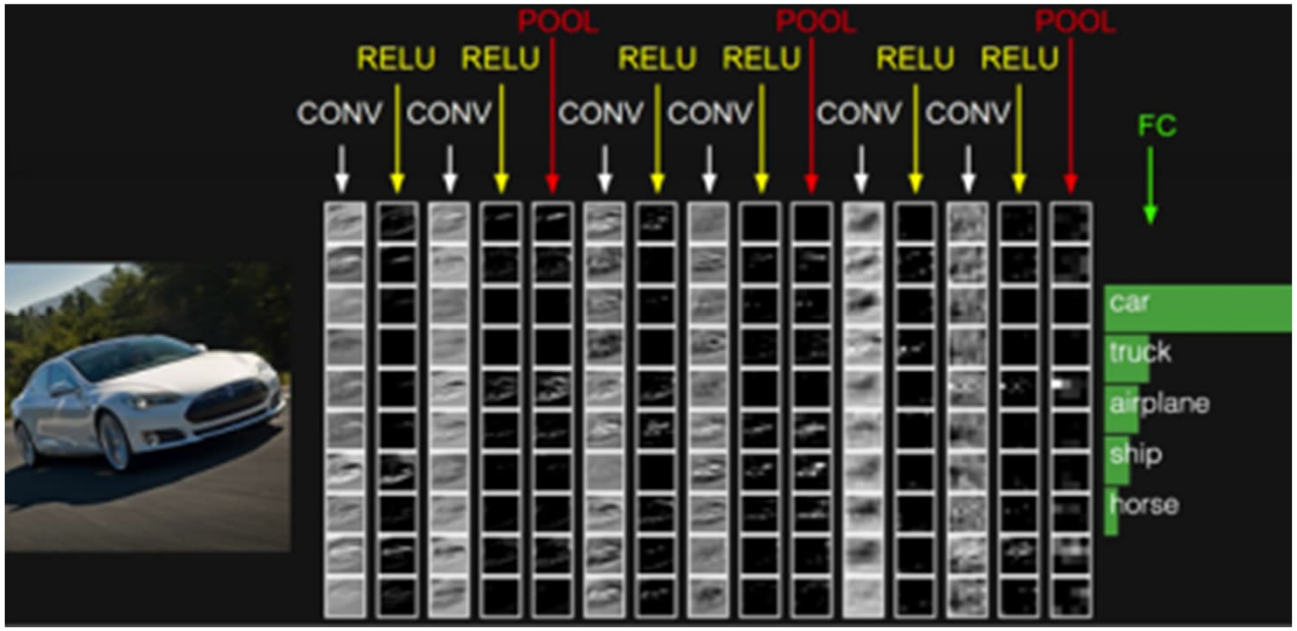
Convolutional Neural Network

LeNet, AlexNet, VGG

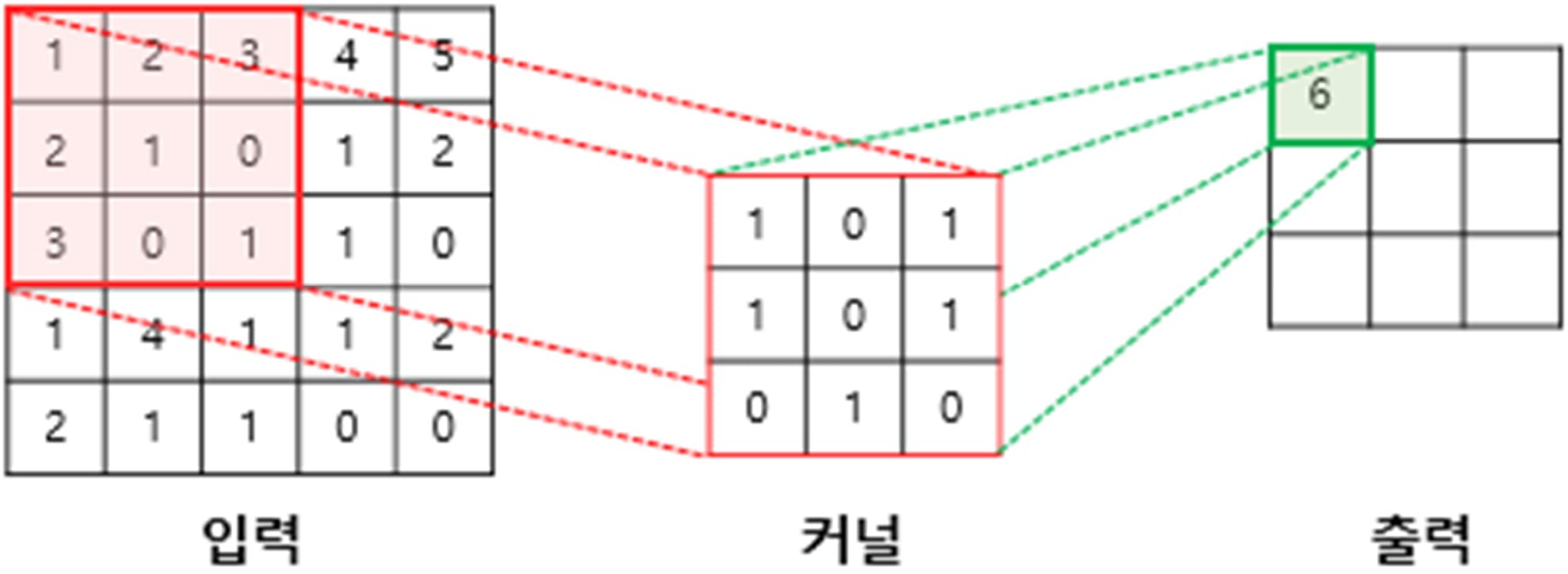
예과 1학년 김성현

Convolutional Neural Network

- Convolution layer(합성곱층)
- Pooling layer(풀링층)

