

투빅스 11기 정규과정

ToBig's 10기 박성진

Convolutional Neural Networks

Contents

Unit 01 | Intro

Unit 02 | Layers in CNN

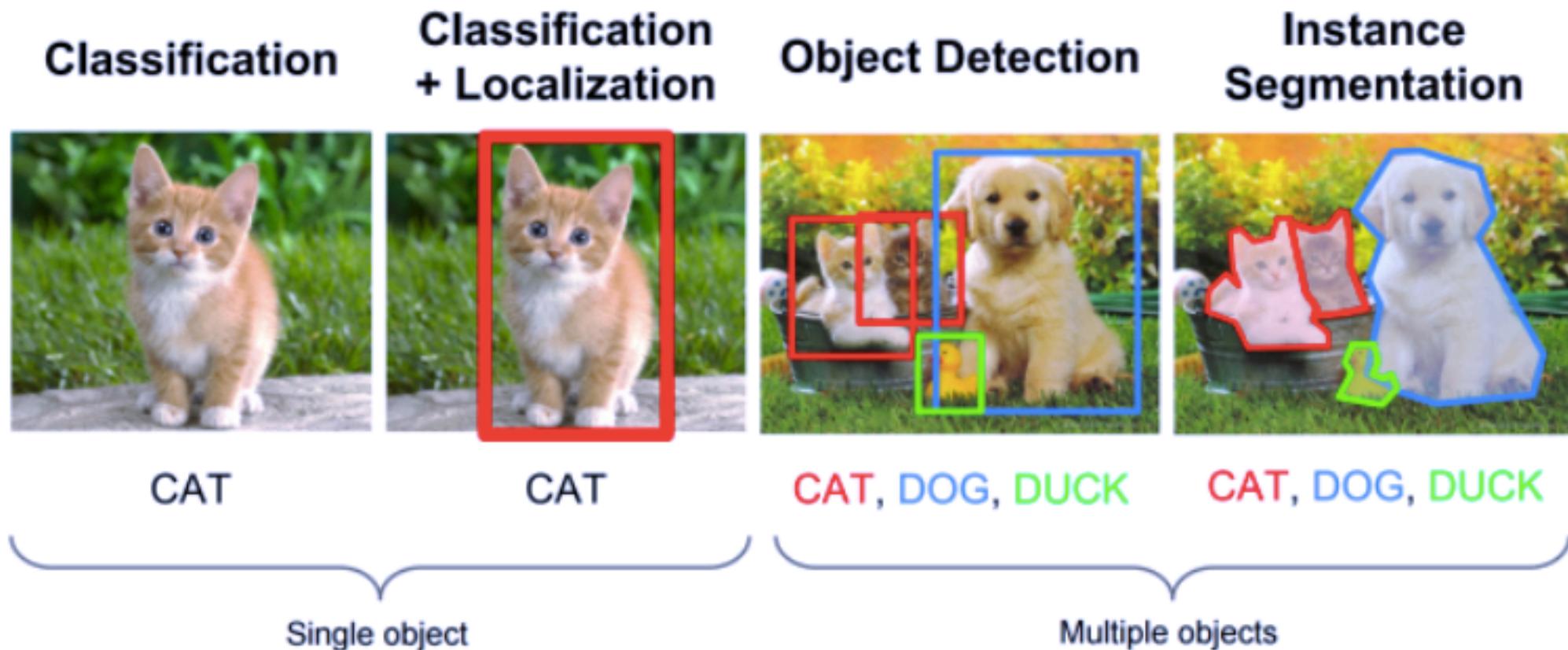
Unit 03 | Convolution

Unit 04 | Sub-Sampling

Unit 05 | Summary

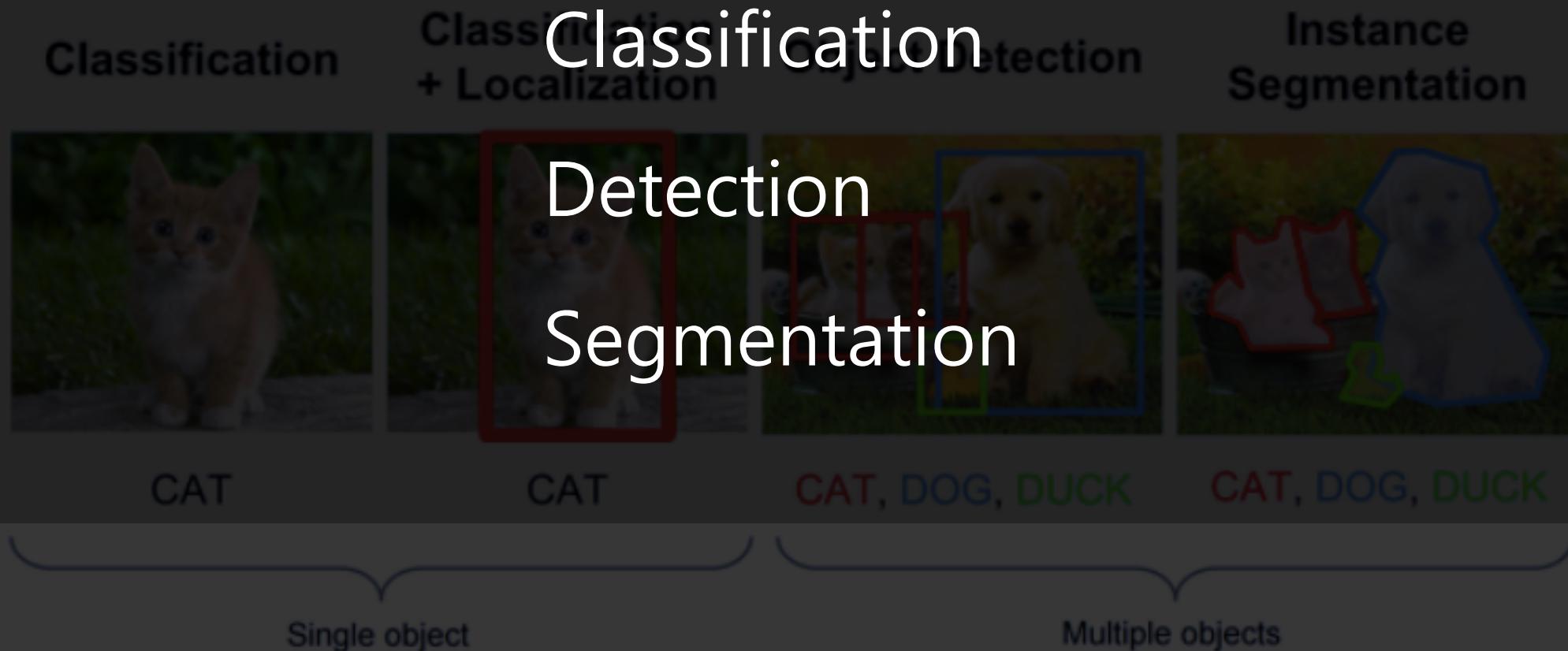
UNIT 01. Intro

◆ Intro - Applications of CNN



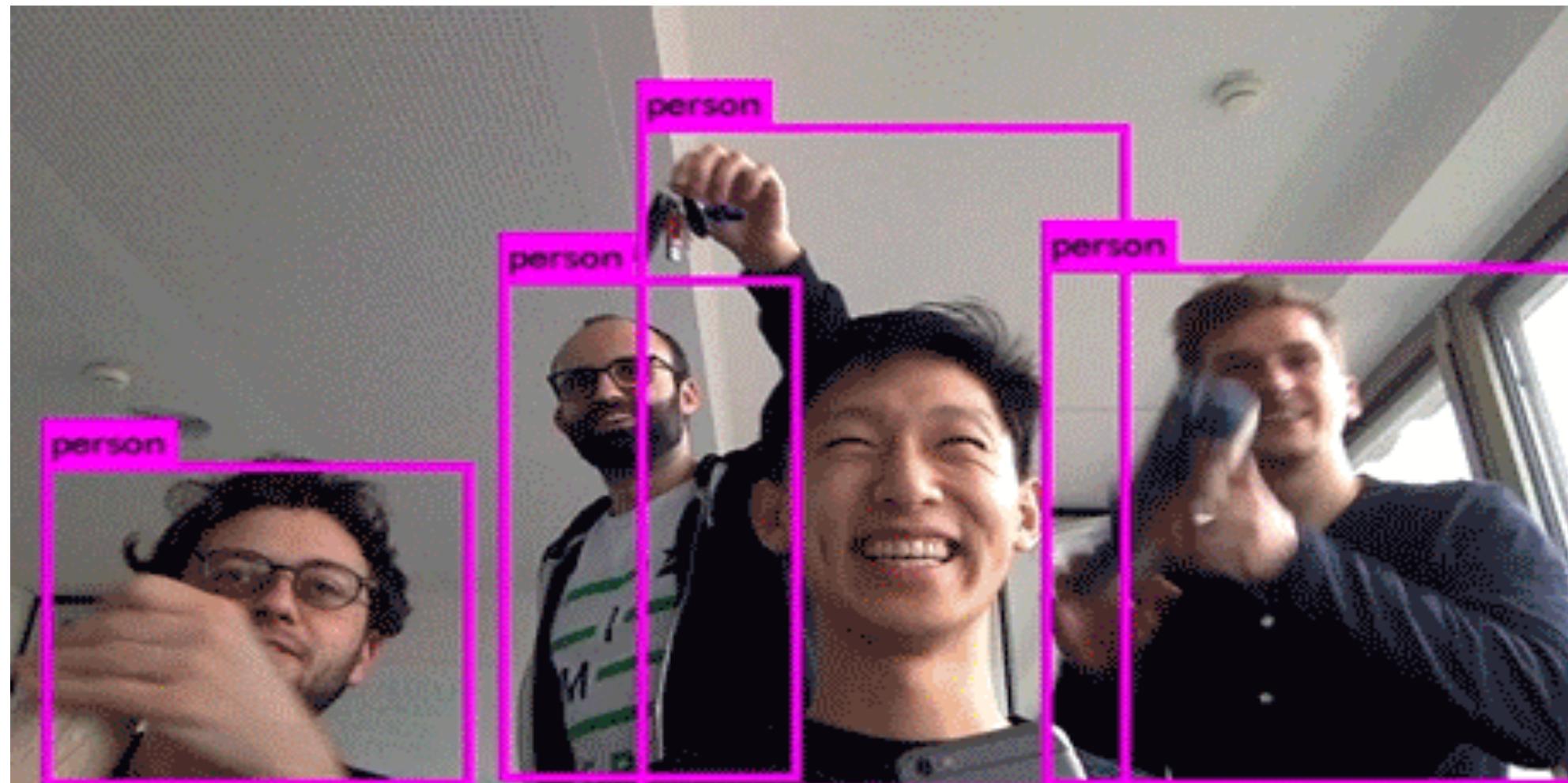
UNIT 01. Intro

◆ Intro - Applications of CNN



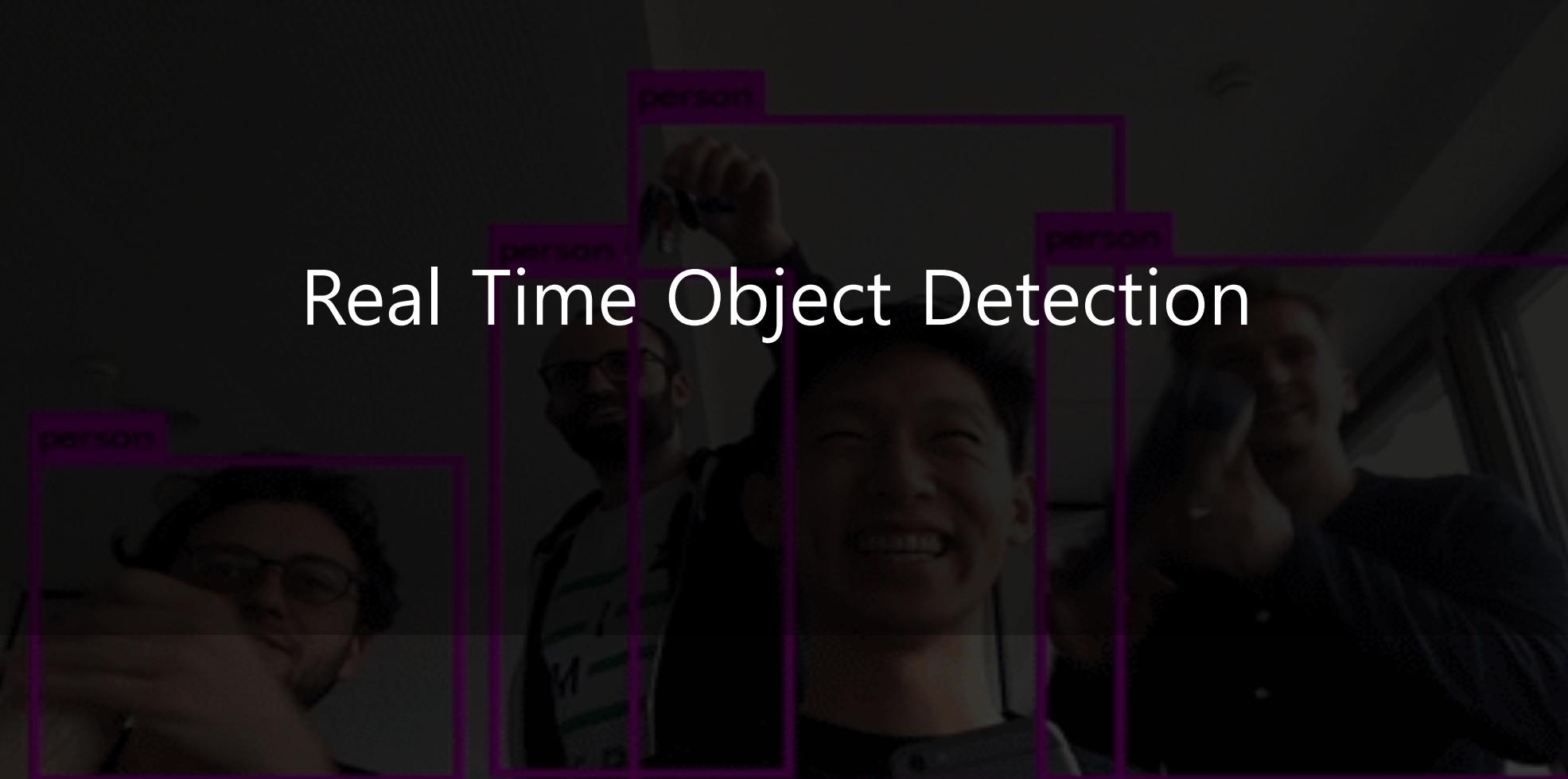
UNIT 01. Intro

◆ Intro - Applications of CNN



UNIT 01. Intro

◆ Intro - Applications of CNN



Real Time Object Detection

UNIT 01. Intro

◆ Intro - Applications of CNN



This flower has long thin yellow petals and a lot of yellow anthers in the center



