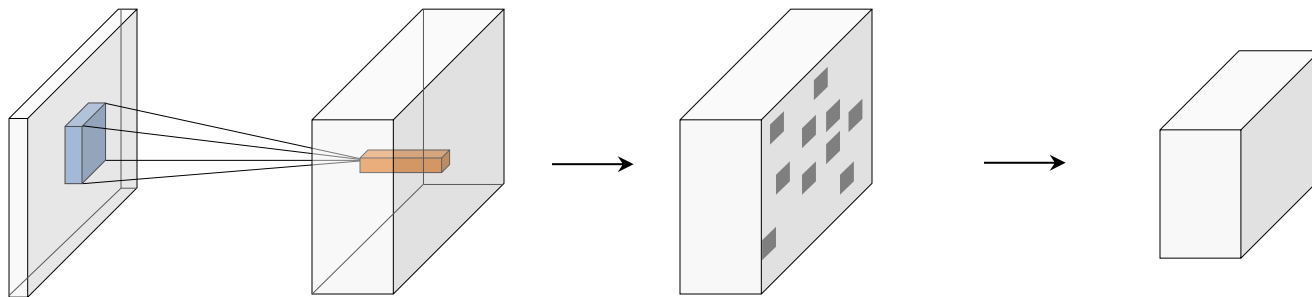
<b>5</b>	<b>5</b>	<b>4</b>
<b>5</b>	<b>5</b>	<b>4</b>
<b>7</b>	<b>3</b>	<b>7</b>

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:

$$H \times W \times C_1 \times C_2$$

0

0

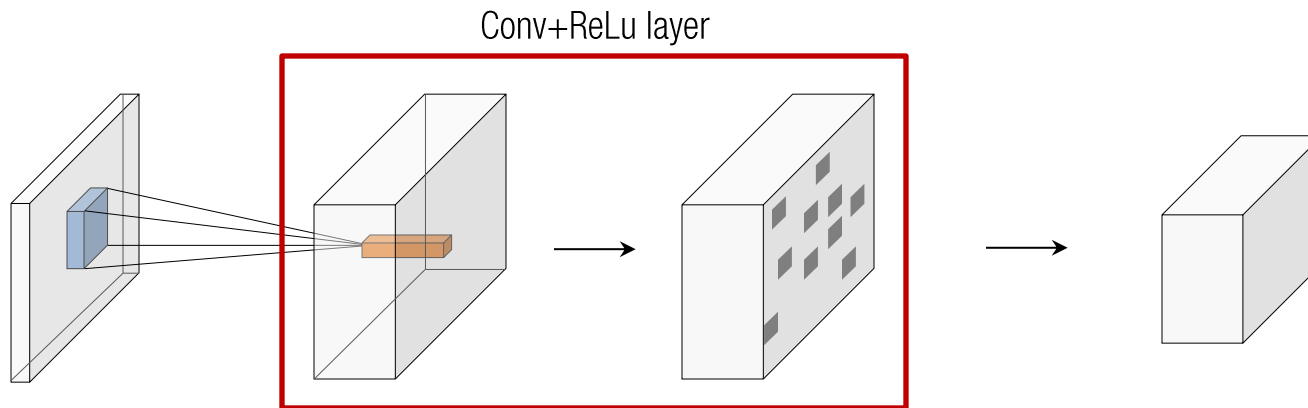
# of biases:

$$1 \times C_2$$

0

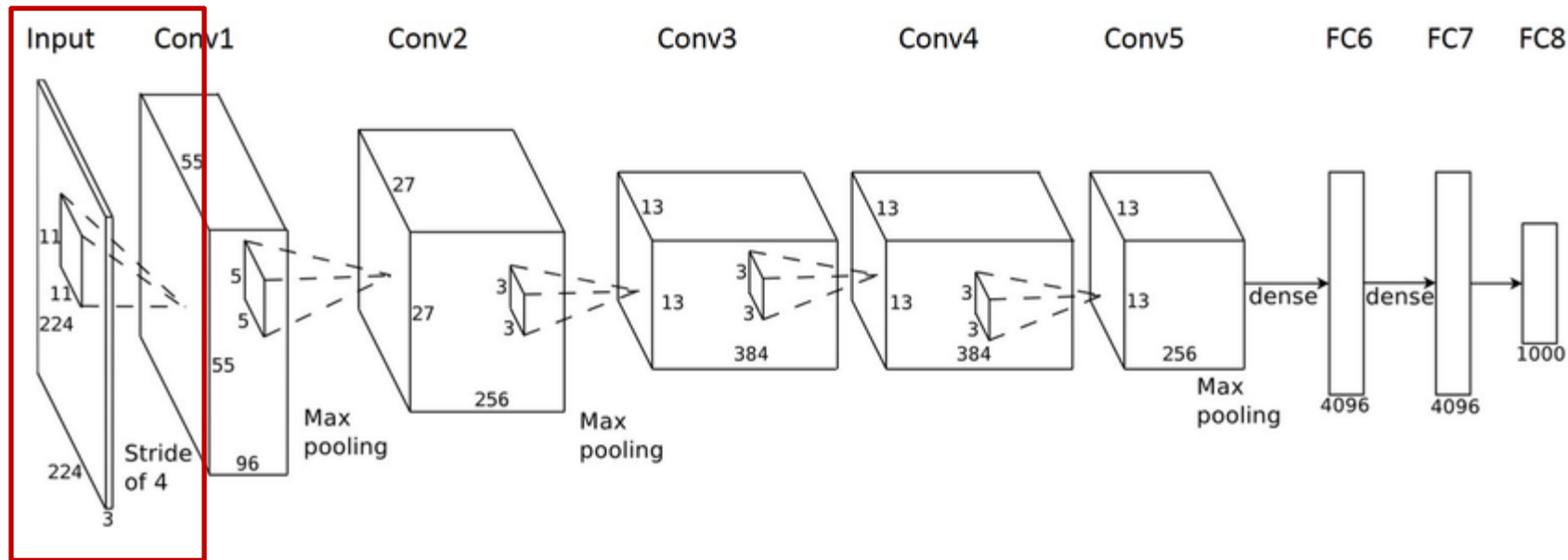
0

# CONV+RELU+POOL



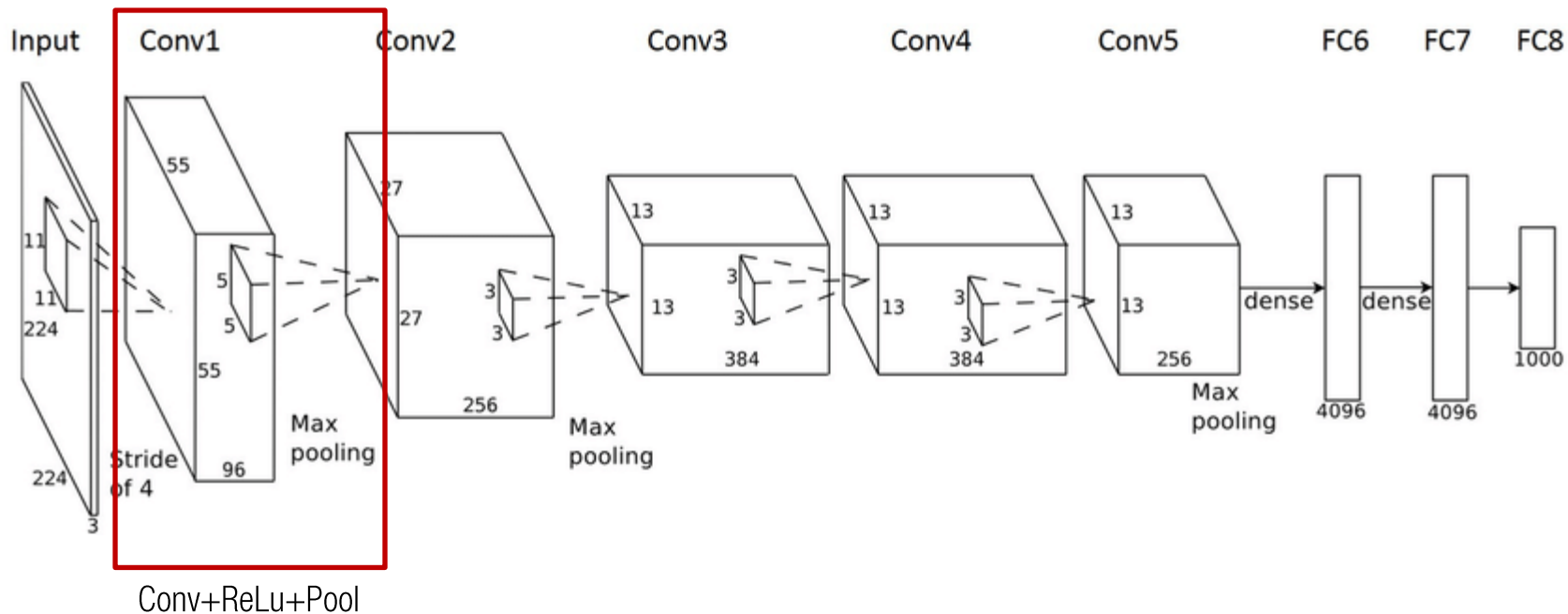
Operations:		Conv	ReLU	Max-pool
# of units:	$H \times W \times C_1$	$H \times W \times C_2$	$H \times W \times C_2$	$H_1 \times W_1 \times C_2$
# of weights:	$H \times W \times C_1 \times C_2$	0	0	0
# of biases:	$1 \times C_2$	0	0	0