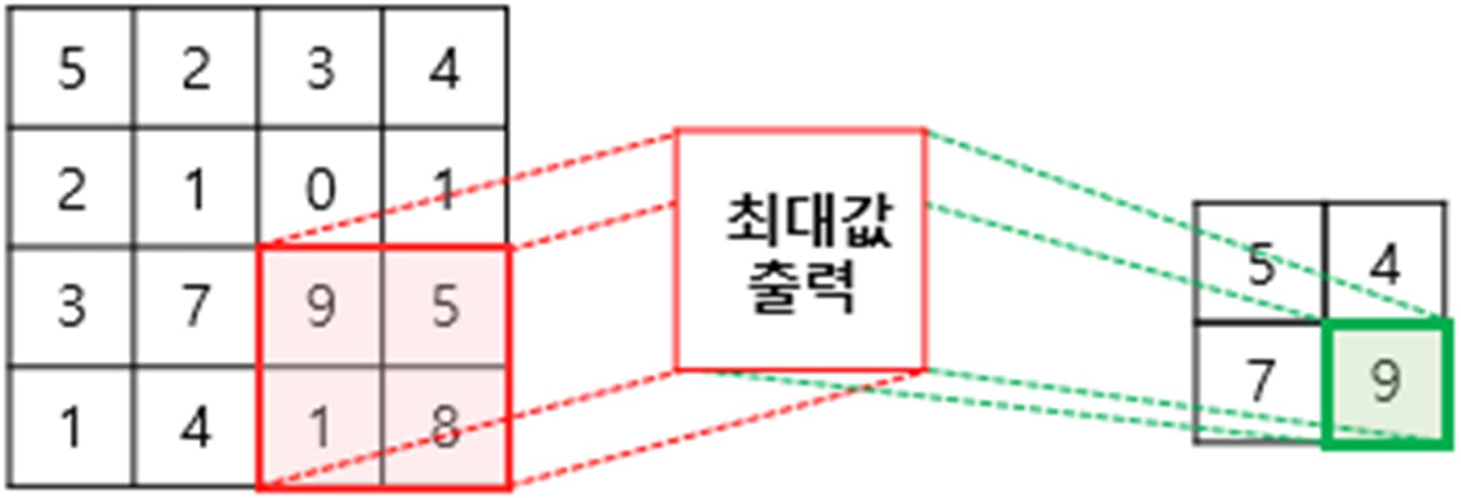


Convolution operation(합성곱 연산)



-> Activating function(**활성화 함수**)

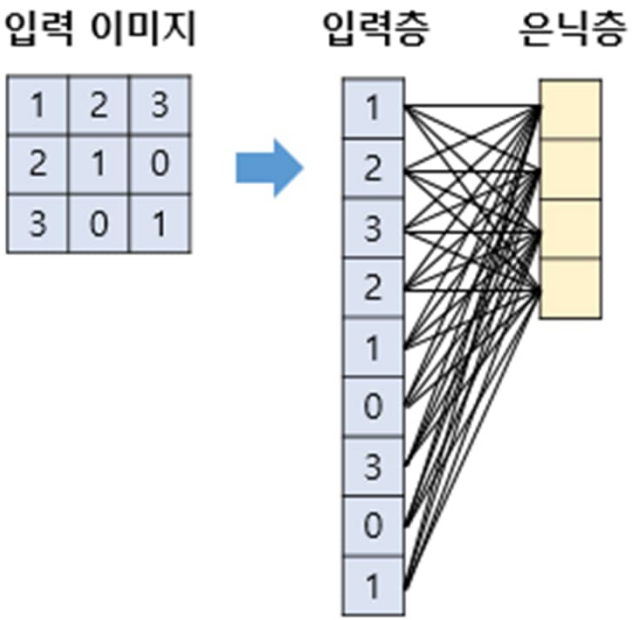
Pooling(풀링)



