



Convolutional Neural Networks

Convolutional Neural Networks

■ Reduction of Model Complexity

- Partially connected & shared weights
- Convolutional Neural Network (CNN)
- Specialized to sequential data
 - Images, Speech, Text

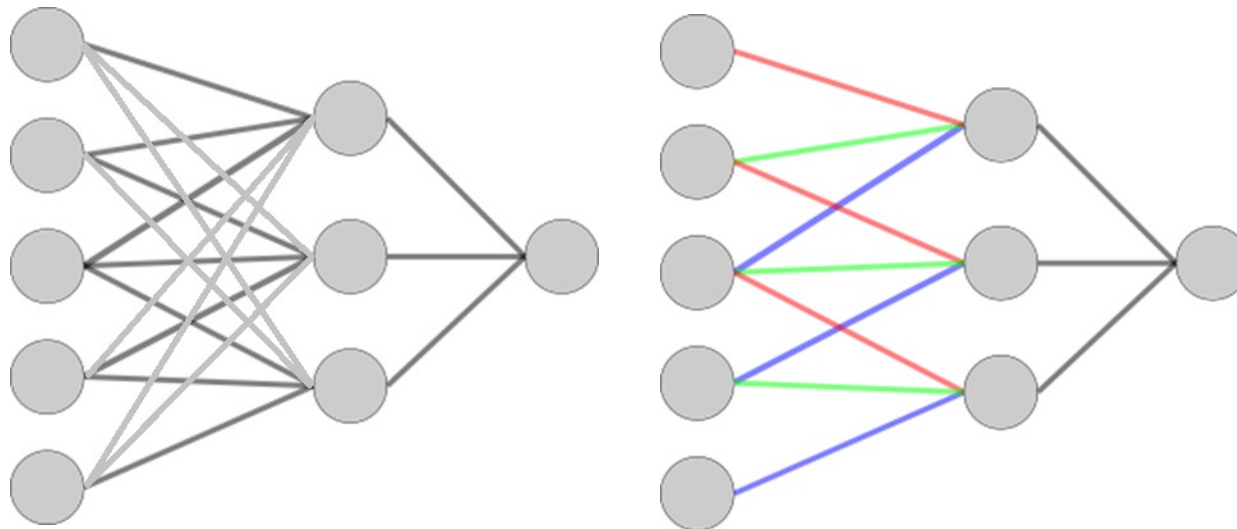


Image Classification

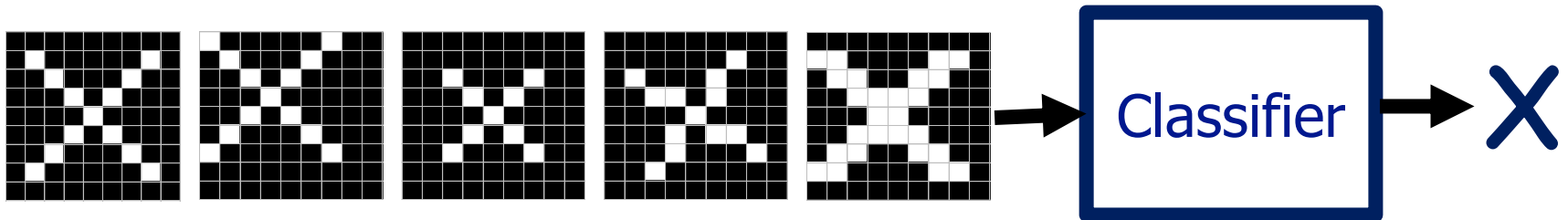
- **X, O Classification**



A two-dimensional
array of pixels

Image Classification

translation



scaling rotation weight

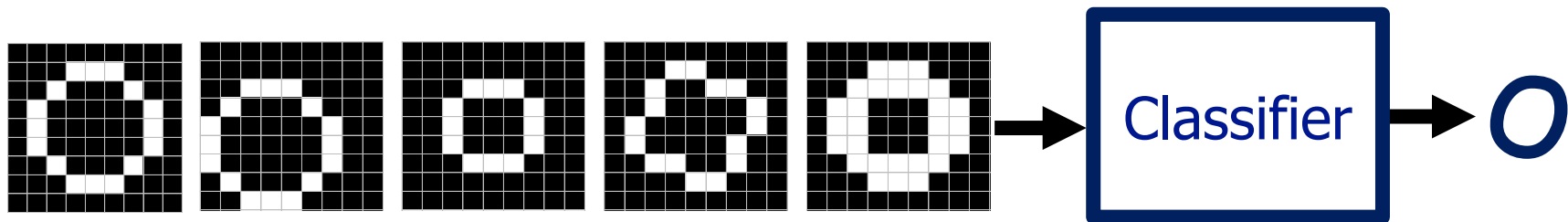
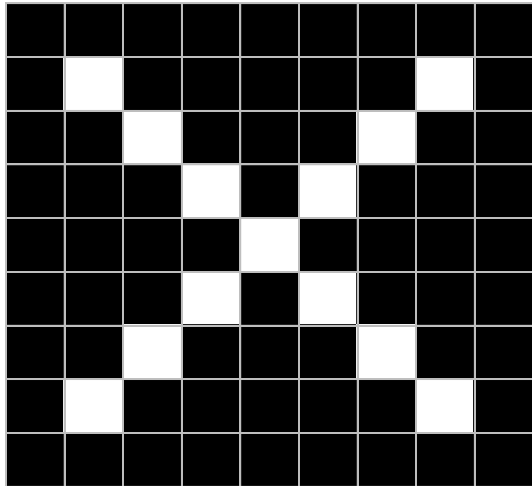


Image Classification

- **Same?**
 - How to determine both are the same?



