AI - FOUNDATION AND APPLICATION

Instructor:
Assoc. Prof. Dr. Truong Ngoc Son

Chapter 4
Convolution Neural Network

Outline

Motivation – Convolution operation

Convolution is a function derived from two given functions by integration which expresses how the shape of one is modified by the other

$$y(t) = x(t)(*)h(t) = \int_{\tau=-\infty}^{+\infty} x(\tau)h(t-\tau)d\tau$$

Discrete convolution

$$y(t) = x(t)(*)h(t) = \sum_{\tau=-\infty}^{+\infty} x(\tau)h(t-\tau)$$

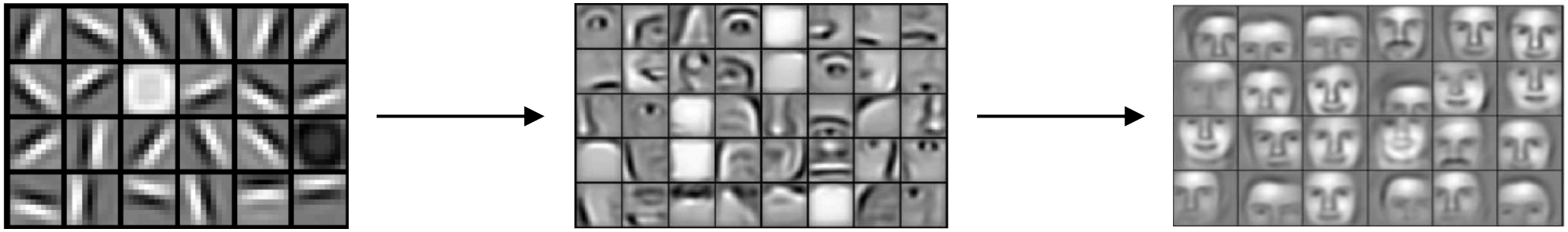
Motivation – Convolution operation

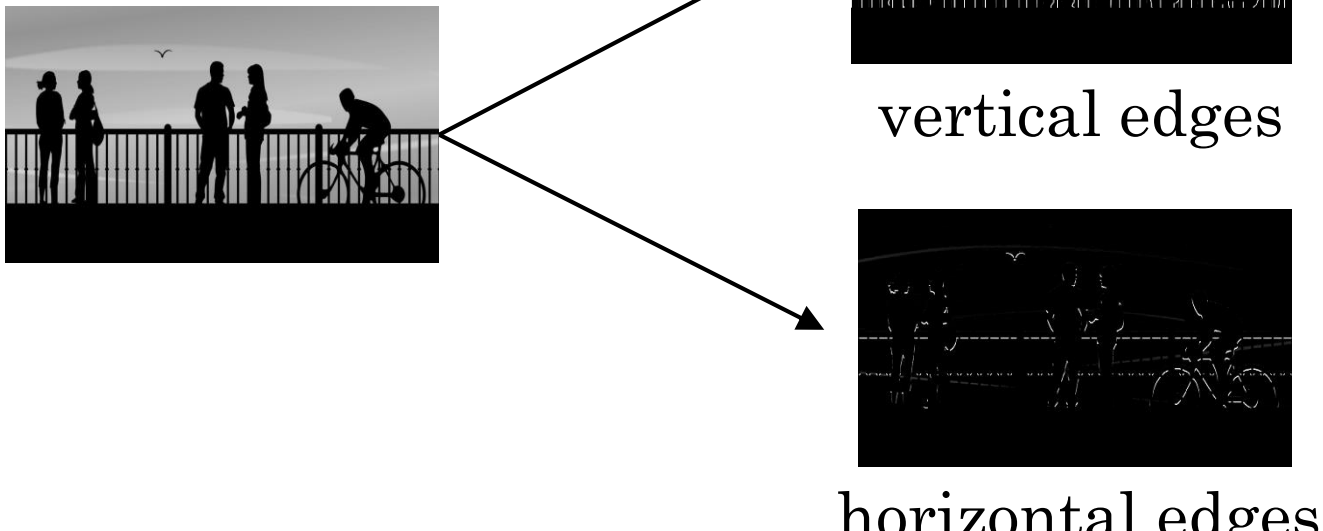
Convolution in 2-D space

$$y(n_1, n_2) = \sum_{k_2=0}^{N-1} \sum_{k_1=0}^{M-1} x(k_1, k_2) h(n_1 - k_1, n_2 - k_2)$$