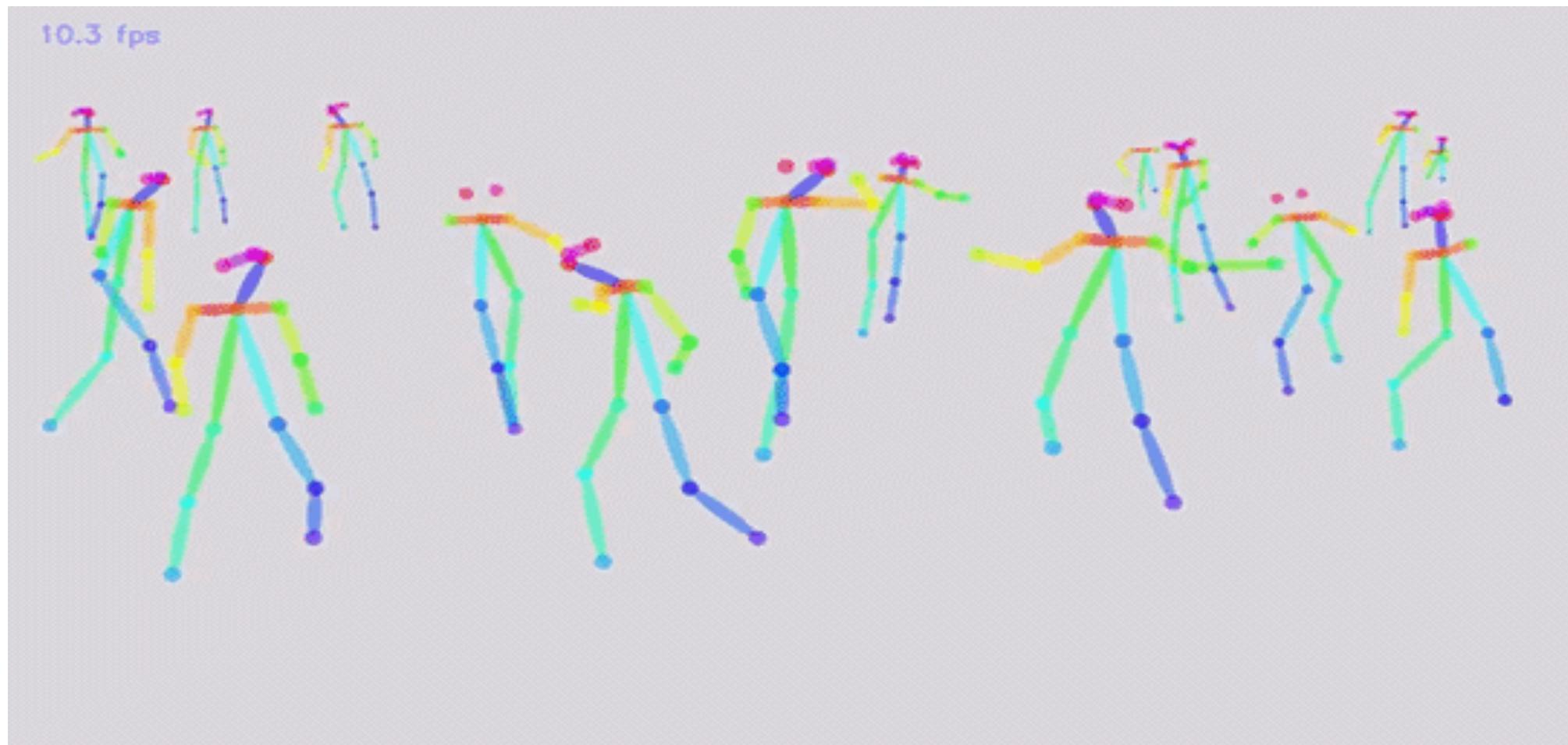
UNIT 01. Intro

◆ Intro - Applications of CNN



UNIT 01. Intro

◆ Intro - Applications of CNN



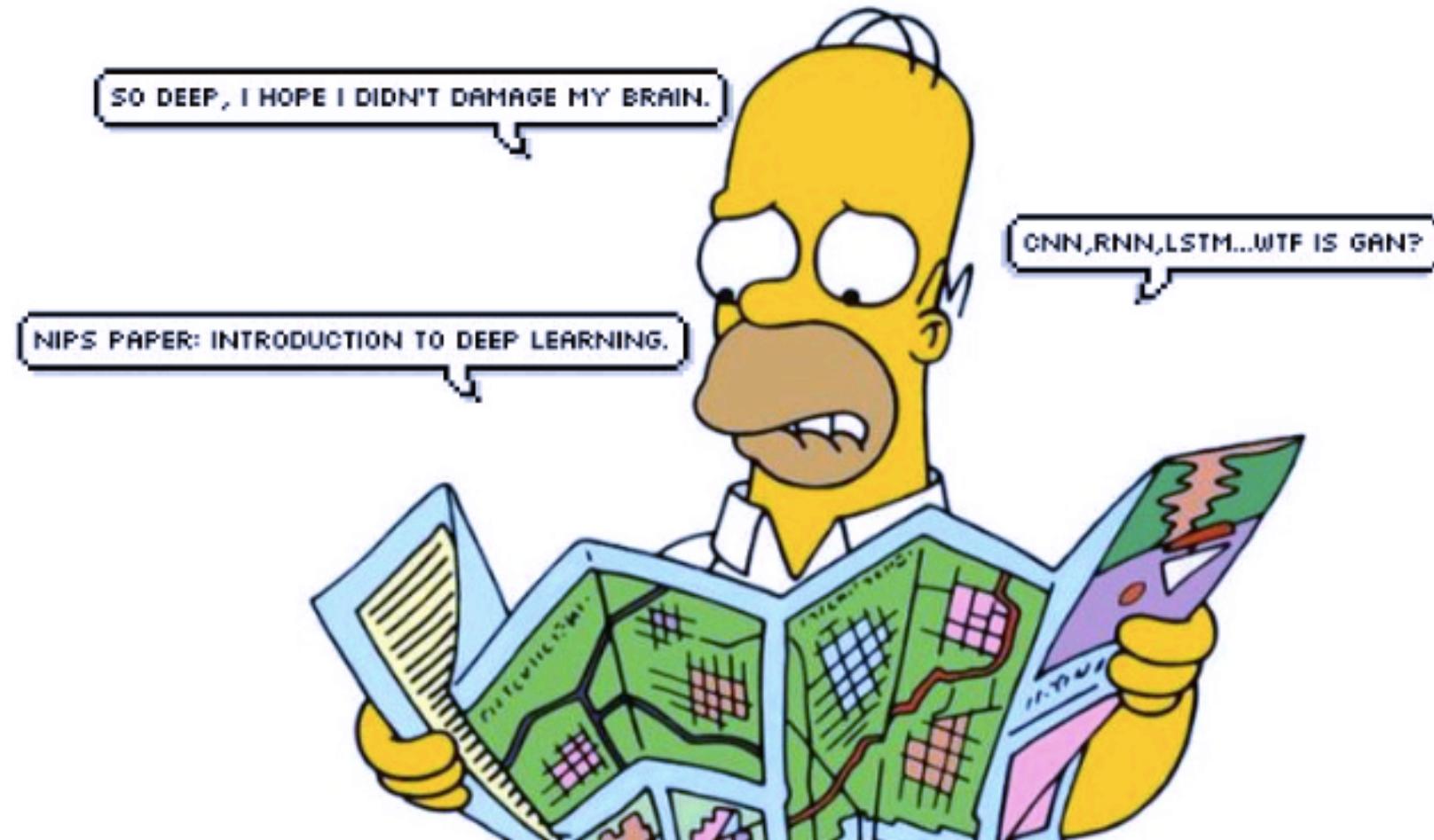
UNIT 01. Intro

◆ Intro - Applications of CNN



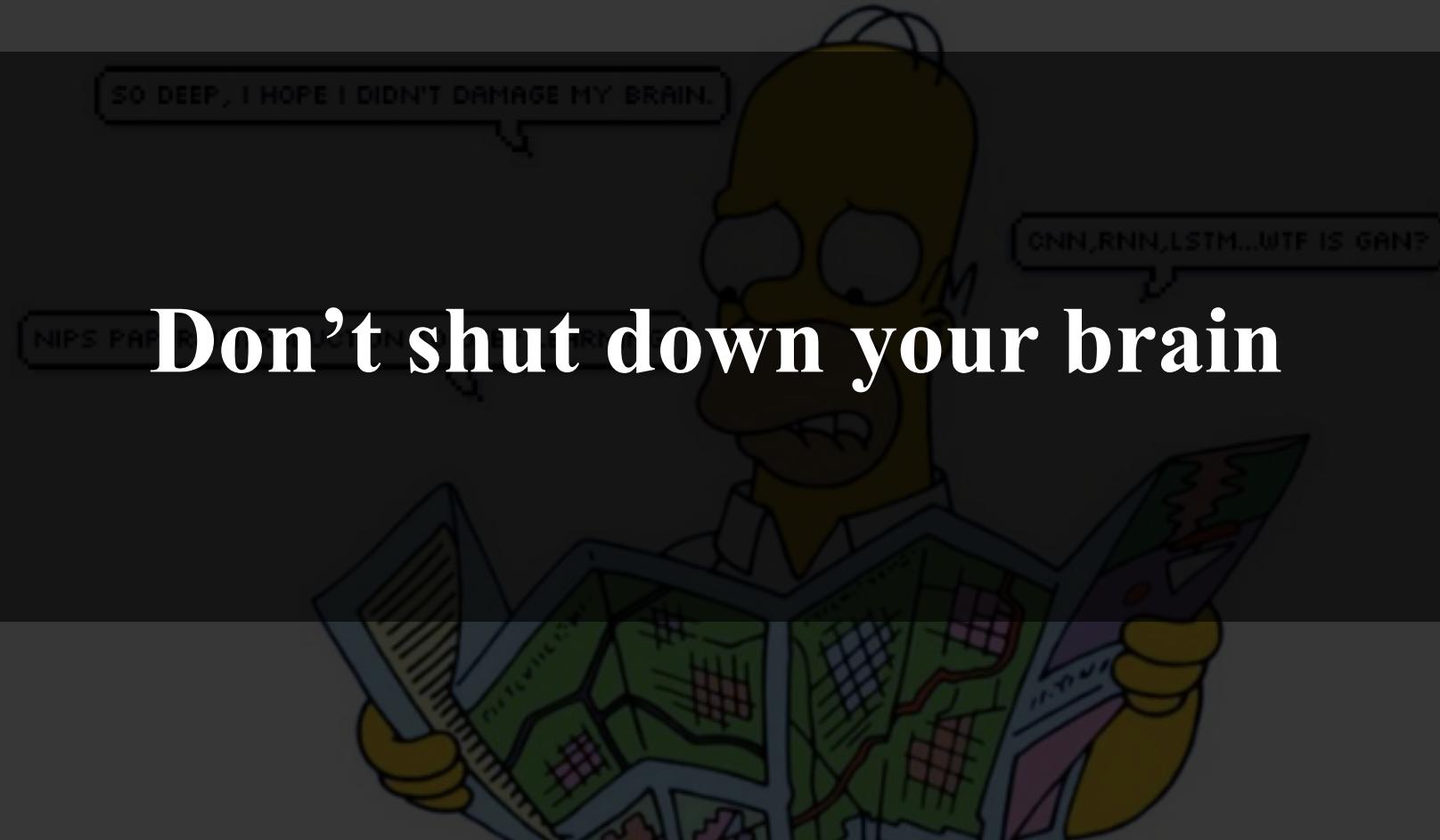
UNIT 01. Intro

◆ Intro - Applications of CNN



UNIT 01. Intro

◆ Intro - Applications of CNN



Contents

Unit 01 | Intro

Unit 02 | Layers in CNN

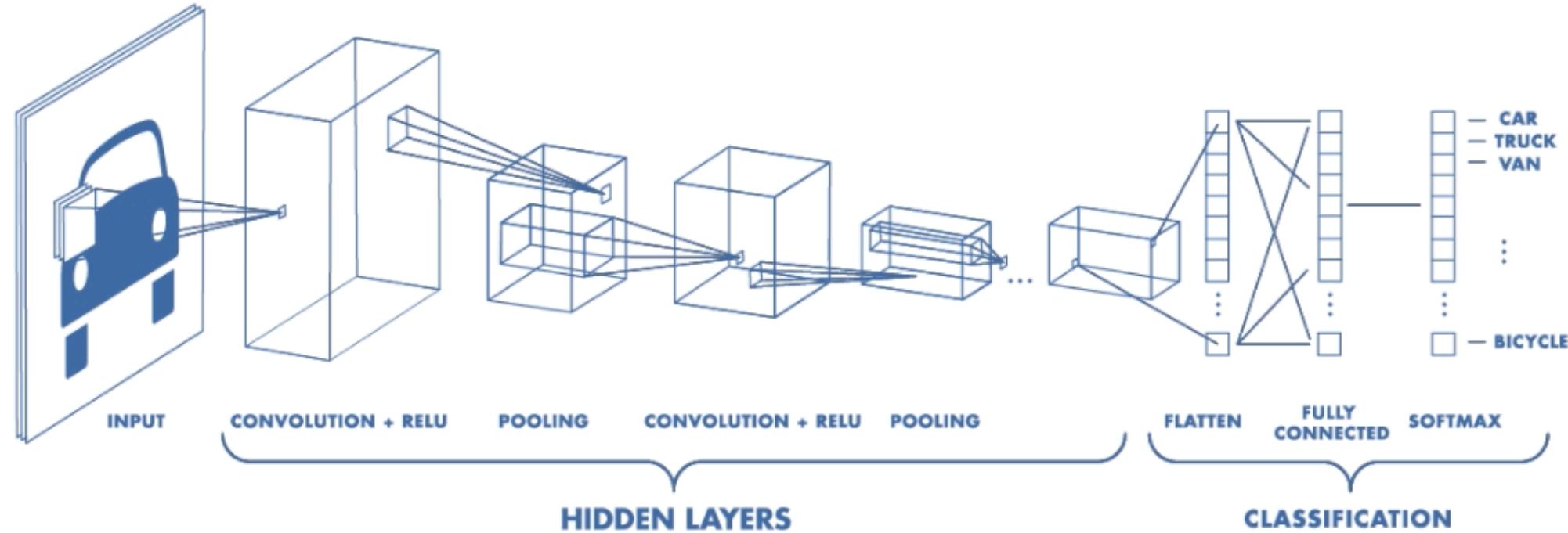
Unit 03 | Convolution

Unit 04 | Sub-Sampling

Unit 05 | Summary

UNIT 02. Layers in CNN

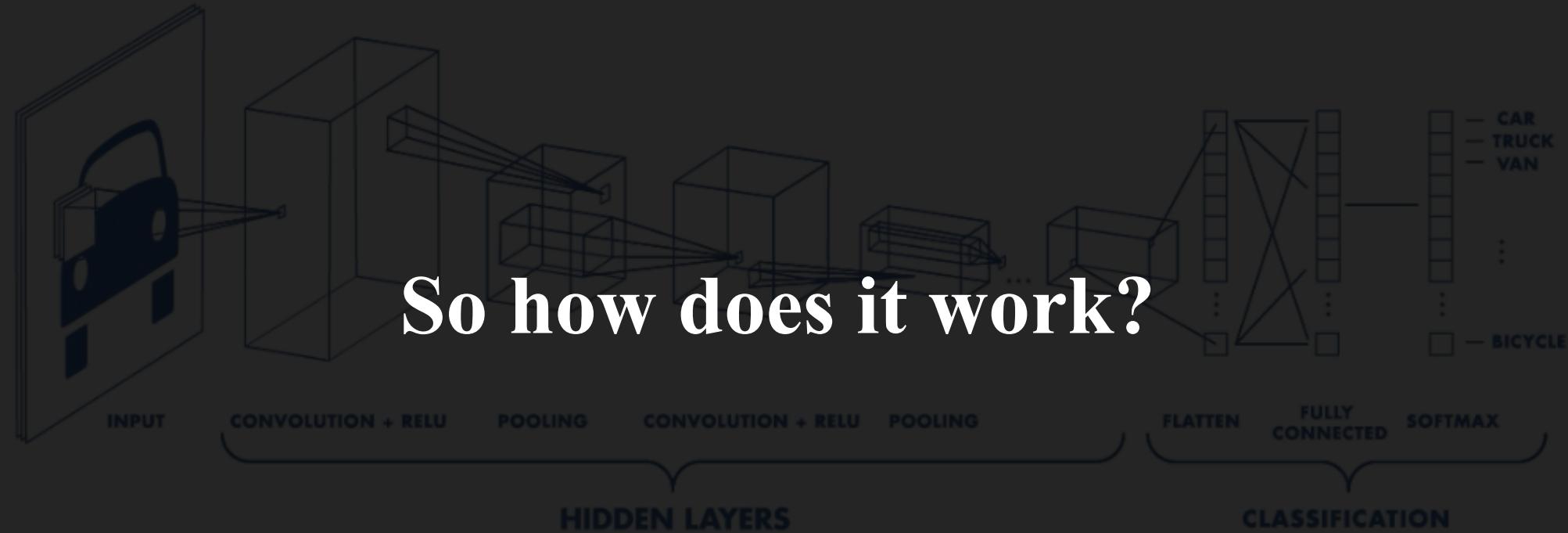
◆ Layers in CNN



- This is Pretty much **everything** about the convolutional neural network
- **Convolution** + **Subsampling** + **Full Connection**

UNIT 02. Layers in CNN

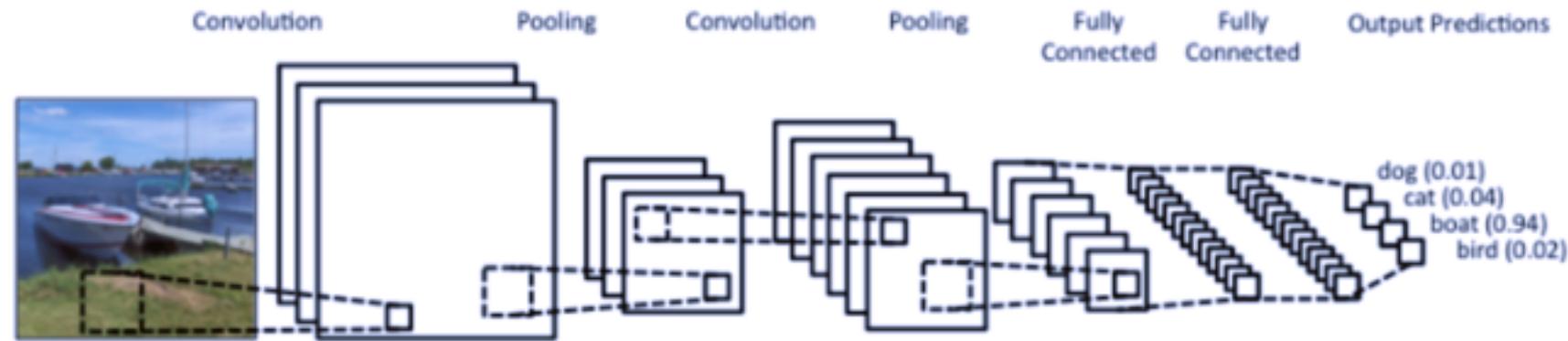
◆ Layers in CNN



- This is Pretty much **everything** about the convolutional neural network
- Convolution + Subsampling + Full Connection

UNIT 02. Layers in CNN

◆ Layers in CNN



- CNN = **Convolutions** followed by **subsampling** and **fully connected layer**
- Feature Extraction <- Convolution + subsampling layers
- Classifier <- Fully connected layer

이미지의 특징 추출
위상 변환
결과 예측

UNIT 02. Layers in CNN

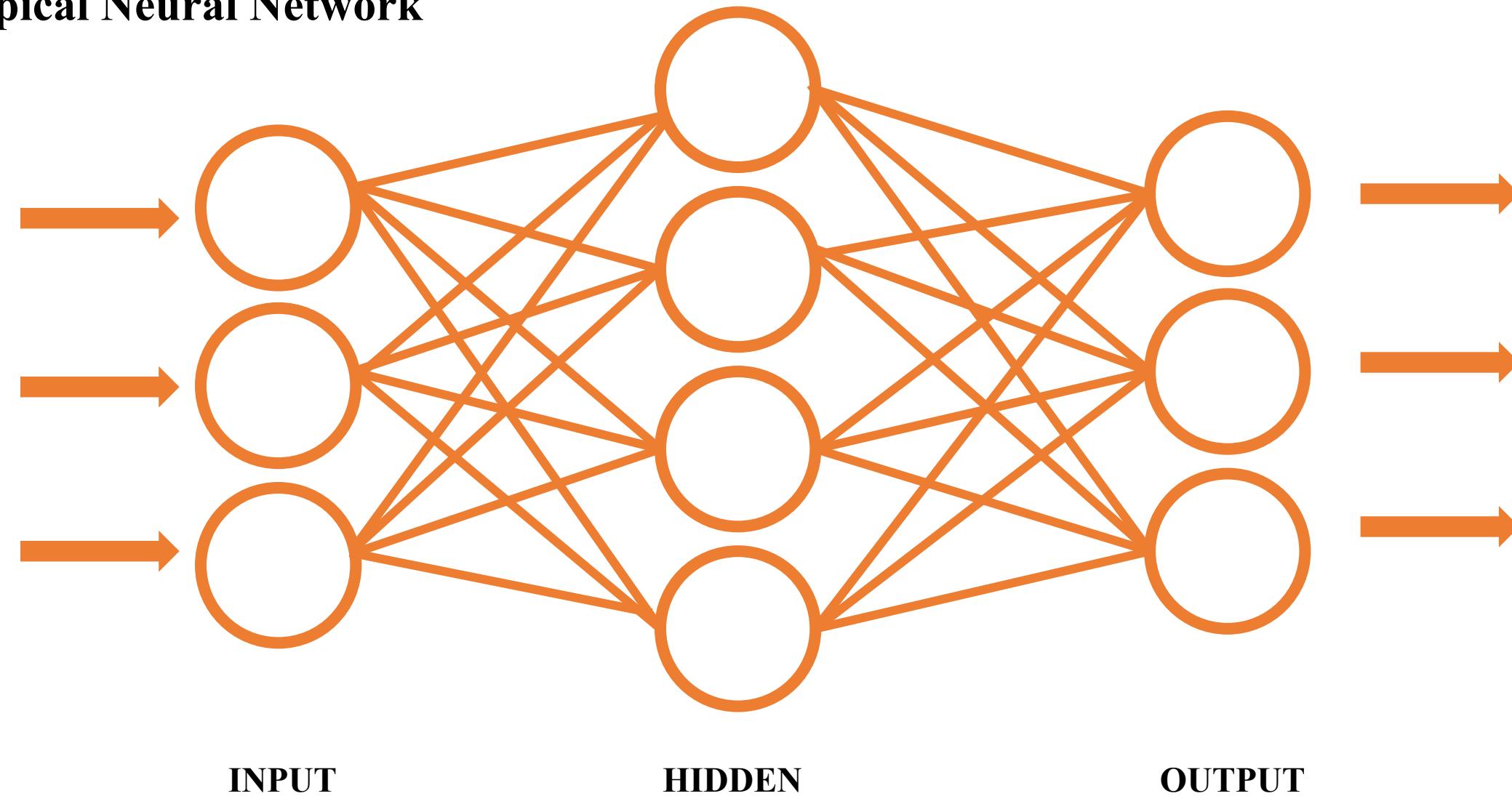
◆ Layers in CNN

Why is so powerful?

- Local connectivity(receptive field)
- Shared Weights and Biases
- Compositionality (구조성적 연속성)