=

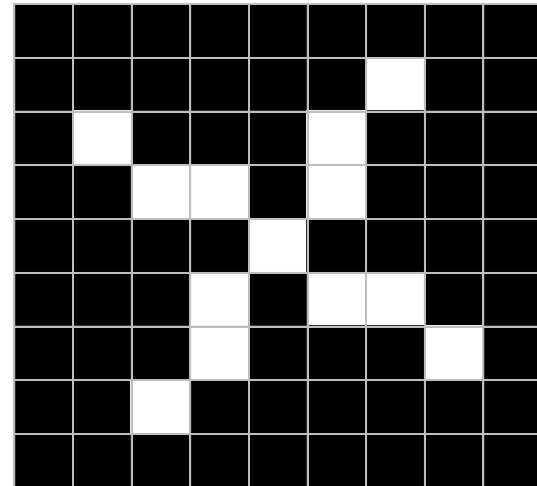


Image Classification

- **Both are partially matching**
 - Critical LOCAL features are the same

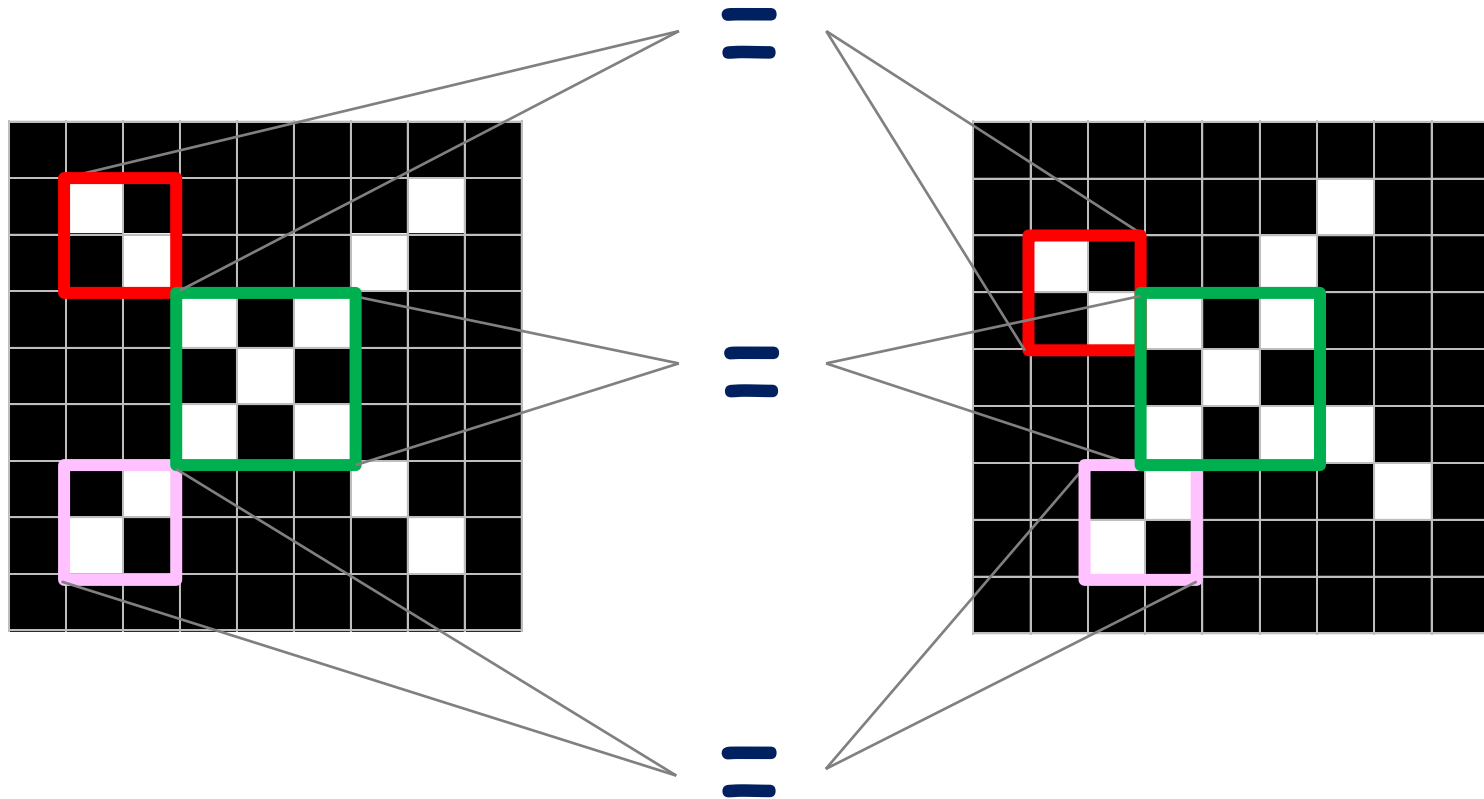
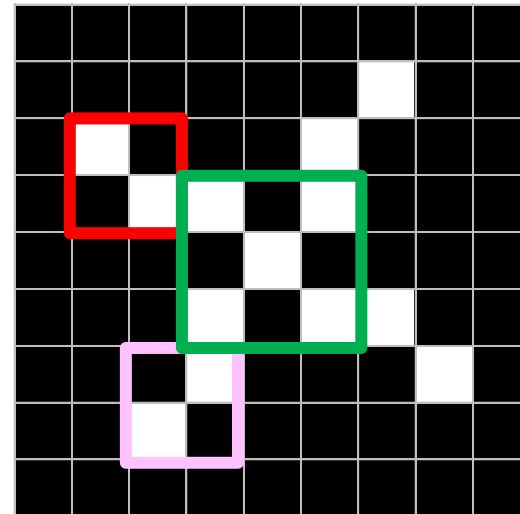
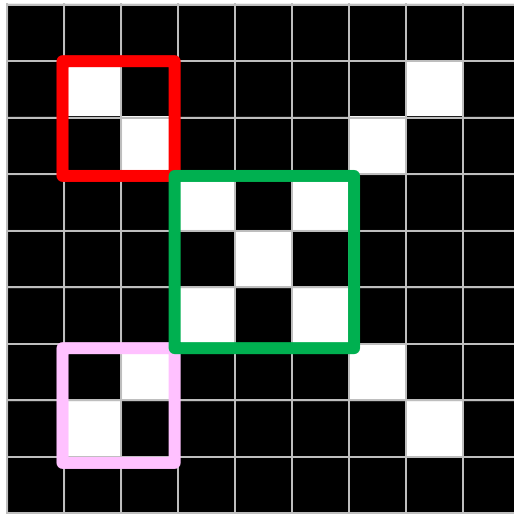


Image Classification

- **Local Features**

- How about finding local features, and
- Matching them



Feature Extraction

- **Convolution**

- A way to find out local features

1	0	1
0	1	1
1	1	0

 *

1	0	1
0	1	0
1	0	1

 = 4

$$I * K = \sum K_{ij} \times I_{ij}$$

Feature Extraction

■ Convolution

-1	-1	1
-1	1	-1
1	-1	-1

$$(I * K)_{xy} = \sum_{i=1}^w \sum_{j=1}^w K_{ij} \cdot I_{x+i-1, y+j-1}$$

0	0	0	0	0	0	0	0
0	0	0	1	1	0	0	0
0	0	1	0	1	1	0	0
0	1	0	0	0	0	1	0
0	1	0	0	0	0	1	0
0	0	1	0	1	1	0	0
0	0	0	1	1	0	0	0
0	0	0	0	0	0	0	0

	-1						

Feature Extraction

■ Convolution

-1	-1	1
-1	1	-1
1	-1	-1

$$(I * K)_{xy} = \sum_{i=1}^w \sum_{j=1}^w K_{ij} \cdot I_{x+i-1, y+j-1}$$

0	0	0	0	0	0	0	0
0	0	0	1	1	0	0	0
0	0	1	0	1	1	0	0
0	1	0	0	0	0	1	0
0	1	0	0	0	0	1	0
0	0	1	0	1	1	0	0
0	0	0	1	1	0	0	0
0	0	0	0	0	0	0	0

	-1						
	-2						

Feature Extraction

■ Convolution

-1	-1	1
-1	1	-1
1	-1	-1

$$(I * K)_{xy} = \sum_{i=1}^w \sum_{j=1}^w K_{ij} \cdot I_{x+i-1, y+j-1}$$

0	0	0	0	0	0	0	0
0	0	0	1	1	0	0	0
0	0	1	0	1	1	0	0
0	1	0	0	0	0	1	0
0	1	0	0	0	0	1	0
0	0	1	0	1	1	0	0
0	0	0	1	1	0	0	0
0	0	0	0	0	0	0	0

	-1						
	-2						
	1						

Feature Extraction

■ Convolution

-1	-1	1
-1	1	-1
1	-1	-1

$$(I * K)_{xy} = \sum_{i=1}^w \sum_{j=1}^w K_{ij} \cdot I_{x+i-1, y+j-1}$$

0	0	0	0	0	0	0	0
0	0	0	1	1	0	0	0
0	0	1	0	1	1	0	0
0	1	0	0	0	0	1	0
0	1	0	0	0	0	1	0
0	0	1	0	1	1	0	0
0	0	0	1	1	0	0	0
0	0	0	0	0	0	0	0

	-1	-2	0	-2	-1	1	
	-2	3	-2	-2	-2	-2	
	1	-1	-2	0	-4	-1	
	-1	-3	0	-2	0	1	
	-2	-1	-4	0	2	-2	
	1	-2	0	0	-3	-1	

Feature Extraction

Convolution

-1	-1	1
-1	1	-1
1	-1	-1

$$(I * K)_{xy} = \sum_{i=1}^w \sum_{j=1}^w K_{ij} \cdot I_{x+i-1, y+j-1}$$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	1	0	0	0	0
0	0	0	1	0	1	1	0	0	0
0	0	1	0	0	0	0	1	0	0
0	0	1	0	0	0	0	1	0	0
0	0	0	1	0	1	1	0	0	0
0	0	0	0	1	1	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

0	0	-1	-2	0	1	0	0
0	-1	-2	0	-2	-1	1	0
-1	-2	3	-2	-2	-2	-2	1
-2	1	-1	-2	0	-4	-1	0
0	-1	-3	0	-2	0	1	-2
1	-2	-1	-4	0	2	-2	-1
0	1	-2	0	0	-3	-1	0
0	0	1	0	-2	-1	0	0