# ALEX NET

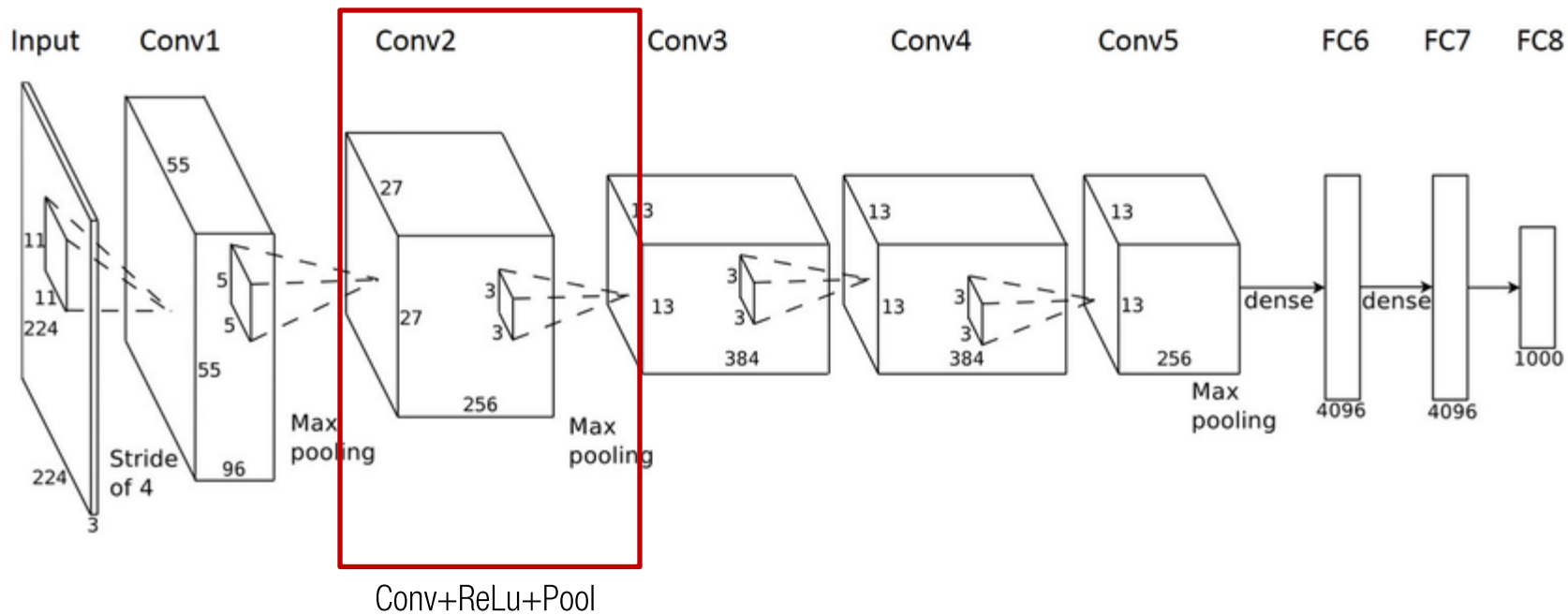


Conv+ReLu

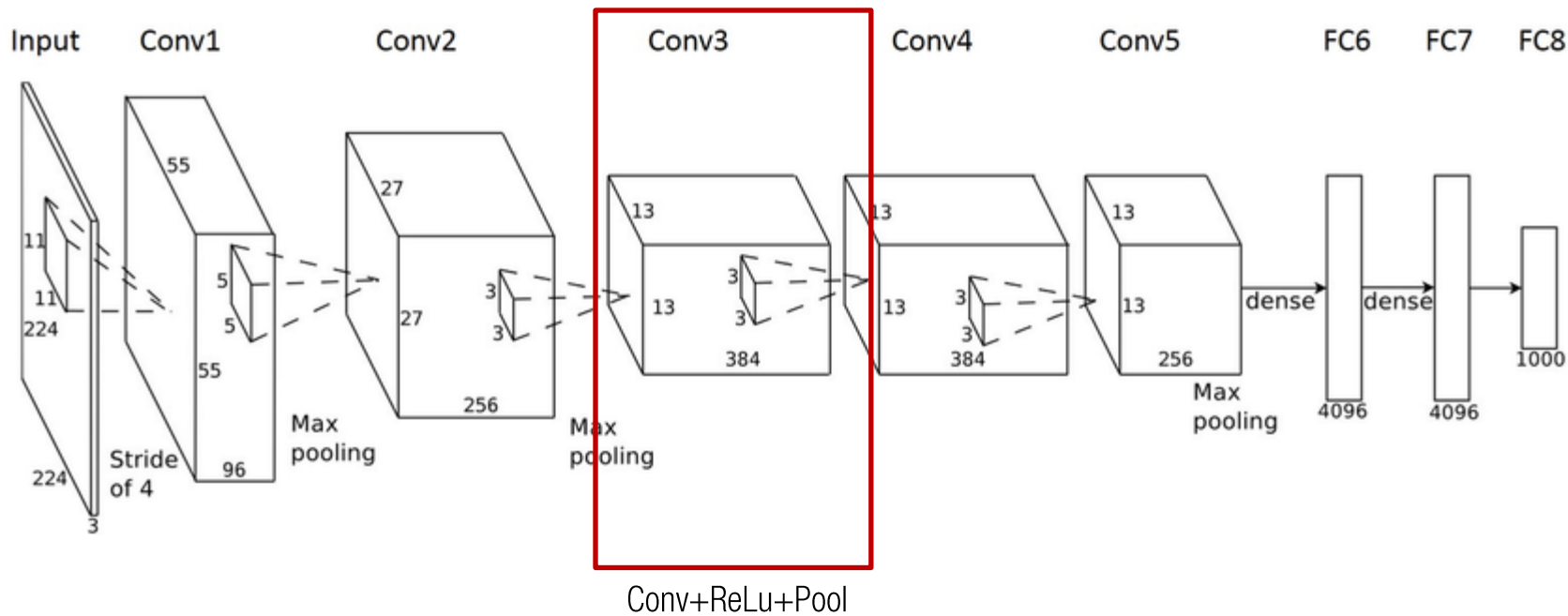
# ALEX NET



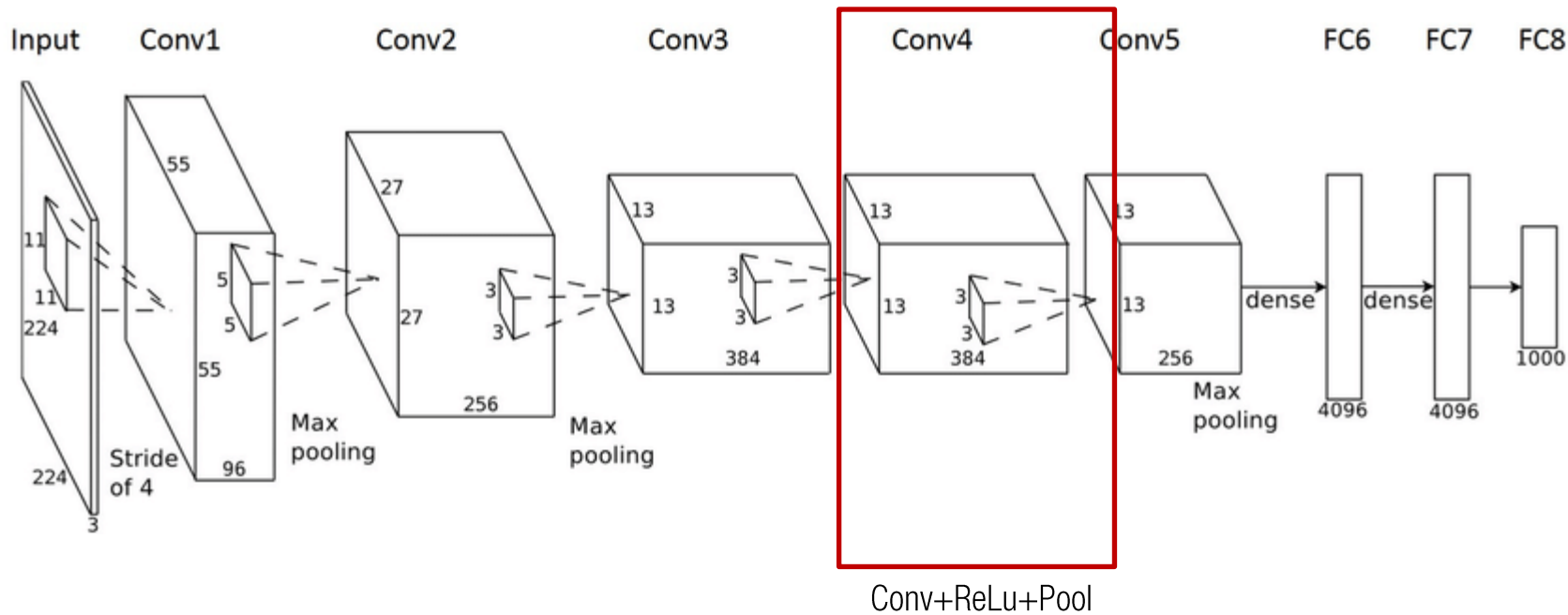
# ALEX NET



# ALEX NET

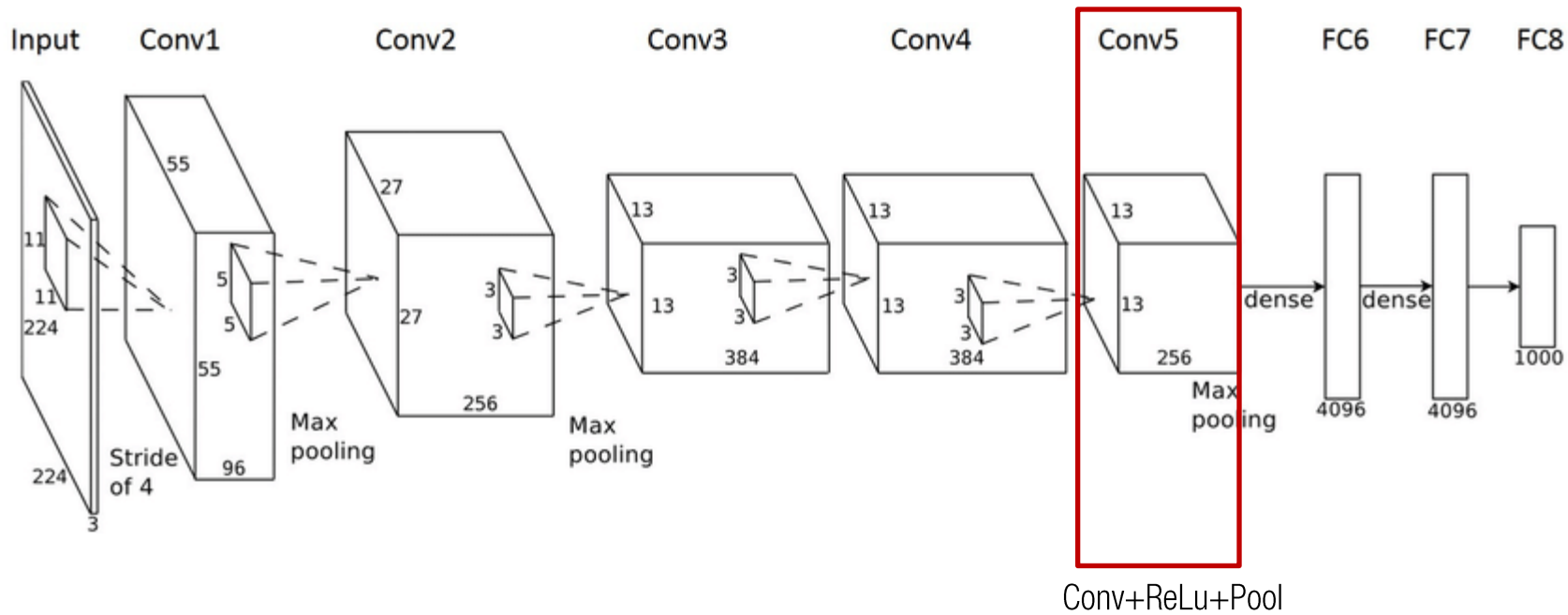


# ALEX NET

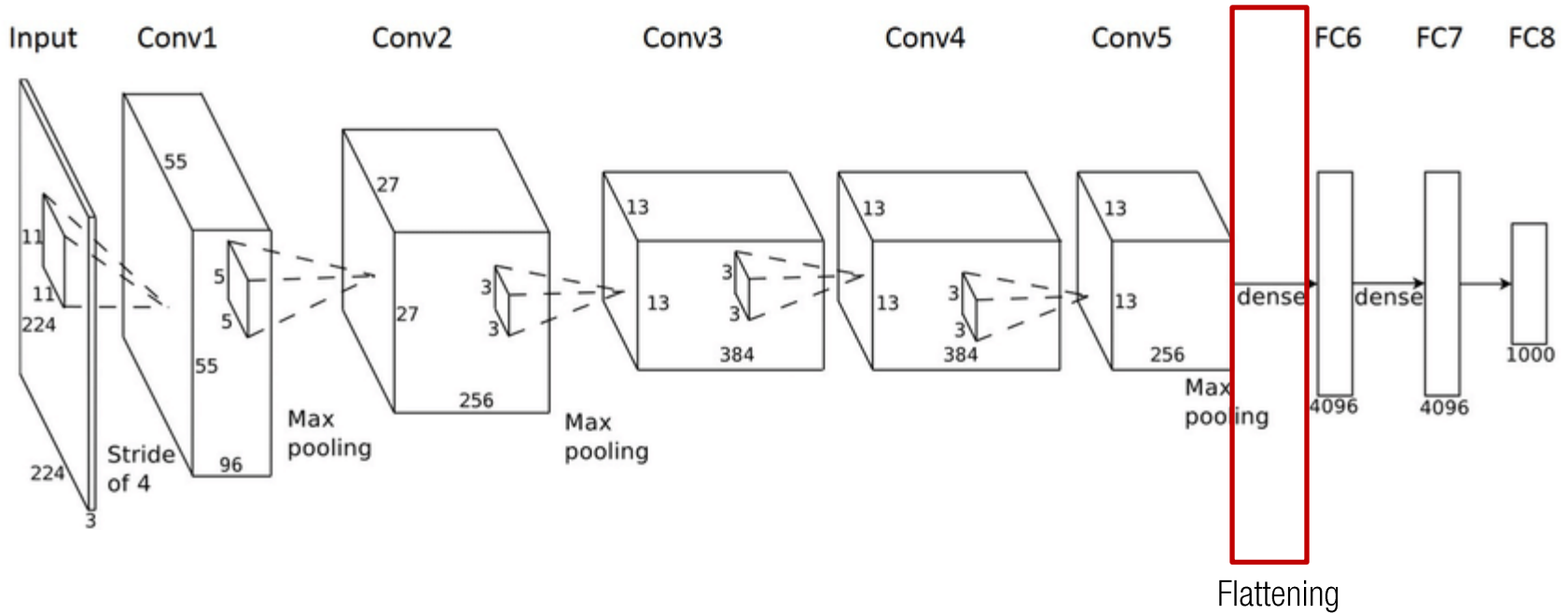




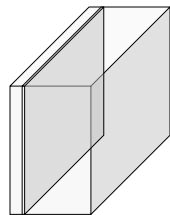
# ALEX NET



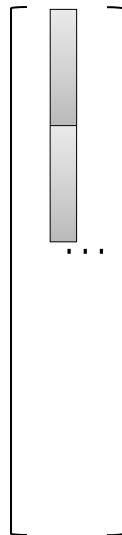
# TENSOR FLATTENING



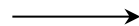
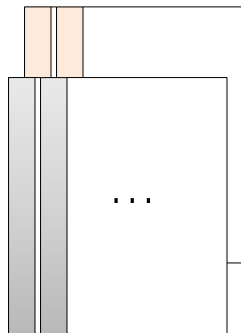
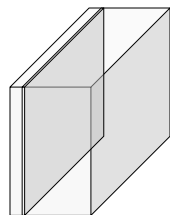
# *TENSOR FLATTENING*