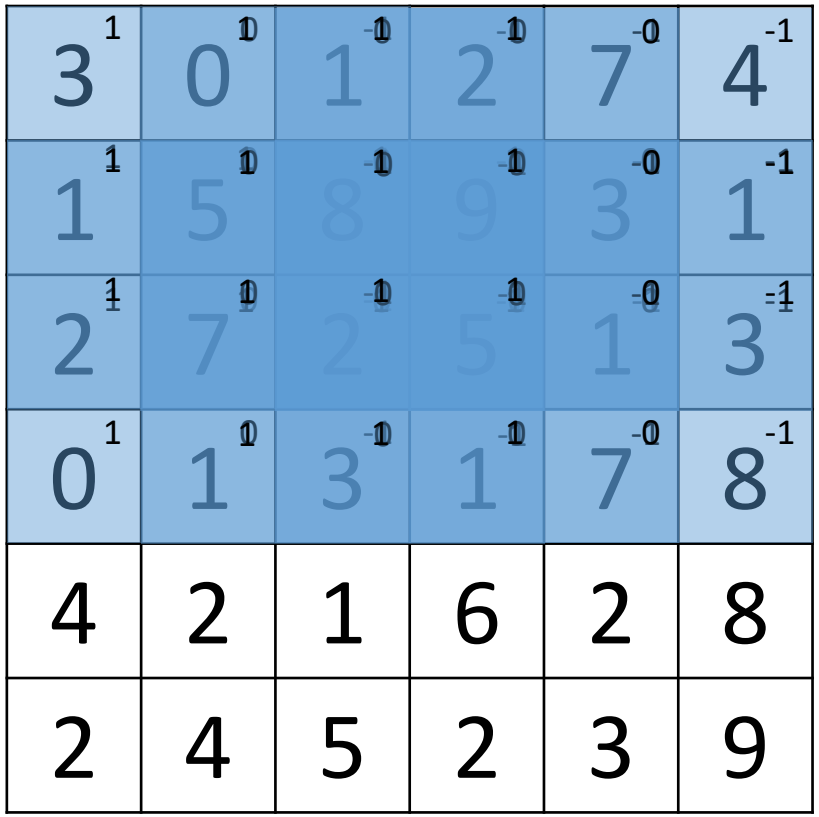
$$0 \leq n_1 \leq N-1, 0 \leq n_2 \leq M-1.$$

Computer Vision Problem





Vertical edge detection



*

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

=

0	-2	-4	-7
-3	-2	-3	-16

Vertical edge detection

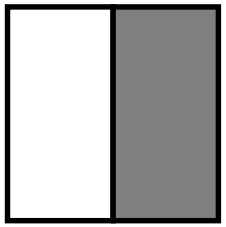
10	10	10	0	0	0
10	10	10	0	0	0
10	10	10	0	0	0
10	10	10	0	0	0
10	10	10	0	0	0
10	10	10	0	0	0

*

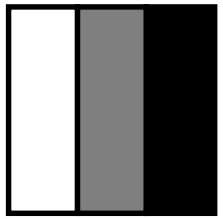
1	0	-1
1	0	-1
1	0	-1

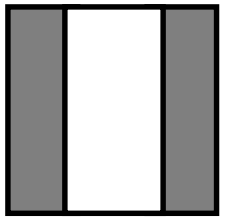
=

0	30	30	0
0	30	30	0
0	30	30	0
0	30	30	0



*





Vertical edge detection examples

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

*

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

=

0	30	30	0
0	30	30	0
0	30	30	0
0	30	30	0

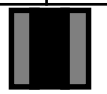
0	0	0	10	10	10
0	0	0	10	10	10
0	0	0	10	10	10
0	0	0	10	10	10
0	0	0	10	10	10
0	0	0	10	10	10