Feature Extraction

■ Threshold

-1	-1	1
-1	1	-1
1	-1	-1

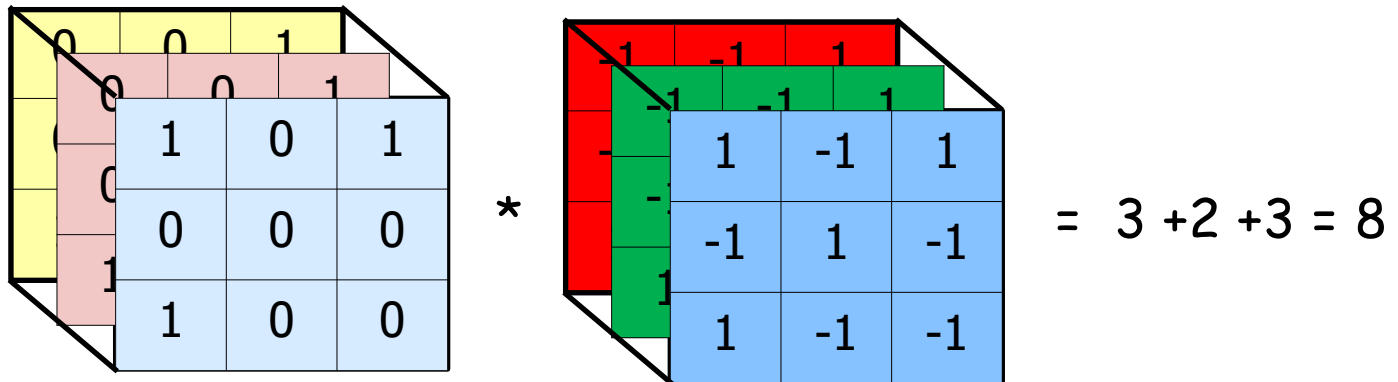
$$(I * K)_{xy} = \sum_{i=1}^w \sum_{j=1}^w K_{ij} \cdot I_{x+i-1, y+j-1}$$

0	0	0	0	0	0	0	0
0	0	0	1	1	0	0	0
0	0	1	0	1	1	0	0
0	1	0	0	0	0	1	0
0	1	0	0	0	0	1	0
0	0	1	0	1	1	0	0
0	0	0	1	1	0	0	0
0	0	0	0	0	0	0	0

0	0	0	0	0	1	0	0
0	0	0	0	0	0	1	0
0	0	3	0	0	0	0	1
0	1	0	0	0	0	0	0
0	0	0	0	0	0	1	0
1	0	0	0	0	2	0	0
0	1	0	0	0	0	0	0
0	0	1	0	0	0	0	0

Feature Extraction

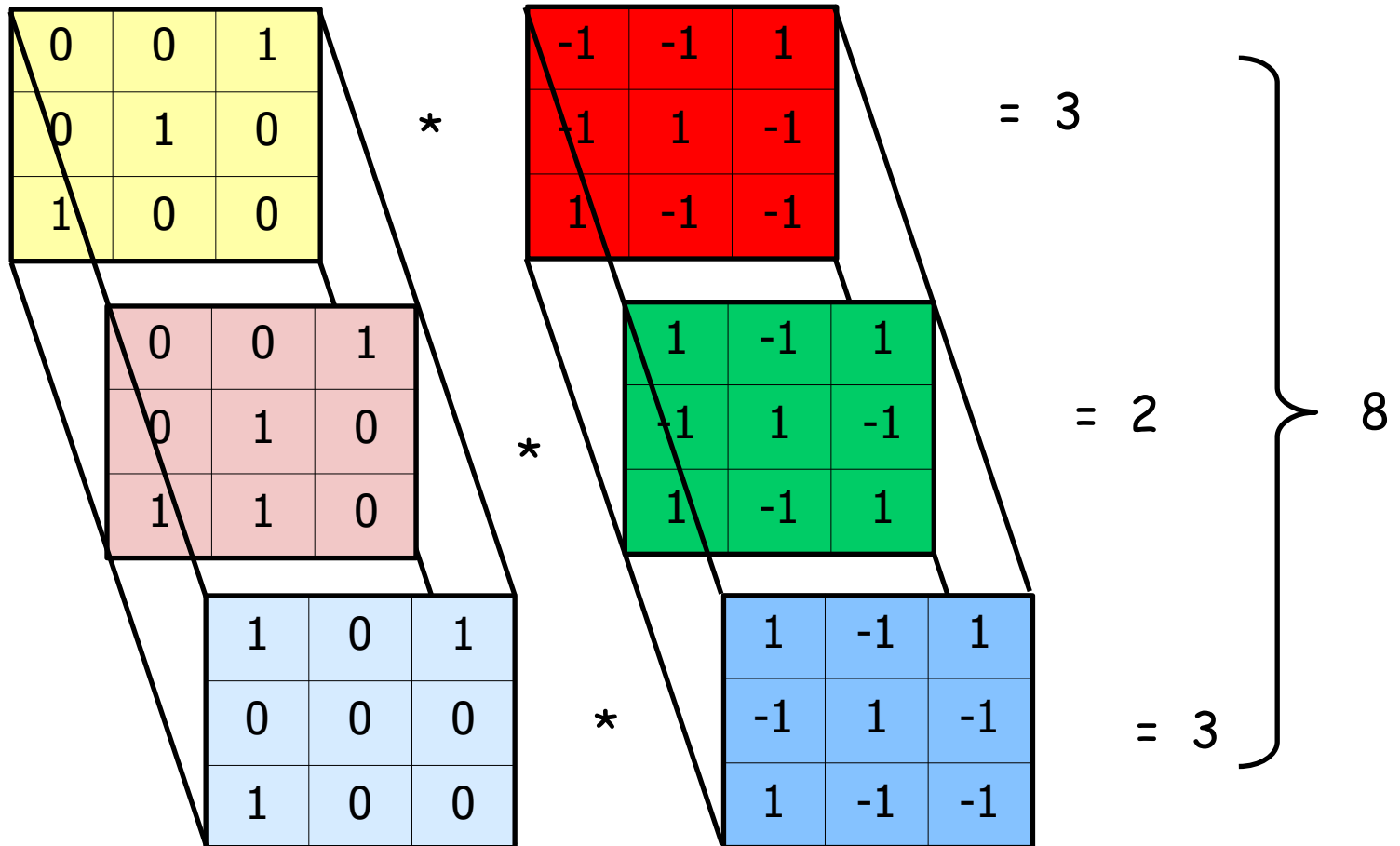
■ Convolution (3D)