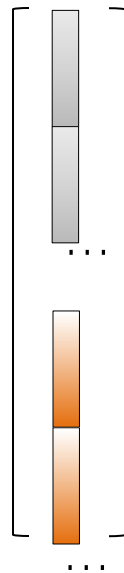
Vectorize



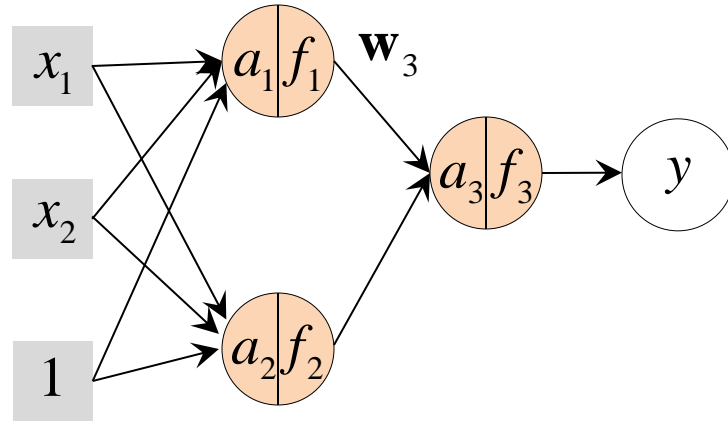
# *TENSOR FLATTENING*



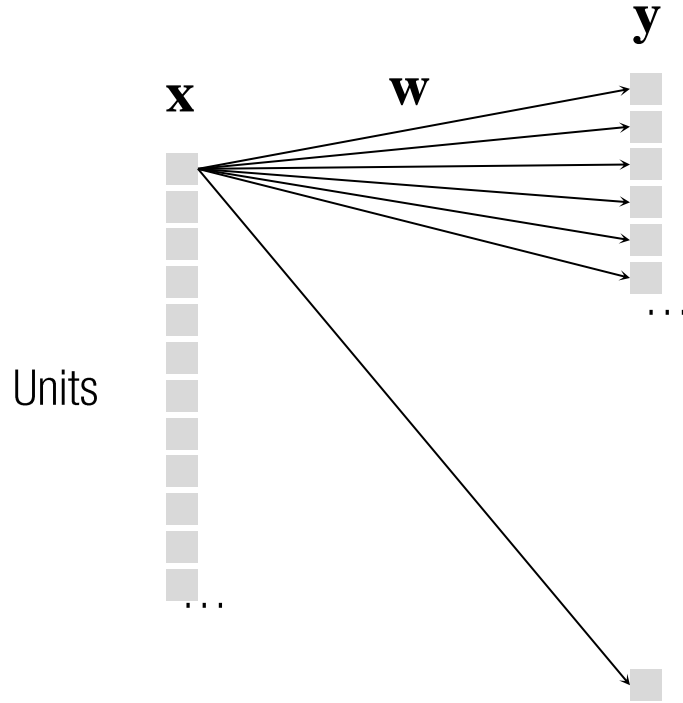
Vectorize



# *RECALL: HIDDEN LAYER*

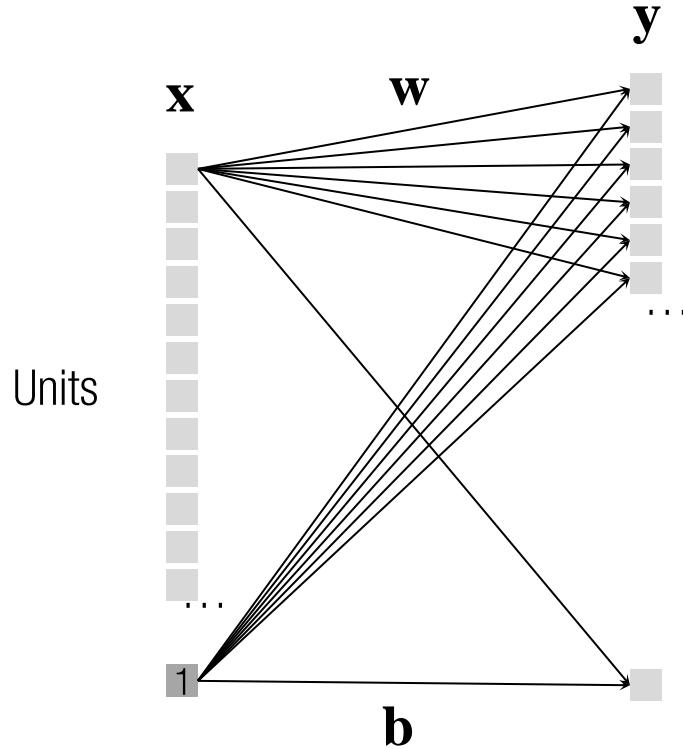


# *FULLY CONNECTED LAYER (FC)*



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

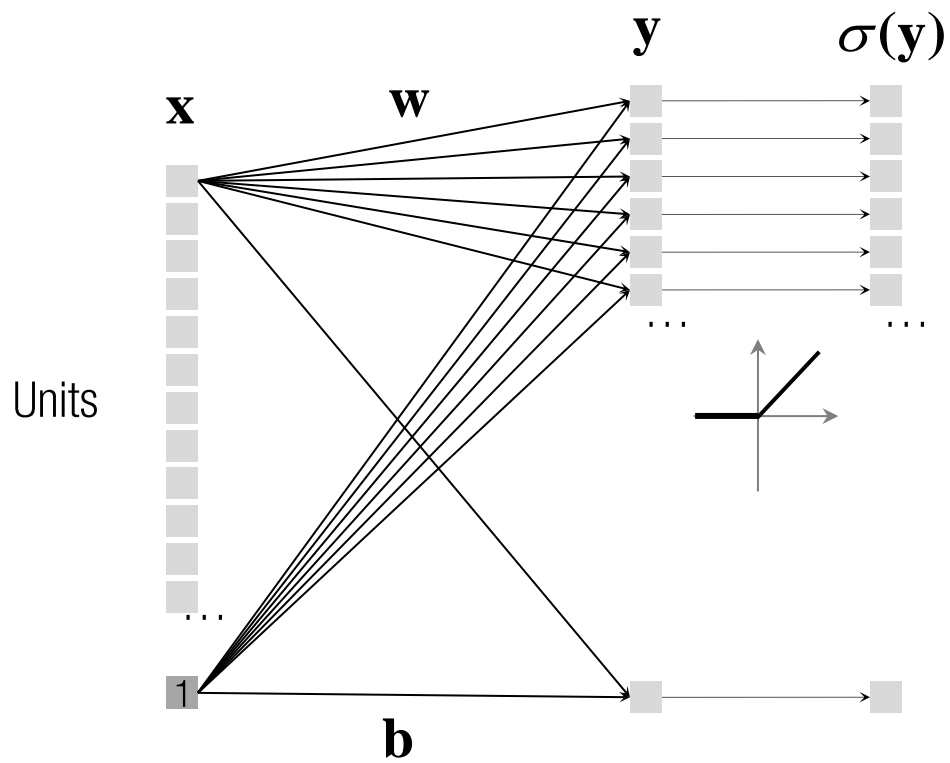
# *FULLY CONNECTED LAYER (FC)*



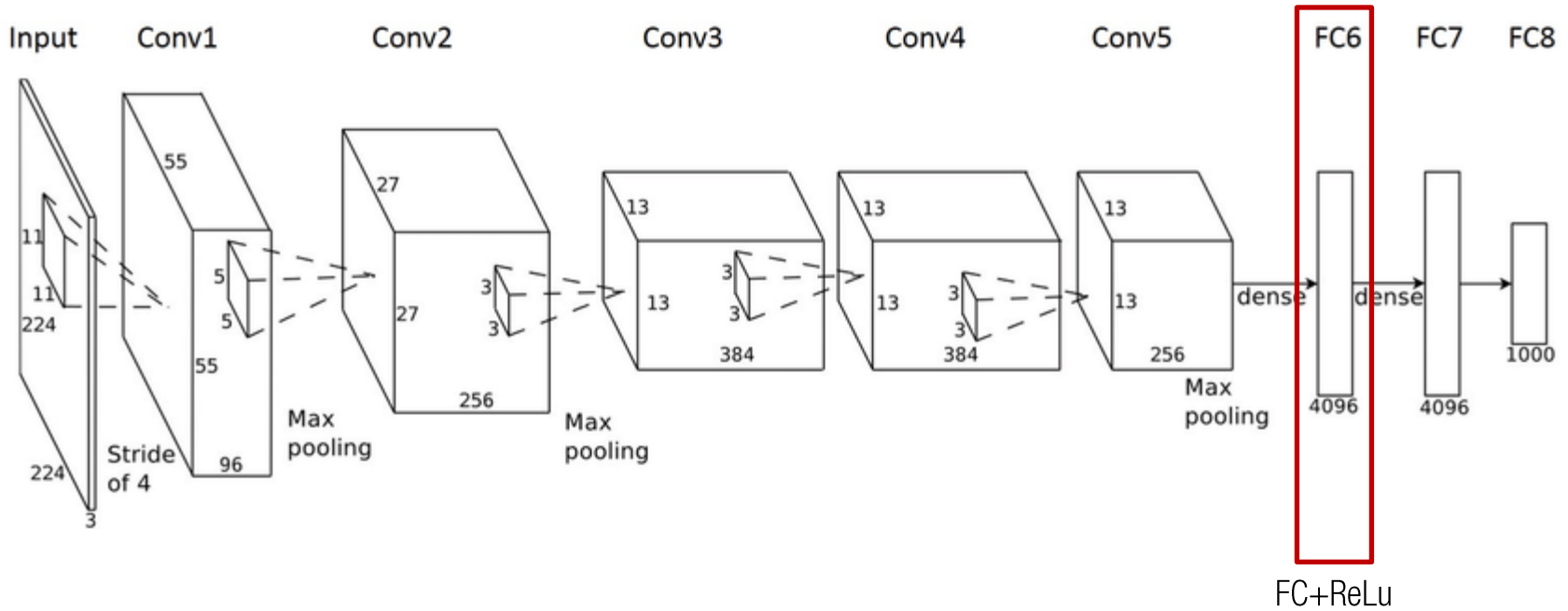
$$\mathbf{y} = \mathbf{W} \begin{bmatrix} \mathbf{x} \\ 1 \end{bmatrix} + \mathbf{b}$$