*

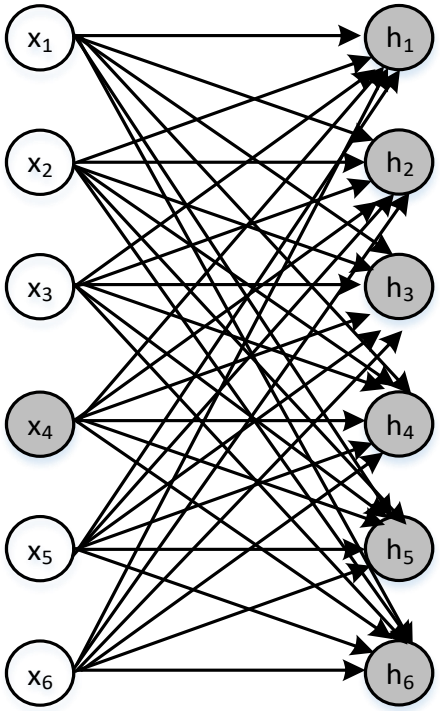
1	0	-1
1	0	-1
1	0	-1

=

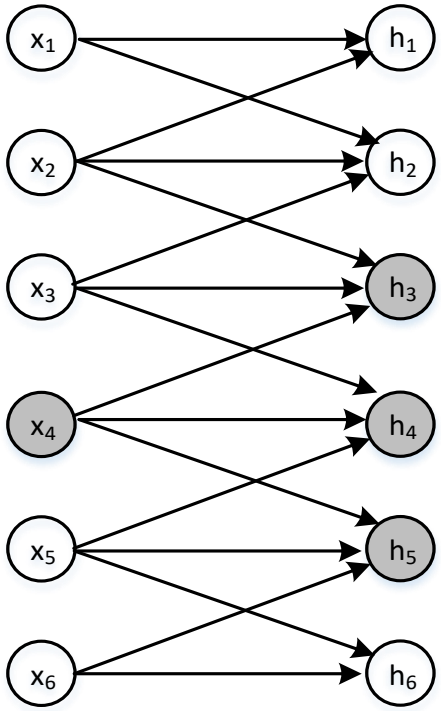
0	-30	-30	0
0	-30	-30	0
0	-30	-30	0
0	-30	-30	0