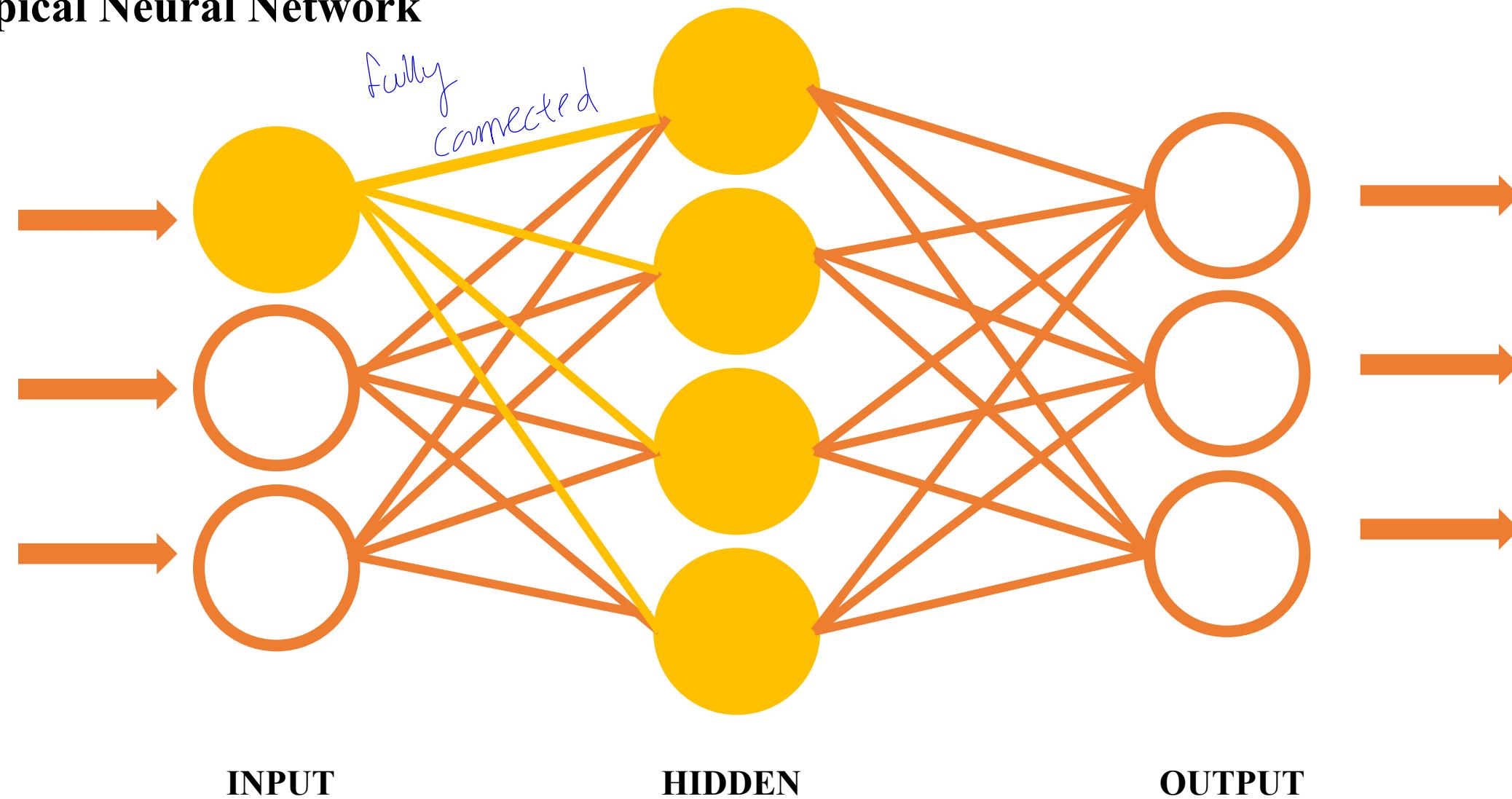
UNIT 02. Layers in CNN

◆ Typical Neural Network



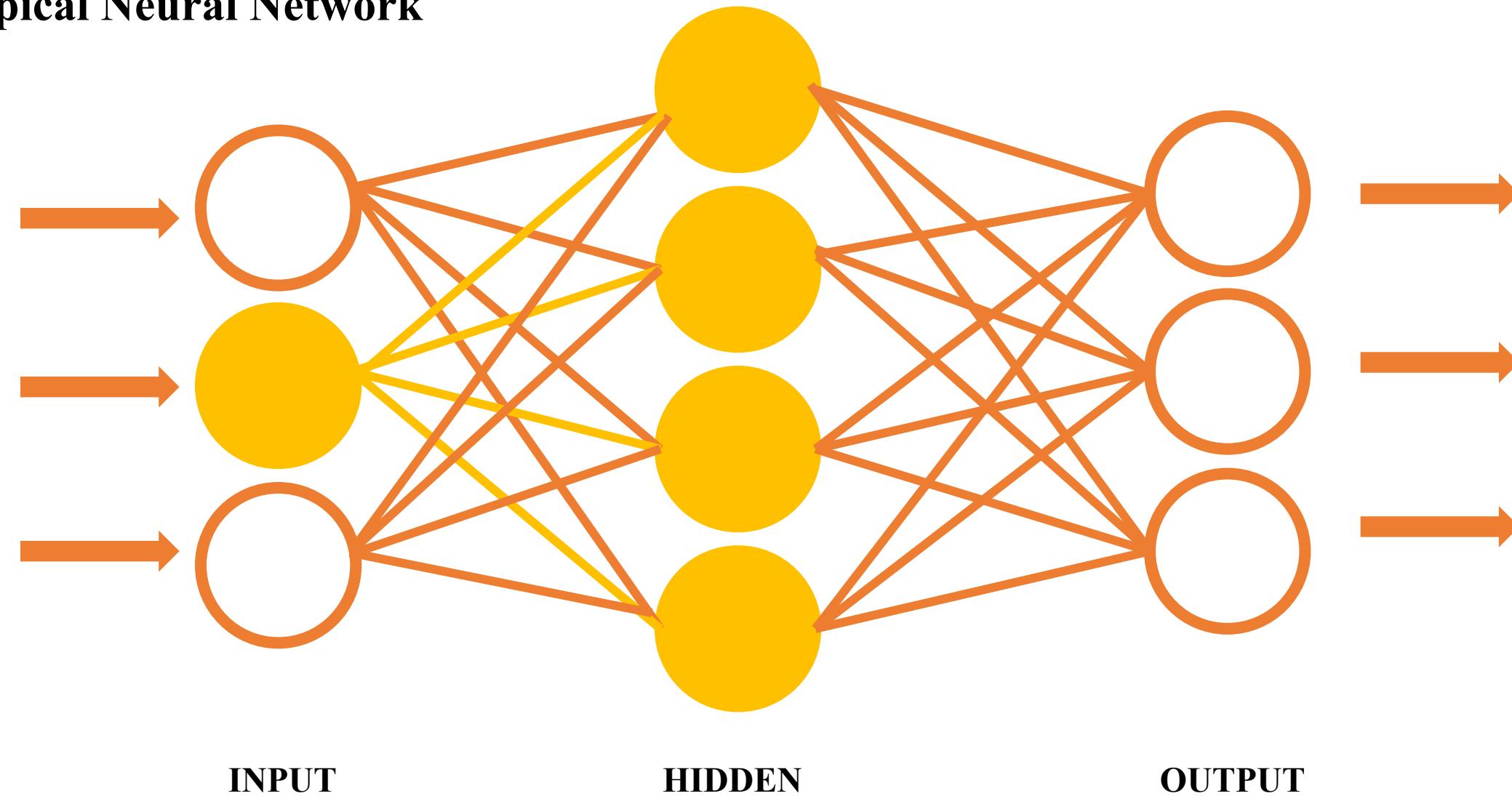
UNIT 02. Layers in CNN

◆ Typical Neural Network



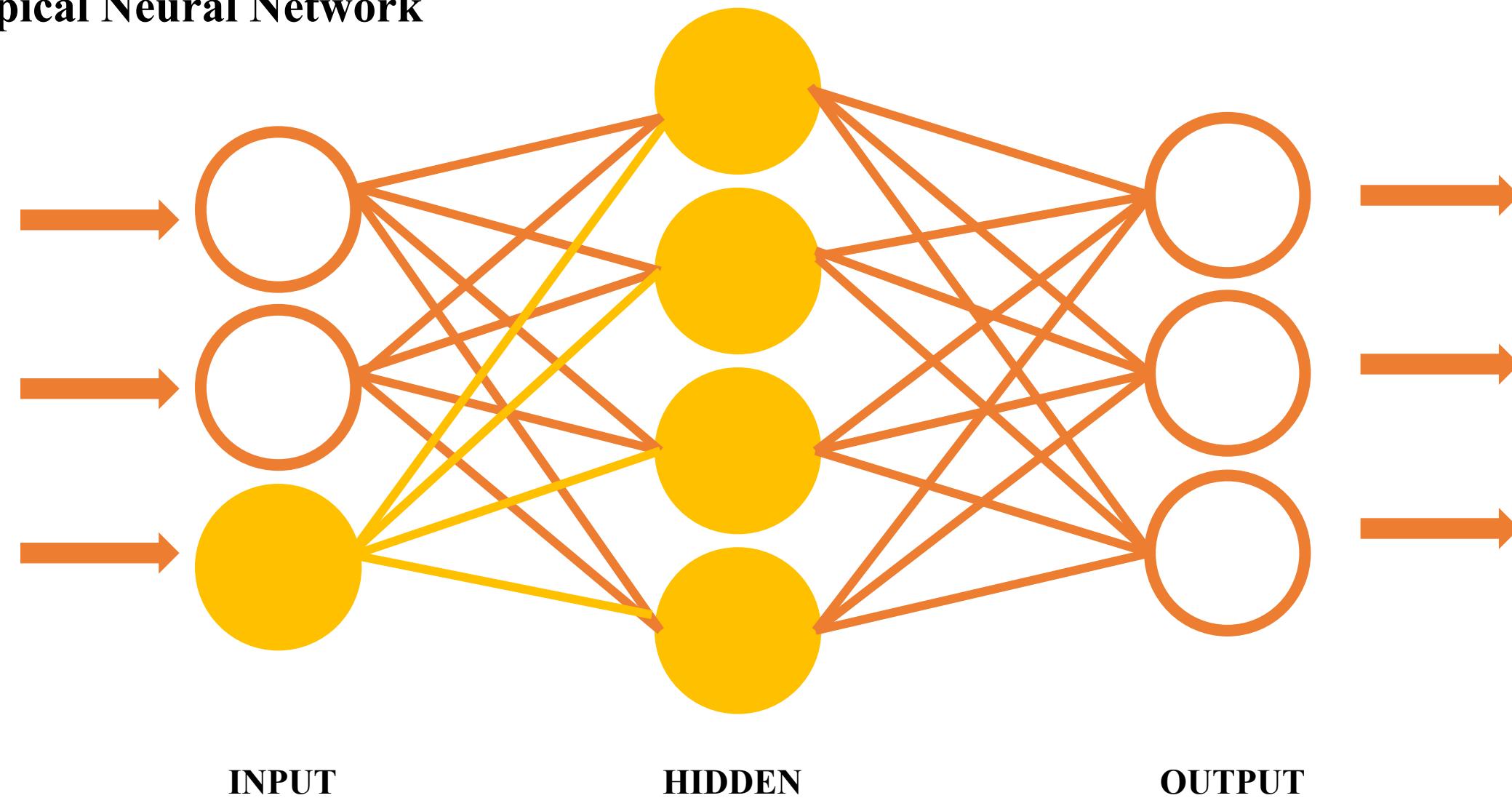
UNIT 02. Layers in CNN

◆ Typical Neural Network



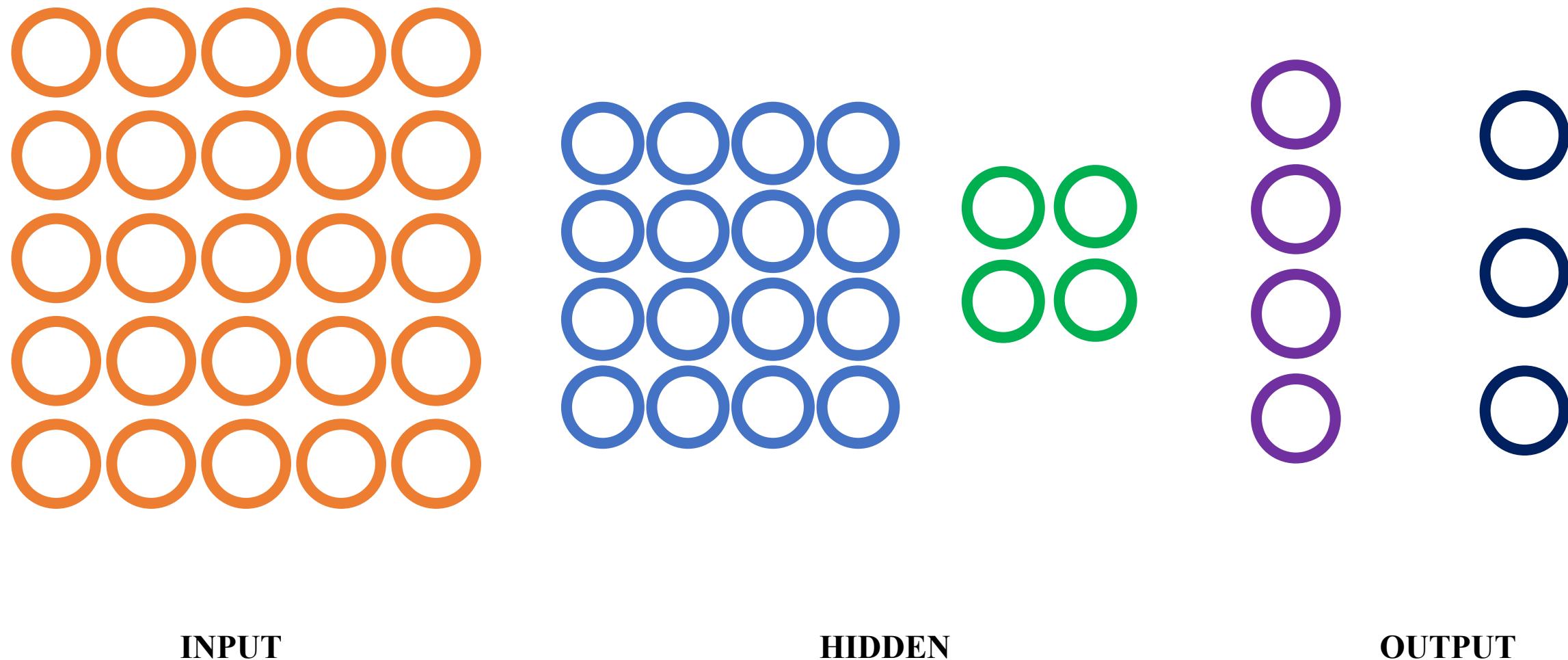
UNIT 02. Layers in CNN

◆ Typical Neural Network



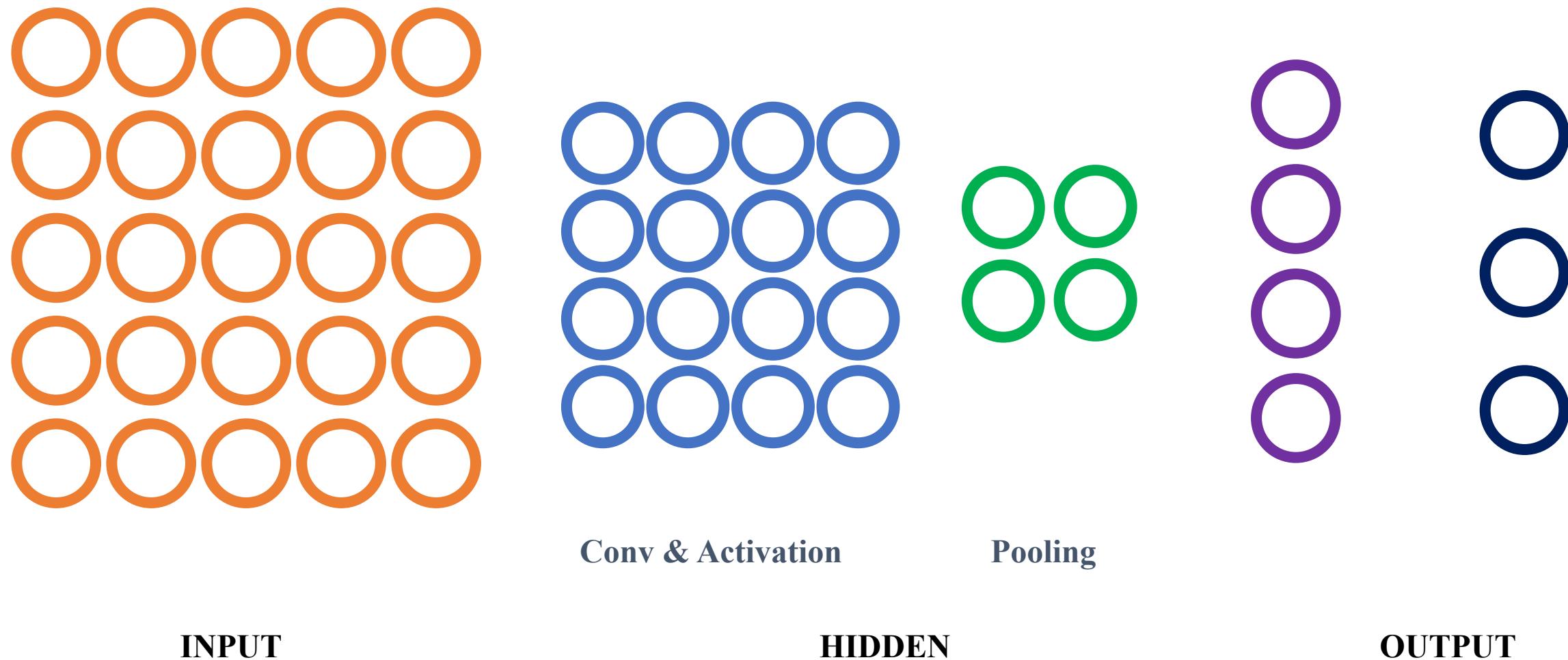
UNIT 02. Layers in CNN

◆ Convolutional Neural Network



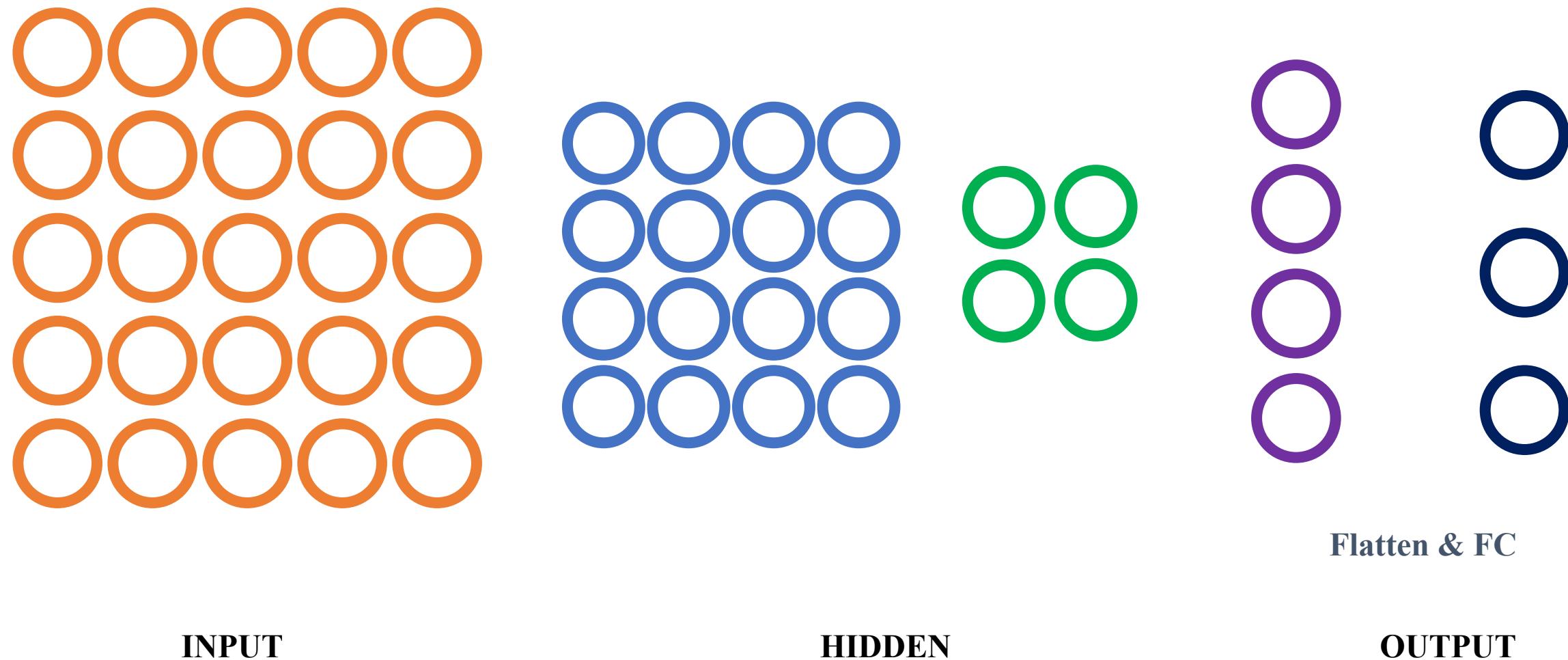
UNIT 02. Layers in CNN

◆ Convolutional Neural Network



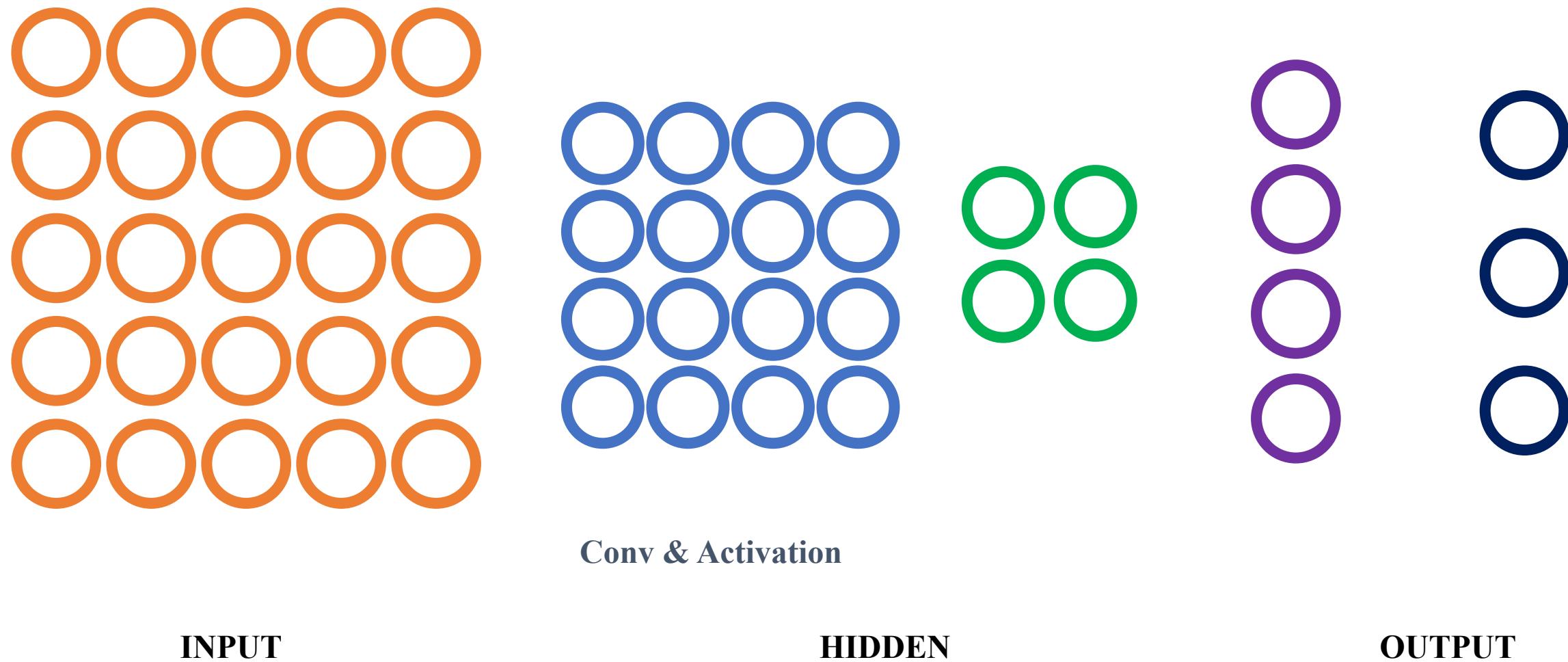
UNIT 02. Layers in CNN

◆ Convolutional Neural Network



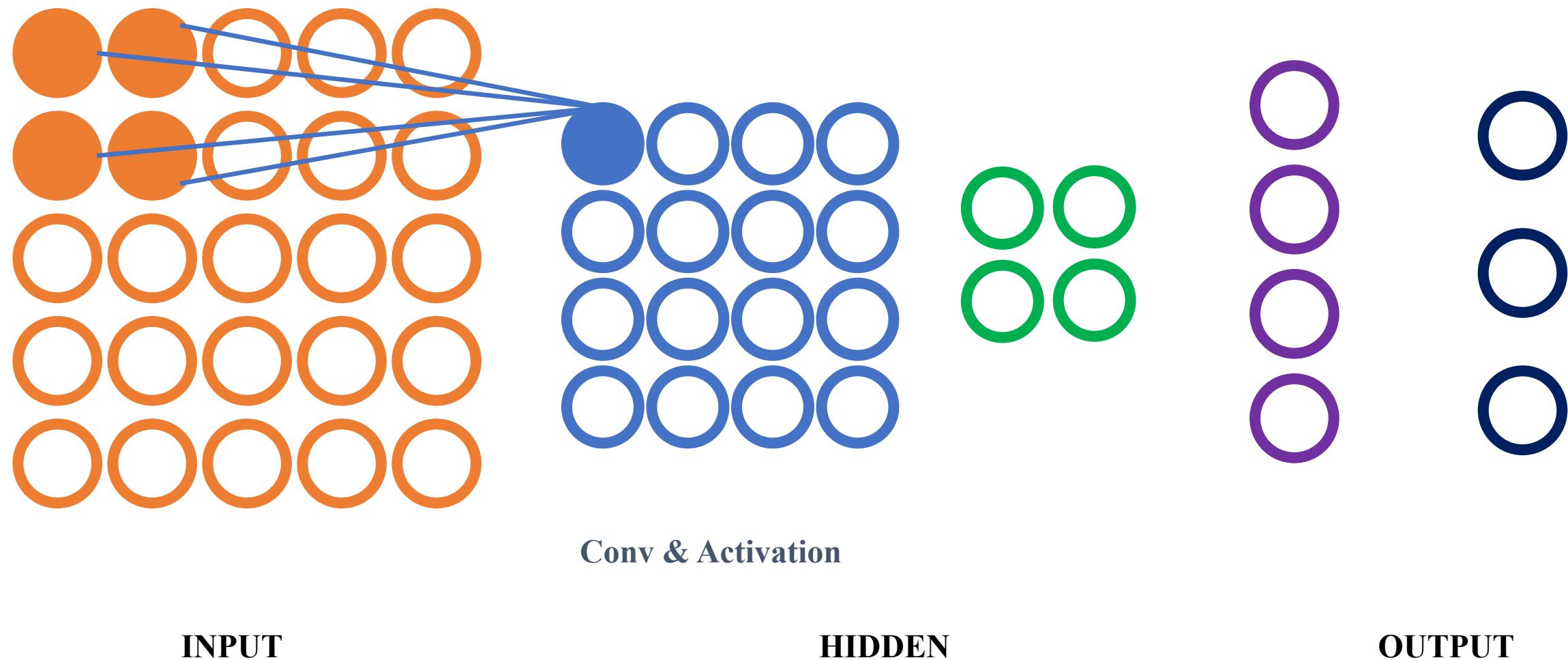
UNIT 02. Layers in CNN

◆ Convolutional Neural Network