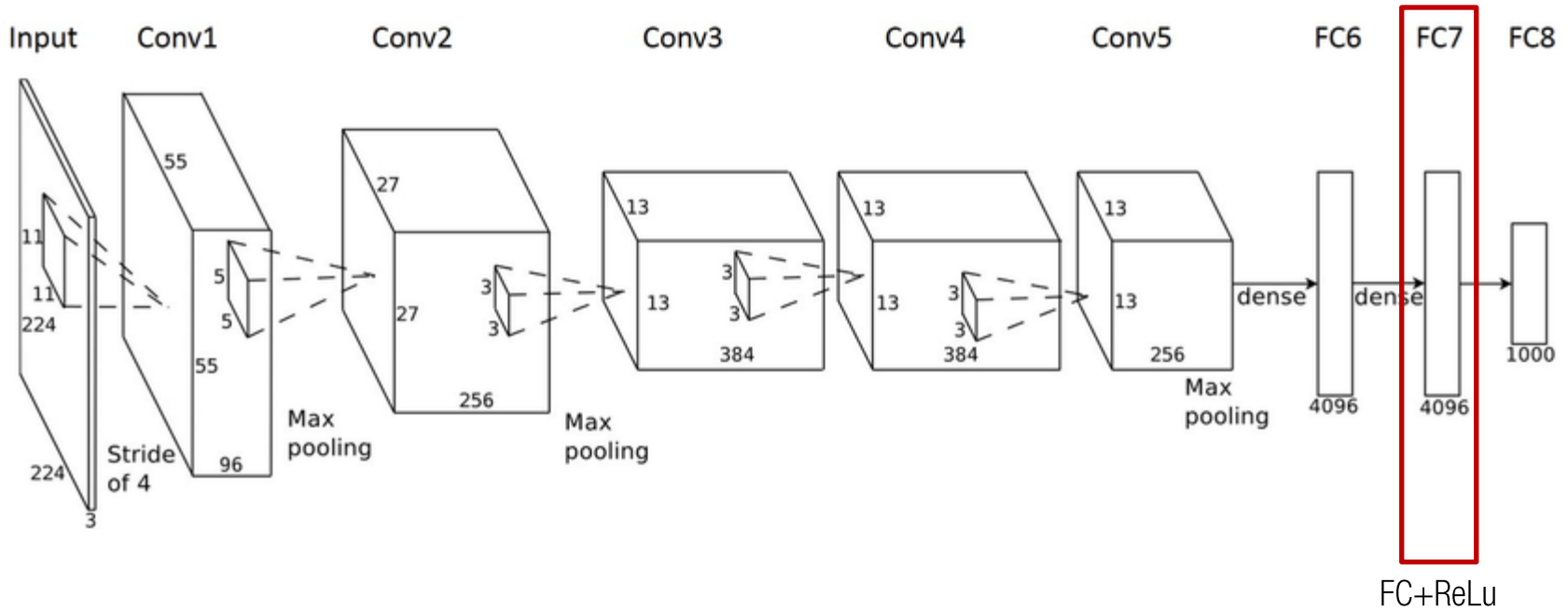
# ***FC+RELU***



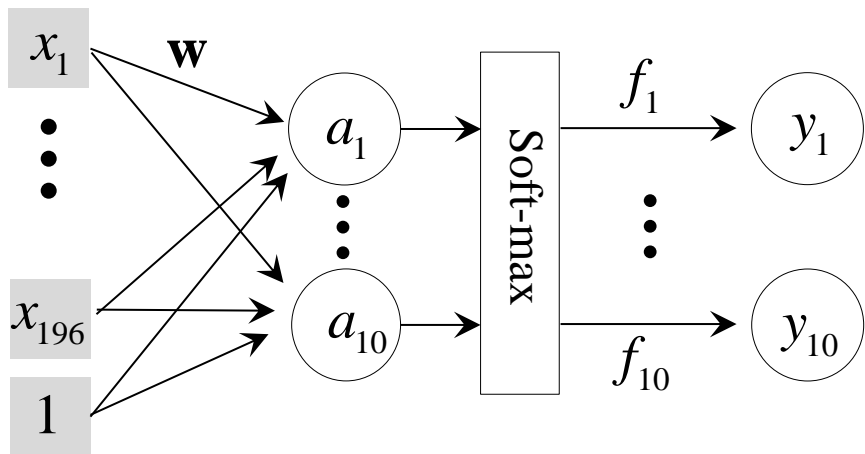
# TENSOR FLATTENING



# TENSOR FLATTENING

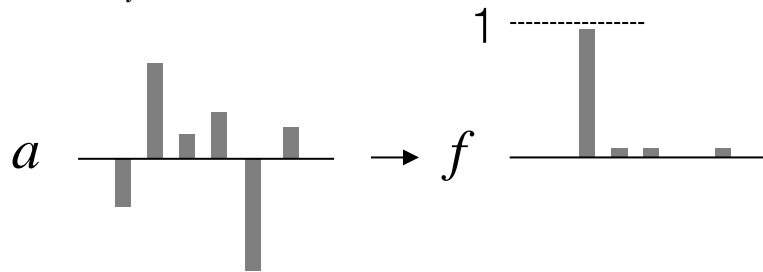


# RECALL: SOFT-MAX



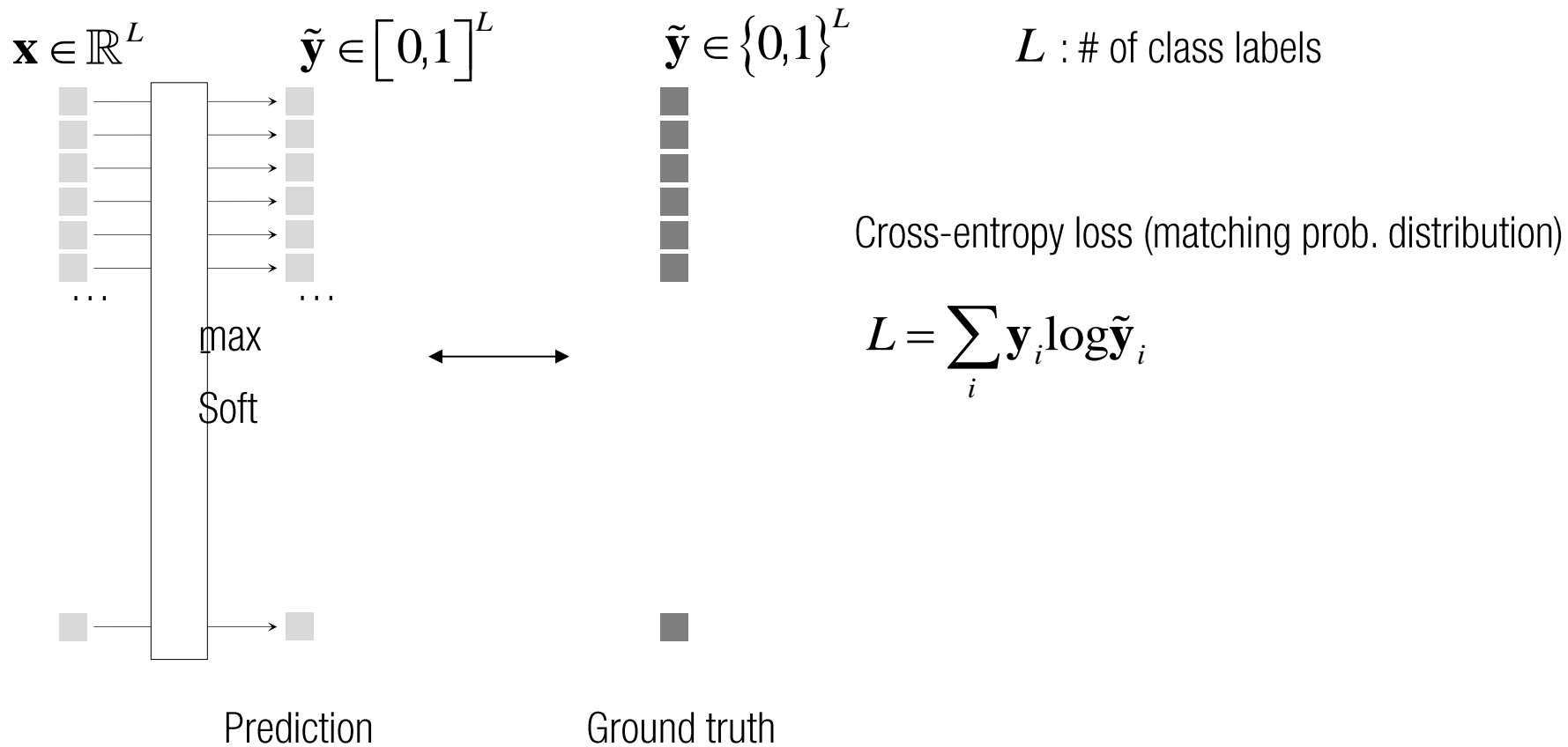
Soft-max (winners-take-all):