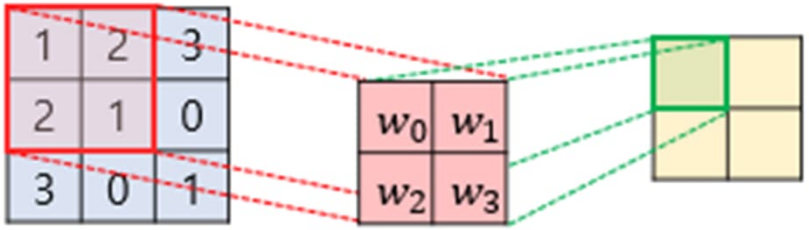
Max pooling

이미지 처리

Multilayer perceptron

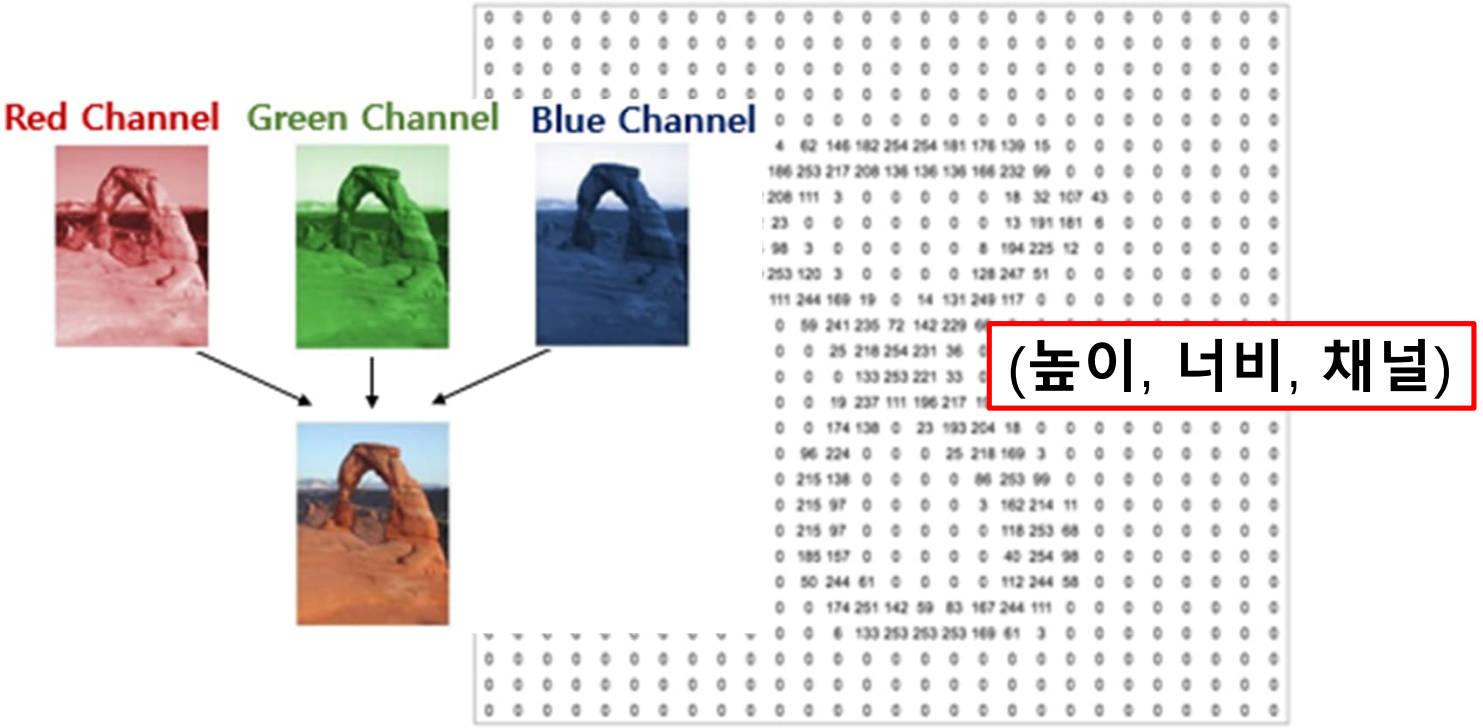


Convolutional Neural Network

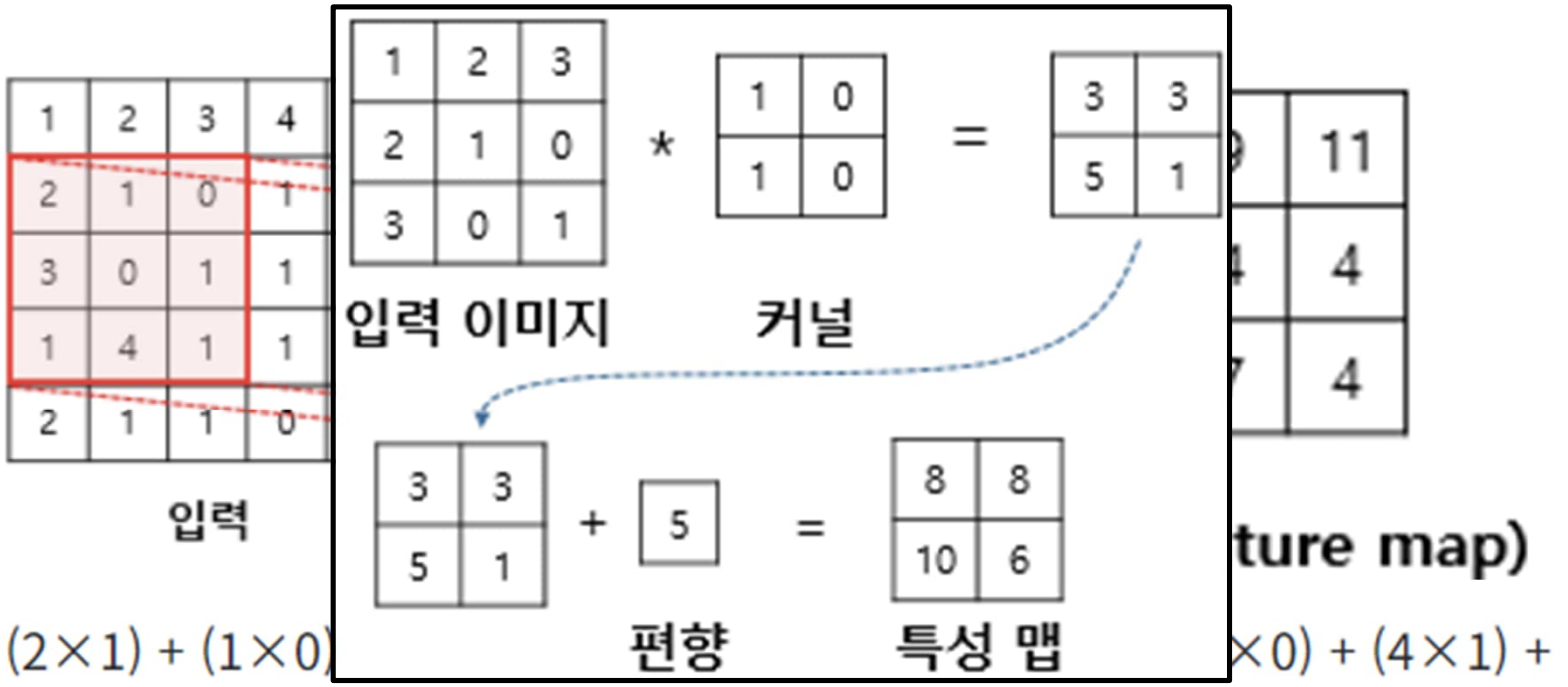


세부 사항 / 용어

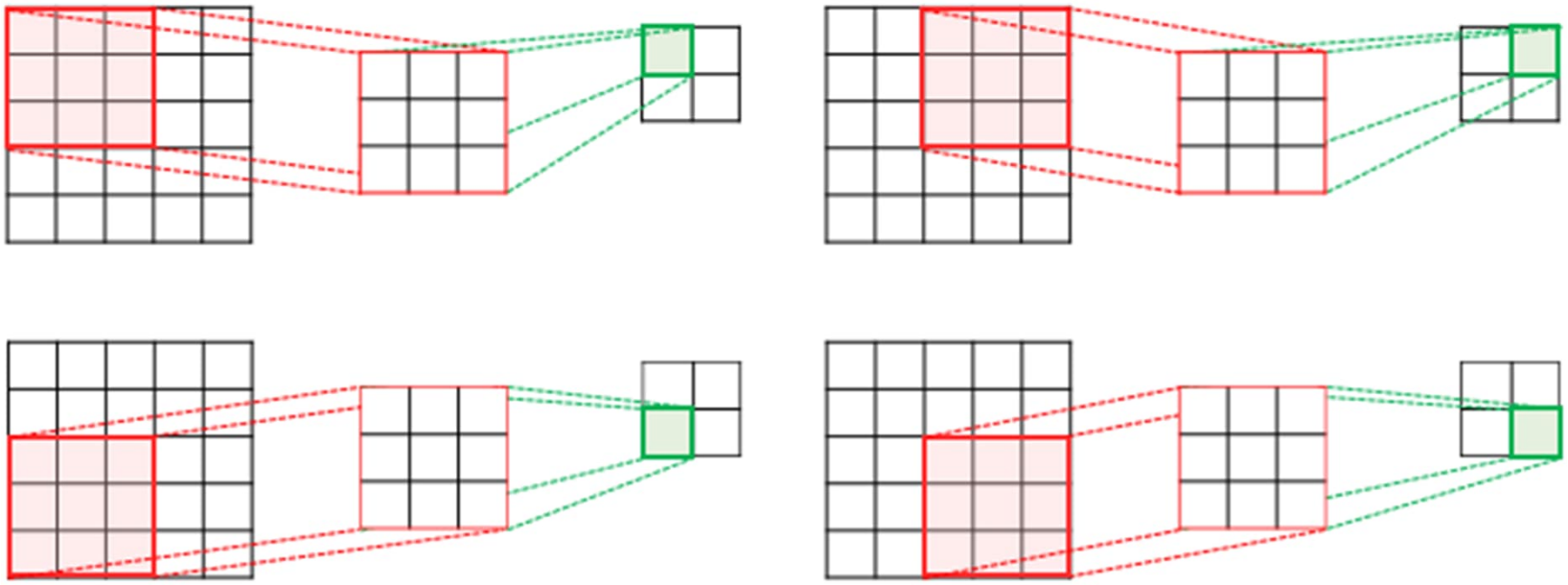
Channel(채널)