Convolution Neural Network

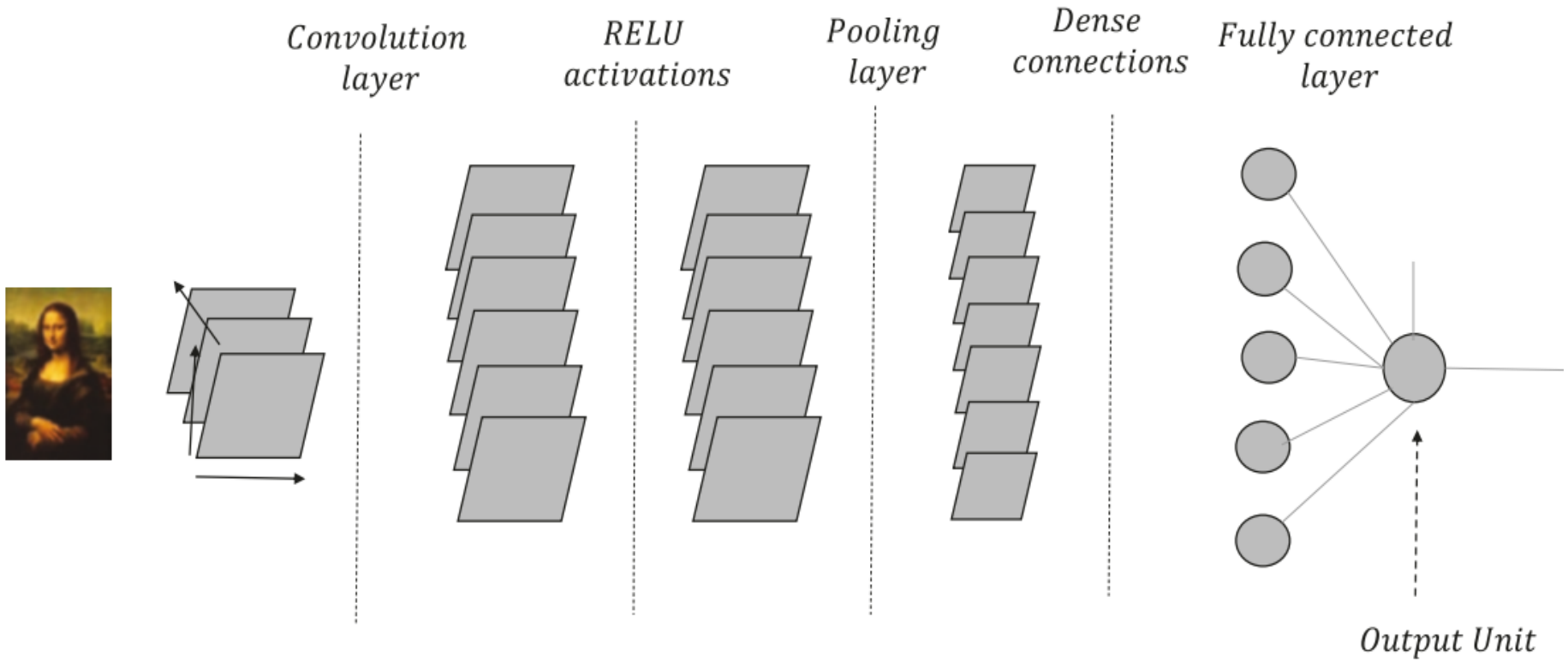


(a) Fully connecter layer



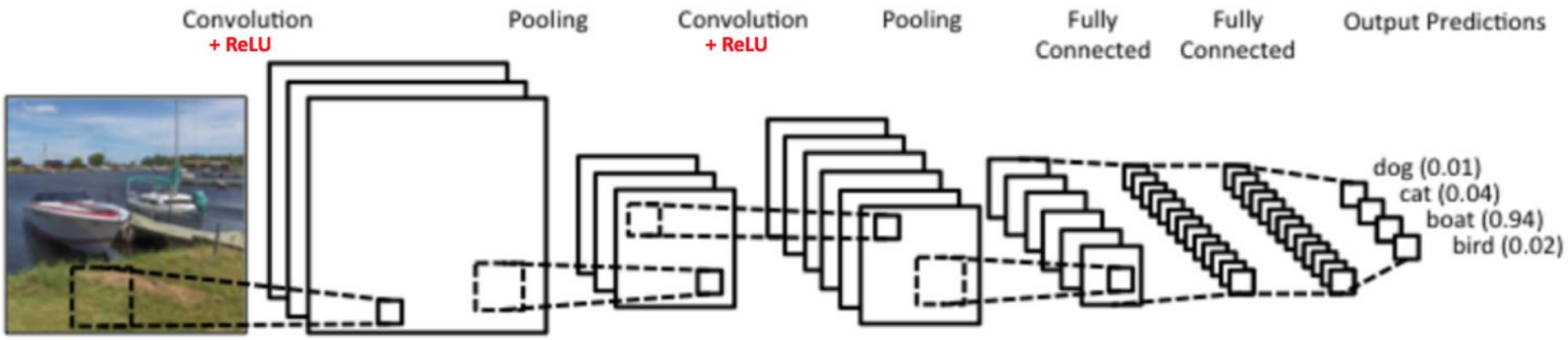
(b) Convolution layer

Convolution Neural Network