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

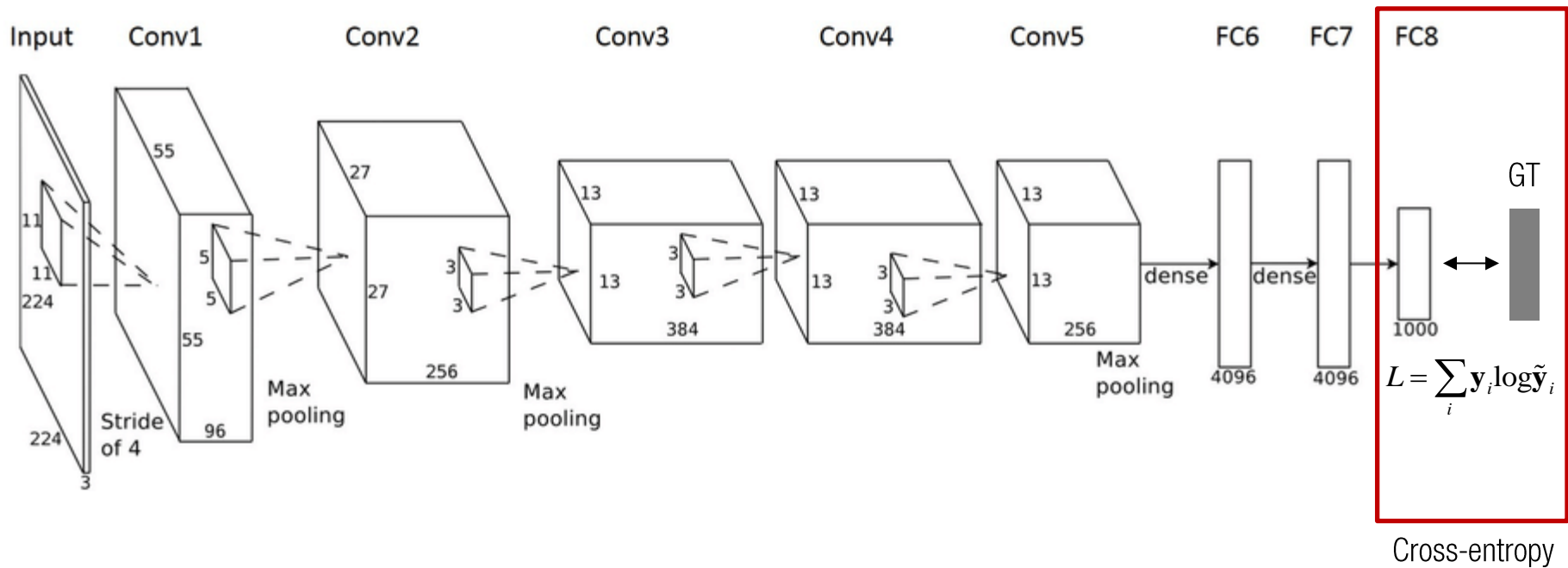


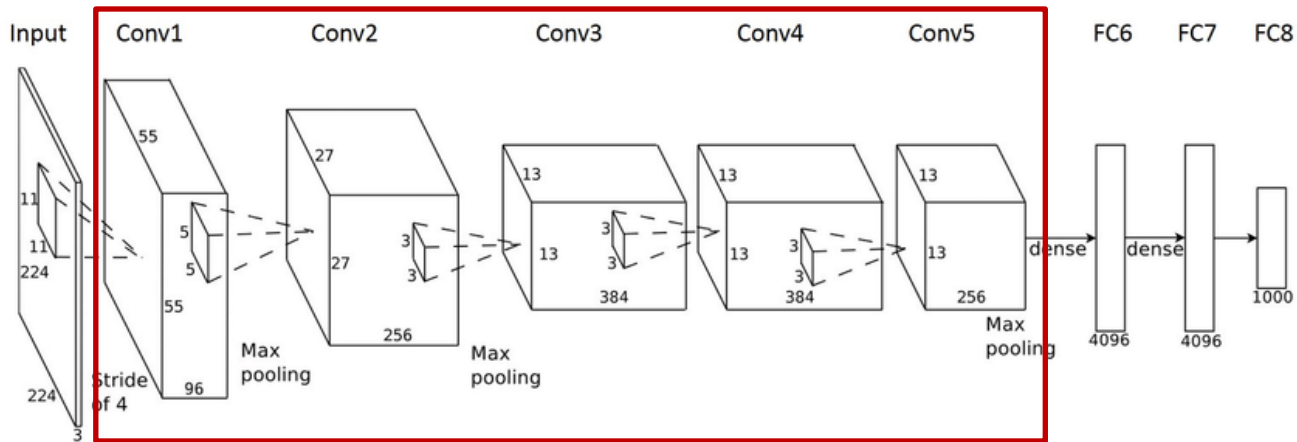
$$\frac{\partial f_i}{\partial a_j} = \begin{cases} f_i(1 - f_i) & i = j \\ -f_i f_j & i \neq j \end{cases}$$

# ***ERROR MEASURE (LOSS)***



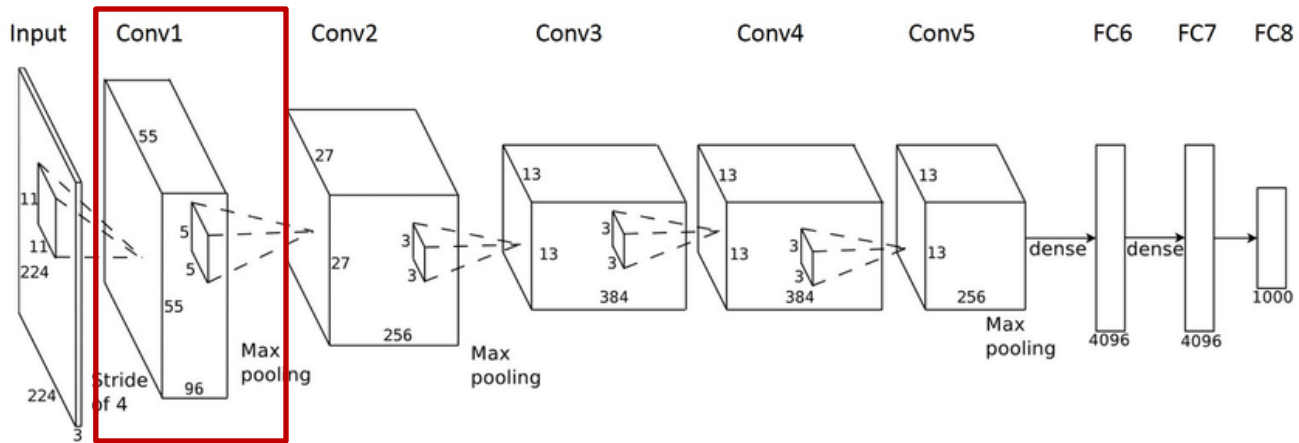
# TENSOR FLATTENING





Local feature extraction

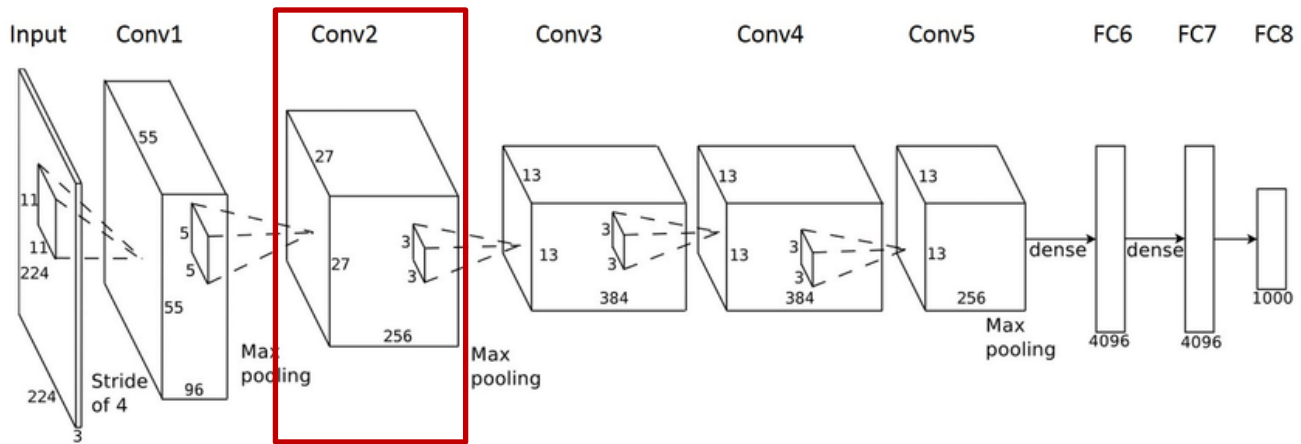




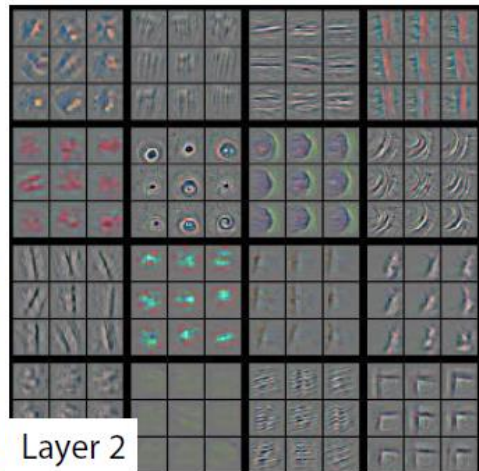
Layer 1



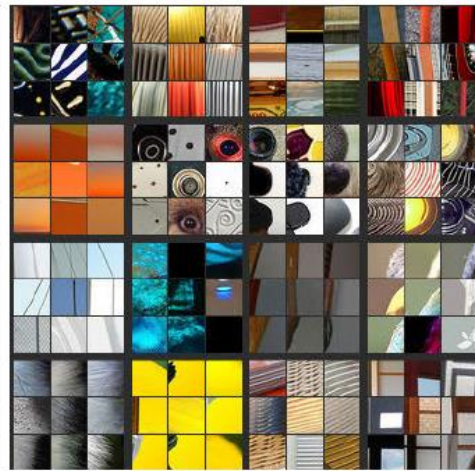


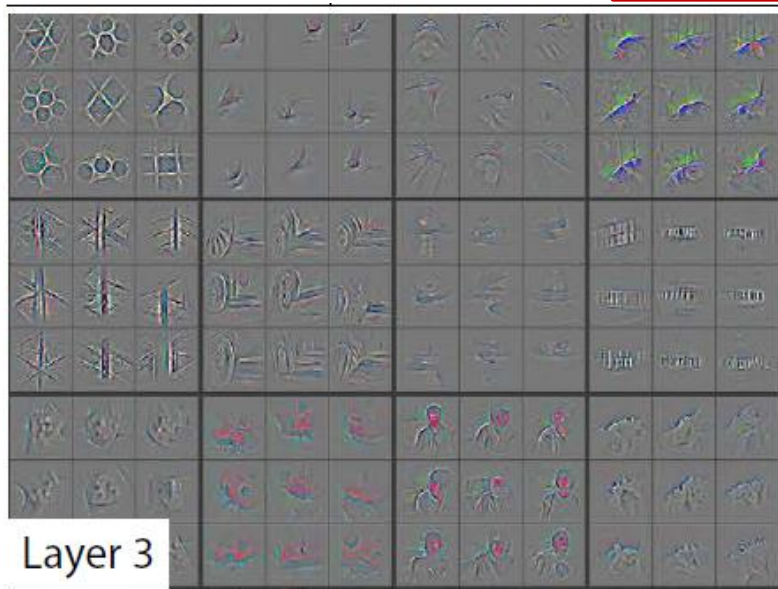
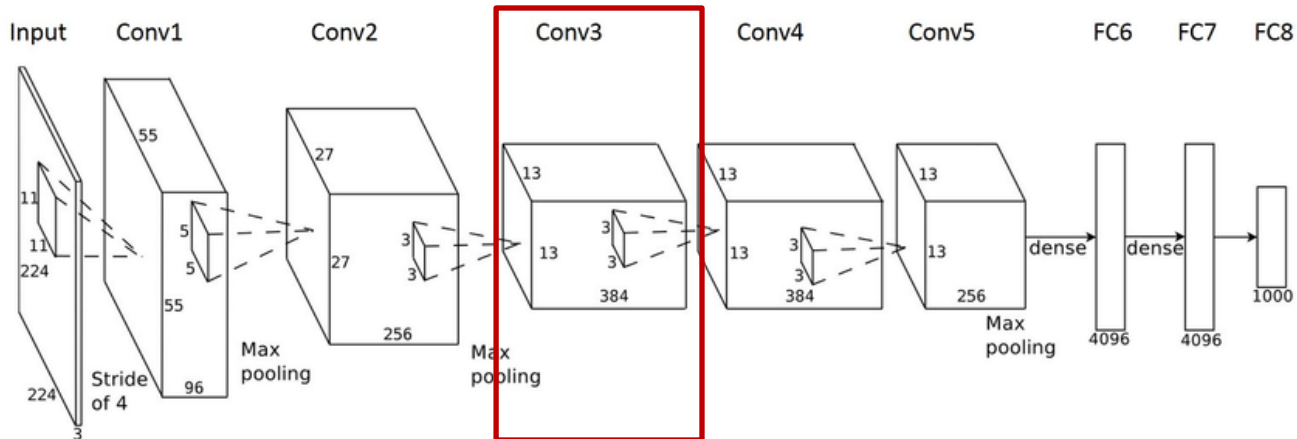