$$I * K = \sum_{channel} \sum_{i,j} K_{ij}^{channel} \times I_{ij}^{channel}$$

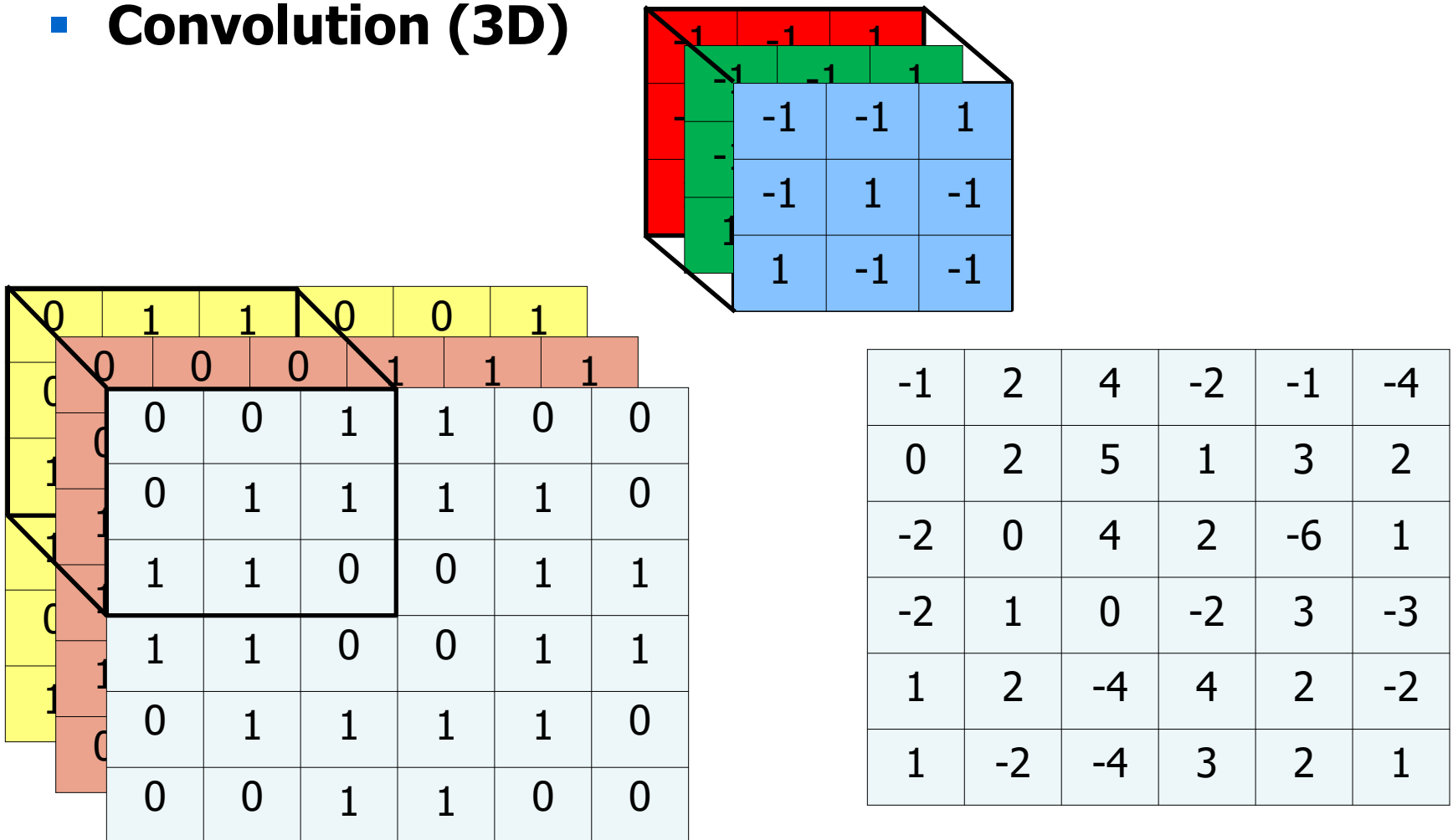
Feature Extraction

■ Convolution (3D)



Feature Extraction

■ Convolution (3D)



Feature Extraction

- Examples of Convolution

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

Horizontal Line



Input



Output
(Feature Map)

Feature Extraction

■ Pooling

- Subsampling from m by m pixels into 1 pixels
- Max, averaging or L^p pooling

0	0	0	0	0	1	0	0
0	0	0	0	0	0	1	0
0	0	3	0	0	0	0	1
0	1	0	0	0	0	0	0
0	0	0	0	0	0	1	0
1	0	0	0	0	2	0	0
0	1	0	0	0	0	0	0
0	0	1	0	0	0	0	0

Feature map

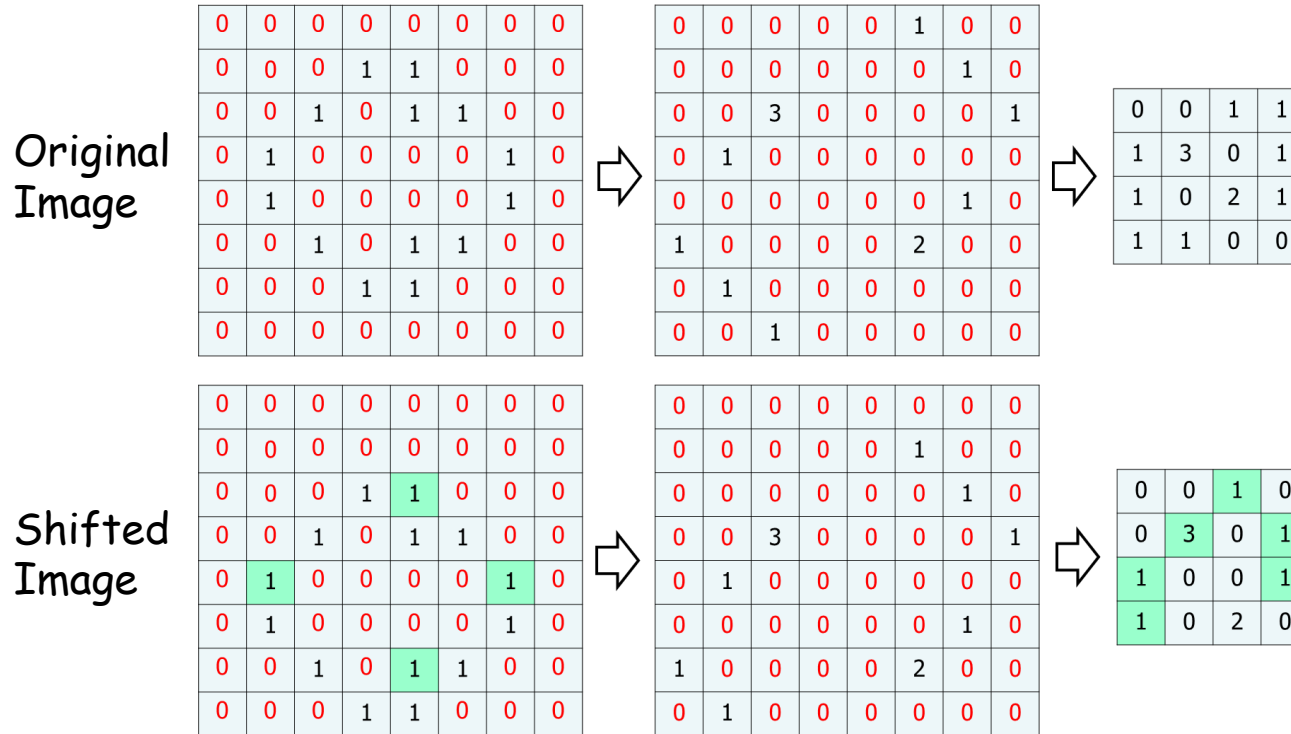
0	0	1	1
1	3	0	1
1	0	2	1
1	1	0	0

Subsampled
feature map

Feature Extraction

■ Advantage of Pooling

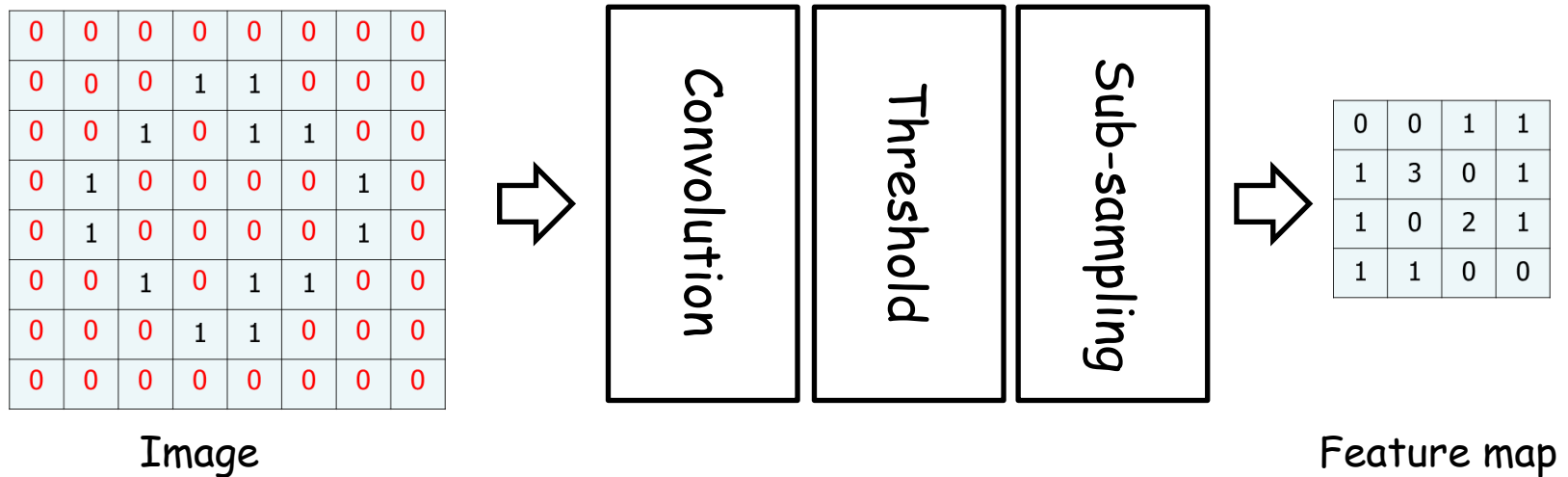
- Reducing the number of parameters
- Generating more robust feature maps: Shift Invariant



More
non-zero values
are matched !!