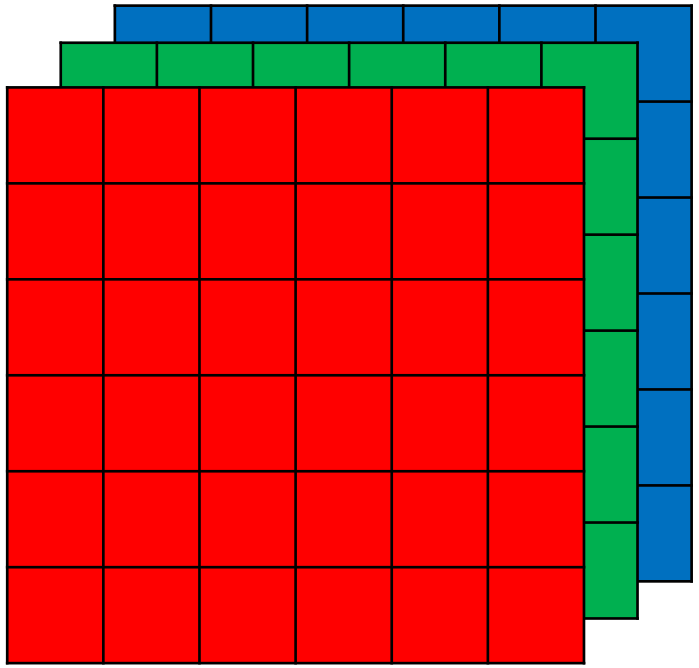


Convolution Neural Network

- Input layer
- Convolution layer
- Activation function
- Pooling layer
- Fully connected layers