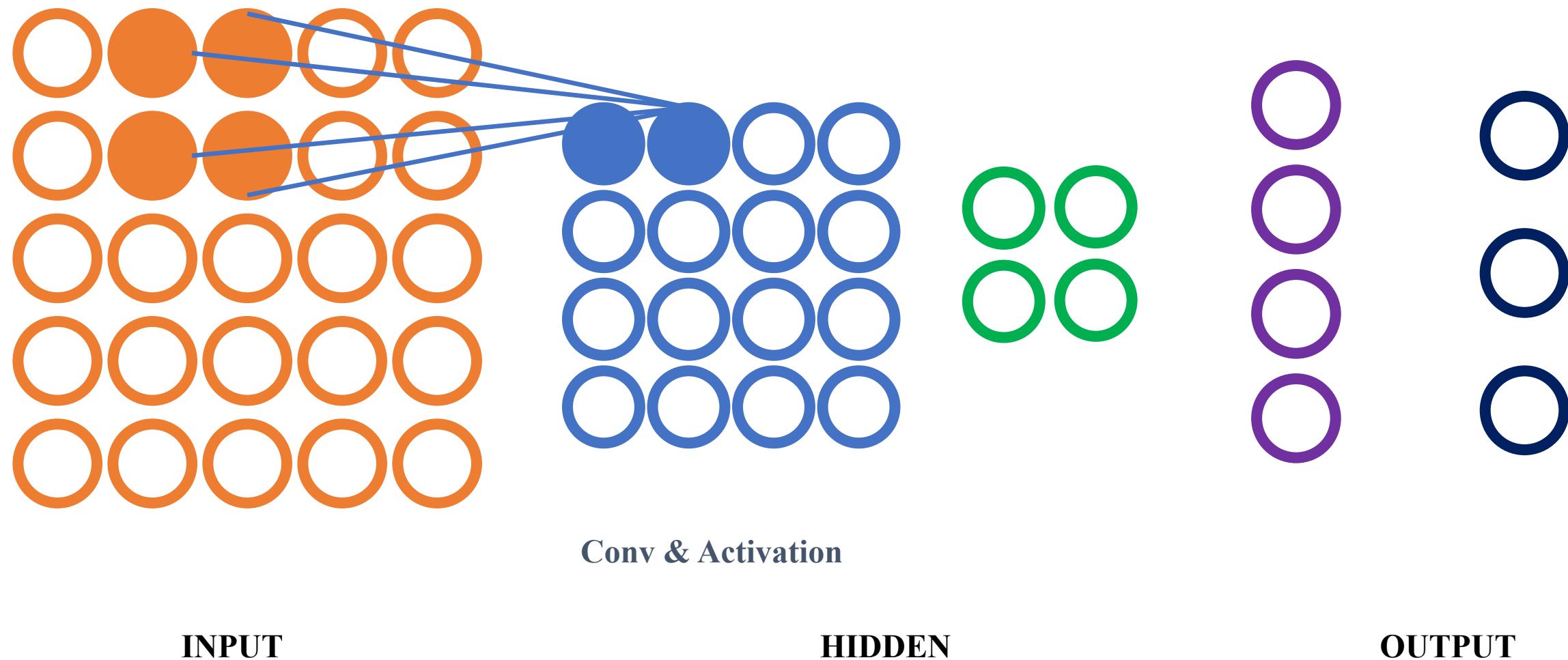
UNIT 02. Layers in CNN

◆ Convolutional Neural Network



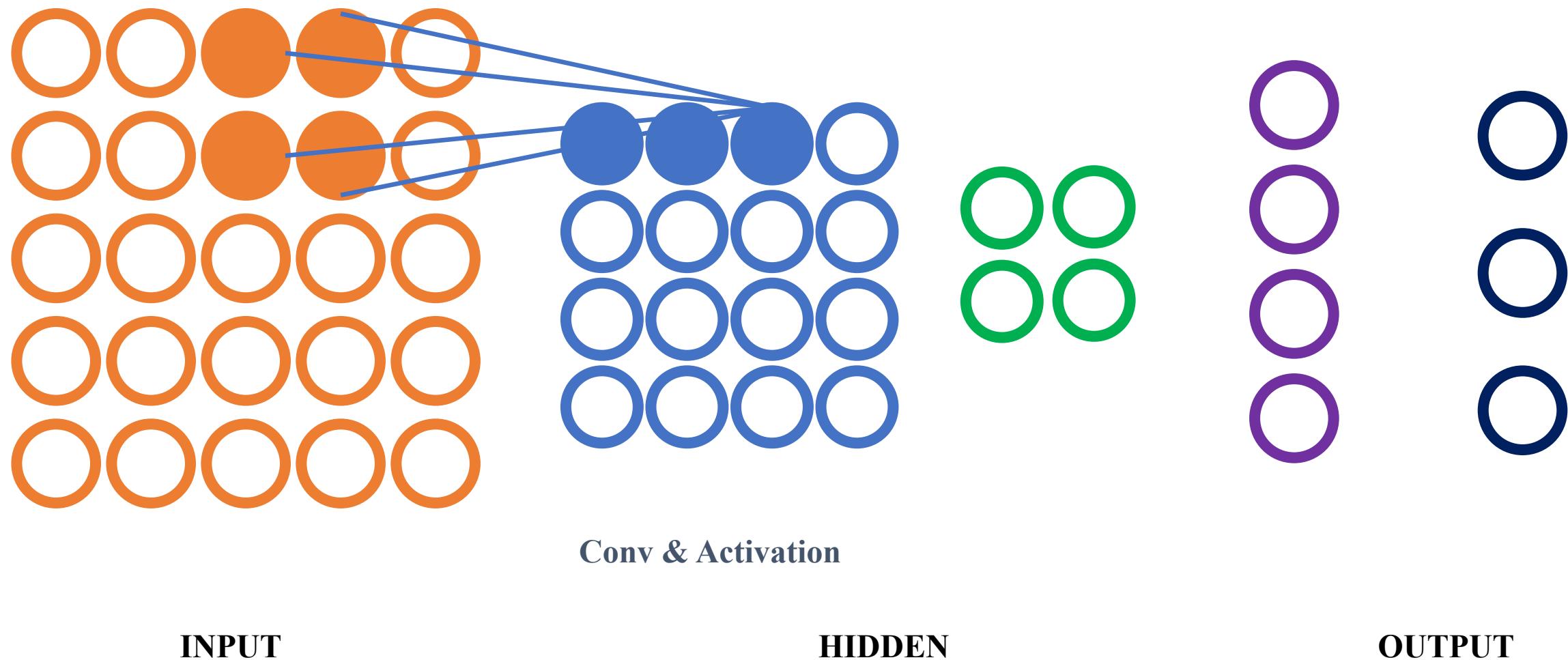
UNIT 02. Layers in CNN

◆ Convolutional Neural Network



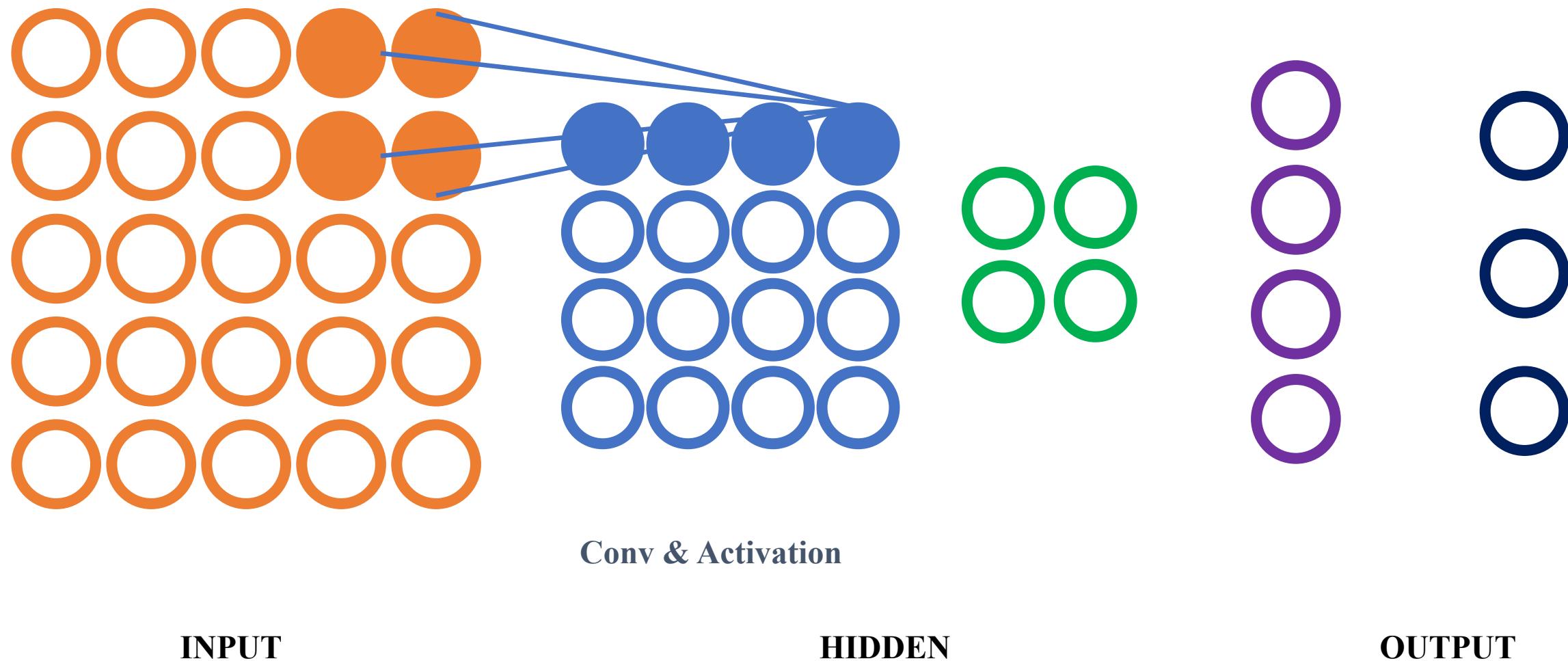
UNIT 02. Layers in CNN

◆ Convolutional Neural Network



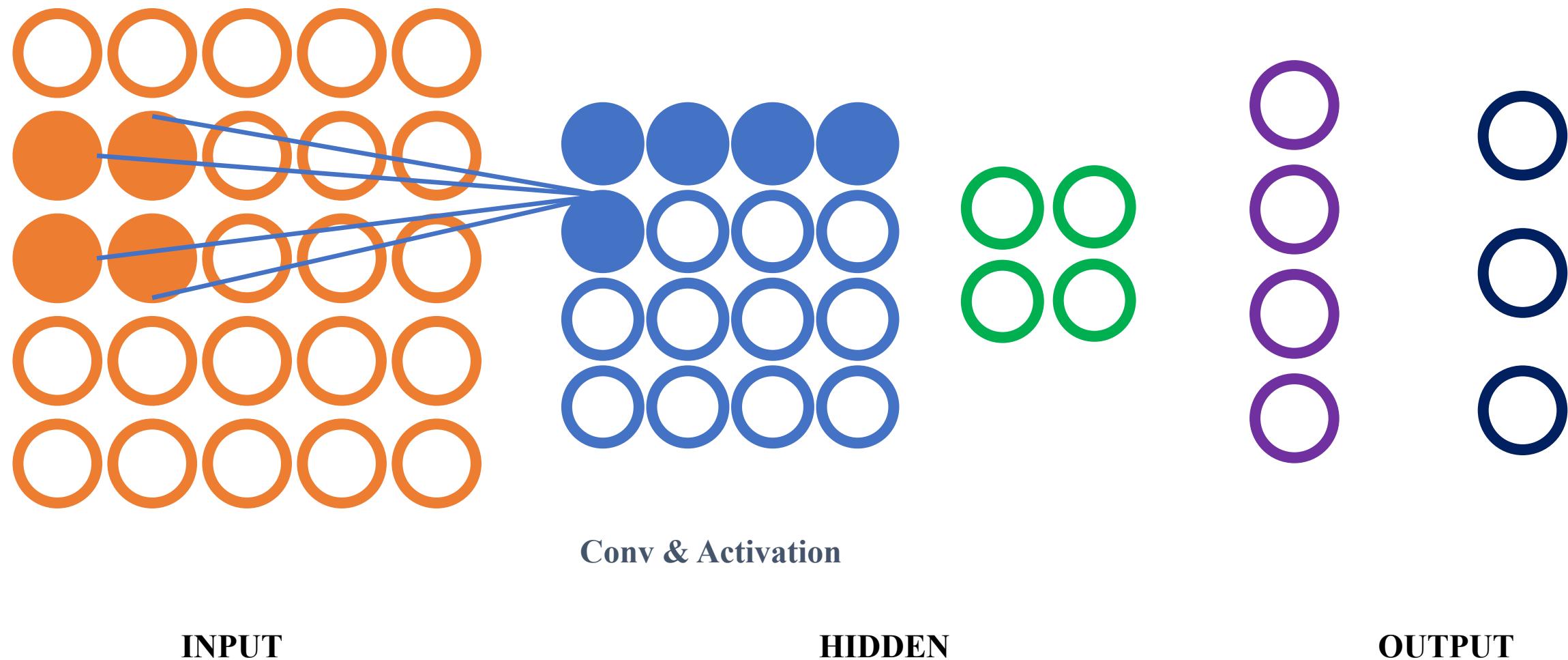
UNIT 02. Layers in CNN

◆ Convolutional Neural Network



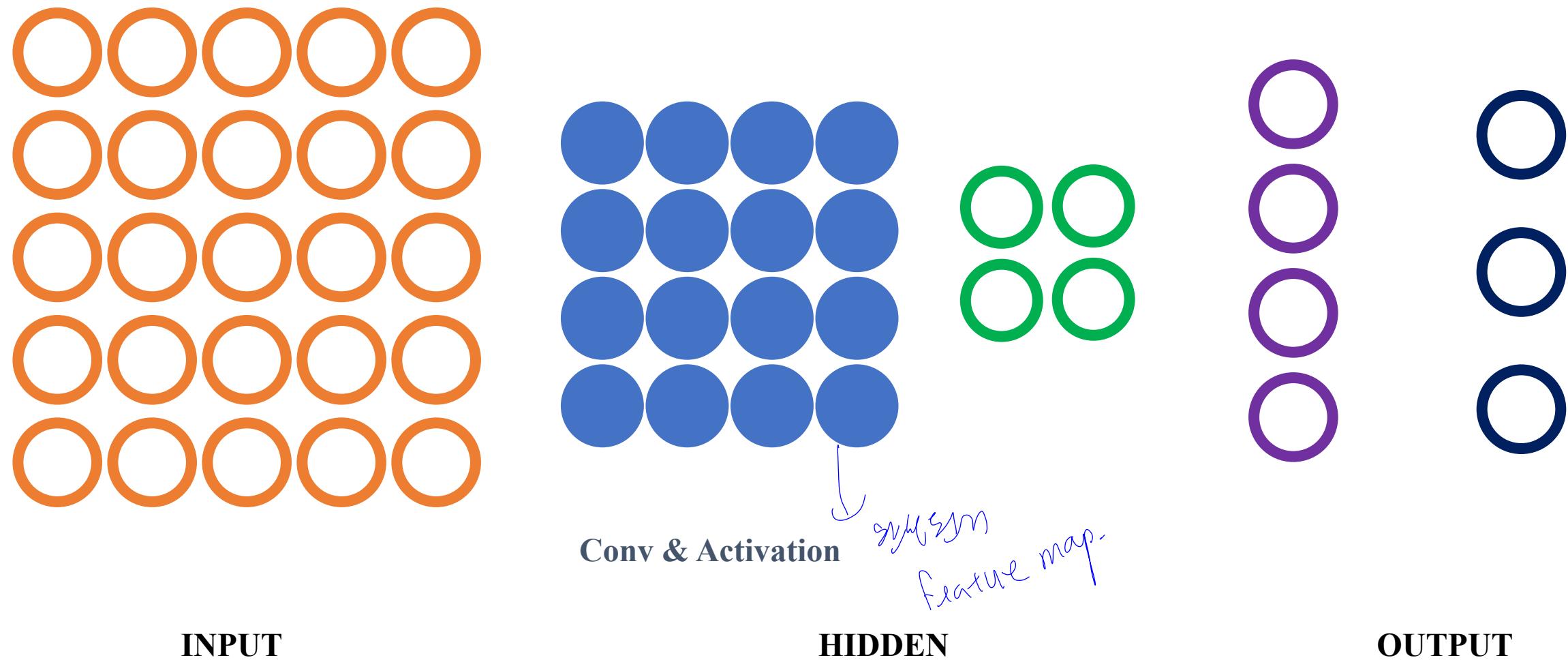
UNIT 02. Layers in CNN

◆ Convolutional Neural Network



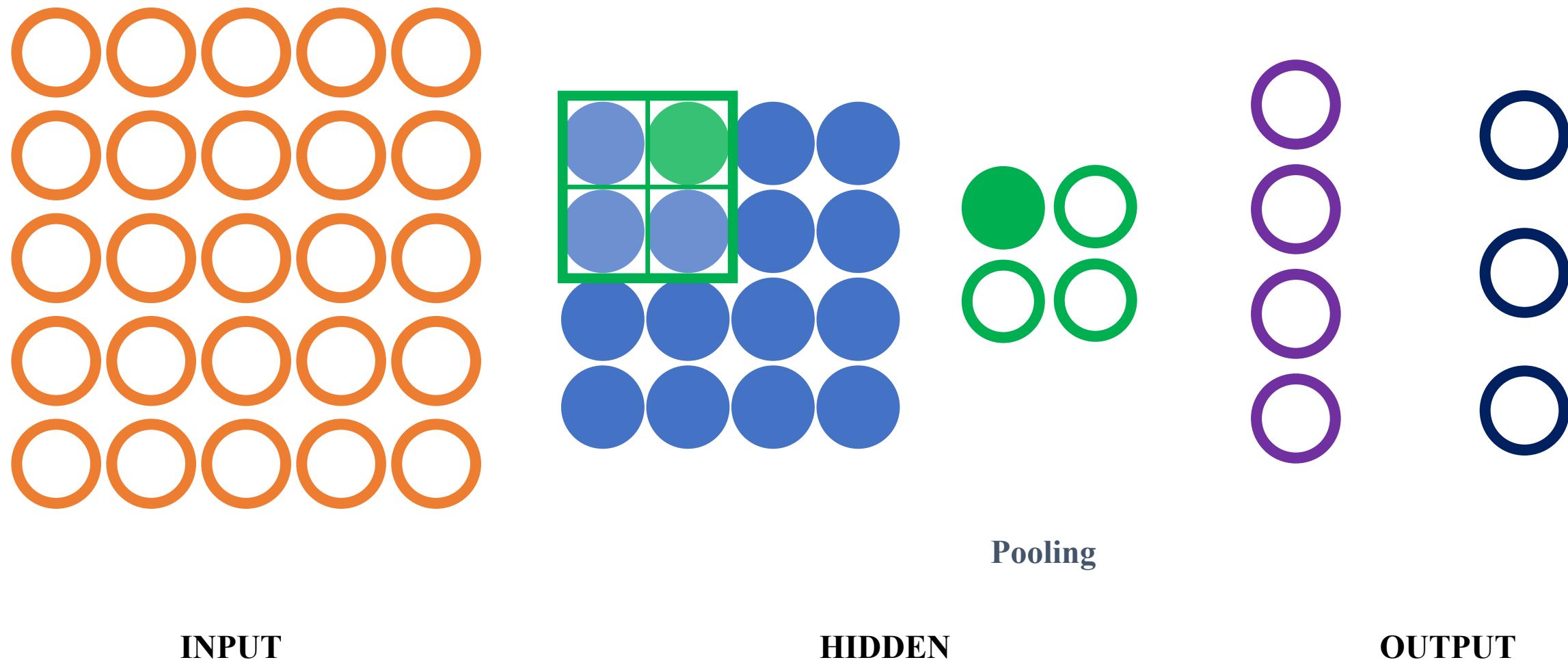
UNIT 02. Layers in CNN

◆ Convolutional Neural Network



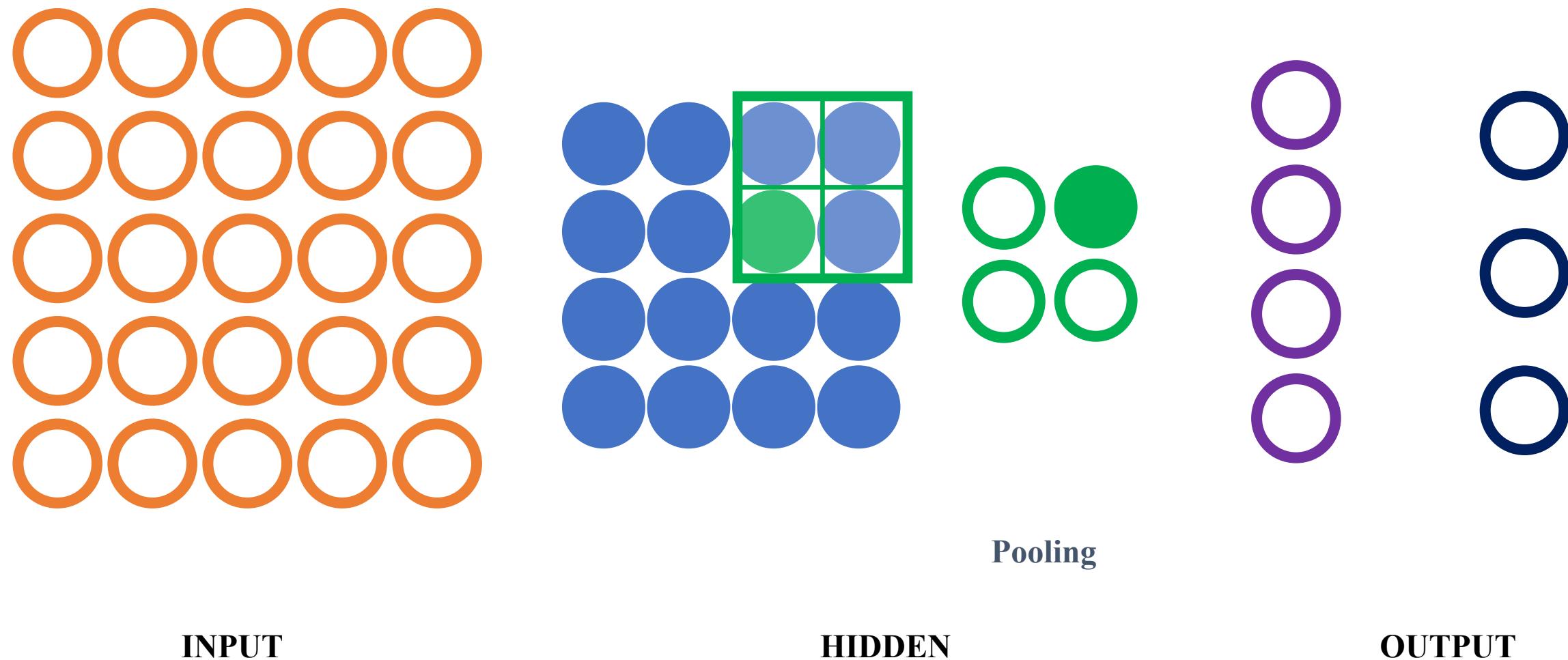
UNIT 02. Layers in CNN

◆ Convolutional Neural Network



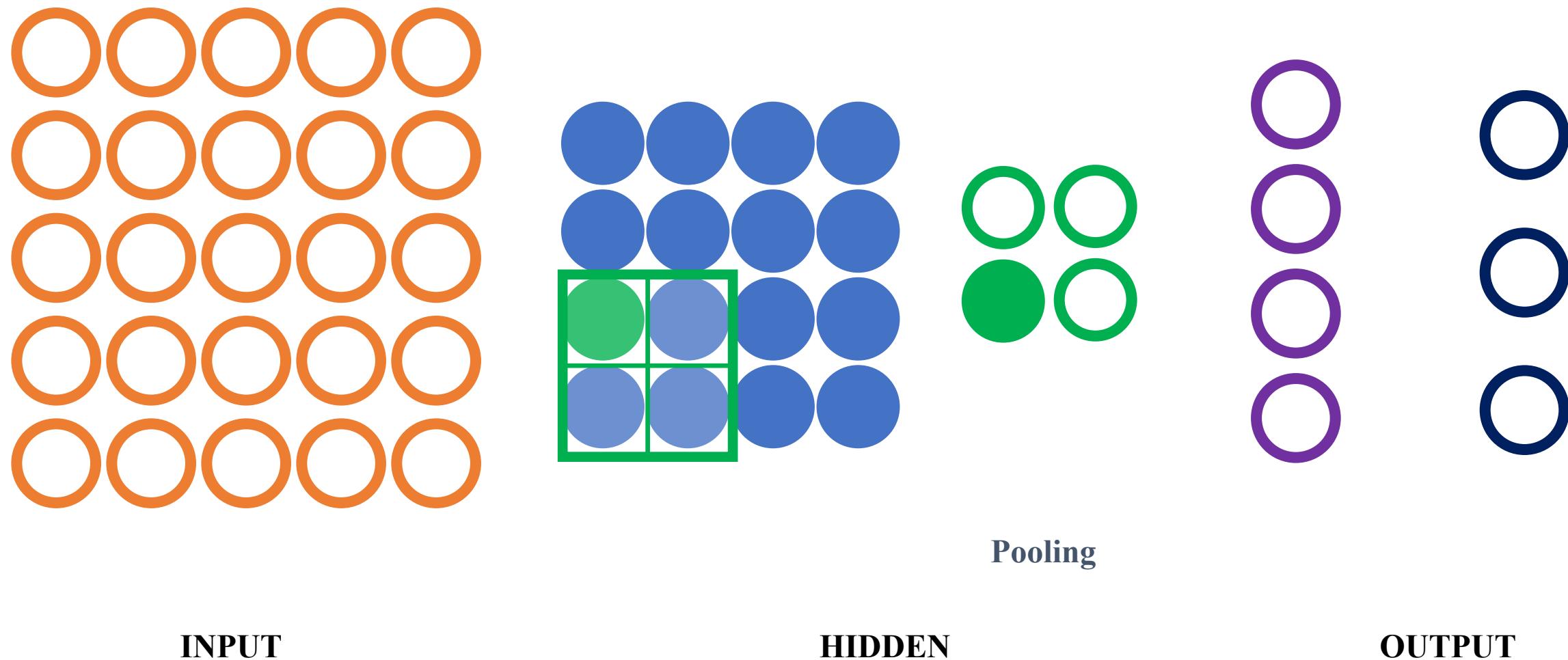
UNIT 02. Layers in CNN