Kernel(커널) / Feature map(특성 맵)



Stride(스트라이드)



Stride = 2

Padding(패딩)

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

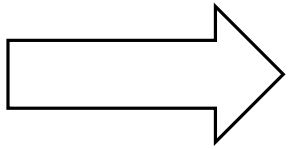
패딩 전

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

패딩 후

Feature map의 크기 계산

- I_h : 입력의 높이
- I_w : 입력의 너비
- K_h : 커널의 높이
- K_w : 커널의 너비
- S : 스트라이드
- O_h : 특성 맵의 높이
- O_w : 특성 맵의 너비



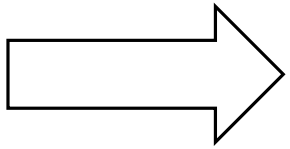
$$O_h = floor(\frac{I_h - K_h}{S} + 1)$$

$$O_w = floor(\frac{I_w - K_w}{S} + 1)$$

Feature map의 크기 계산

- I_h : 입력의 높이
- I_w : 입력의 너비
- K_h : 커널의 높이
- K_w : 커널의 너비
- S : 스트라이드
- O_h : 특성 맵의 높이
- O_w : 특성 맵의 너비

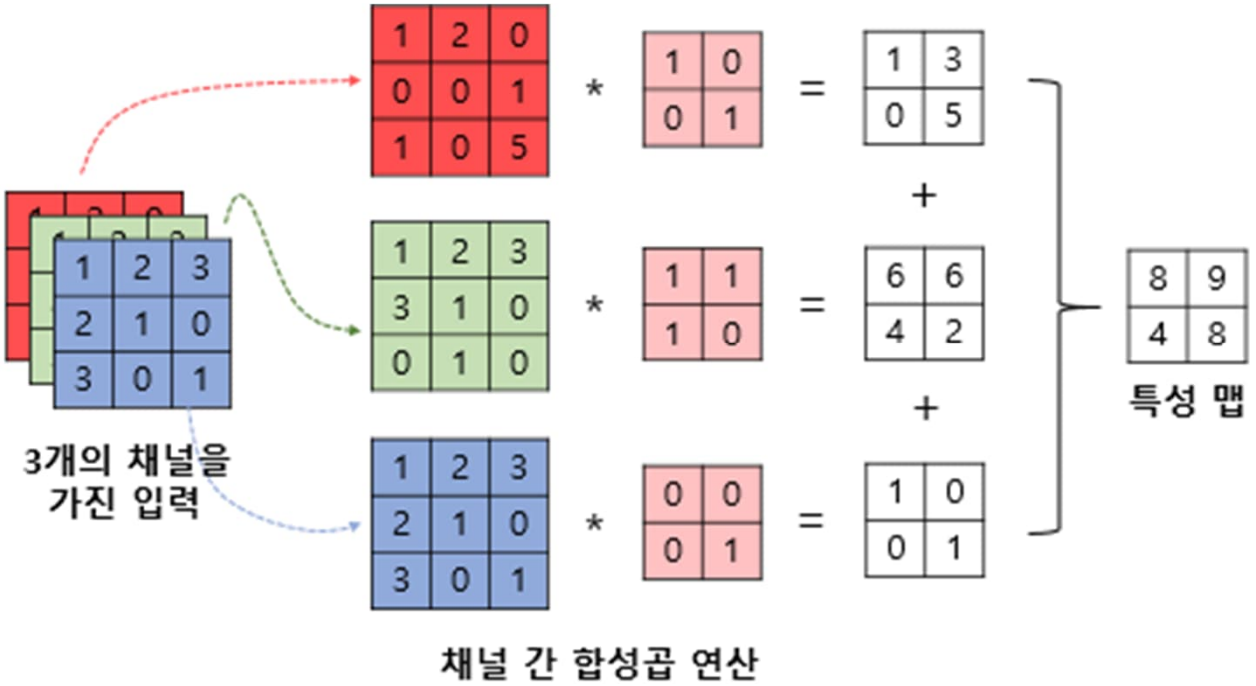
+ P : **패딩의 폭**



$$O_h = floor(\frac{I_h - K_h + 2P}{S} + 1)$$

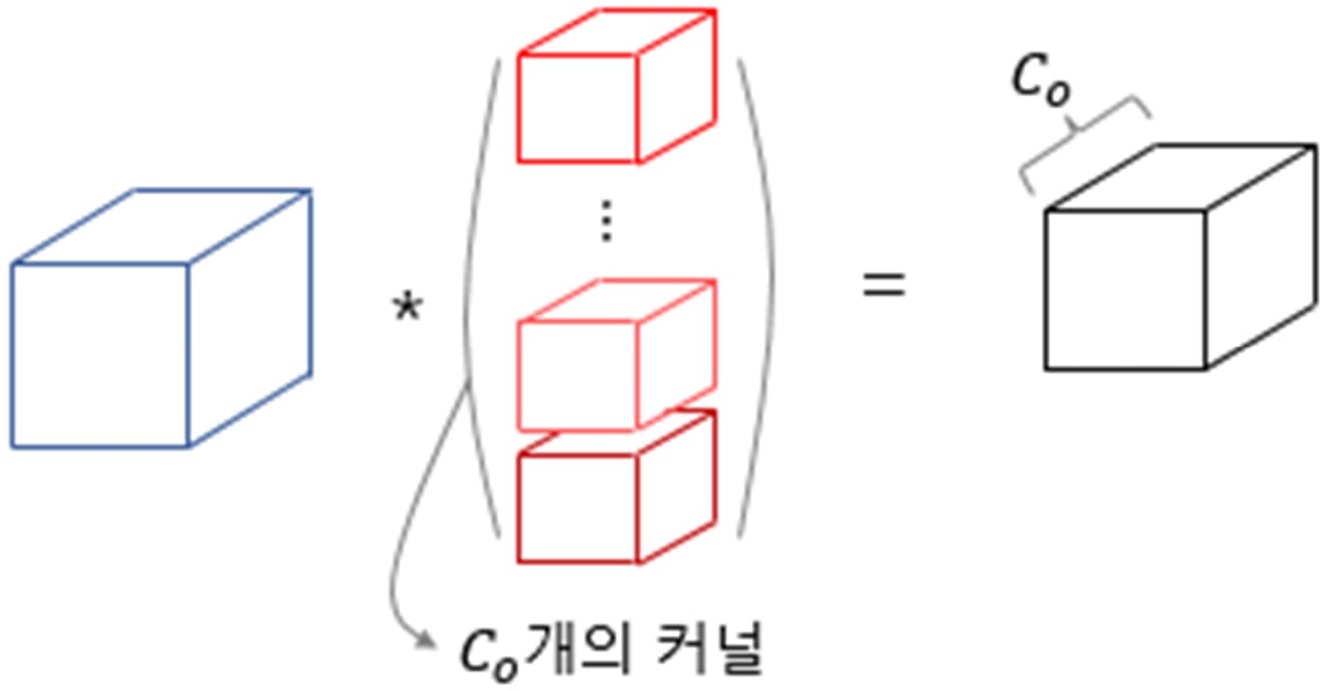
$$O_w = floor(\frac{I_w - K_w + 2P}{S} + 1)$$