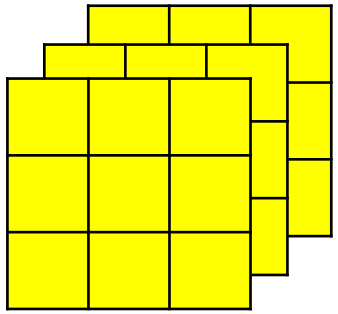


Convolution layer



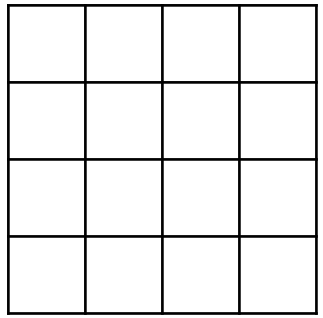
6 x 6 x 3

*



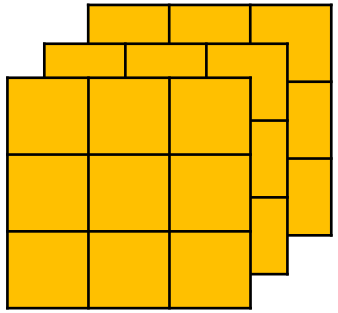
3 x 3 x 3

=



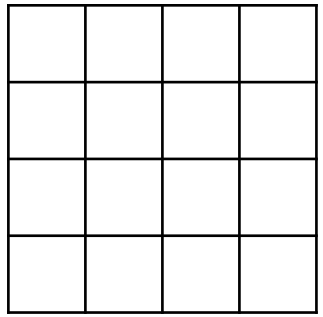
4 x 4

*



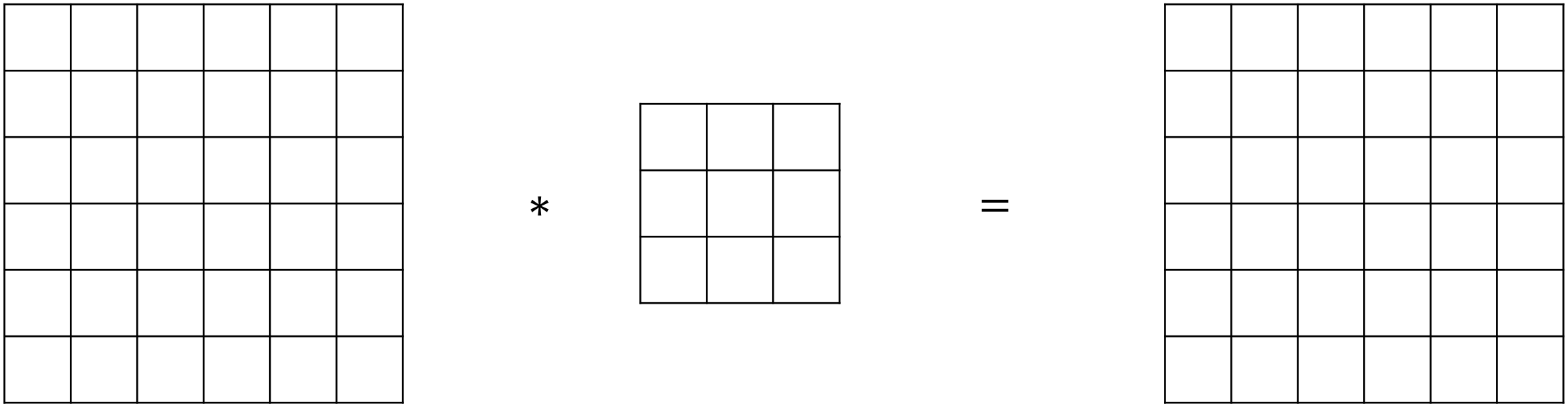
3 x 3 x 3

=



4 x 4

Padding

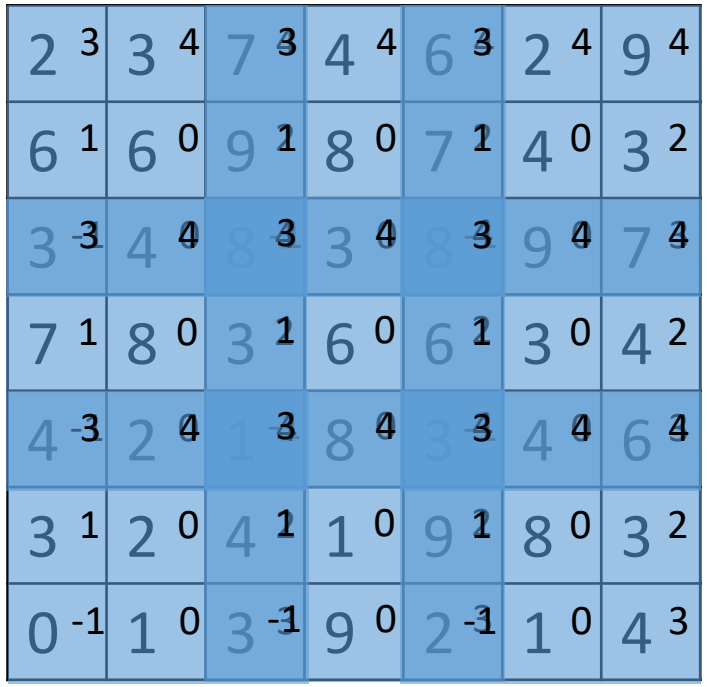


Valid and Same convolutions

"Valid": No padding is required

"Same": Pad so that output size is the same as the input size.

Strided convolution



*

3	4	4
1	0	2
-1	0	3

=

Summary of convolutions

$n \times n$ image $\quad$ $f \times f$ filter

padding p $\quad$ stride s

$$\left\lfloor \frac{n+2p-f}{s} + 1 \right\rfloor \times \left\lfloor \frac{n+2p-f}{s} + 1 \right\rfloor$$

Technical note on cross-correlation vs. convolution

Convolution in math textbook:

