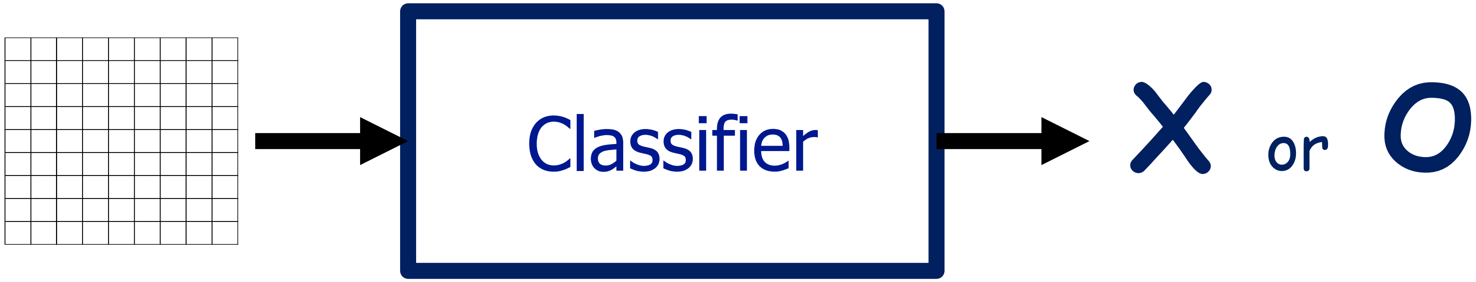
Convolutional Neural Networks

Image Classification

- **X, O Classification**



Classifier

X or O

A two-dimensional
array of pixels

Image Classification

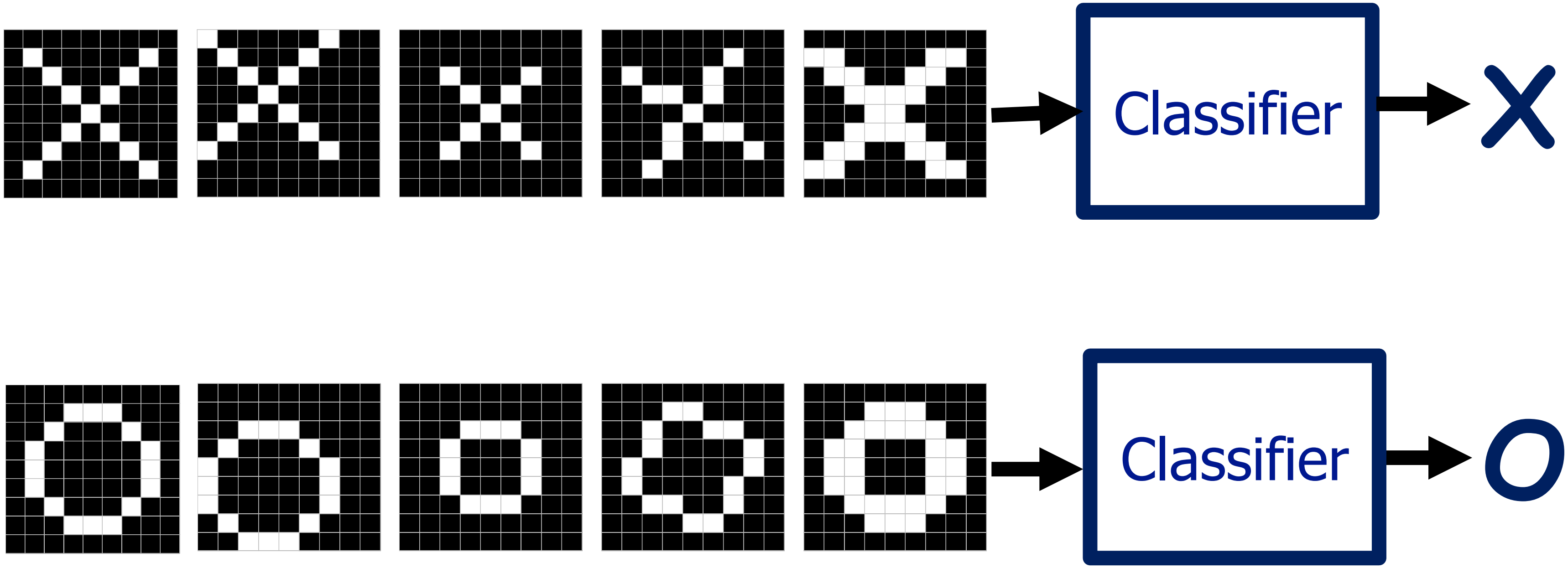


Image Classification

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

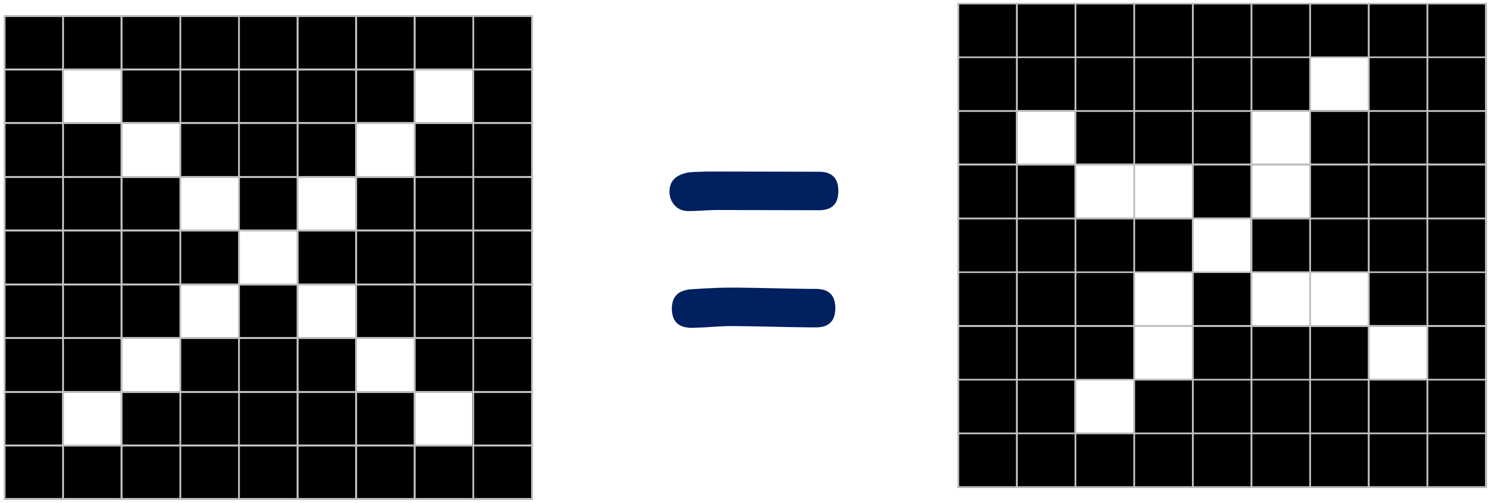
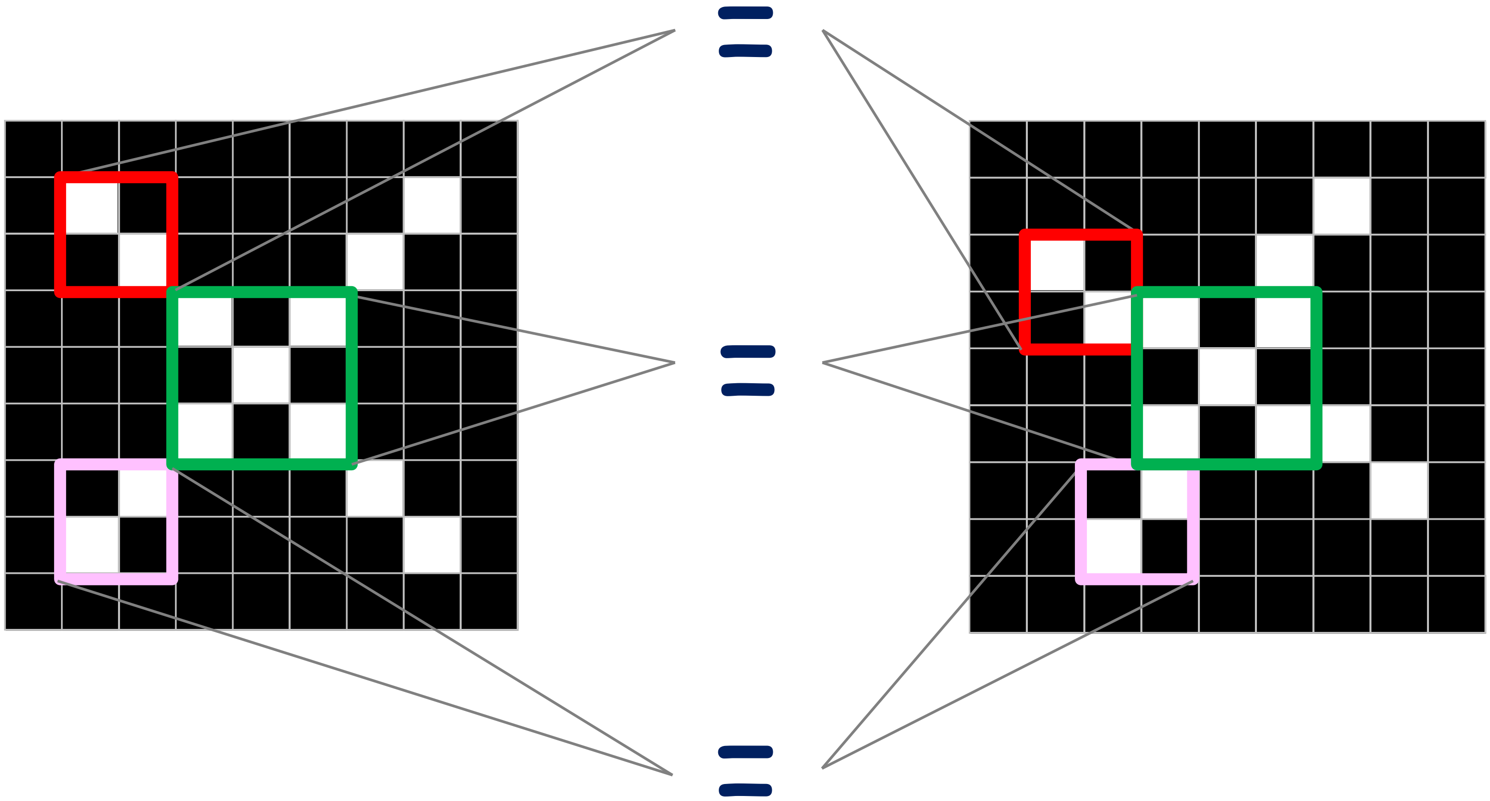


Image Classification

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