Layer 1

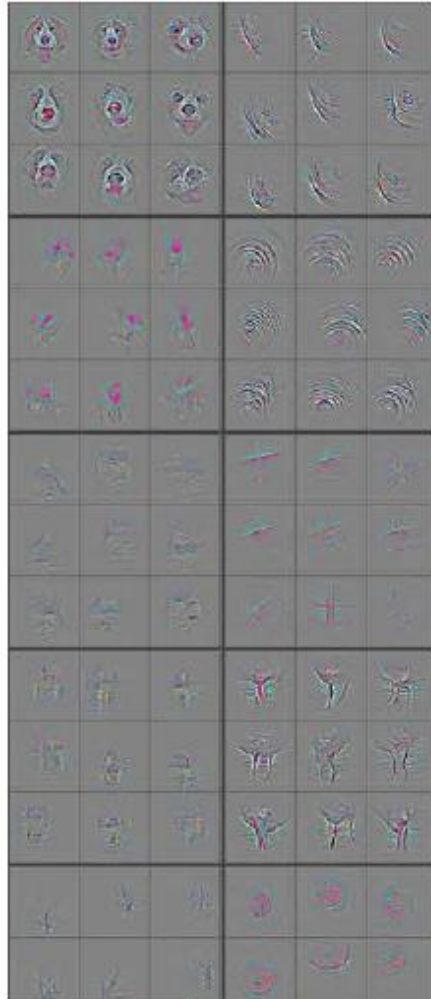


Layer 2

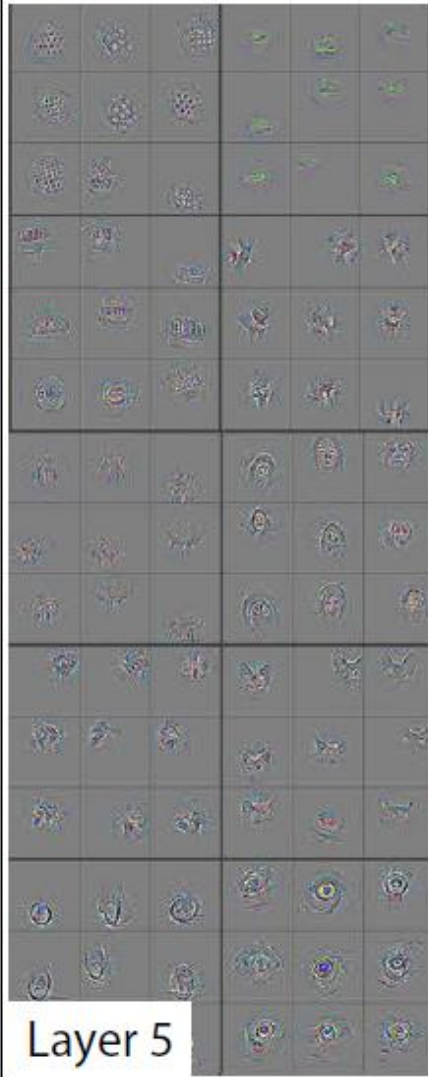






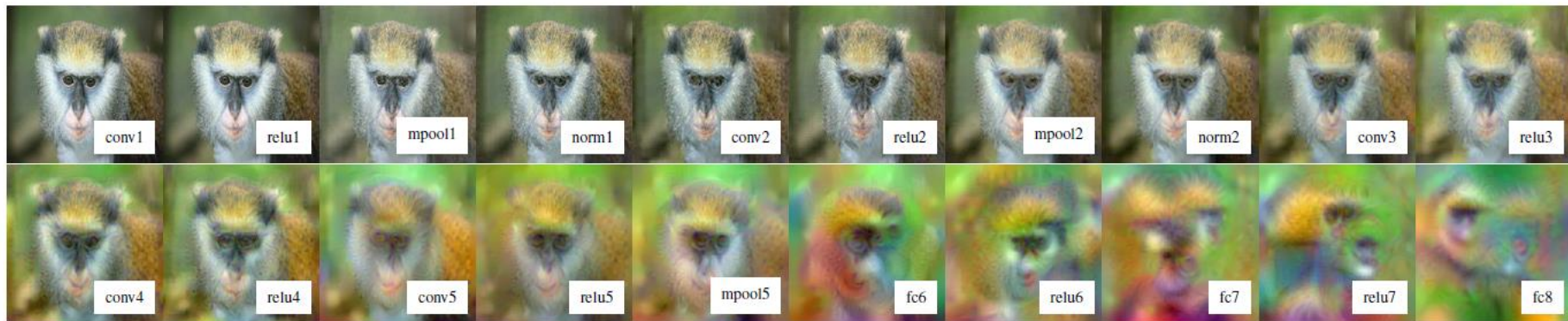
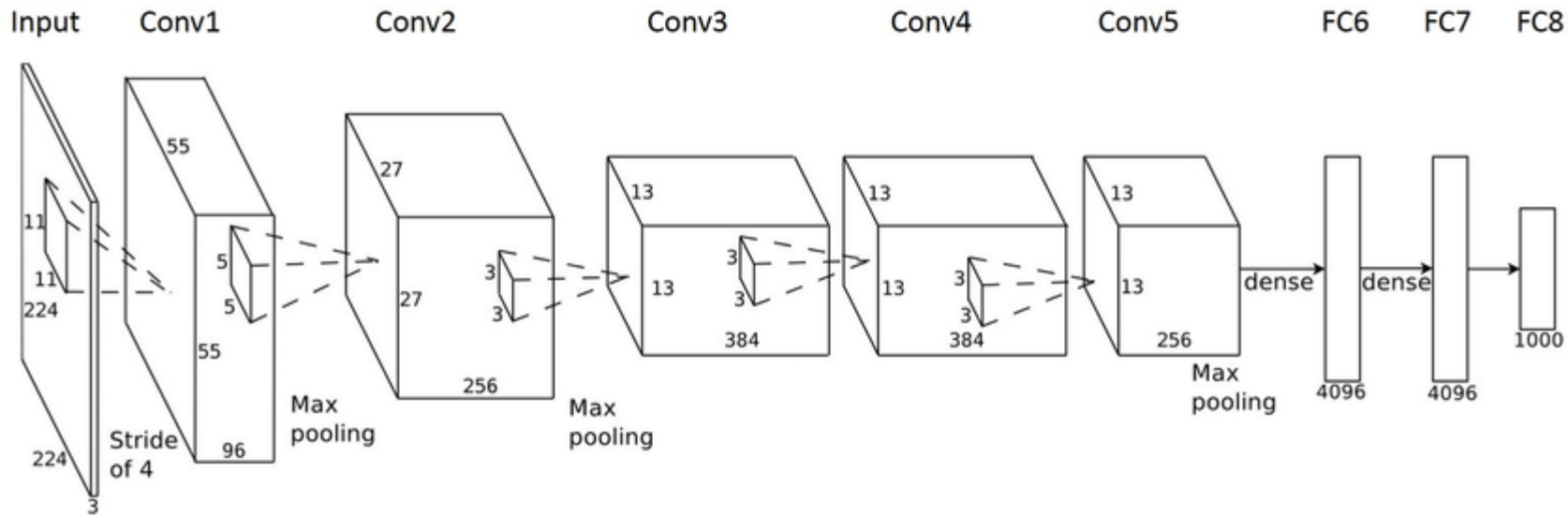