다수의 채널(입력)



3-채널 입력 -> 3-채널 커널

다수의 채널(출력)



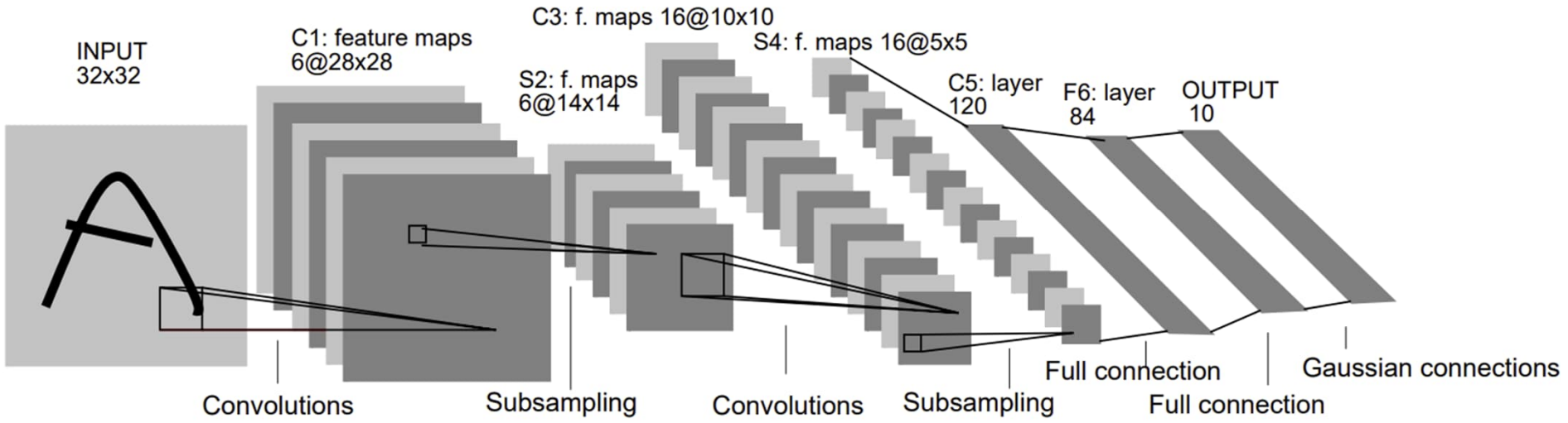
Co-**채널 입력** -> Co**개의 커널**

구체적 예

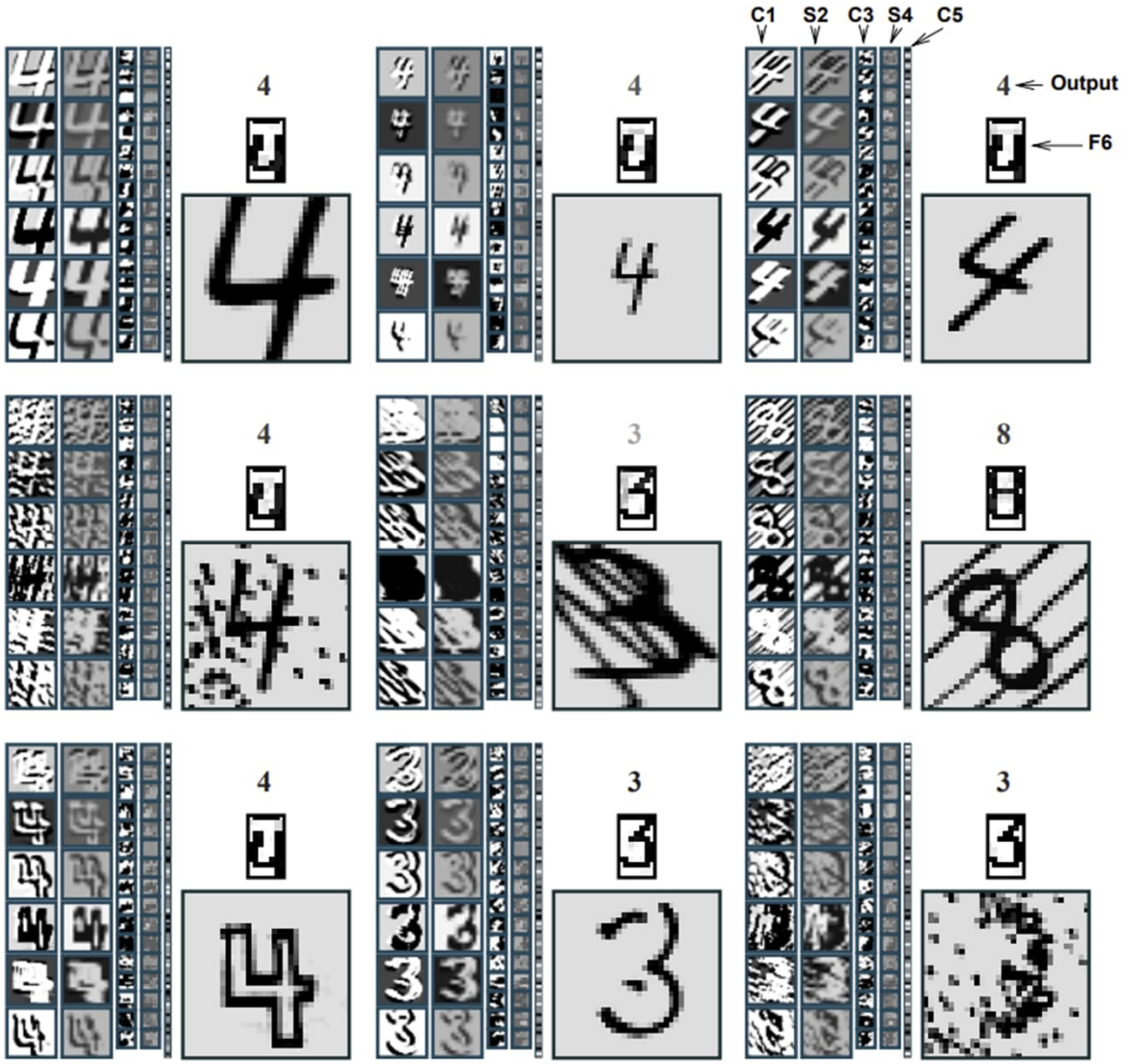
– LeNet, AlexNet, VGG

LeNet-5

- LeCun et al., 1998
- 손글씨 인식

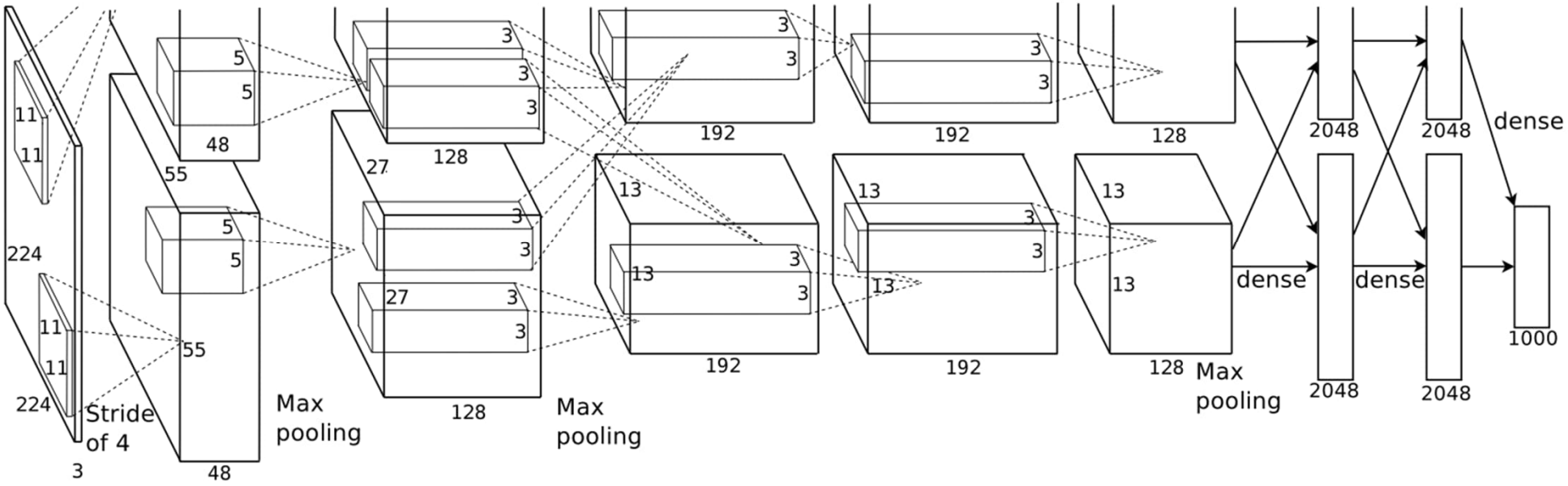


LeNet-

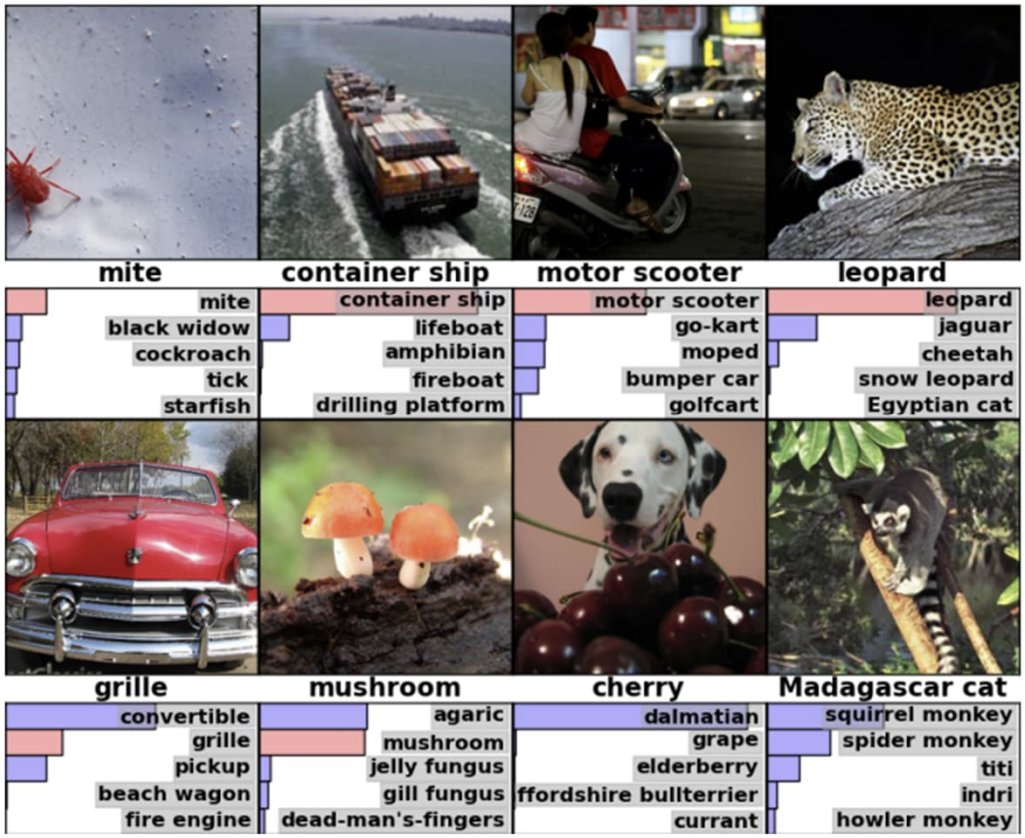


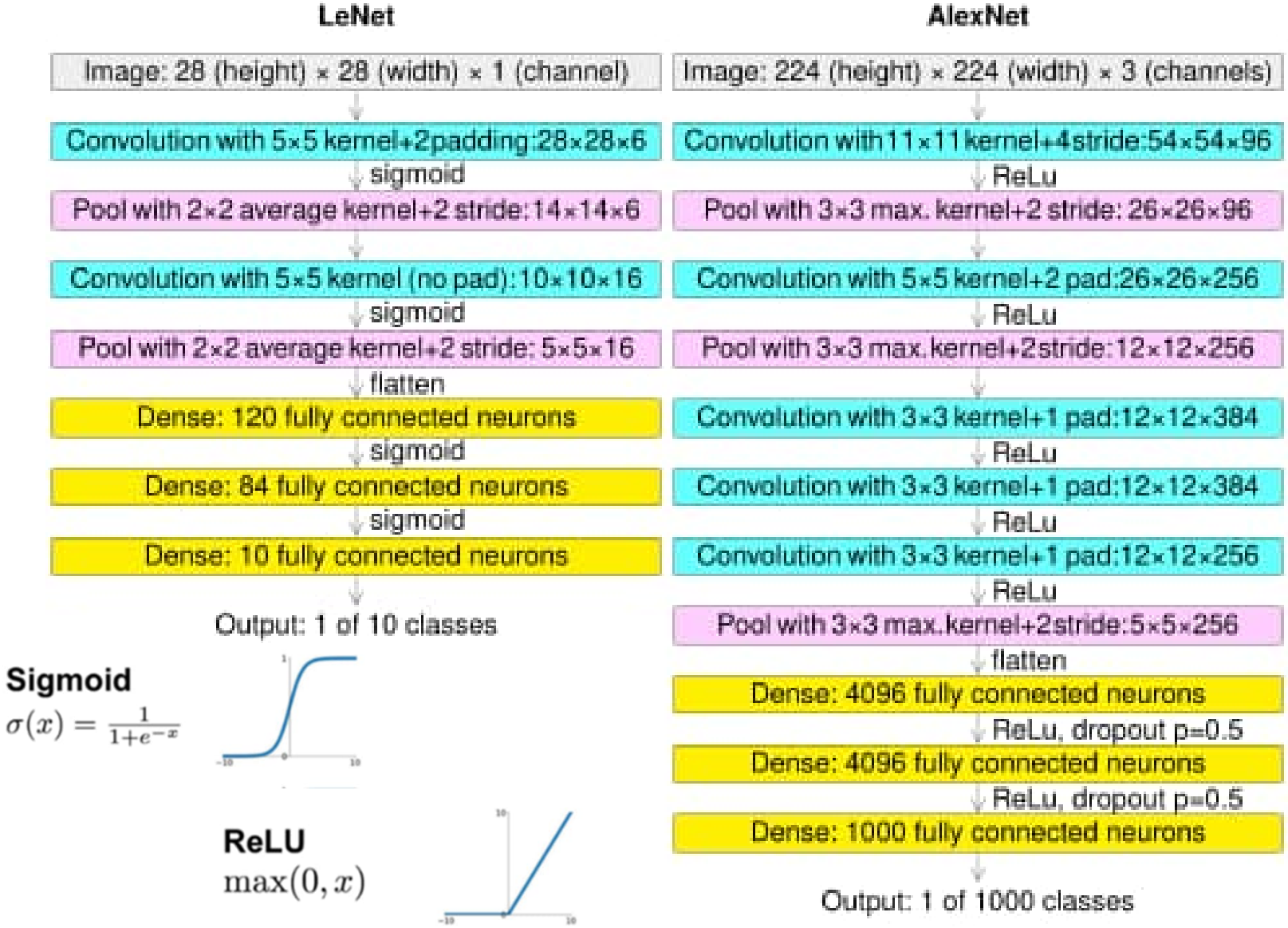
AlexNet

- Alex Krizhevsky
- ImageNet Large Scale Visual Recognition Challenge, 2012



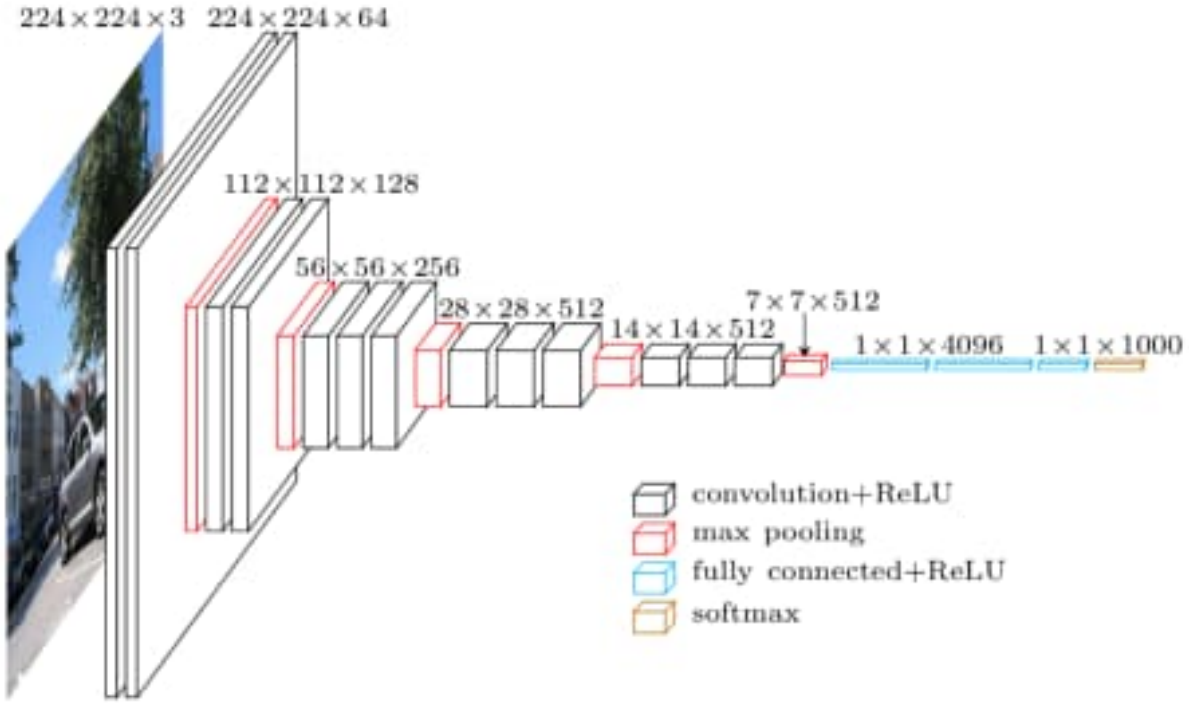
AlexNet

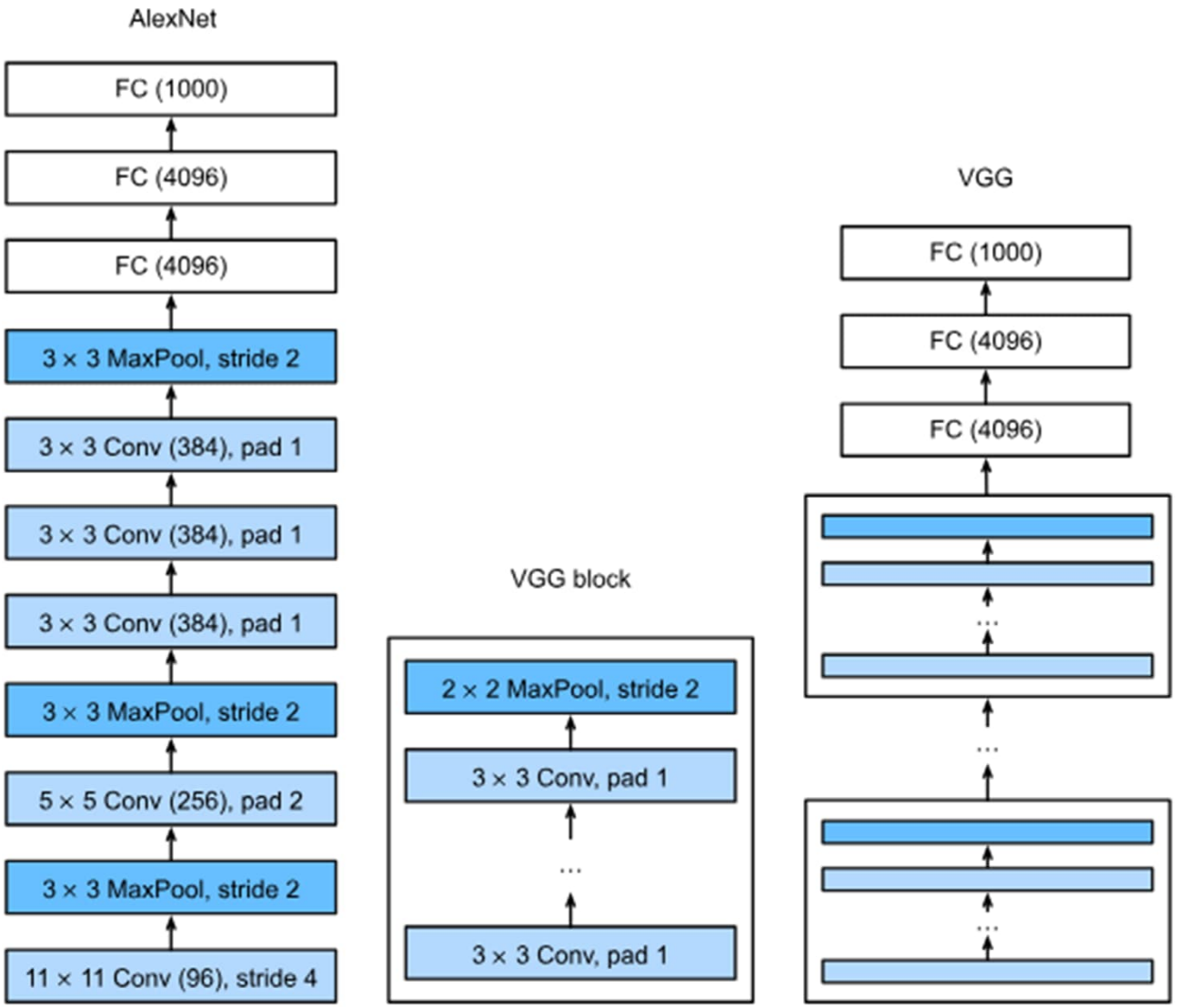

mite
container ship
motor scooter
leopard
mite
black widow
cockroach
tick
starfish