Feature Extraction

- **Convolution**
  - A way to find out local features

$$\begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 0 \end{bmatrix} * \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} = 4$$

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

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

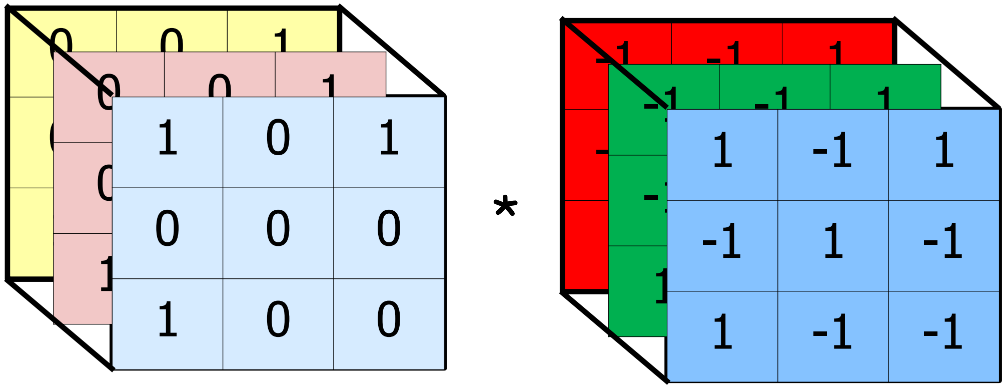
$$(I * K)_{xy} = \sum_{i=1}^{w}\sum_{j=1}^{w} K_{ij} \cdot I_{x+i-1,y+j-1}$$