◆ Convolutional Neural Network



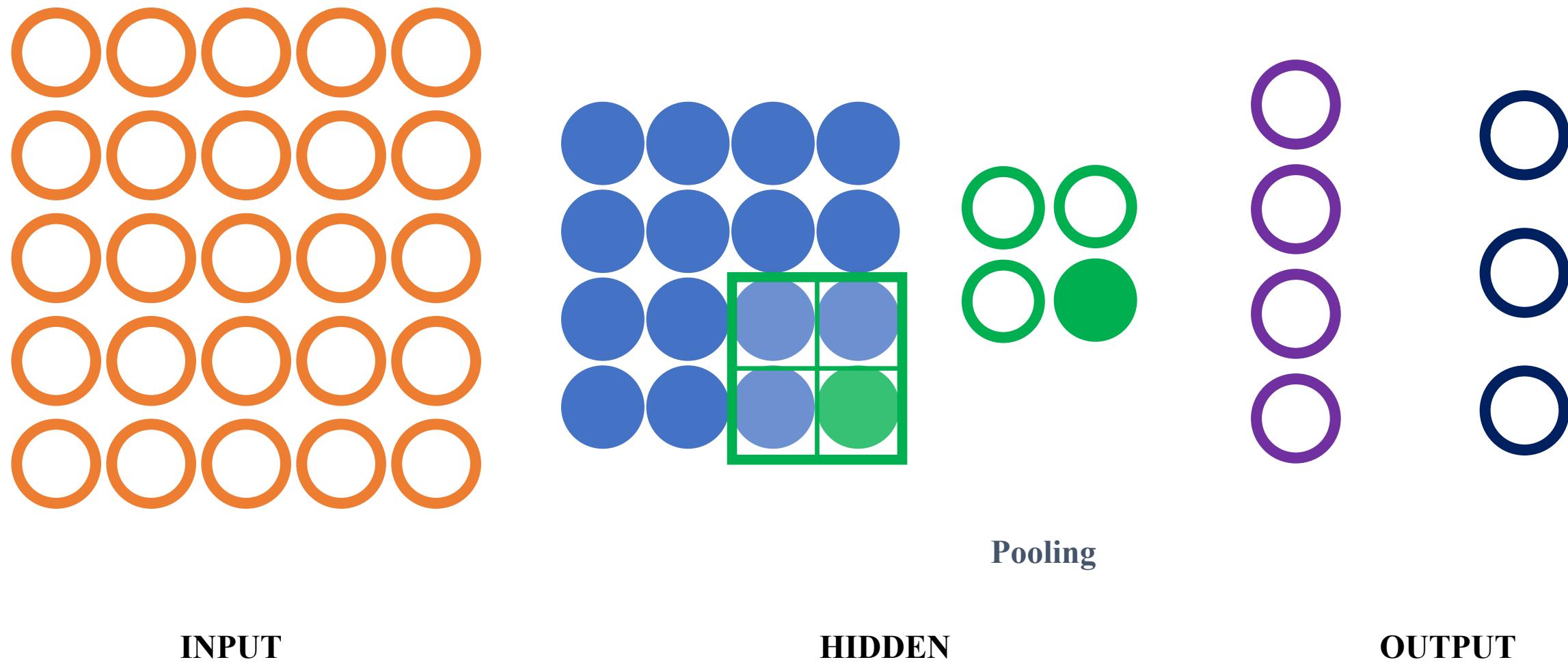
UNIT 02. Layers in CNN

◆ Convolutional Neural Network



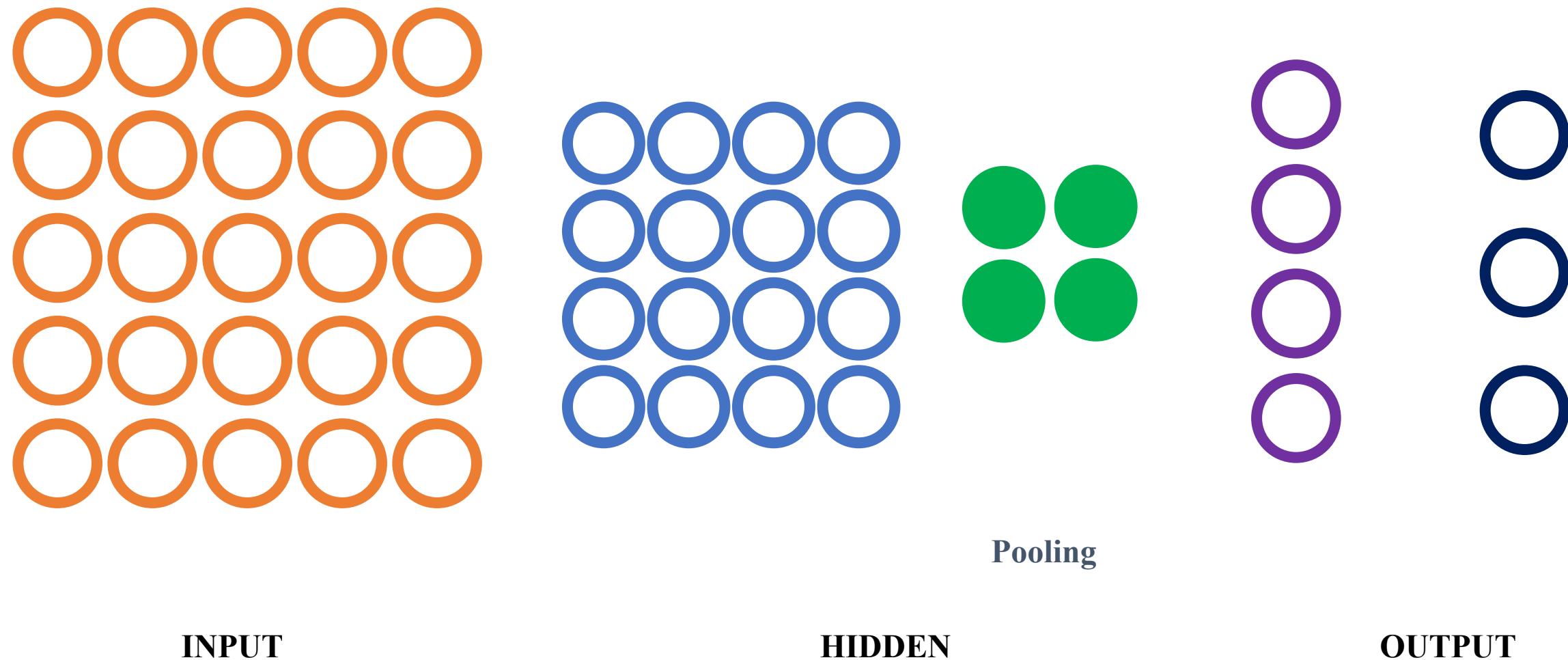
UNIT 02. Layers in CNN

◆ Convolutional Neural Network



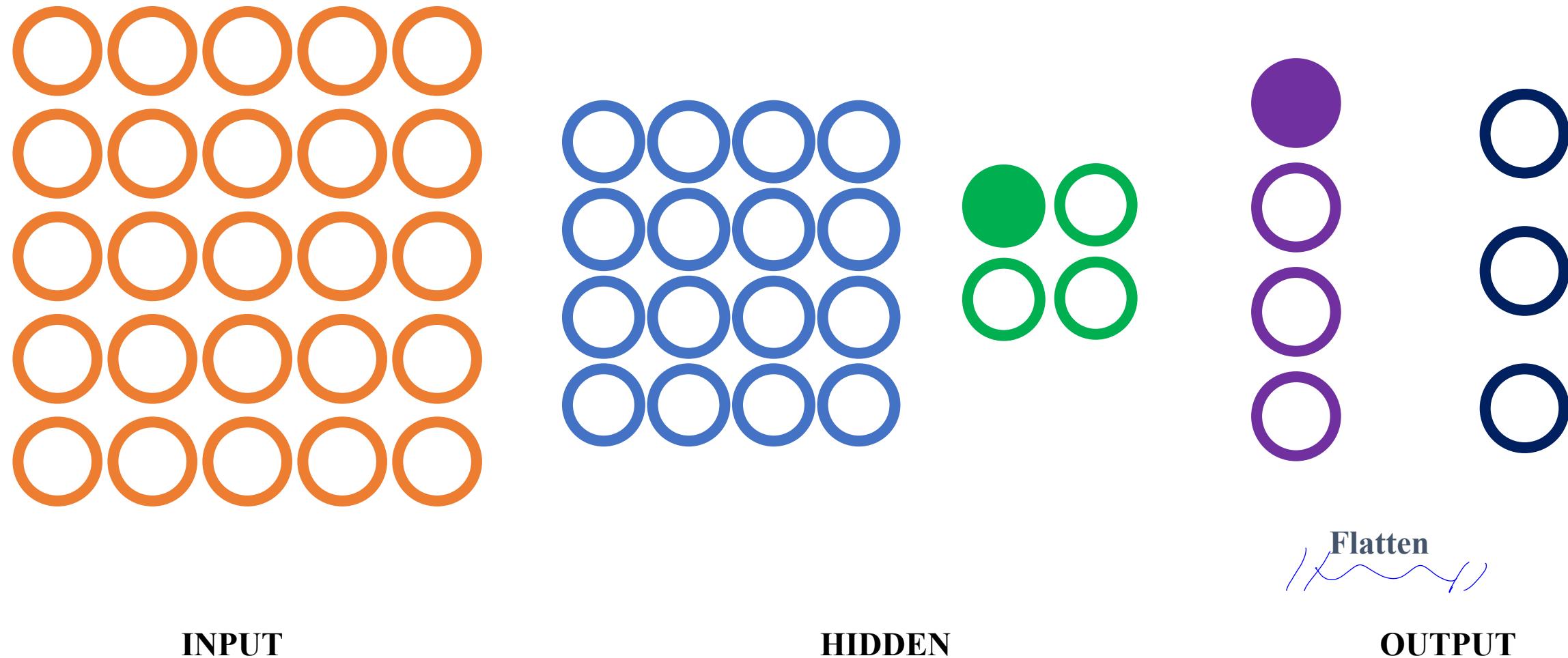
UNIT 02. Layers in CNN

◆ Convolutional Neural Network



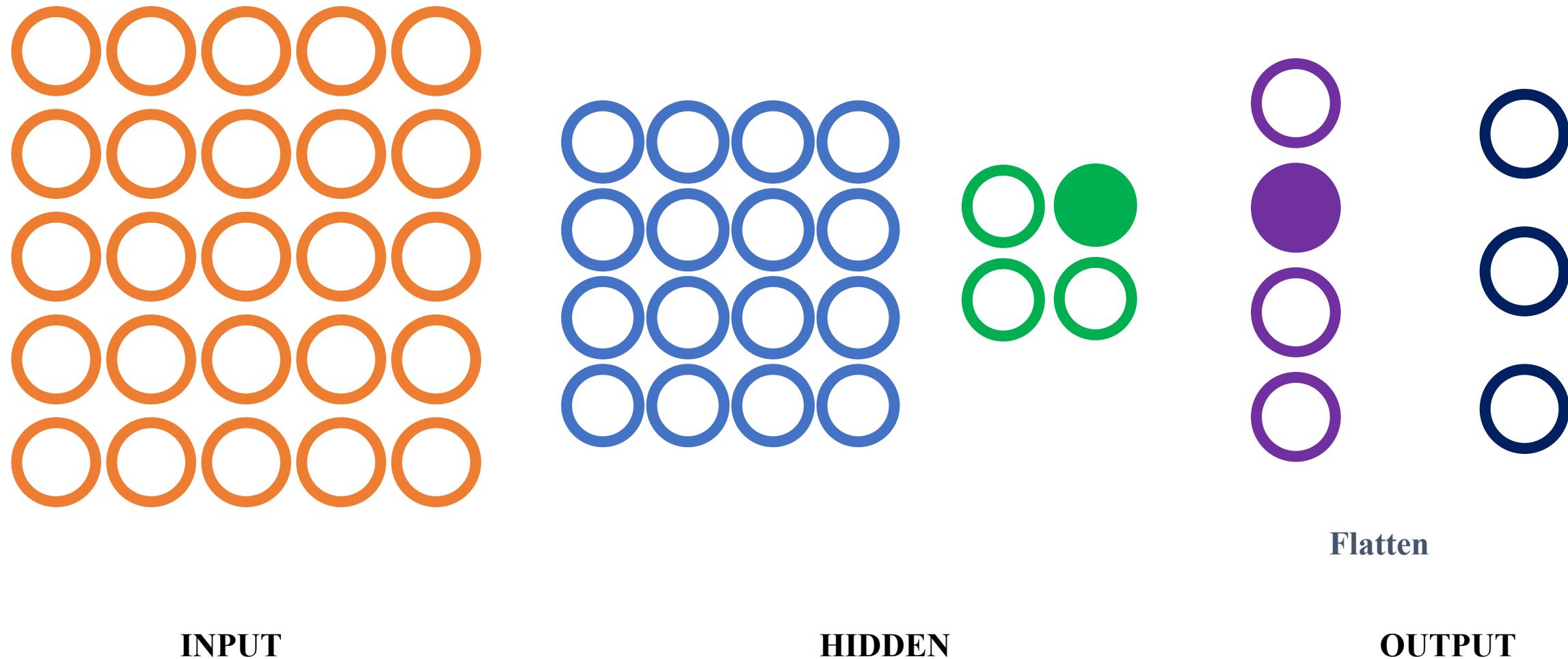
UNIT 02. Layers in CNN

◆ Convolutional Neural Network



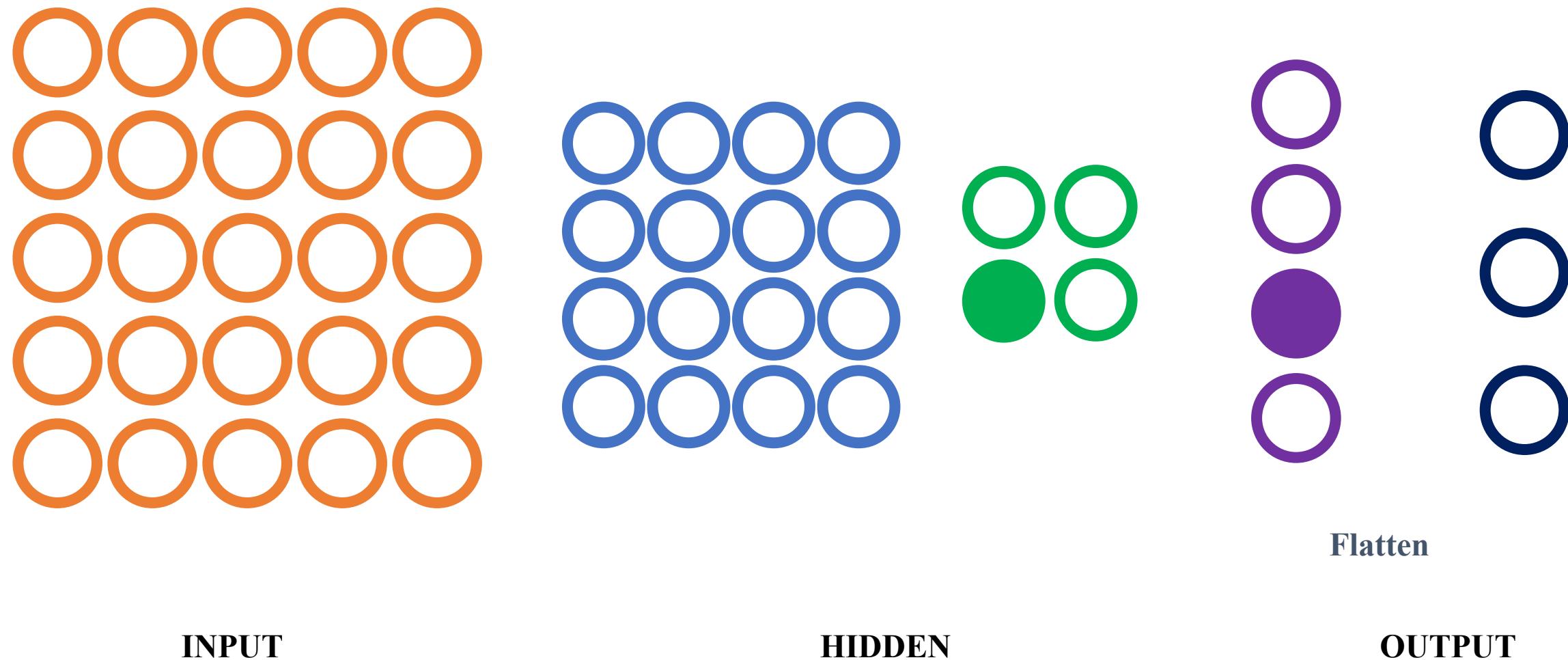
UNIT 02. Layers in CNN

◆ Convolutional Neural Network



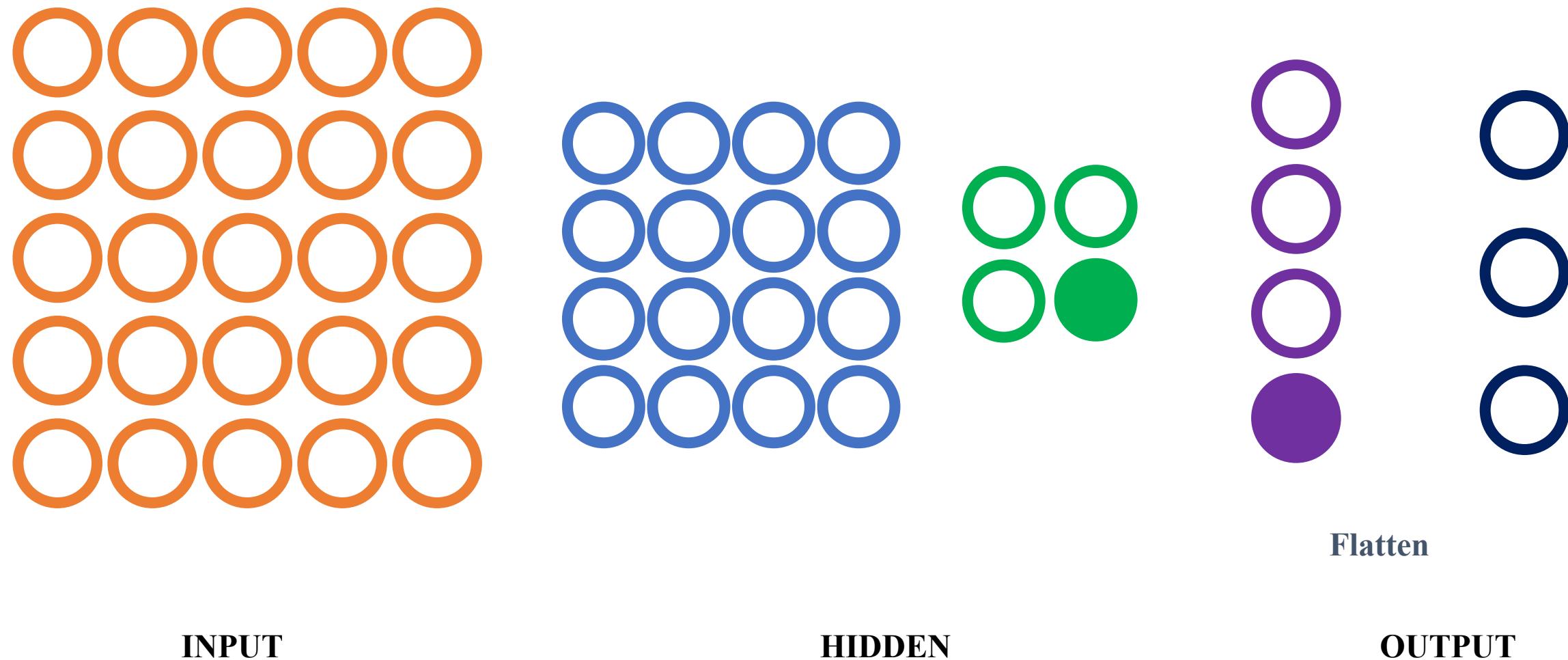
UNIT 02. Layers in CNN

◆ Convolutional Neural Network



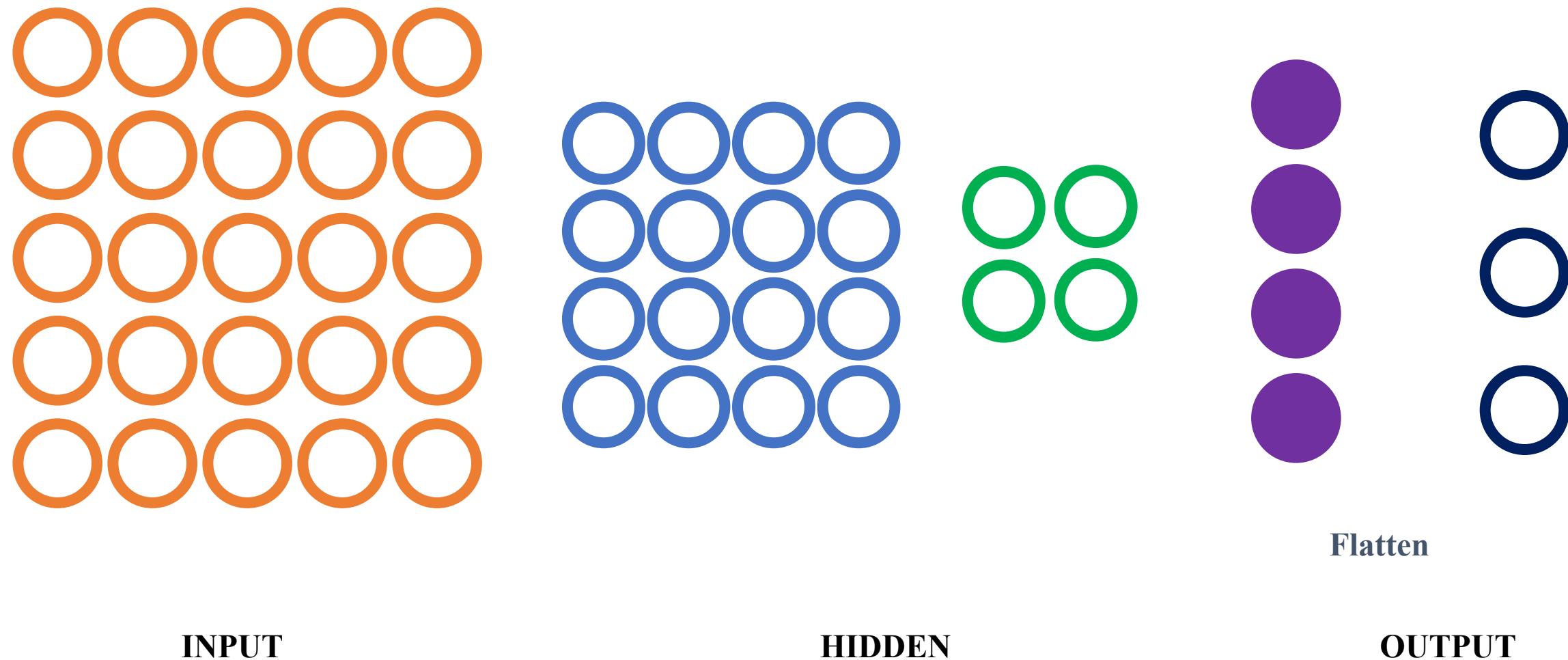
UNIT 02. Layers in CNN

◆ Convolutional Neural Network



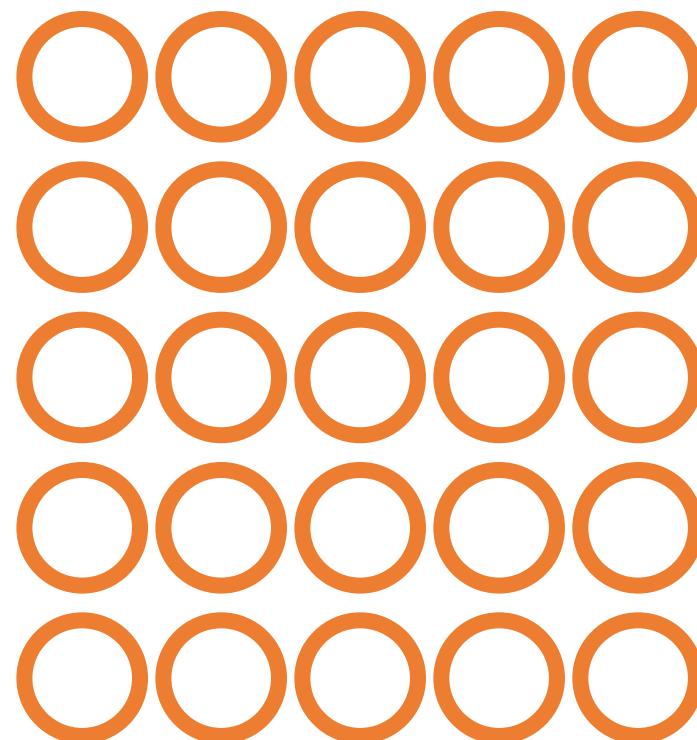