2	3	7	4	6	2
6	6	9	8	7	4
3	4	8	3	8	9
7	8	3	6	6	3
4	2	1	8	3	4
3	2	4	1	9	8

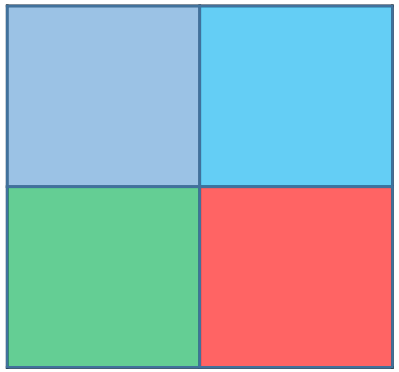
$*$

3	4	5
1	0	2
-1	9	7

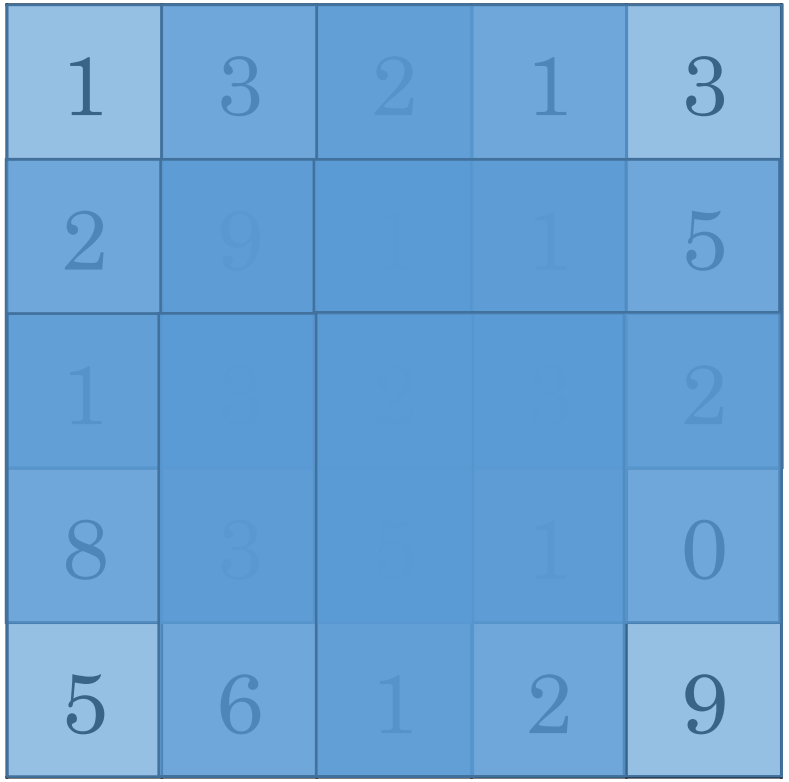
7	2	5
9	0	4
-1	1	3

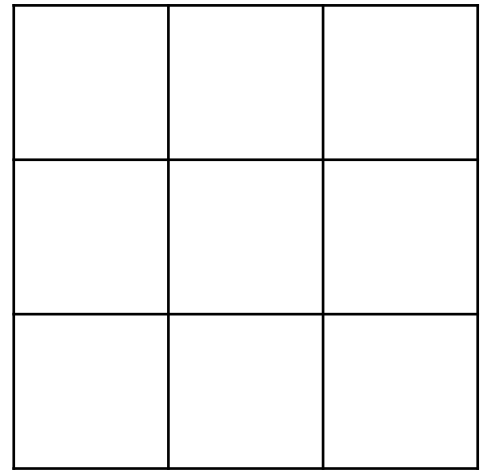
Pooling layer: Max pooling

1	3	2	1
2	9	1	1
1	3	2	3
5	6	1	2

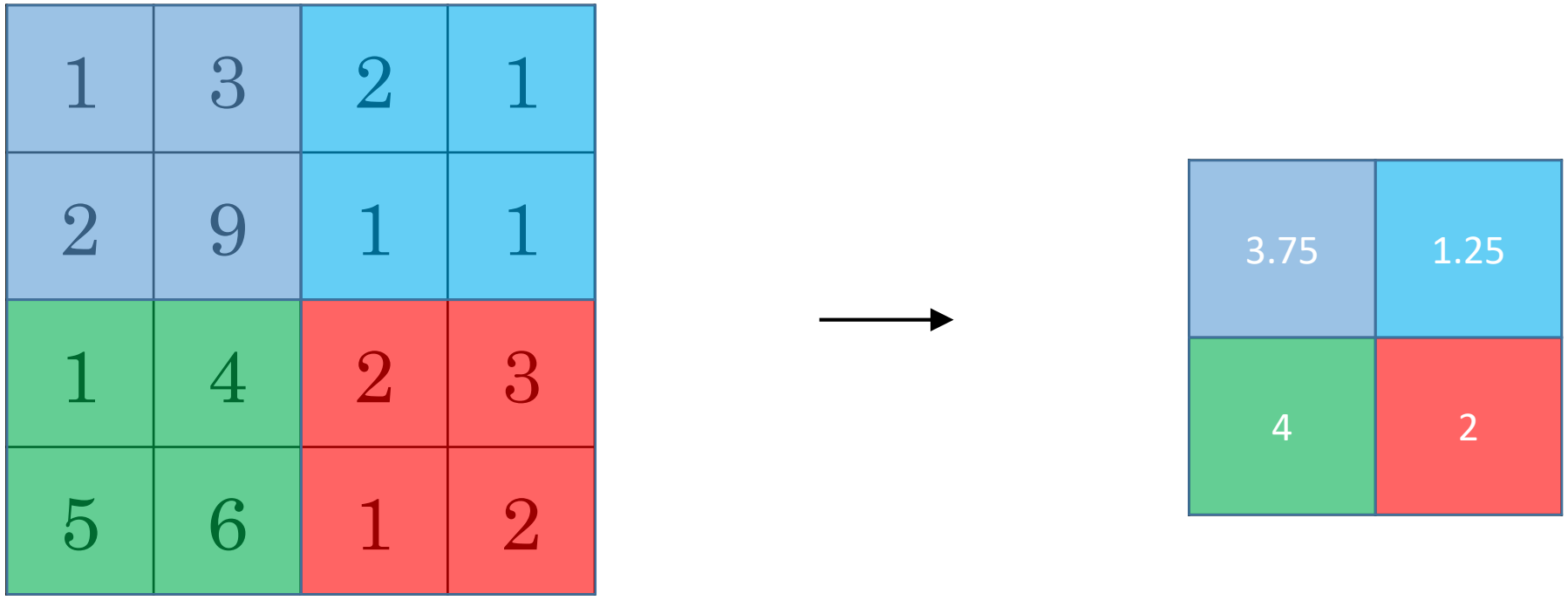


Pooling layer: Max pooling