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)**

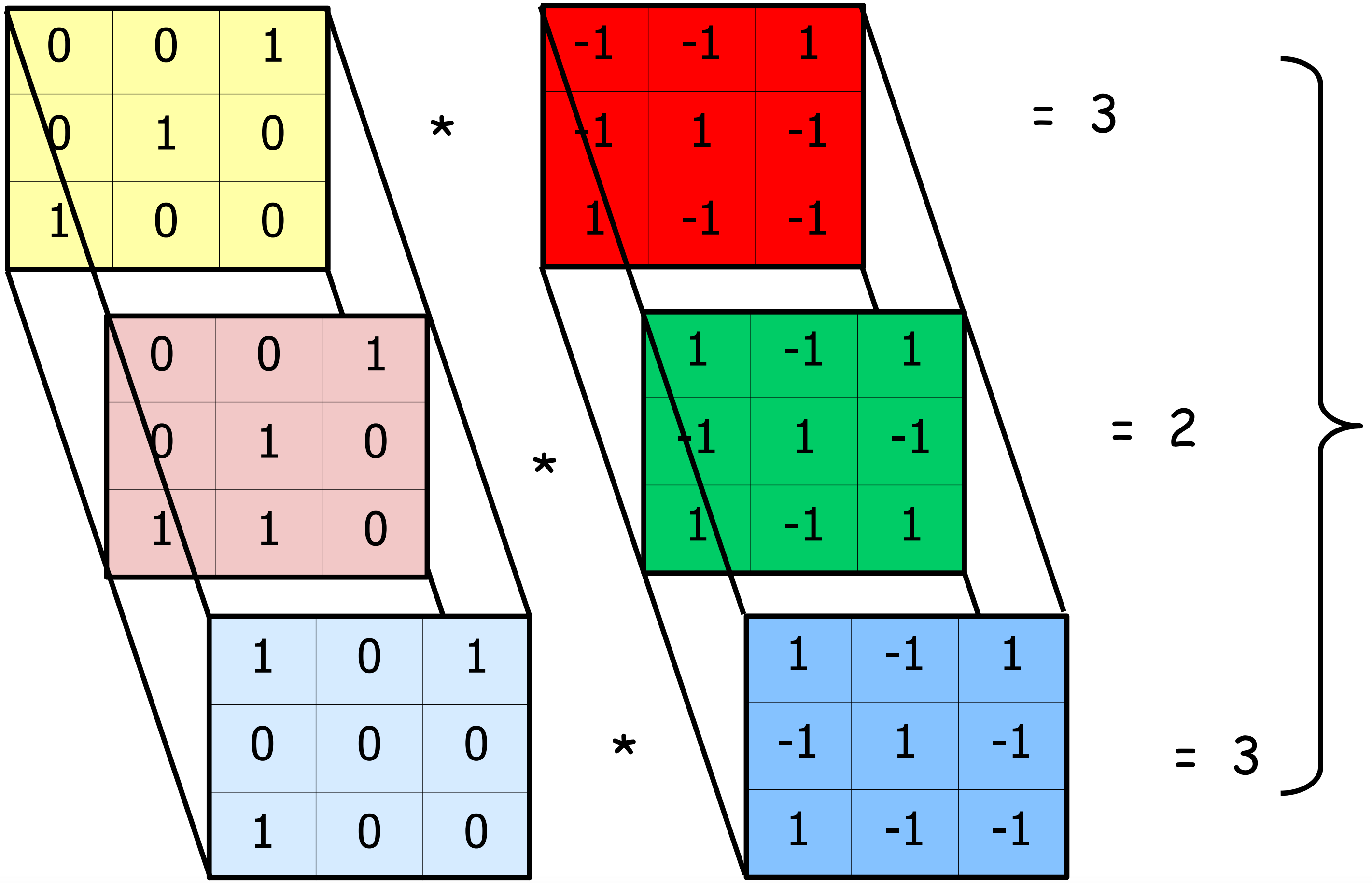


= 3 +2 +3 = 8

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

Feature Extraction

- **Convolution (3D)**



$$\begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} * \begin{bmatrix} -1 & -1 & 1 \\ -1 & 1 & -1 \\ 1 & -1 & -1 \end{bmatrix} = 3$$

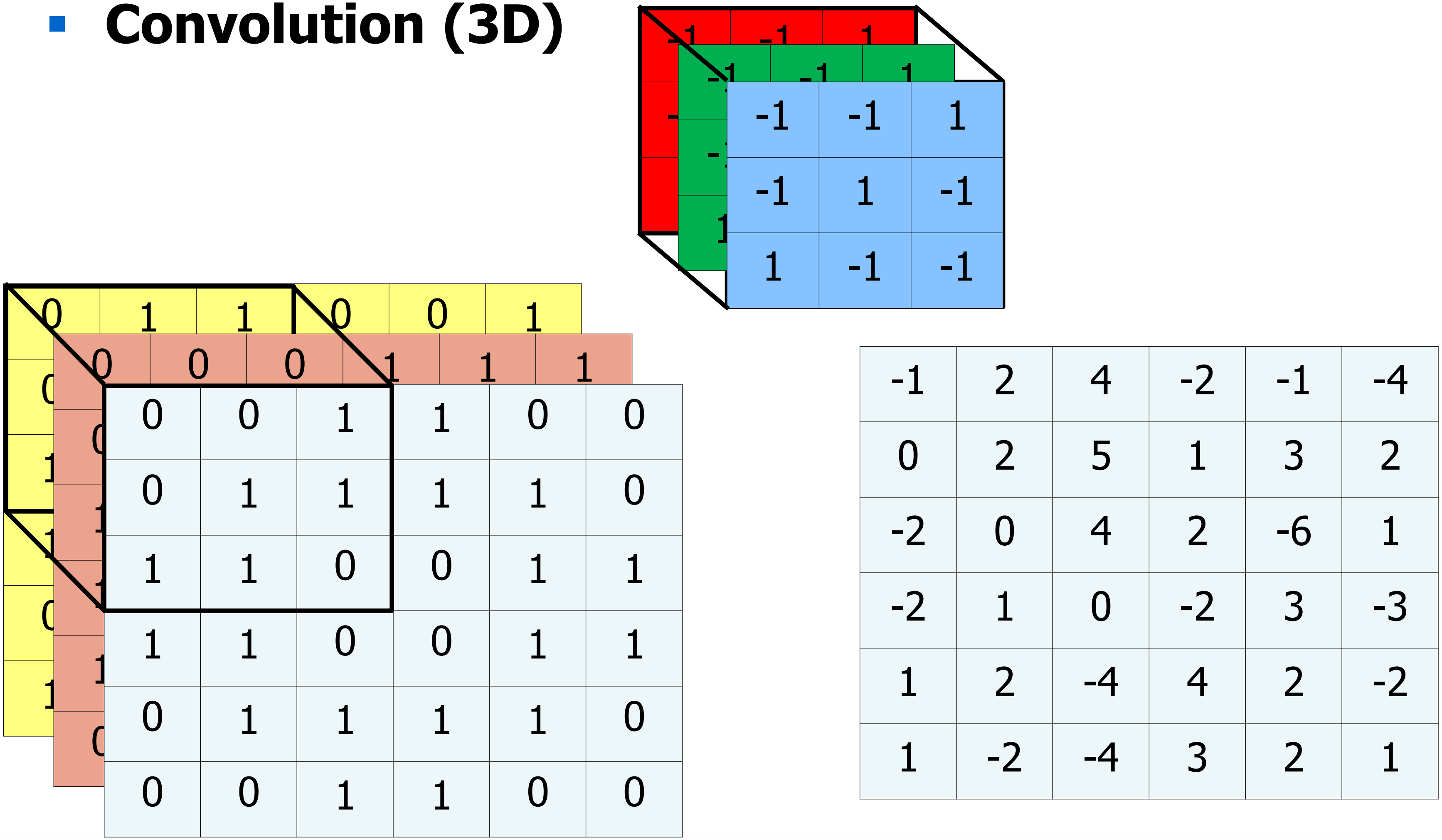
$$\begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 1 & 0 \end{bmatrix} * \begin{bmatrix} 1 & -1 & 1 \\ -1 & 1 & -1 \\ 1 & -1 & 1 \end{bmatrix} = 2$$

$$\begin{bmatrix} 1 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix} * \begin{bmatrix} 1 & -1 & 1 \\ -1 & 1 & -1 \\ 1 & -1 & -1 \end{bmatrix} = 3$$