UNIT 02. Layers in CNN

◆ Convolutional Neural Network

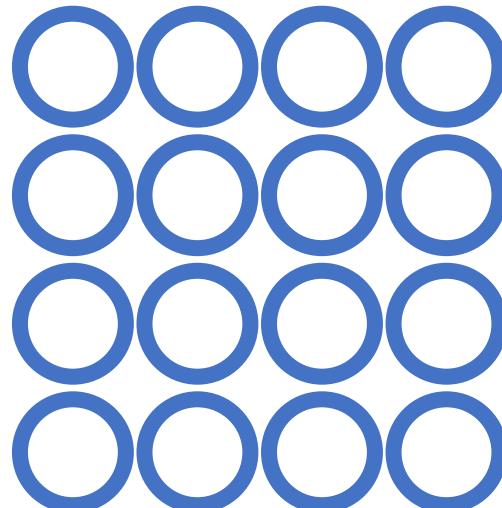


UNIT 02. Layers in CNN

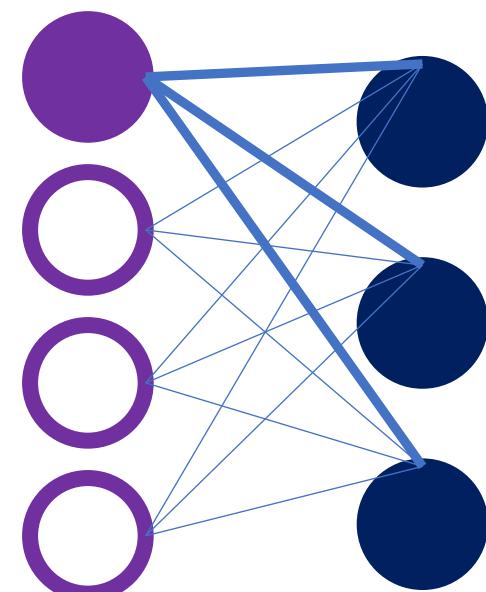
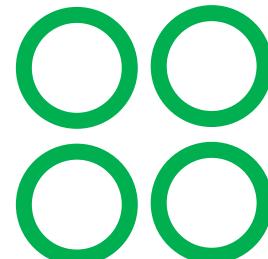
◆ Convolutional Neural Network



INPUT



HIDDEN

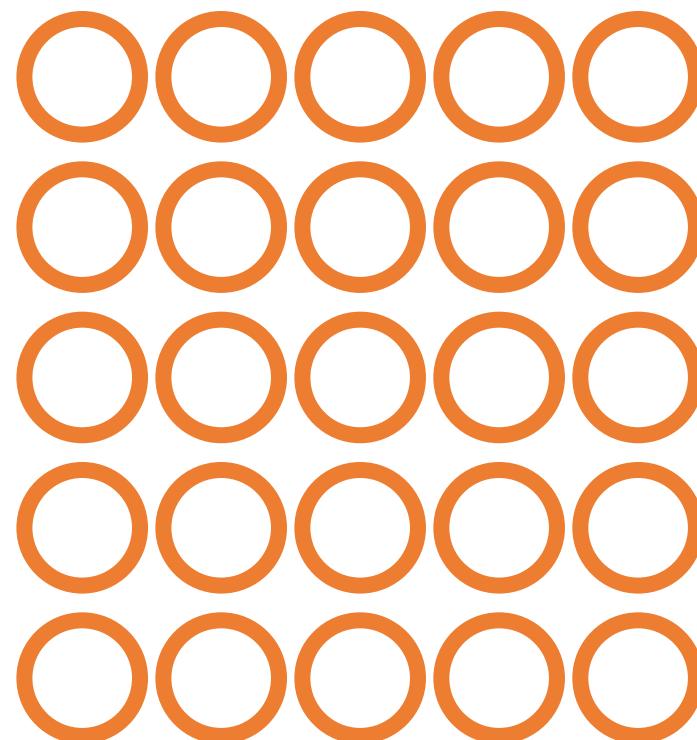


FC

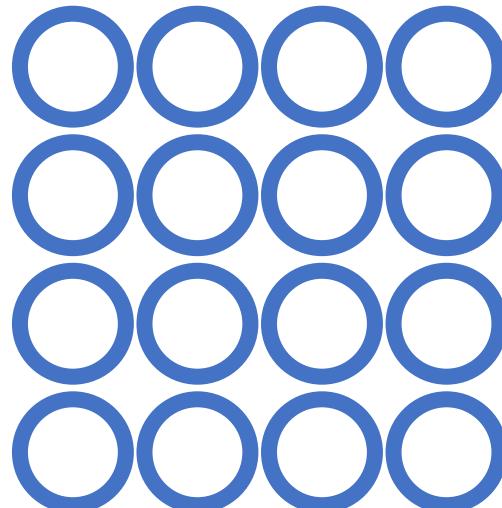
OUTPUT

UNIT 02. Layers in CNN

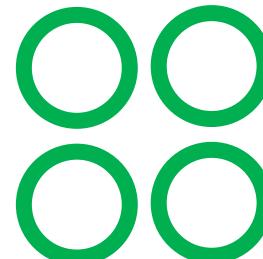
◆ Convolutional Neural Network



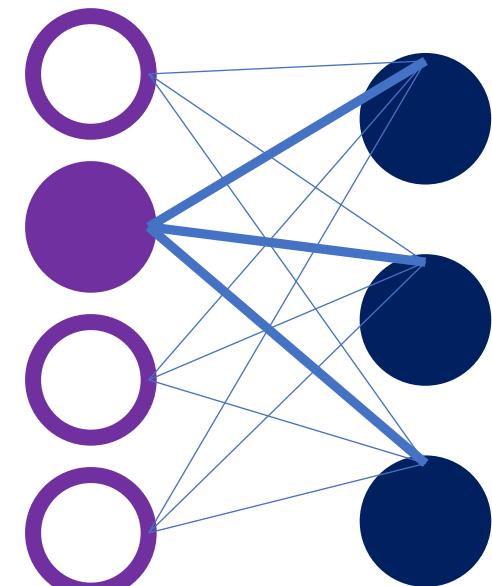
INPUT



HIDDEN



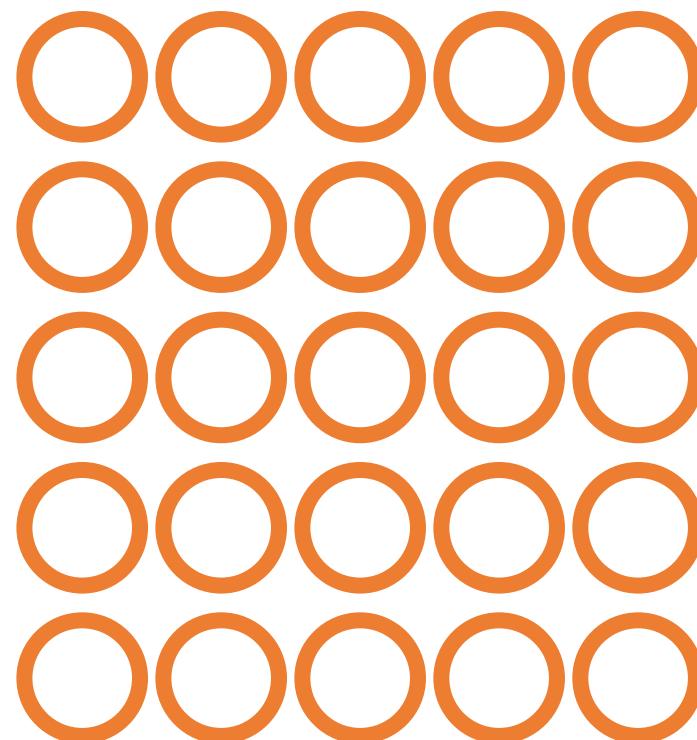
FC



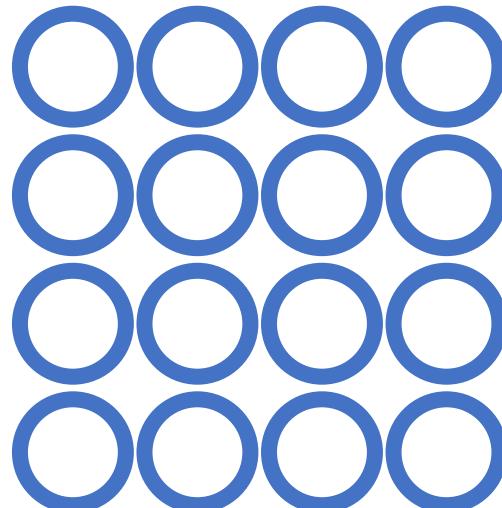
OUTPUT

UNIT 02. Layers in CNN

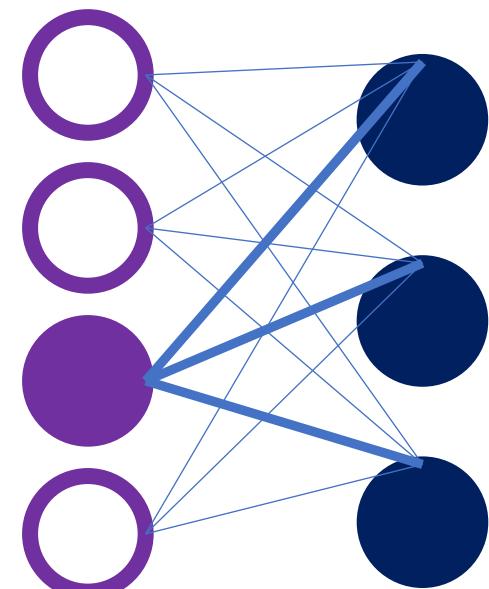
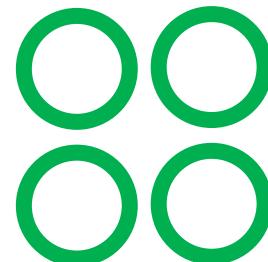
◆ Convolutional Neural Network