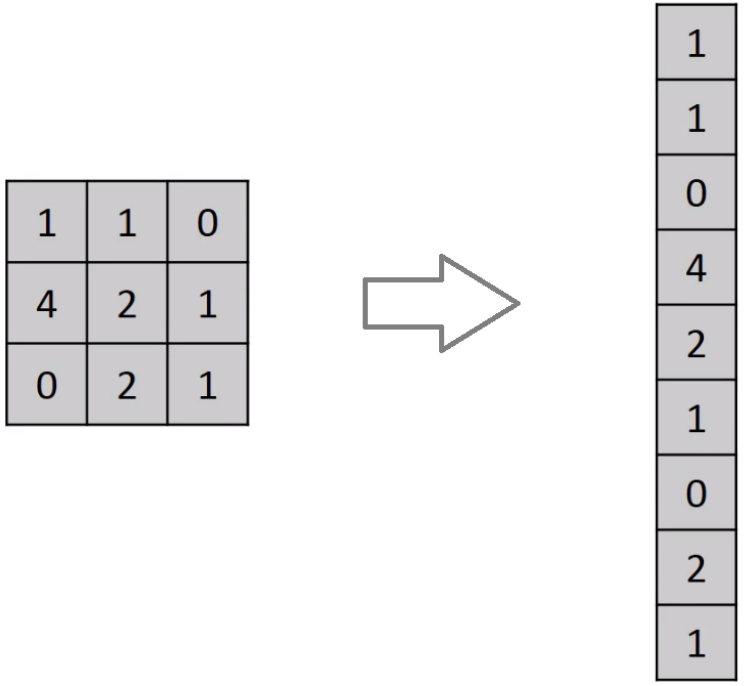


Pooling layer: Average pooling



Fully-connected layer (FC)

Flatten



Summary of pooling

Hyperparameters:

f : filter size

s : stride

Max or average pooling

CNN

- 1. Lenet-5
- 2. Alex-net
- 3. VGG
- 4. ResNet
- 5. Inception
- 6. MobileNet
- 7. Data Augmentation
- 8. Object detection: R-CNN
- 9. Object detection: Yolo
- Transfer Learning ()

PYTHON CODE