$$\left.\begin{matrix} 3 \\ 2 \\ 3 \end{matrix}\right\} 8$$

Feature Extraction

- **Convolution (3D)**



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

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

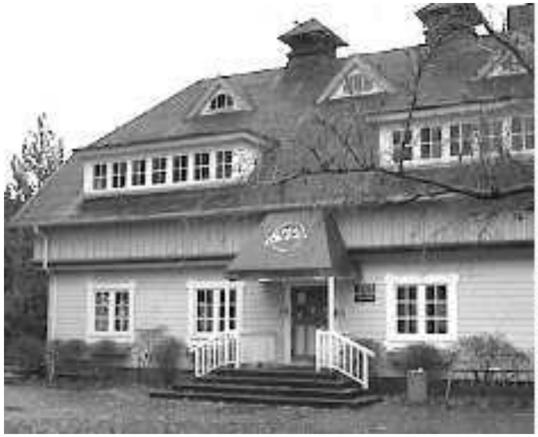
-1	2	4	-2	-1	-4
0	2	5	1	3	2
-2	0	4	2	-6	1
-2	1	0	-2	3	-3
1	2	-4	4	2	-2
1	-2	-4	3	2	1

Feature Extraction

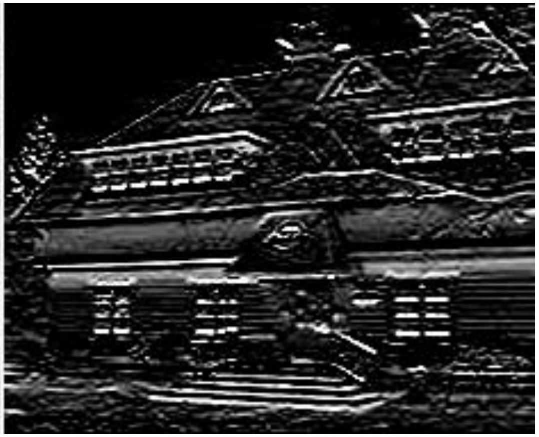
- **Examples of Convolution**

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

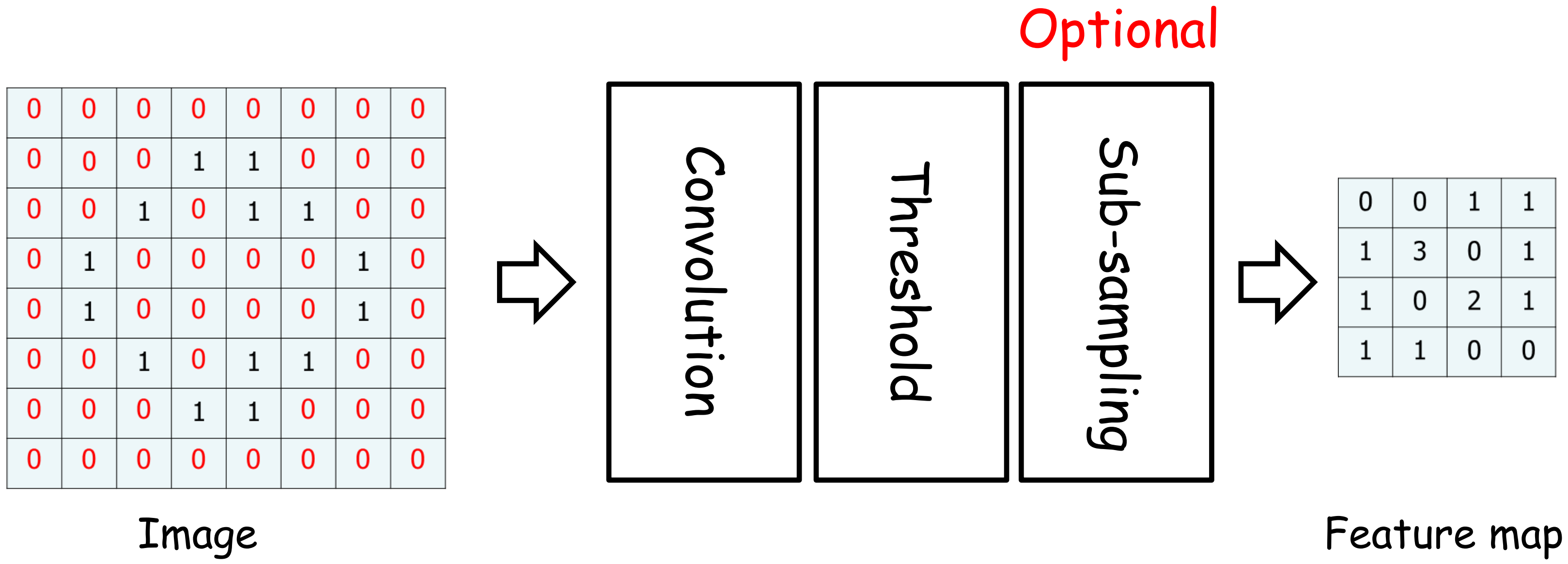
Feature map

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

Subsampled feature map

Convolutional Neural Networks

- **Feature Extraction**

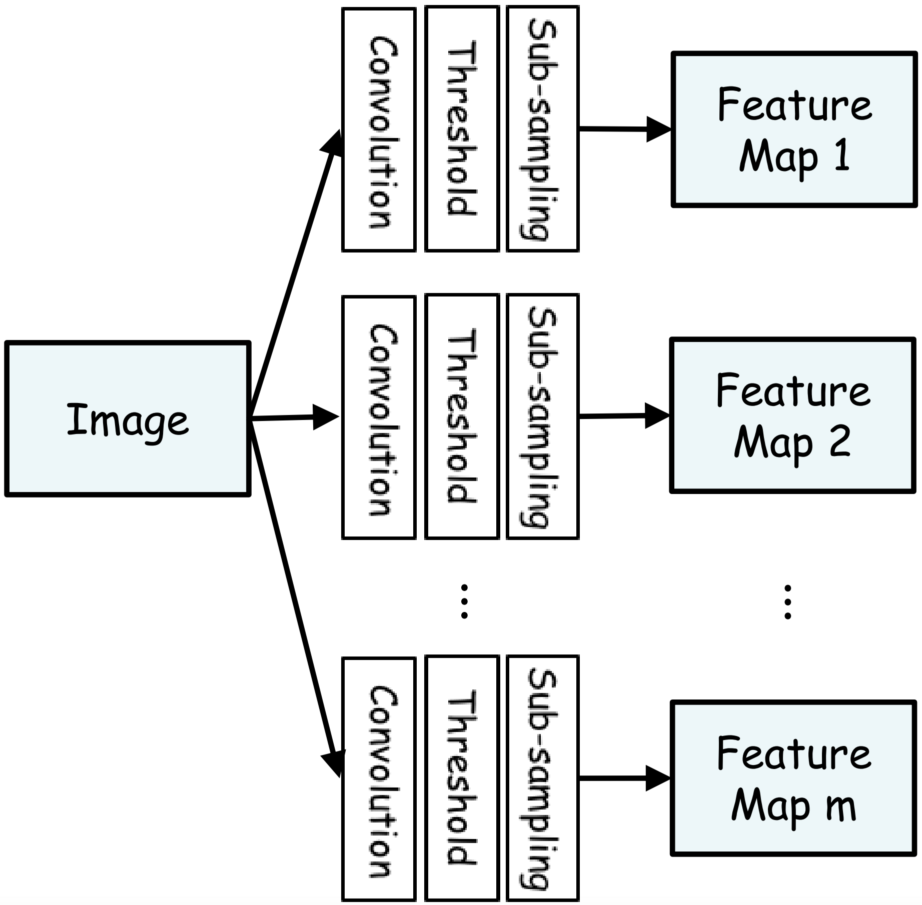


0	0	0	0	0	0	0	0
0	0	0	1	1	0	0	0
0	0	1	0	1	1	0	0
0	1	0	0	0	0	1	0
0	1	0	0	0	0	1	0
0	0	1	0	1	1	0	0
0	0	0	1	1	0	0	0
0	0	0	0	0	0	0	0