INPUT



HIDDEN

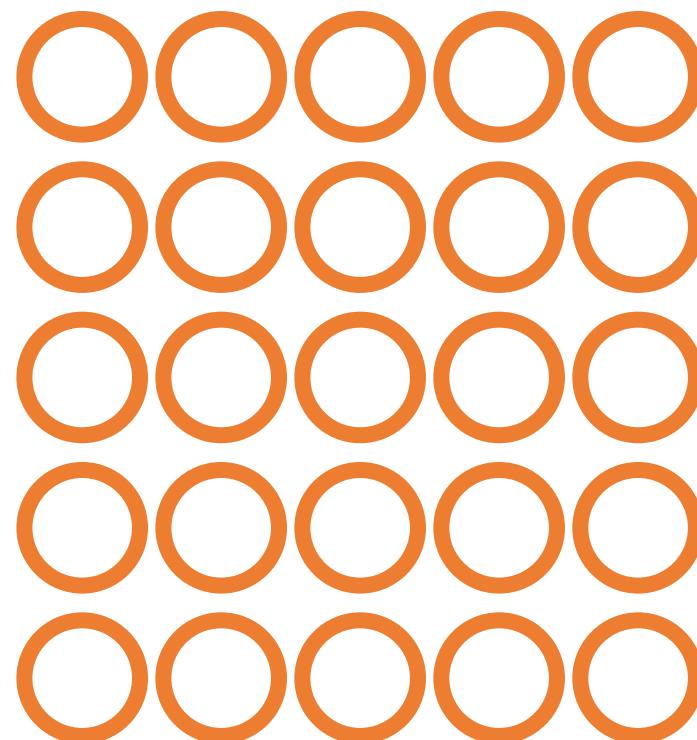


FC

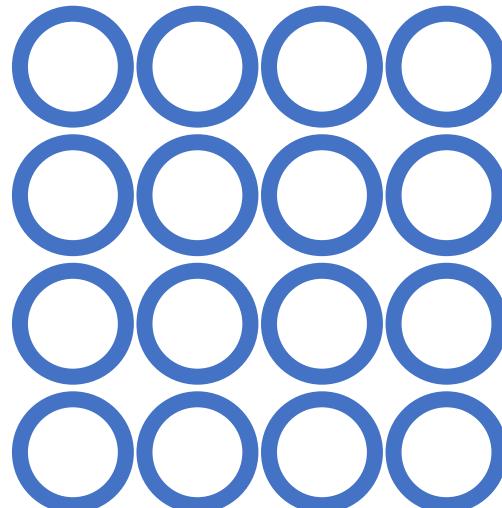
OUTPUT

UNIT 02. Layers in CNN

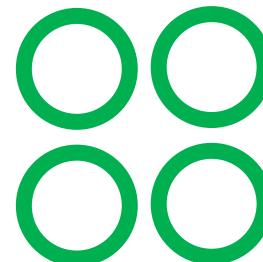
◆ Convolutional Neural Network



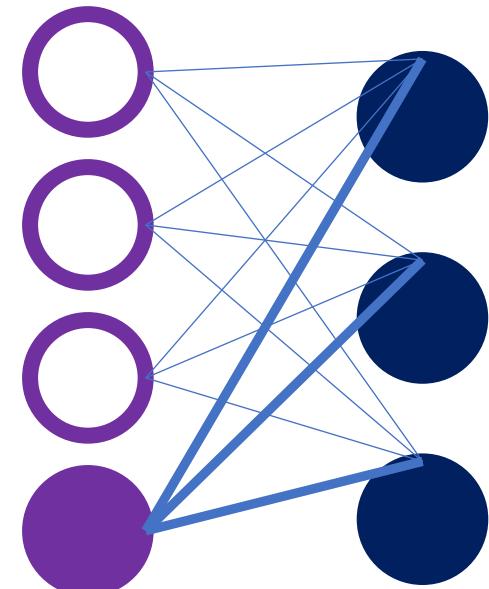
INPUT



HIDDEN



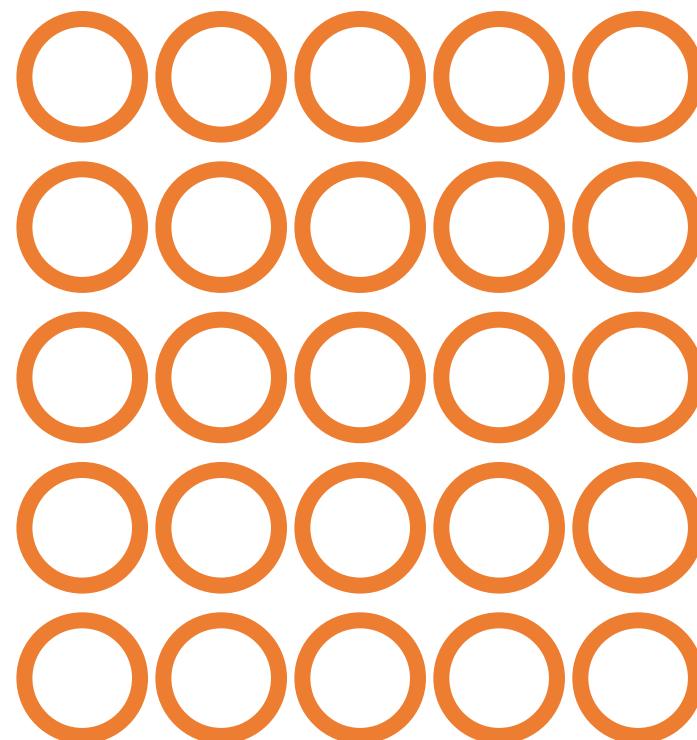
FC



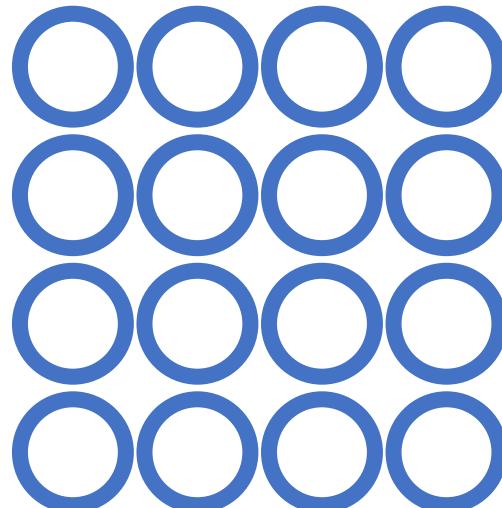
OUTPUT

UNIT 02. Layers in CNN

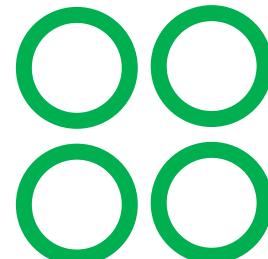
◆ Convolutional Neural Network



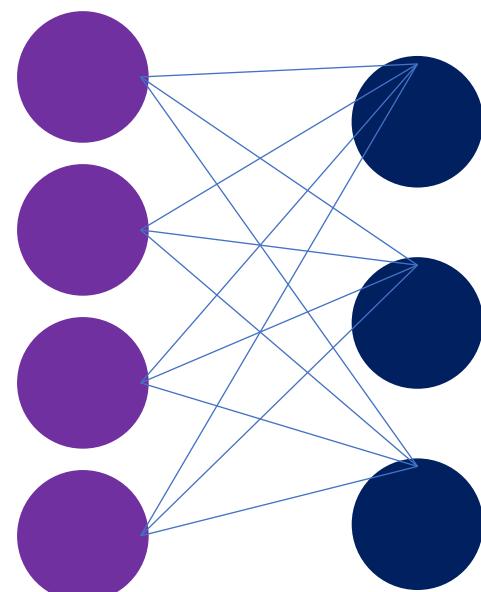
INPUT



HIDDEN



Flatten & FC



OUTPUT

Contents

Unit 01 | Intro

Unit 02 | Layers in CNN

Unit 03 | Convolution

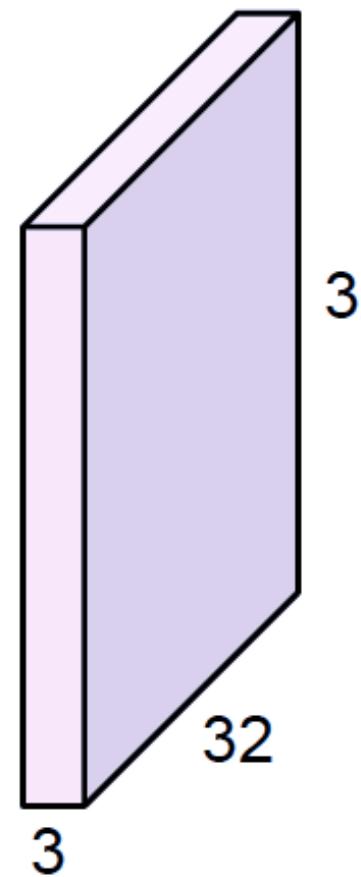
Unit 04 | Sub-Sampling

Unit 05 | Summary

UNIT 03. Convolution

◆ Convolution

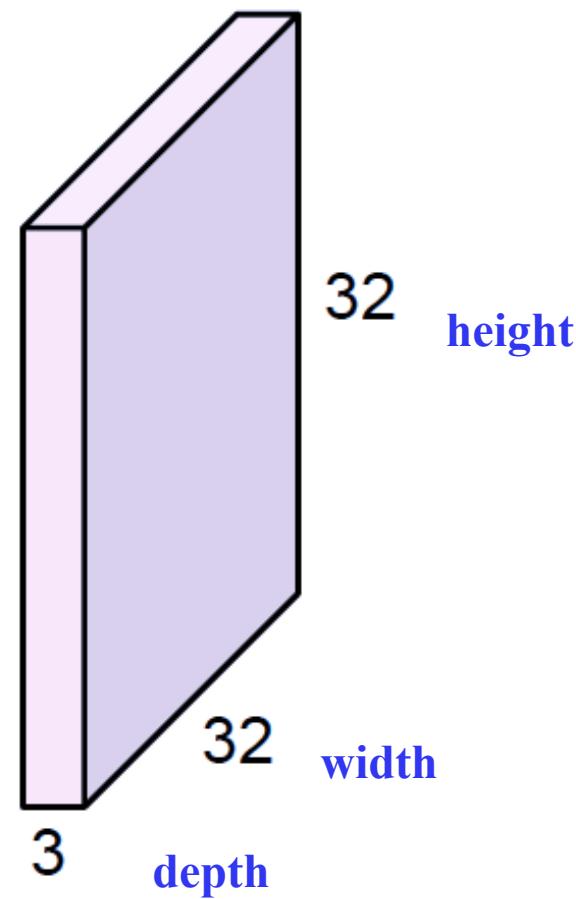
32x32x3 image



UNIT 03. Convolution

◆ Convolution

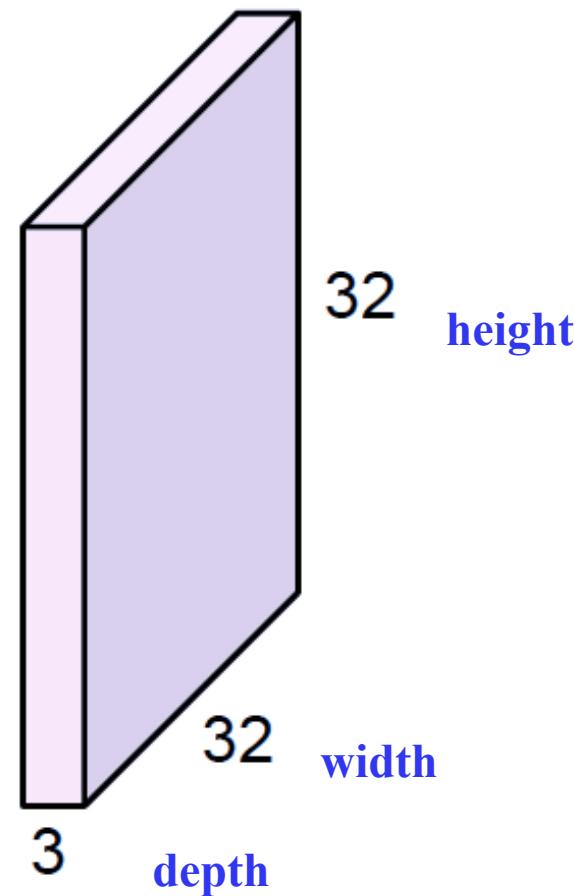
32x32x3 image



UNIT 03. Convolution

◆ Convolution

32x32x3 image



5x5x3 filter

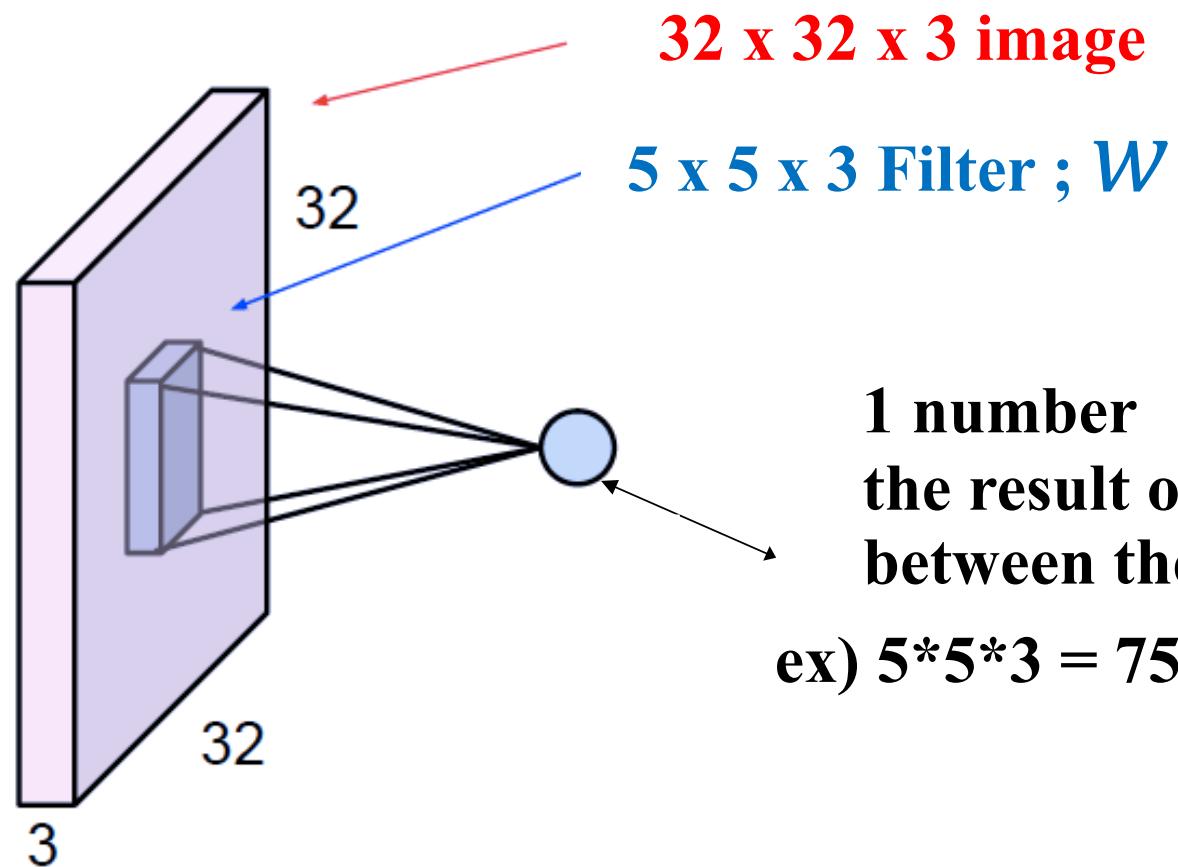


Convolve the filter with the image
“Slide over the image spatially, computing
dot products”

UNIT 03. Convolution

◆ Convolution

32x32x3 image



32 x 32 x 3 image

5 x 5 x 3 Filter ; W

1 number
the result of taking a dot product
between the filter and a local receptive field
ex) $5 \times 5 \times 3 = 75$ -dimensional dot product + bias

UNIT 03. Convolution

◆ Convolution

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

Input Volumn (7x7)

1	0	1
0	1	0
1	0	1

Filter (3 x 3)

Convolved Feature Volumn (5 x 5)

UNIT 03. Convolution

◆ Convolution

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

Dot Product

1	0	1
0	1	0
1	0	1



6				

Input Volume

Filter (3 x 3)

Convolved Feature Volume (5 x 5)

UNIT 03. Convolution

◆ Convolution

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

Dot Product

1	0	1
0	1	0
1	0	1



6	9			

Input Volume

Filter (3 x 3)

Convolved Feature Volume (5 x 5)

UNIT 03. Convolution

◆ Convolution

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

Dot Product

1	0	1
0	1	0
1	0	1



6	9	7		

Input Volumn

Filter (3 x 3)

Convolved Feature Volumn (5 x 5)

UNIT 03. Convolution

◆ Convolution

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

Input Volumn

Dot Product

1	0	1
0	1	0
1	0	1



6	9	7	6	6
9	7	7	6	5
6	5	2	3	3
5	4	4	6	9
5	6	6	8	4

Filter (3 x 3)

Convolved Feature Volumn (5 x 5)

UNIT 03. Convolution

◆ Convolution

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

Input Volumn

Dot Product

1	0	0
0	0	0
0	0	0



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

Filter (3 x 3)

Convolved Feature Volumn (5 x 5)

UNIT 03. Convolution

◆ Convolution

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

Input Volumn

Dot Product

0	0	0
0	1	0
0	0	0



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

Filter (3 x 3)

Convolved Feature Volumn (5 x 5)

UNIT 03. Convolution

◆ Convolution

6	9	7	6	6
9	7	7	6	5
6	5	2	3	3
5	4	4	6	9
5	6	6	8	4



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



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



8	13	9	8	7
12	10	10	9	7
8	9	3	4	6
8	6	6	6	11
7	6	11	10	6

UNIT 03. Convolution

◆ Convolution

6	9	7	6	6
9	7	7	6	5
6	5	2	3	3
5	4	4	6	9
5	6	6	8	4



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



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



8	13	9	8	7
12	10	10	9	7
8	9	3	4	6
8	6	6	6	11
7	6	11	10	6

3x3x3 filter

27-dimensional dot product

UNIT 03. Convolution

◆ Convolution

6	9	7	6	6
9	7	7	6	5
6	5	2	3	3
5	4	4	6	9
5	6	6	8	4



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



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



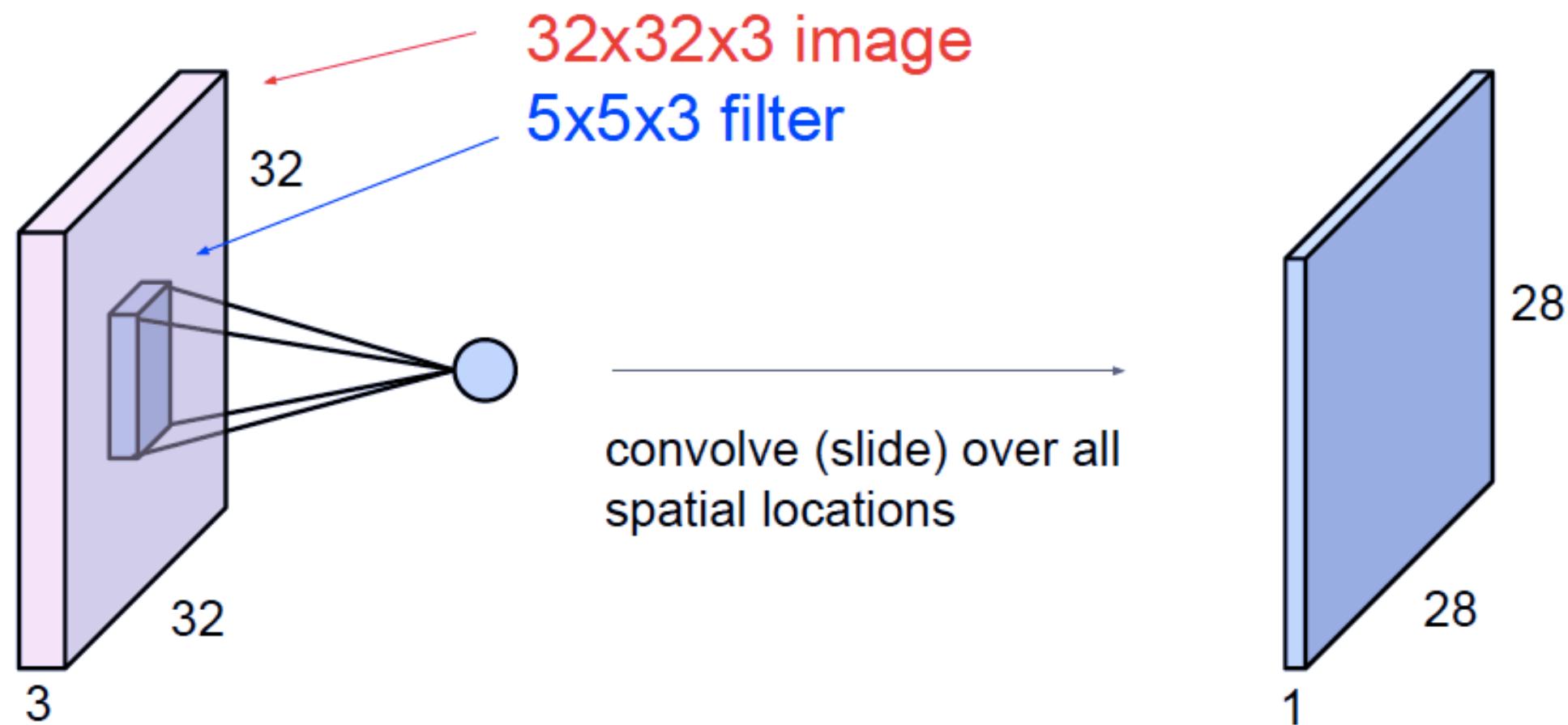
8	13	9	8	7
12	10	10	9	7
8	9	3	4	6
8	6	6	6	11
7	6	11	10	6