Layer 4



Layer 5





<https://www.youtube.com/watch?v=AgkflQ4IGaM>

# VGG-16 (DEPTH MATTERS)

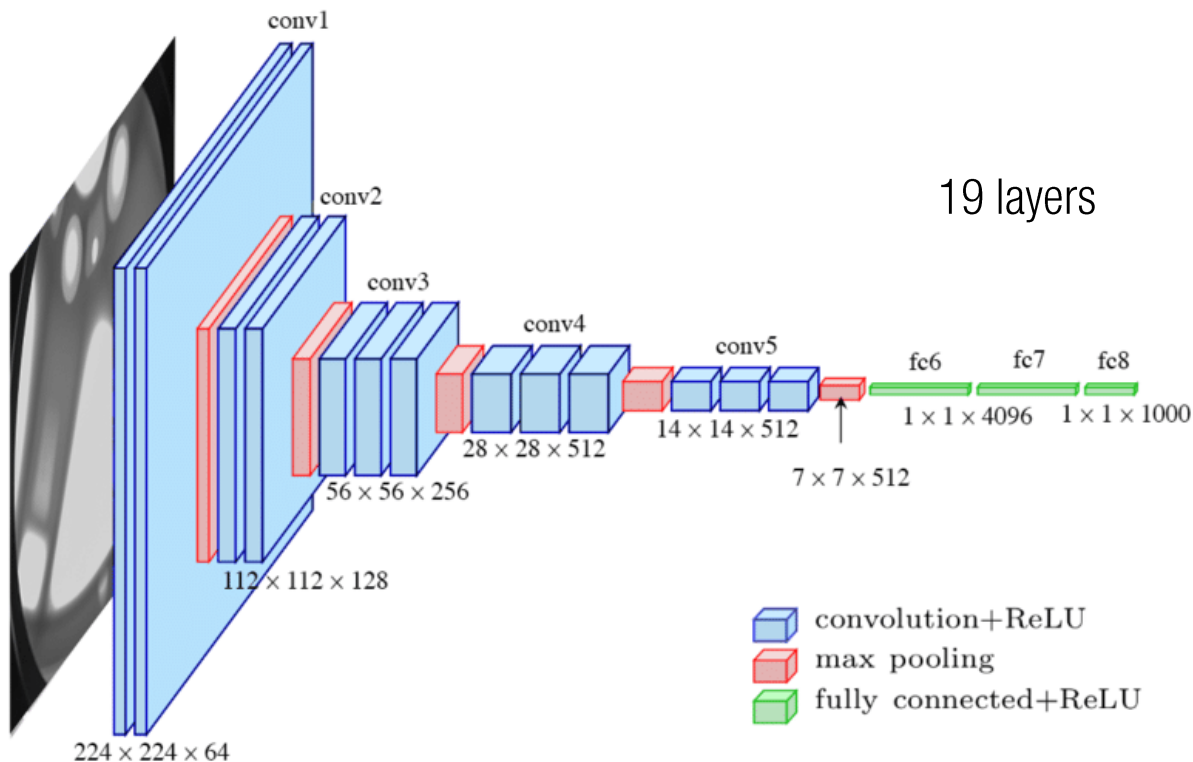




Table 7: **Comparison with the state of the art in ILSVRC classification.** Our method is denoted as “VGG”. Only the results obtained without outside training data are reported.

Method	top-1 val. error (%)	top-5 val. error (%)	top-5 test error (%)
VGG (2 nets, multi-crop & dense eval.)	<b>23.7</b>	<b>6.8</b>	<b>6.8</b>
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)	-	<b>6.7</b>	
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	-

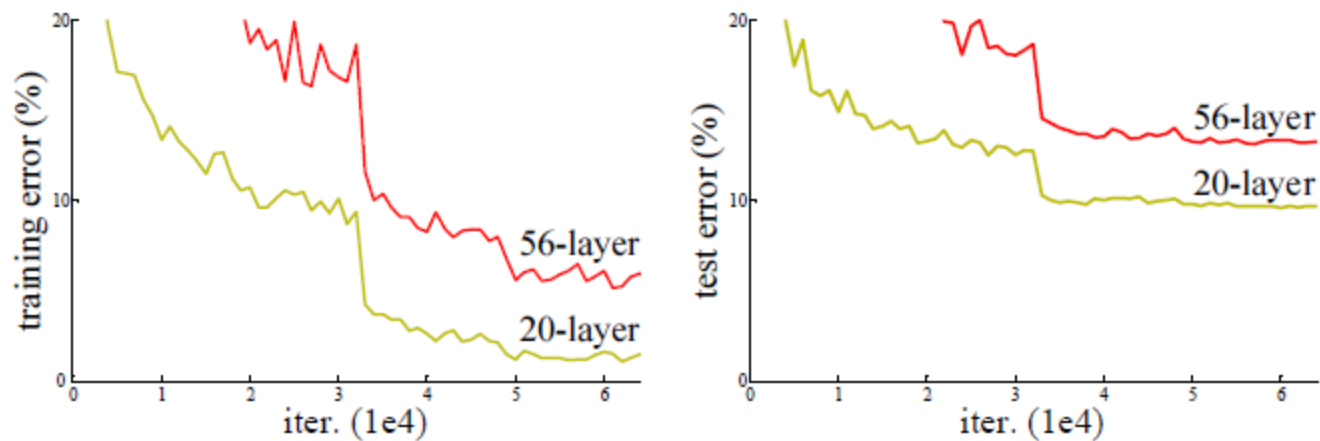
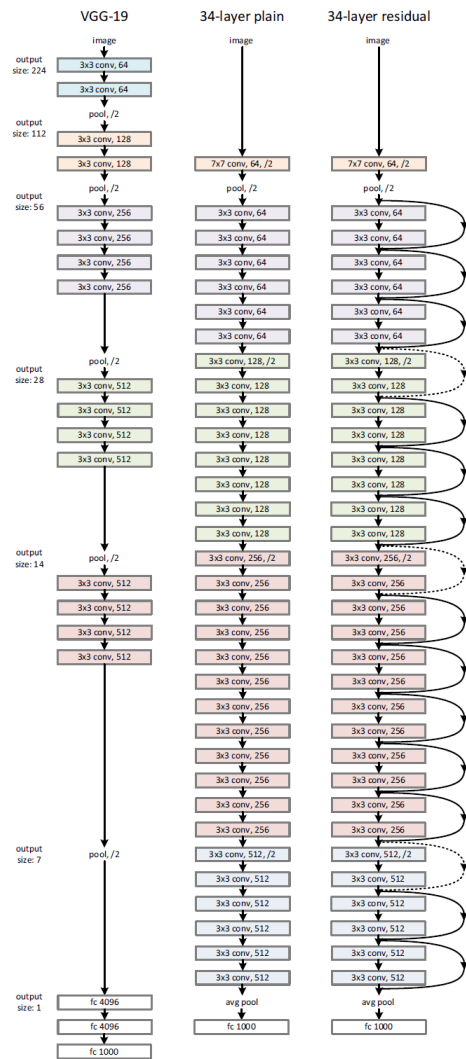
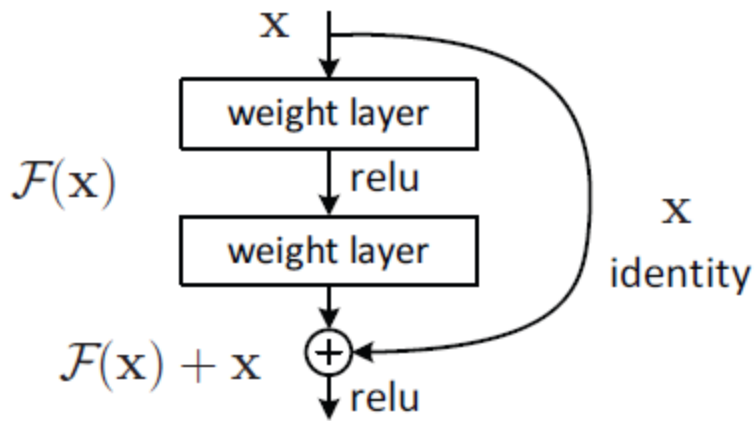


Figure 1. Training error (left) and test error (right) on CIFAR-10 with 20-layer and 56-layer “plain” networks. The deeper network has higher training error, and thus test error. Similar phenomena on ImageNet is presented in Fig. 4.



# RESNET



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
<b>ResNet (ILSVRC'15)</b>	<b>3.57</b>

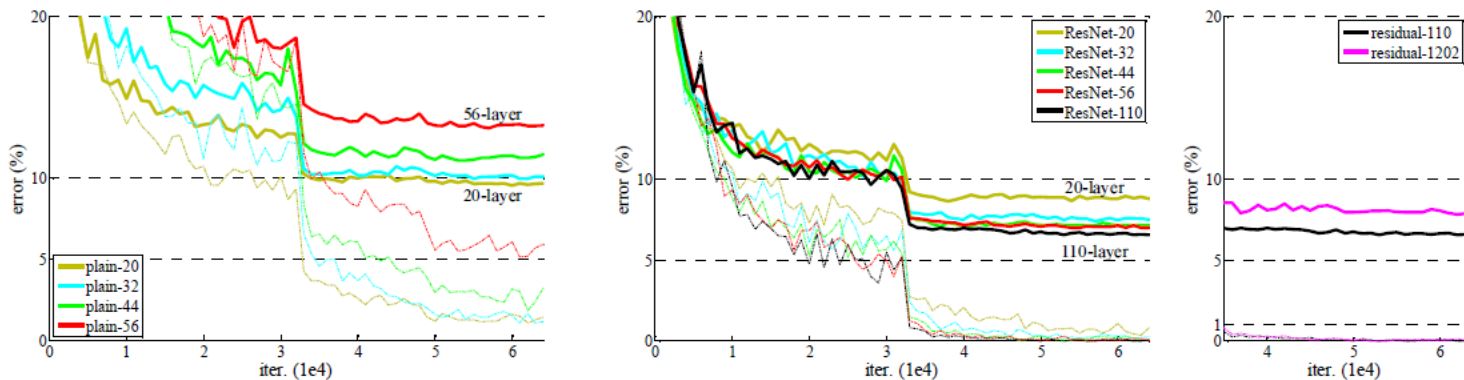


Figure 6. Training on **CIFAR-10**. Dashed lines denote training error, and bold lines denote testing error. **Left:** plain networks. The error of plain-110 is higher than 60% and not displayed. **Middle:** ResNets. **Right:** ResNets with 110 and 1202 layers.