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

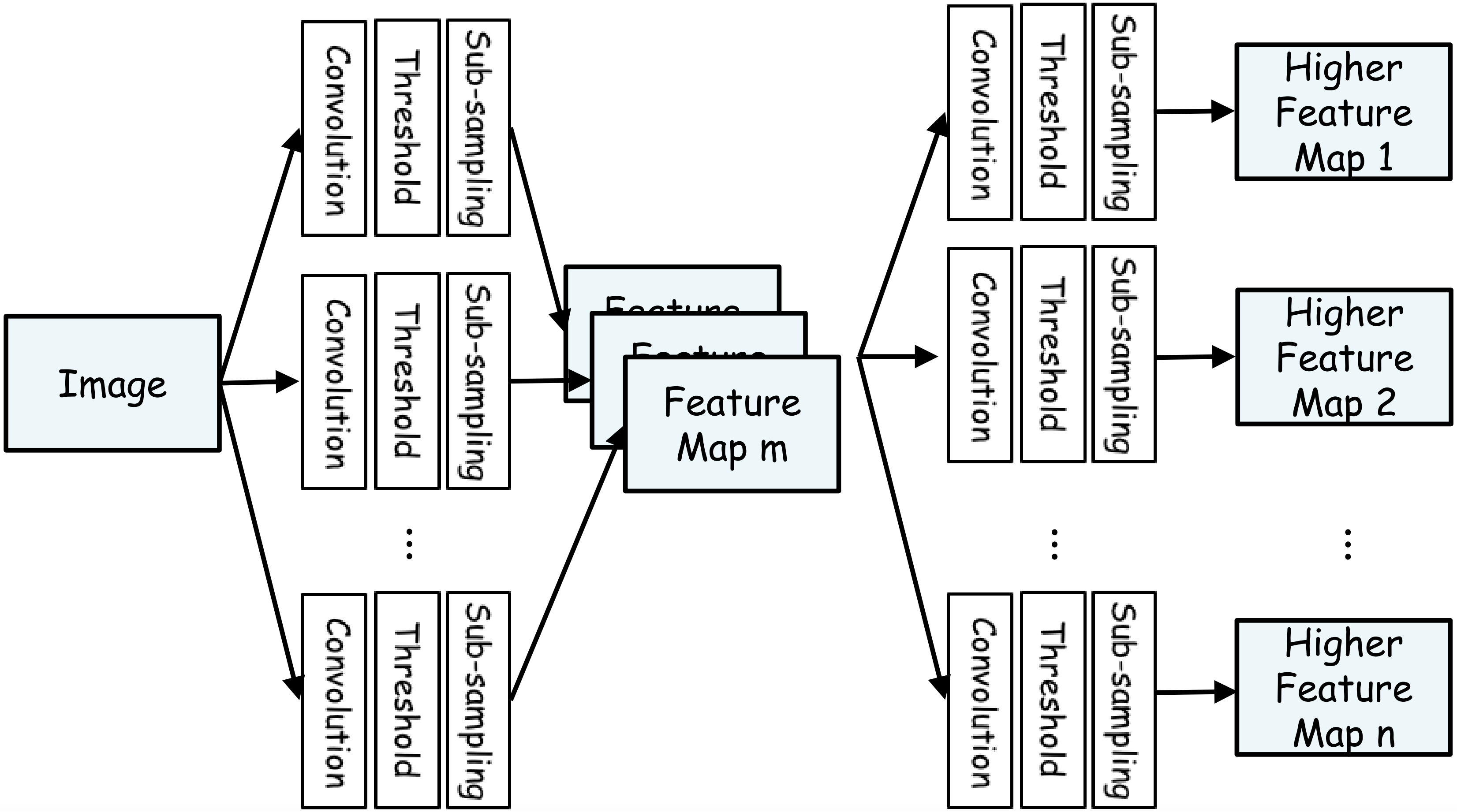
Convolutional Neural Networks

- **Structure**



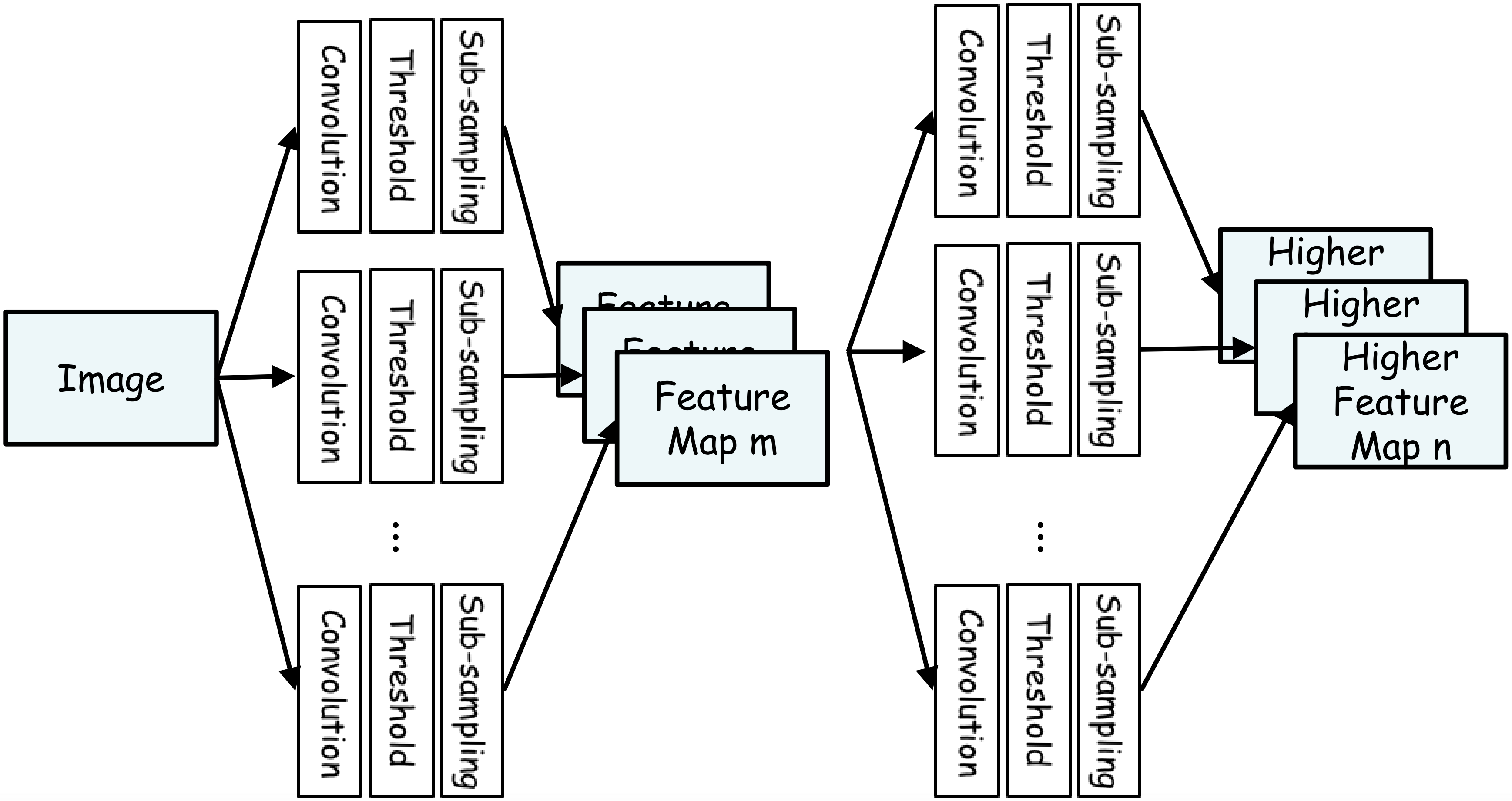
Convolutional Neural Networks

- **Structure**



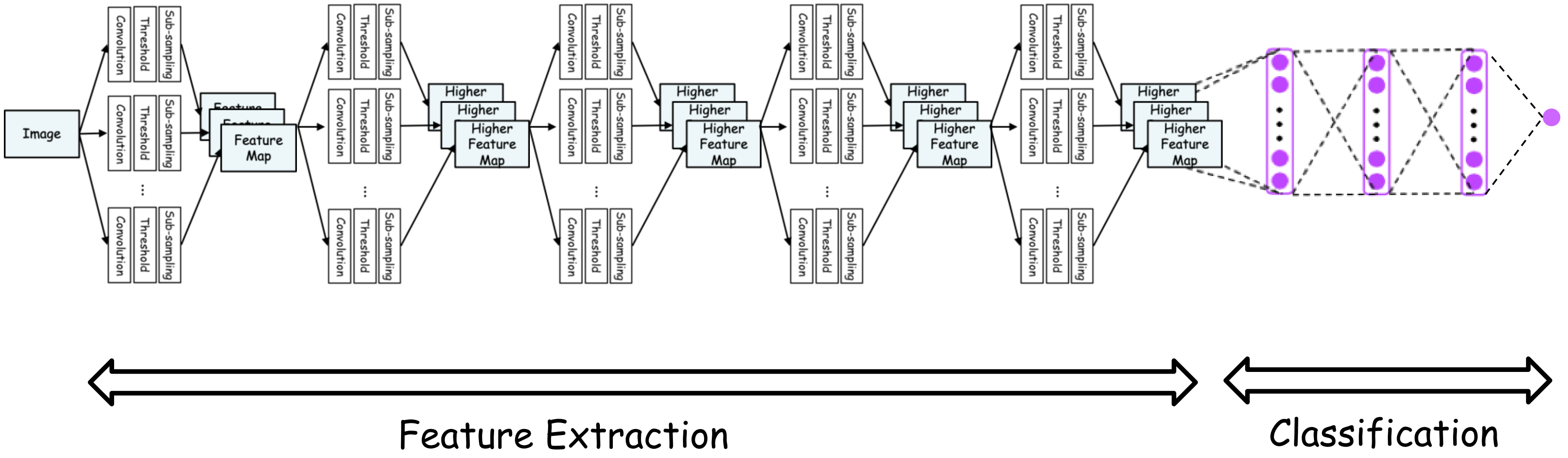