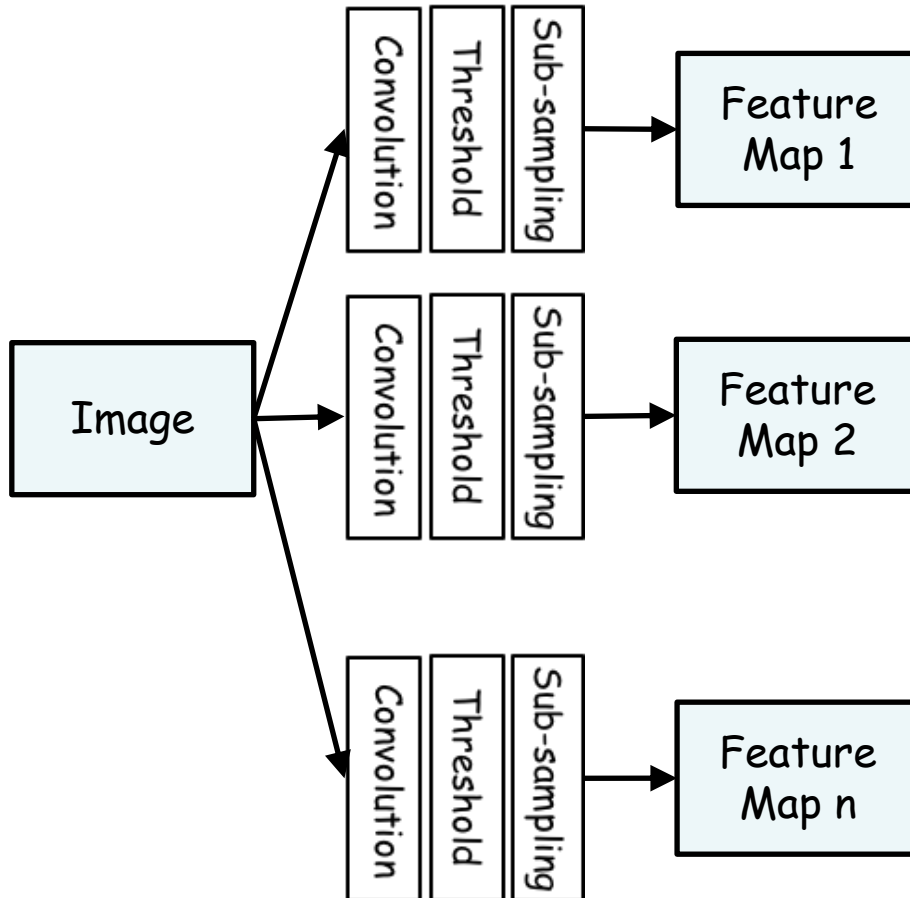
Convolutional Neural Networks

- Feature Extraction



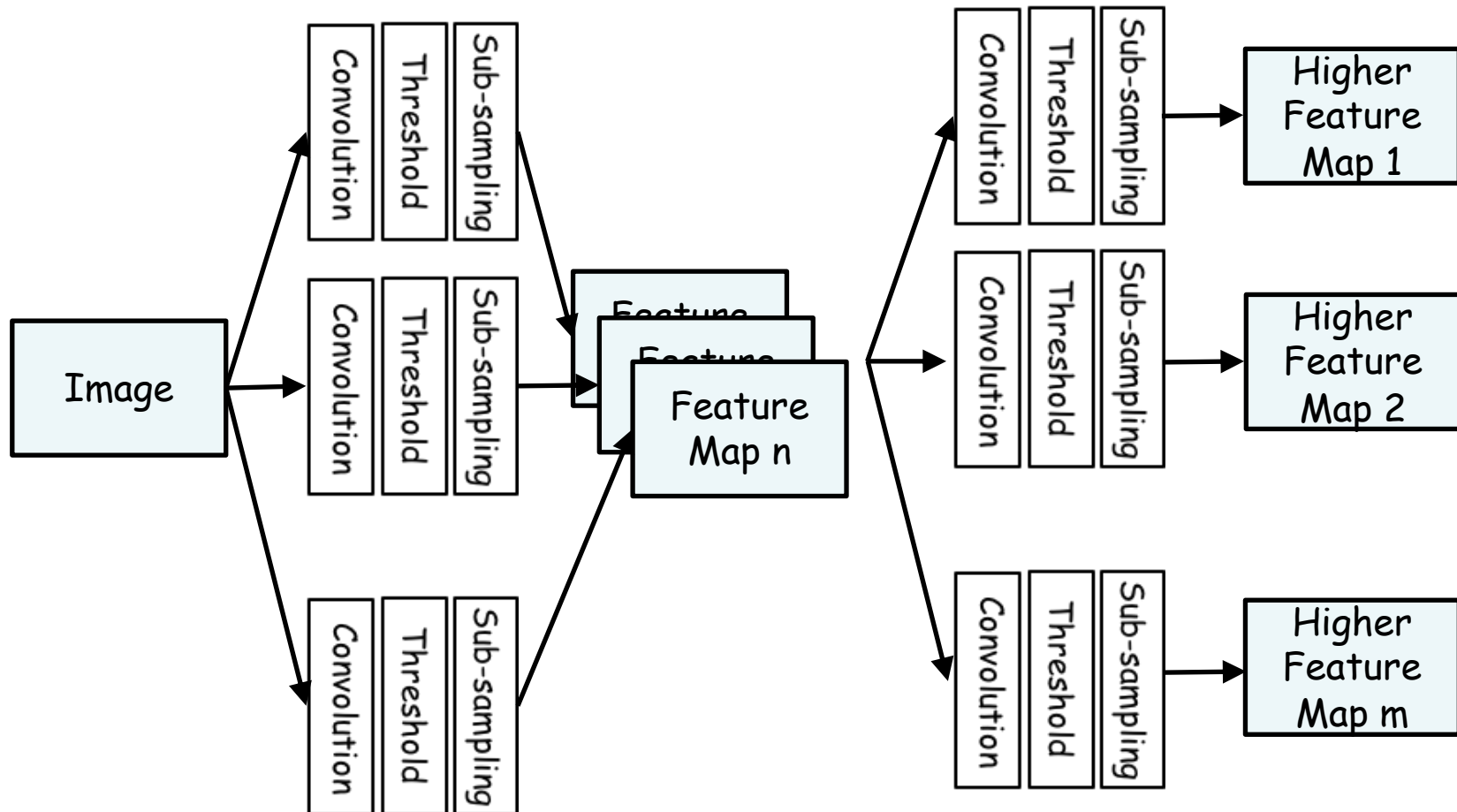
Convolutional Neural Networks

- Structure



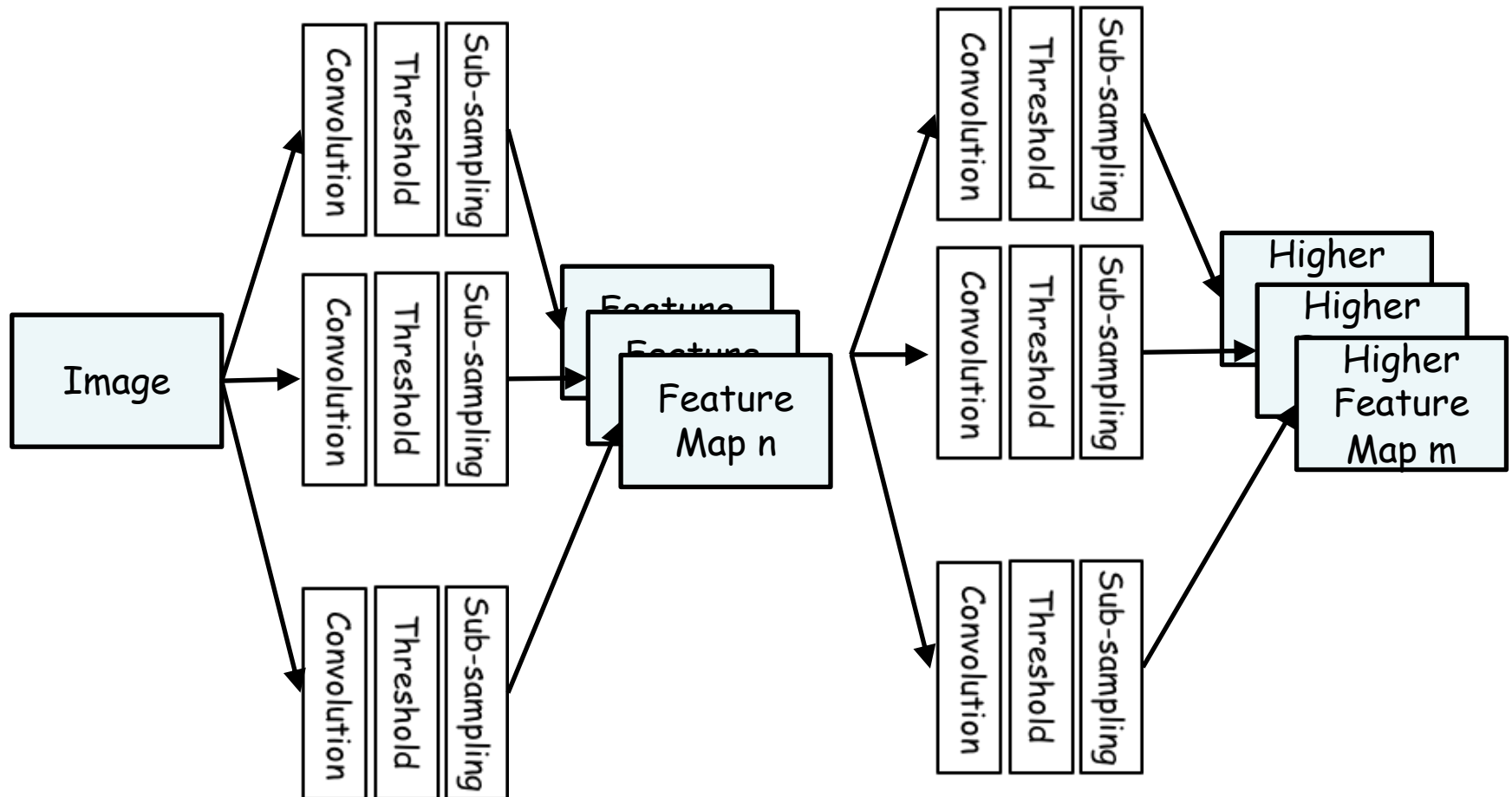
Convolutional Neural Networks

■ Structure



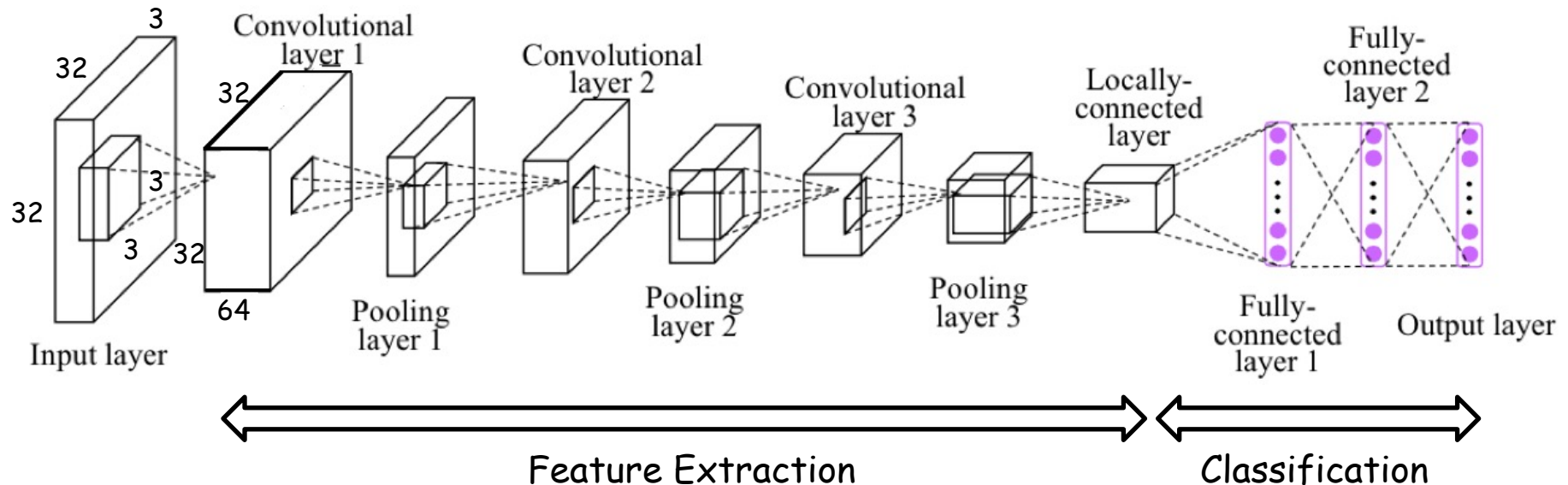
Convolutional Neural Networks

■ Structure



Convolutional Neural Networks

■ Graphical Representation



Convolutional Neural Networks

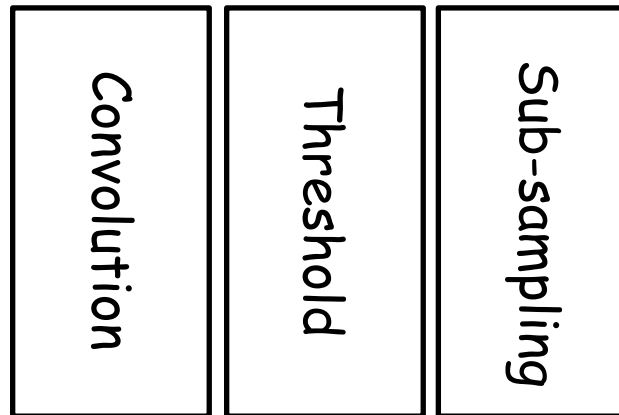
- **Who determines convolution masks?**
 - designed by EXPERTS!!
- **In CNN**
 - CNNs can be converted into neural networks