container ship
lifeboat
amphibian
fireboat
drilling platform
motor scooter
go-kart
moped
bumper car
golfcart
leopard
jaguar
cheetah
snow leopard
Egyptian cat
grille
mushroom
cherry
Madagascar cat
convertible
grille
pickup
beach wagon
fire engine
agaric
mushroom
jelly fungus
gill fungus
dead-man's-fingers
dalmatian
grape
elderberry
ffordshire bullterrier
currant
squirrel monkey
spider monkey
titi
indri
howler monkey


LeNet
Image: 28 (height) × 28 (width) × 1 (channel)
Convolution with 5×5 kernel+2padding:28×28×6
sigmoid
Pool with 2×2 average kernel+2 stride:14×14×6
Convolution with 5×5 kernel (no pad):10×10×16
sigmoid
Pool with 2×2 average kernel+2 stride: 5×5×16
flatten
Dense: 120 fully connected neurons
sigmoid
Dense: 84 fully connected neurons
sigmoid
Dense: 10 fully connected neurons
Output: 1 of 10 classes
Sigmoid
$\sigma(x) = \frac{1}{1+e^{-x}}$
ReLU
$\max(0, x)$
AlexNet
Image: 224 (height) × 224 (width) × 3 (channels)
Convolution with 11×11 kernel+4stride:54×54×96
ReLu
Pool with 3×3 max. kernel+2 stride: 26×26×96
Convolution with 5×5 kernel+2 pad:26×26×256
ReLu
Pool with 3×3 max.kernel+2stride:12×12×256
Convolution with 3×3 kernel+1 pad:12×12×384
ReLu
Convolution with 3×3 kernel+1 pad:12×12×384
ReLu
Convolution with 3×3 kernel+1 pad:12×12×256
ReLu
Pool with 3×3 max.kernel+2stride:5×5×256
flatten
Dense: 4096 fully connected neurons
ReLu, dropout p=0.5
Dense: 4096 fully connected neurons
ReLu, dropout p=0.5
Dense: 1000 fully connected neurons
Output: 1 of 1000 classes

VGG

- Visual Geometry Group, 2014




AlexNet
FC (1000)
FC (4096)
FC (4096)
3 × 3 MaxPool, stride 2
3 × 3 Conv (384), pad 1
3 × 3 Conv (384), pad 1
3 × 3 Conv (384), pad 1
3 × 3 MaxPool, stride 2
5 × 5 Conv (256), pad 2
3 × 3 MaxPool, stride 2
11 × 11 Conv (96), stride 4
VGG block
2 × 2 MaxPool, stride 2
3 × 3 Conv, pad 1
...
3 × 3 Conv, pad 1
VGG
FC (1000)
FC (4096)
FC (4096)
...
...
...