Convolutional Neural Networks

- **Structure**



Convolutional Neural Networks

- **Structure**

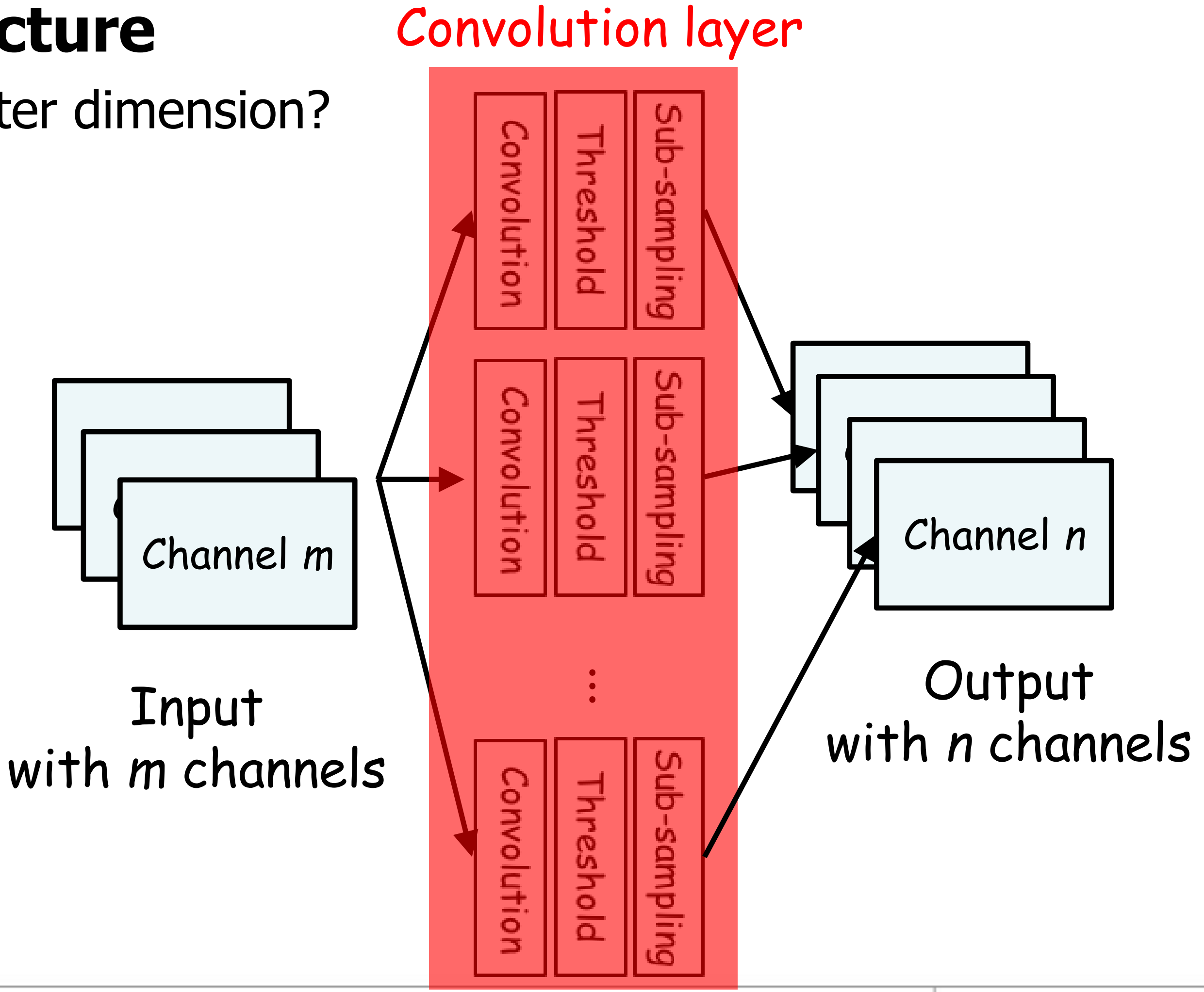


Feature Extraction

Classification

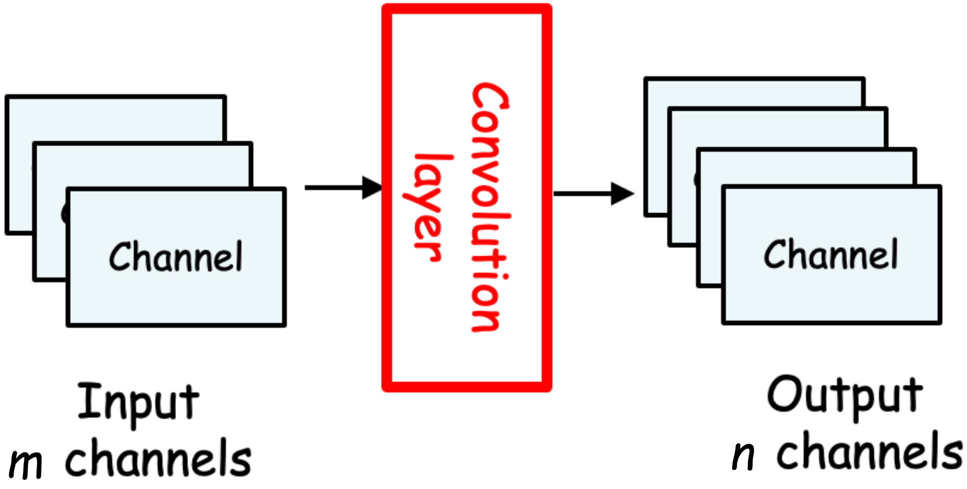
Convolutional Neural Networks

- **Structure**
  - Filter dimension?