 - Convolution masks are converted into connection weights
 - Masks are found with gradient descent methods

Neural Network Representation

- Feature Extraction

0	0	0	0	0	0	0	0
0	0	0	1	1	0	0	0
0	0	1	0	1	1	0	0
0	1	0	0	0	0	1	0
0	1	0	0	0	0	1	0
0	0	1	0	1	1	0	0
0	0	0	1	1	0	0	0
0	0	0	0	0	0	0	0

Image



0	0	1	1
1	3	0	1
1	0	2	1
1	1	0	0

Feature map

Neural Network Representation

■ Convolution & Threshold

- Values in Kernel
=> Connection weights
- Most of them are zeros

input

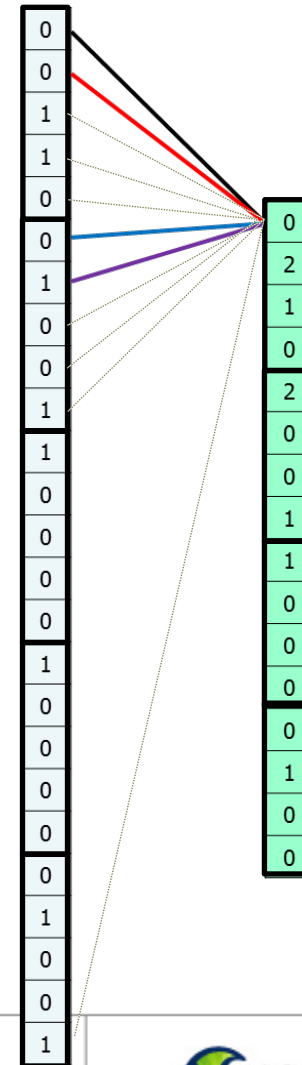
0	0	1	1	0
0	1	0	0	1
1	0	0	0	0
1	0	0	0	0
0	1	0	0	1

0	1
1	0

Kernel

output

0	2	1	0
2	0	0	1
1	0	0	0
0	1	0	0



Neural Network Representation

■ Convolution & Threshold

- Values in Kernel
=> Connection weights
- Most of them are zeros

input

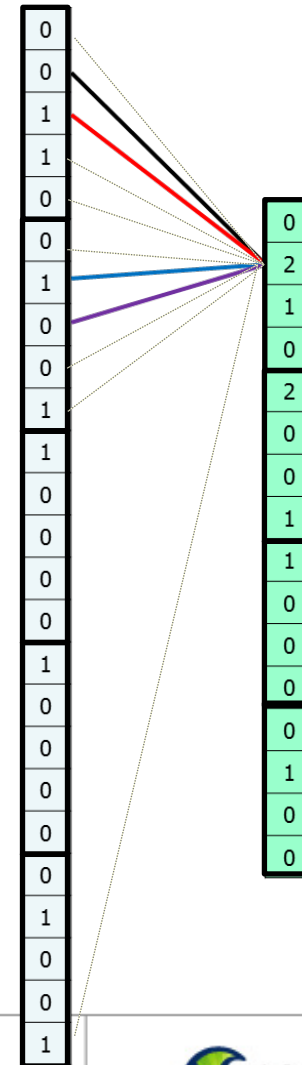
0	0	1	1	0
0	1	0	0	1
1	0	0	0	0
1	0	0	0	0
0	1	0	0	1

0	1
1	0

Kernel

output

0	2	1	0
2	0	0	1
1	0	0	0
0	1	0	0



Neural Network Representation

■ Convolution & Threshold

- Values in Kernel
=> Connection weights
- Most of them are zeros

input

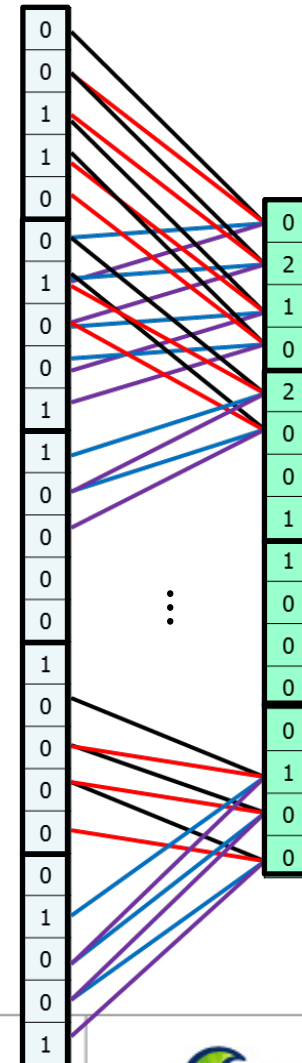
0	0	1	1	0
0	1	0	0	1
1	0	0	0	0
1	0	0	0	0
0	1	0	0	1

0	1
1	0

Kernel

output

0	2	1	0
2	0	0	1
1	0	0	0
0	1	0	0



Neural Network Representation

■ Pooling

- Weight are fixed to 1
- Activation:

$$y = \max(w_1x_1, w_2x_2, w_3x_3, w_4x_4)$$

input

0	0	1	1	0
0	1	0	0	1
1	0	0	0	0
1	0	0	0	0
0	1	0	0	1

output

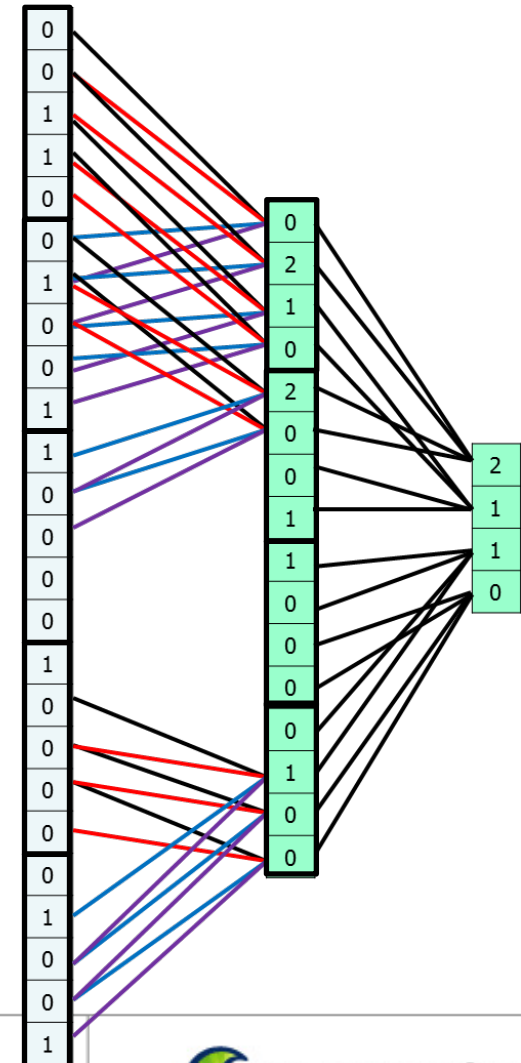
0	2	1	0
2	0	0	1
1	0	0	0
0	1	0	0

0	1
1	0

Kernel

2	1
1	0

Subsampled output

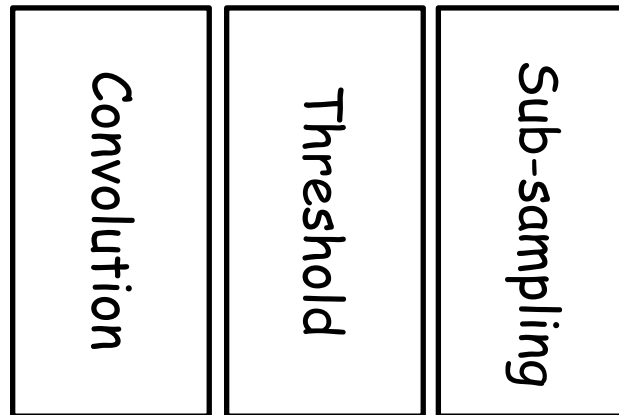


Neural Network Representation

- Feature Extraction

0	0	0	0	0	0	0	0
0	0	0	1	1	0	0	0
0	0	1	0	1	1	0	0
0	1	0	0	0	0	1	0
0	1	0	0	0	0	1	0
0	0	1	0	1	1	0	0
0	0	0	1	1	0	0	0
0	0	0	0	0	0	0	0

Image



0	0	1	1
1	3	0	1
1	0	2	1
1	1	0	0

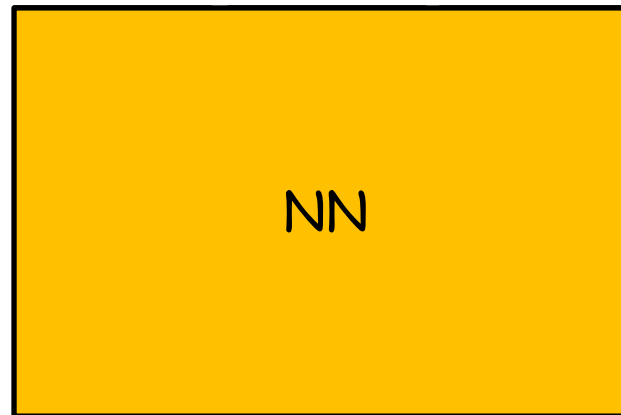
Feature map

Neural Network Representation

- Feature Extraction

0	0	0	0	0	0	0	0
0	0	0	1	1	0	0	0
0	0	1	0	1	1	0	0
0	1	0	0	0	0	1	0
0	1	0	0	0	0	1	0
0	0	1	0	1	1	0	0
0	0	0	1	1	0	0	0
0	0	0	0	0	0	0	0