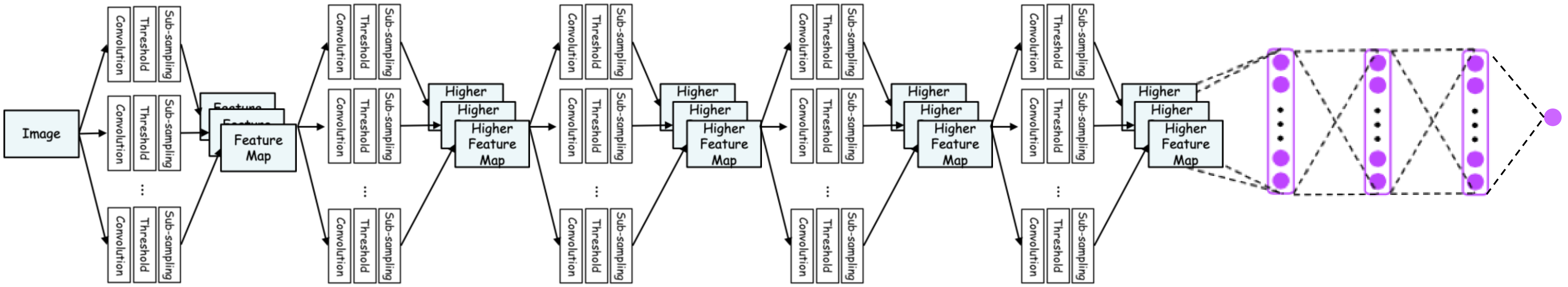
Convolutional Neural Networks

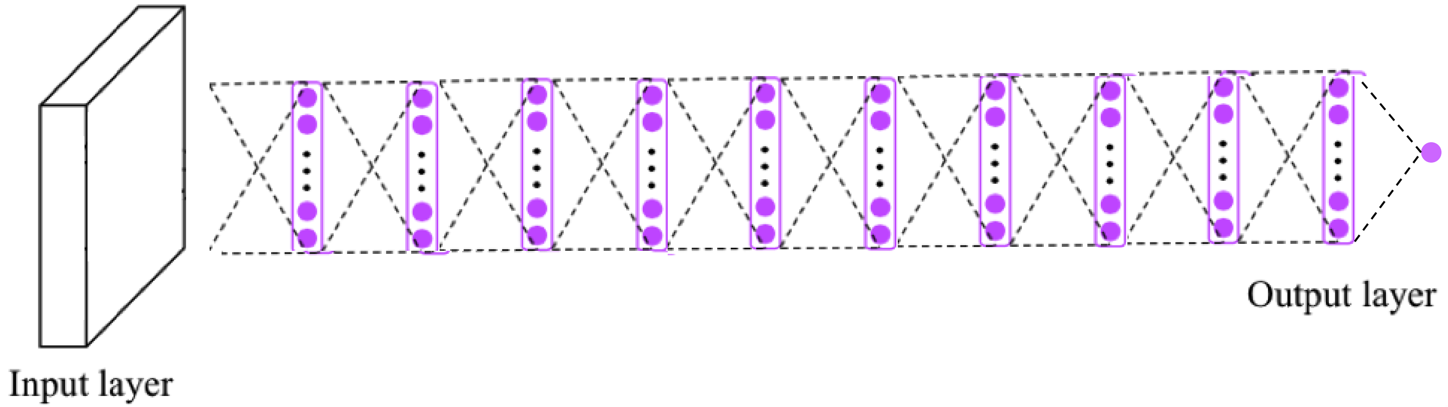
- **Structure**
  - How many convolutions in the convolution layer?
  - What are the dimensions of each filter?
  - How many trainable parameters in the conv. layer?



Convolutional Neural Networks

- **Structure**

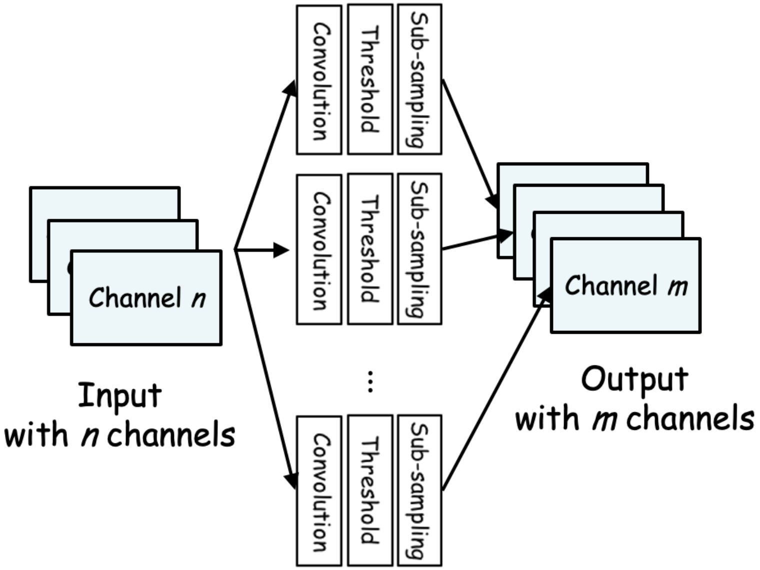




Convolutional Neural Networks

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

Graphical Notation



Convolution

Threshold

Sub-sampling

Convolution

Threshold

Sub-sampling

⋮

Convolution

Threshold

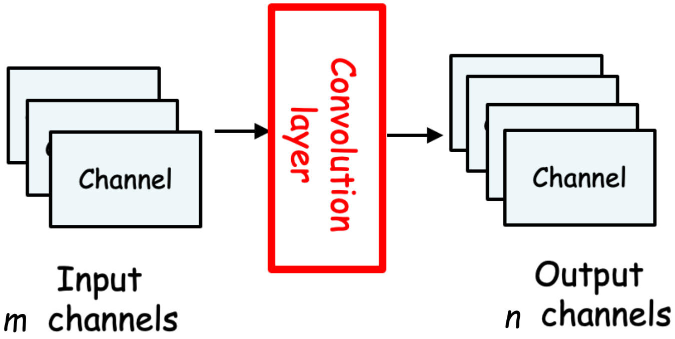
Sub-sampling

Channel *n*

Input
with *n* channels

Channel *m*

Output
with *m* channels



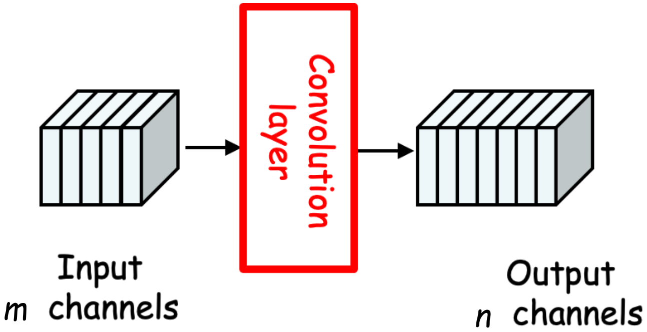
Channel

Input
m channels

Convolution layer

Channel

Output
n channels



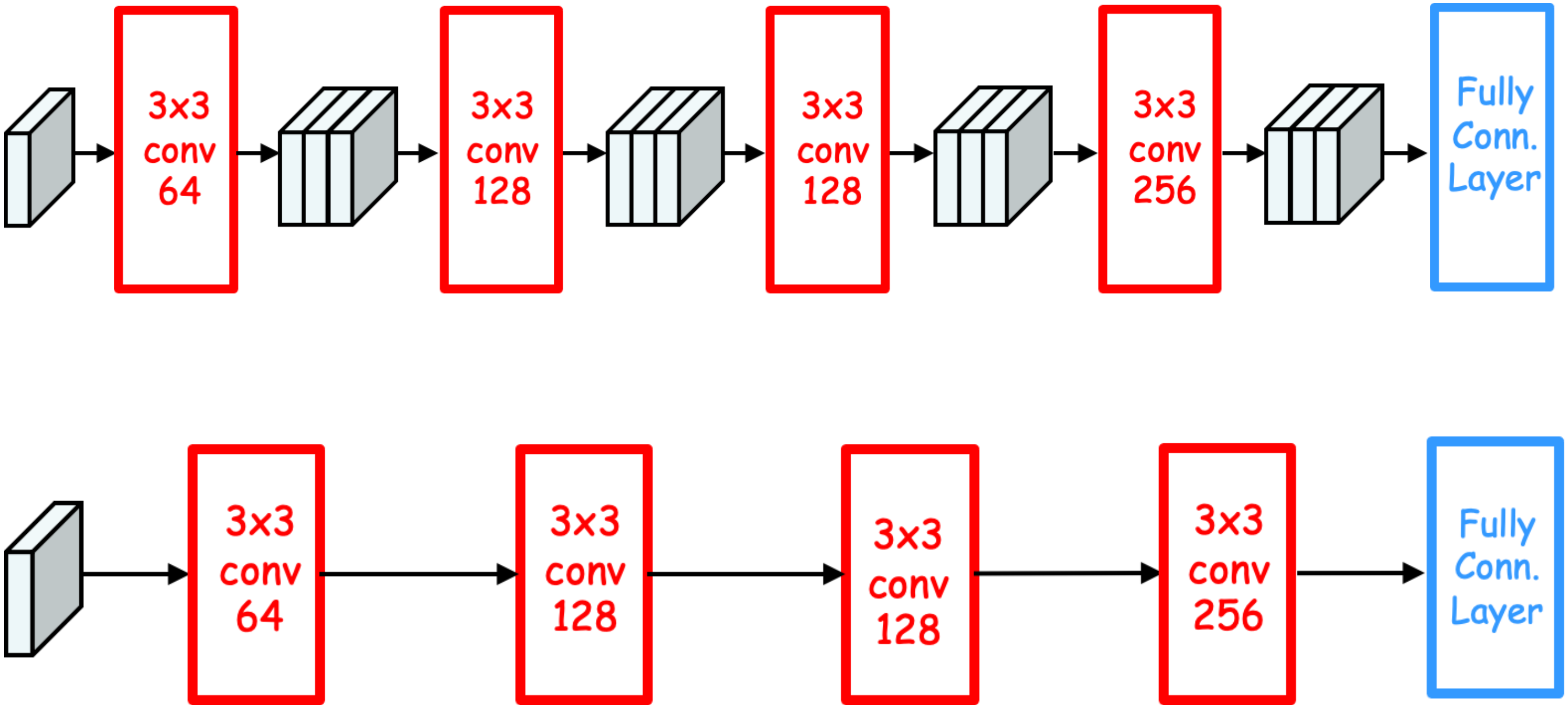
Input
m channels

Convolution layer

Output
n channels

Convolutional Neural Networks

- **Structure**


3x3 conv 64
3x3 conv 128
3x3 conv 128
3x3 conv 256
Fully Conn. Layer
3x3 conv 64
3x3 conv 128
3x3 conv 128
3x3 conv 256
Fully Conn. Layer