Image

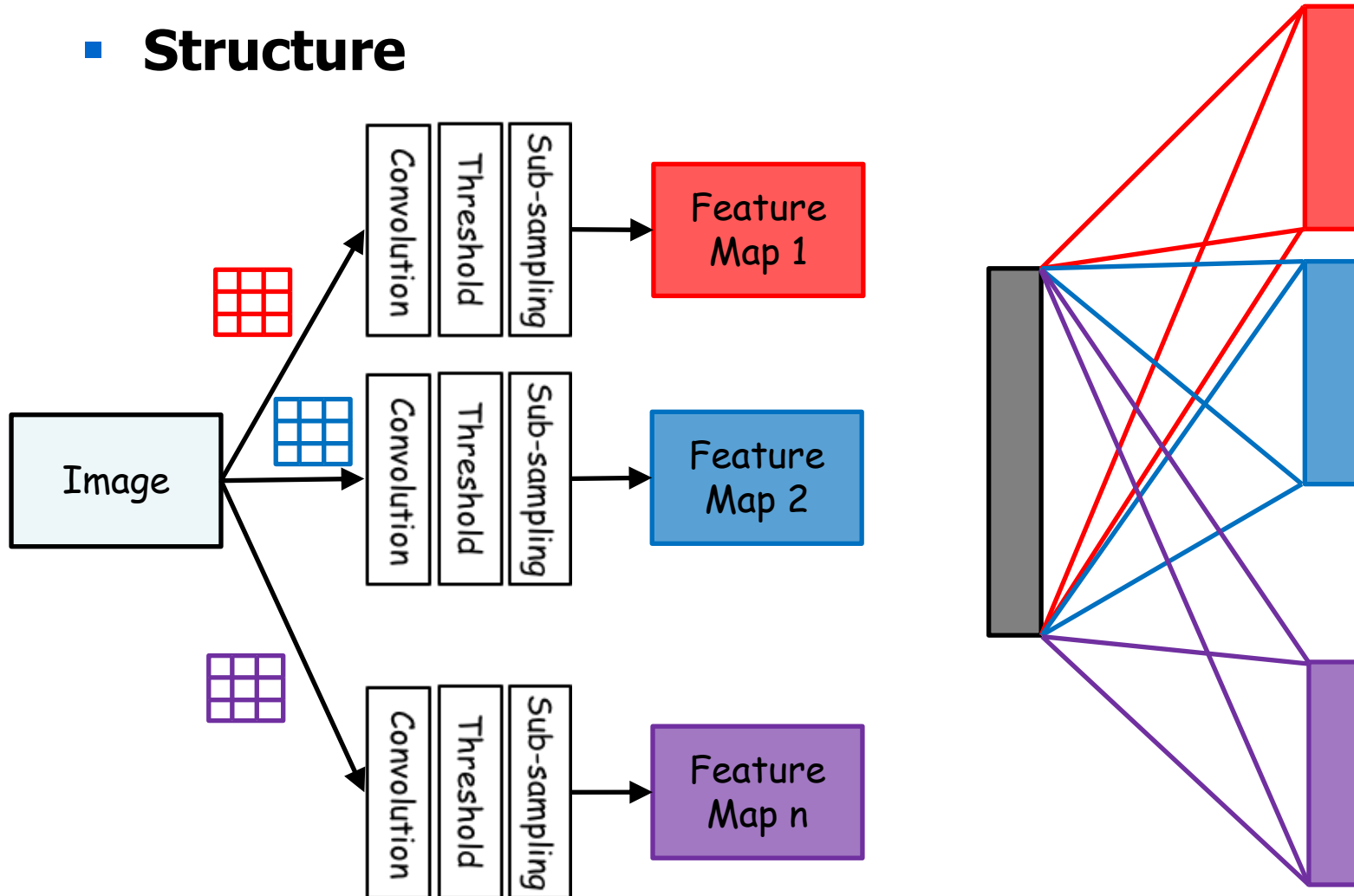


0	0	1	1
1	3	0	1
1	0	2	1
1	1	0	0

Feature map

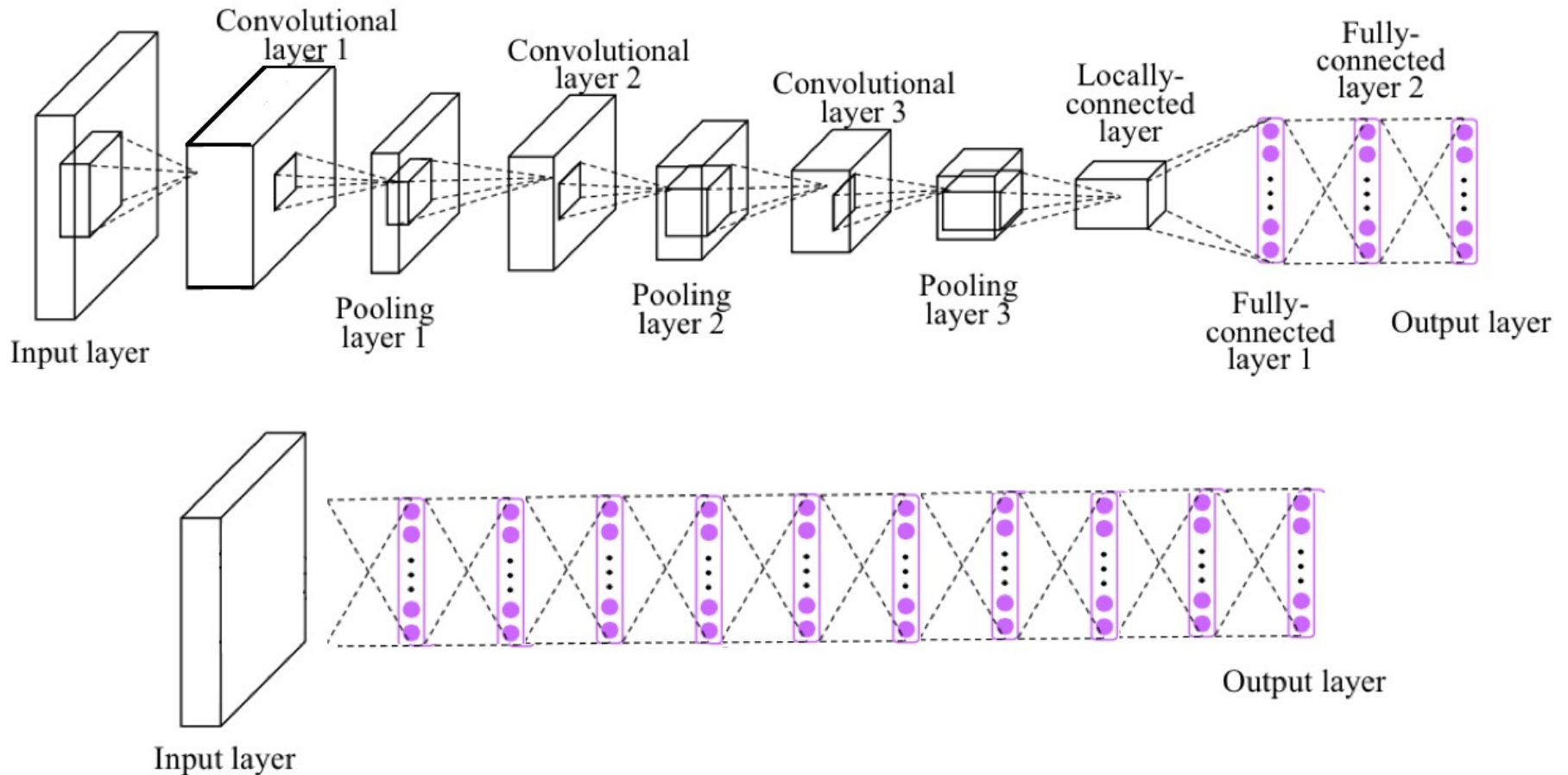
Neural Network Representation

- **Structure**



Neural Network Representation

■ Structure



Convolutional Neural Networks

■ Example