3x3x3 filter

27-dimensional dot product

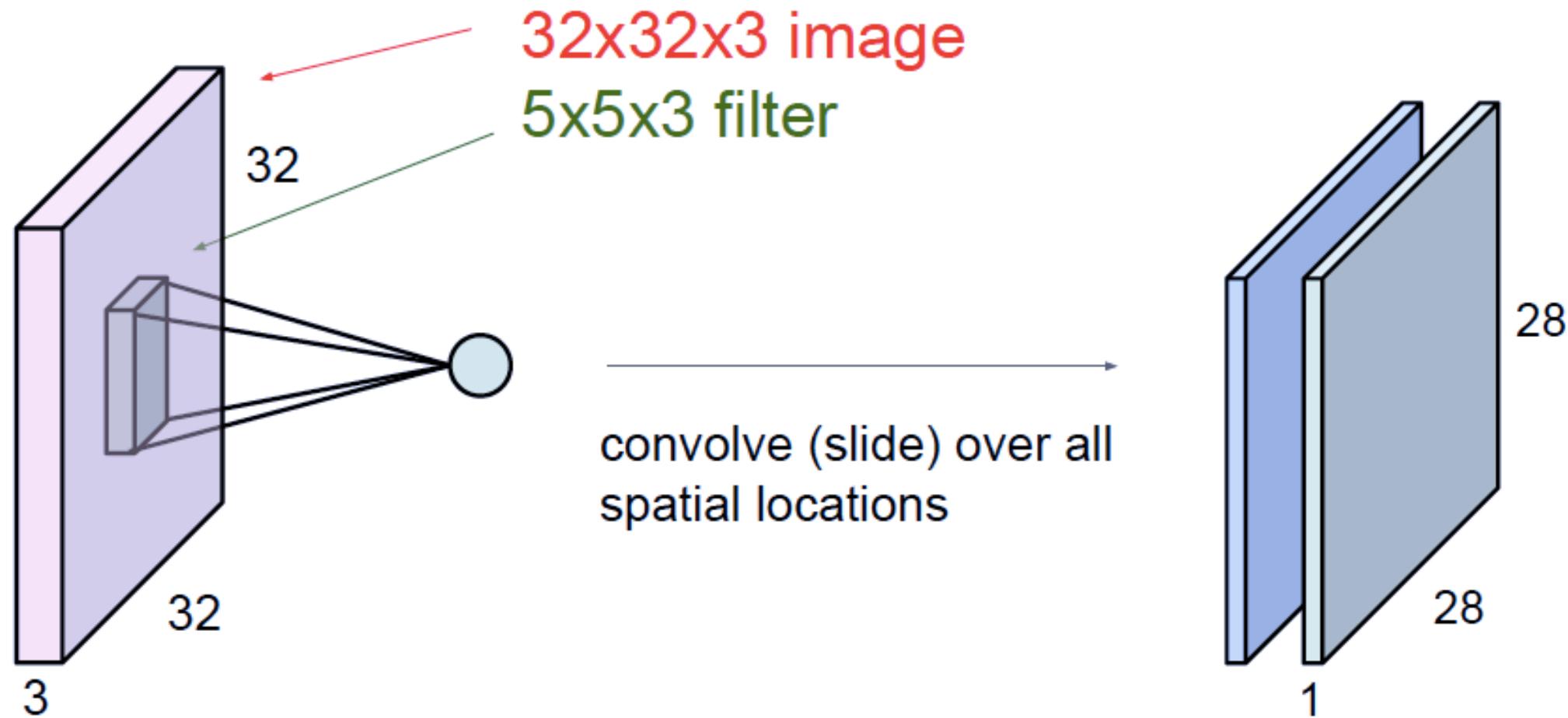
UNIT 03. Convolution

◆ Convolution



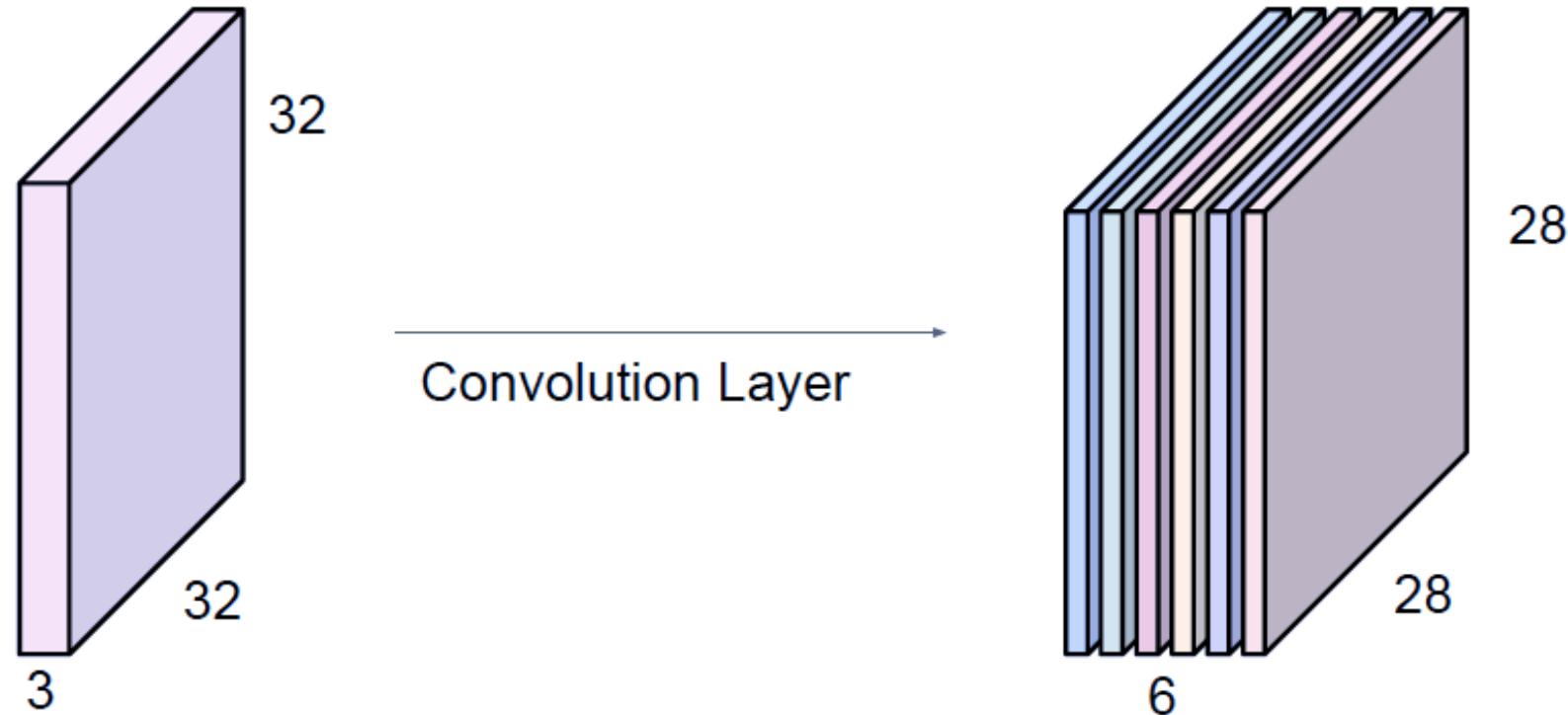
UNIT 03. Convolution

◆ Convolution



UNIT 03. Convolution

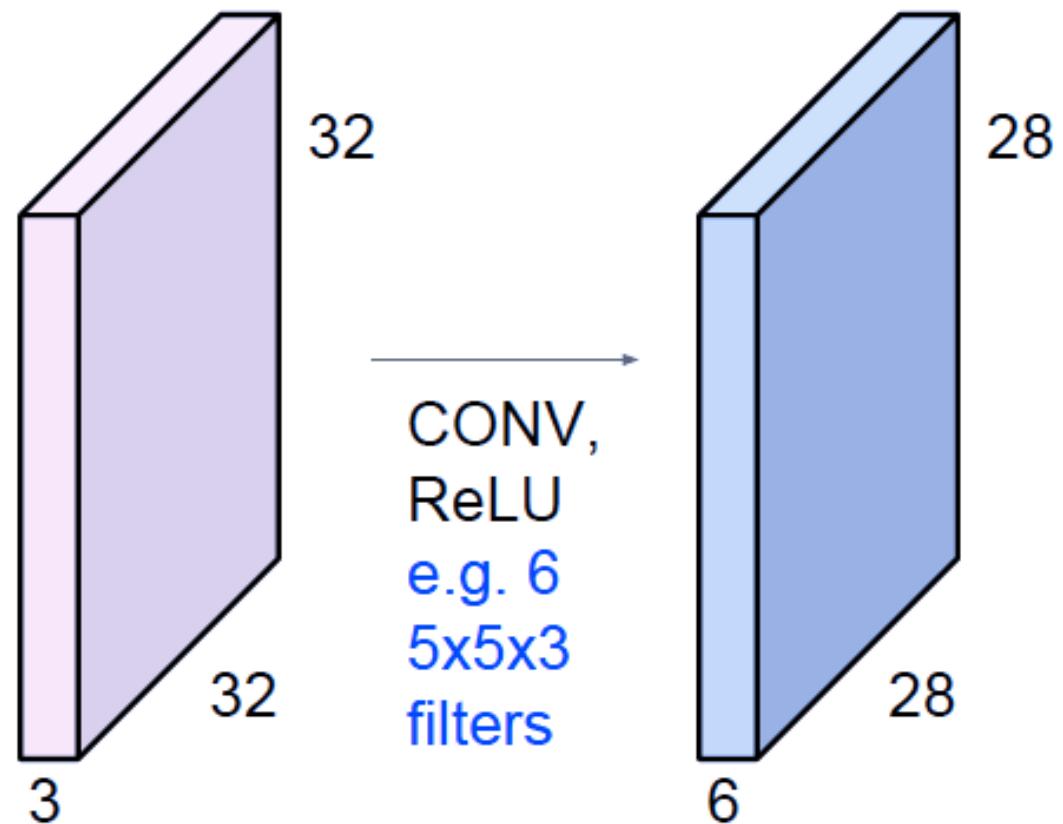
◆ Convolution



- 예를 들어, 6개의 $5 \times 5 \times 3$ filter가 있다면, 6개의 feature map이 생성됨
- 6개의 feature map을 stack up 하여 $28 \times 28 \times 6$ 의 새로운 size의 이미지를 얻음

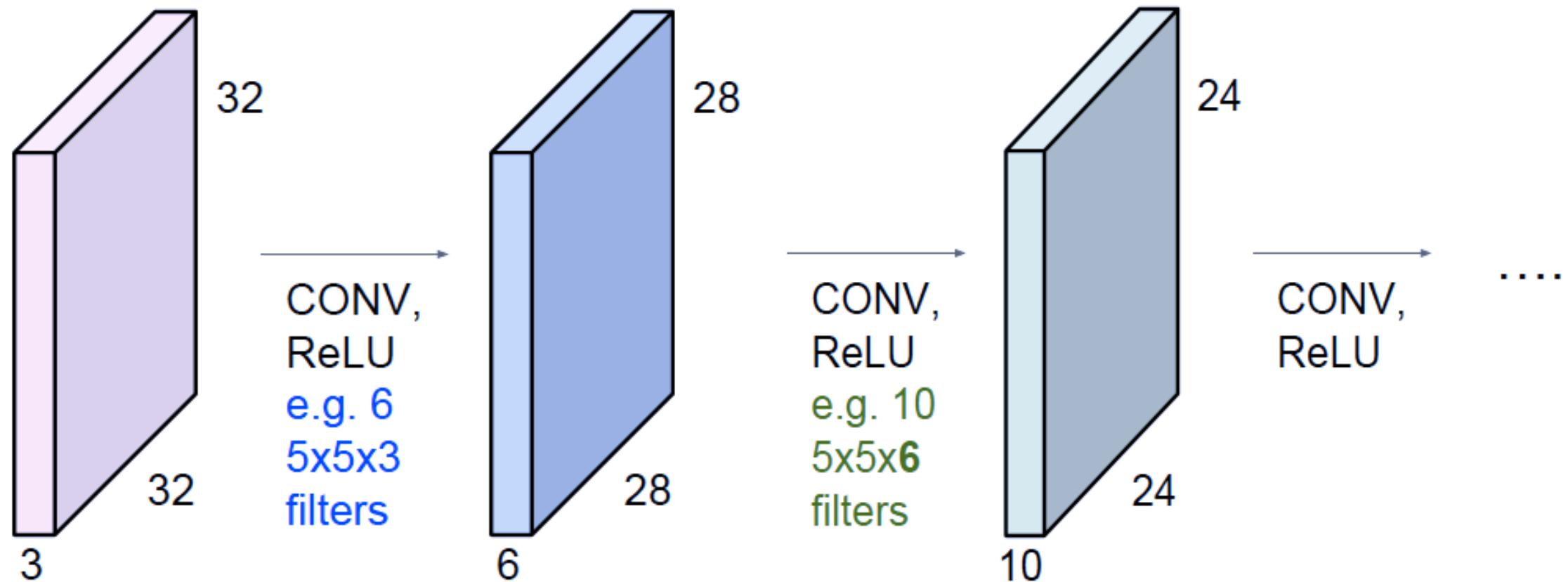
UNIT 03. Convolution

◆ Convolution



UNIT 03. Convolution

◆ Convolution



UNIT 03. Convolution

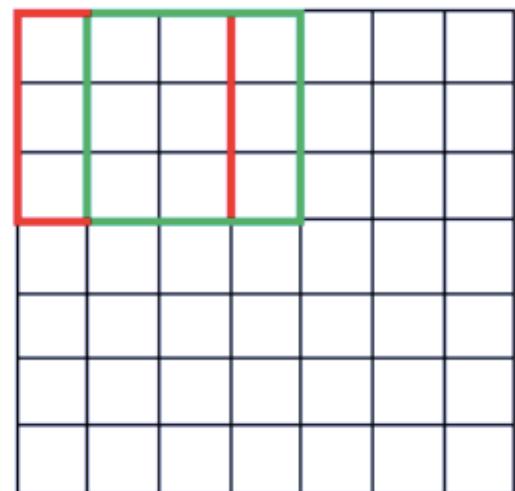
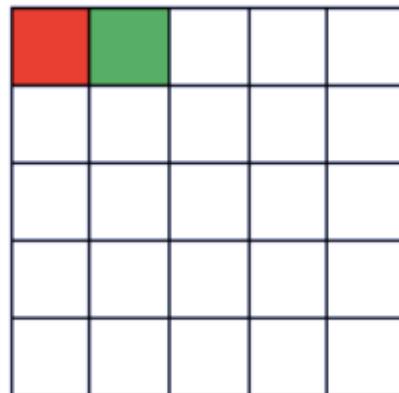
◆ Get familiar with this



UNIT 03. Convolution

◆ Stride

- **Stride =The amount by which the filter shifts**
- **Stride is normally set in a way so that the output volume is an integer and not a fraction**

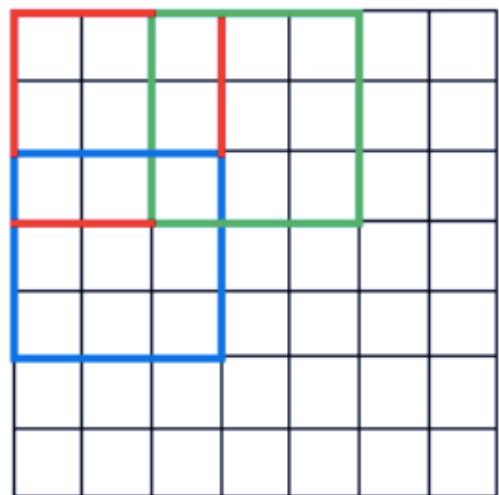
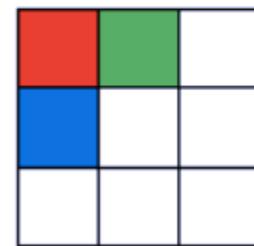
7 x 7 Input Volume5 x 5 Output Volume

Stride = 1

UNIT 03. Convolution

◆ Stride

- Stride = The amount by which the filter shifts
- Stride is normally set in a way so that the output volume is an integer and not a fraction

7 x 7 Input Volume3 x 3 Output Volume

Stride 2 size 3x3

Stride = 2 ↗ filter ↘

UNIT 03. Convolution

◆ Stride

- **Stride =The amount by which the filter shifts**
- **Stride is normally set in a way so that the output volume is an integer and not a fraction**

A large, solid black question mark is centered on the slide.

Stride = 3

UNIT 03. Convolution

◆ Stride

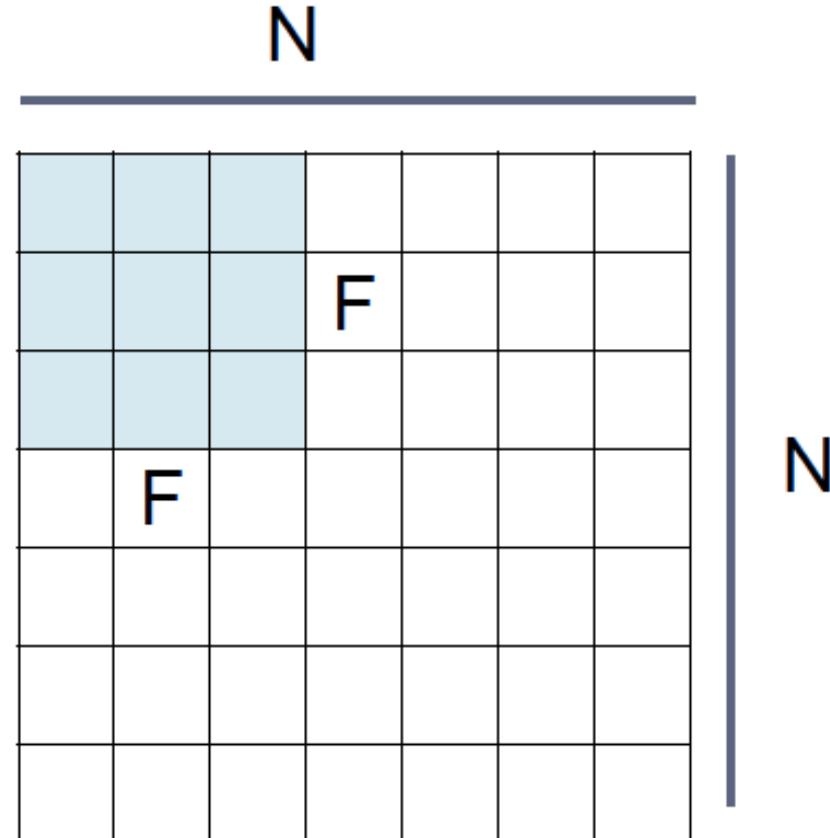
- **Stride =The amount by which the filter shifts**
- **Stride is normally set in a way so that the output volume is an integer and not a fraction**

doesn't fit!
cannot apply 3x3 filter on
7x7 input with stride 3.

Stride = 3

UNIT 03. Convolution

◆ Stride



- **N = Input size**
- **F = Filter size**
- **Output size: $\{(N - F) / \text{stride}\} + 1$**

- Ex) $N=7, F=3$

$$\text{stride}=1 \rightarrow \{(7-3)/1\} + 1 = 5$$

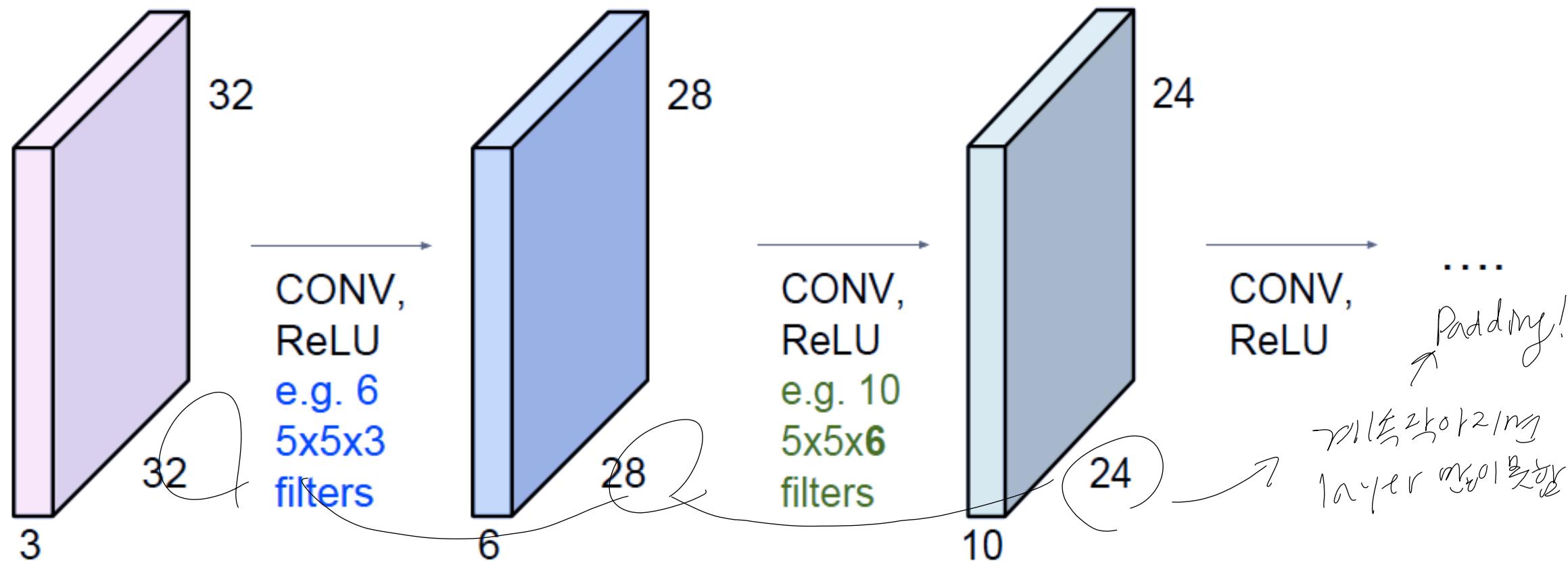
$$\text{Stride}=2 \rightarrow \{(7-3)/2\} + 1 = 3$$

$$\text{Stride}=3 \rightarrow \{(7-3)/3\} + 1 = 2.33 \quad \text{X}$$

UNIT 03. Convolution

◆ Padding

- 32x32 input convolved repeatedly with 5x5 filters shrinks volumes spatially!
(32 -> 28 -> 24 ...). Shrinking too fast is not good, doesn't work well.



UNIT 03. Convolution

◆ Padding

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	0	0	0	0	0	0

- Ex) Input 7x7

Filter 3x3, stride = 1

zero pad with 1 pixel border

what's the output?

- Recall

Output size: $\{(N - F) / \text{stride}\} + 1$

$$\begin{array}{c} 9-3 \\ (6/1) \\ 6+1=7 \end{array}$$

UNIT 03. Convolution

◆ Padding

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	0	0	0	0	0	0

- Ex) Input 7x7

Filter 3x3

pad with 1 pixel border -> what's the output?

- 7x7 output!!
- output = $\{(9-3)/1\} + 1$

UNIT 03. Convolution

◆ Padding

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	0	0	0	0	0	0

- Ex) Input 7x7

Filter 3x3

pad with 1 pixel border -> what's the output?

- 7x7 output!!
- 일반적으로 stride = 1 을 주며, Filter size 가 3인 경우 zero-padding = 1 을 주면
- 공식에 따라, input size 를 보존하는 output0| 생성

UNIT 03. Convolution

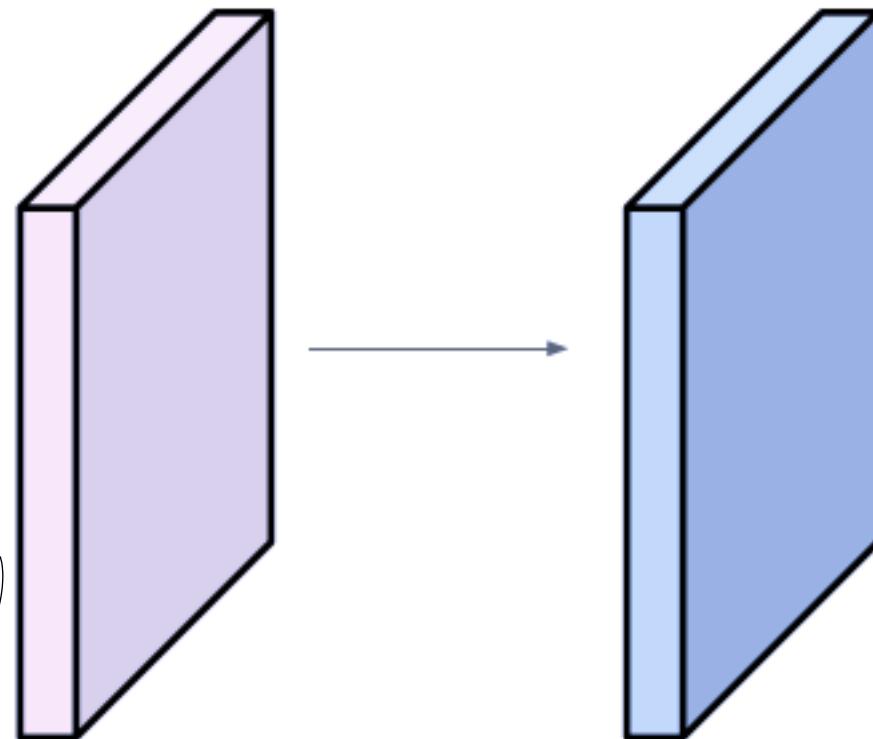
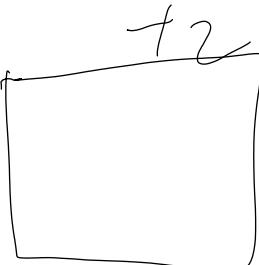
◆ Example

- Input volume: $32 \times 32 \times 3$
- 10 5x5 filters with stride 1, pad 2

- Number of parameters in this layer?

24x24

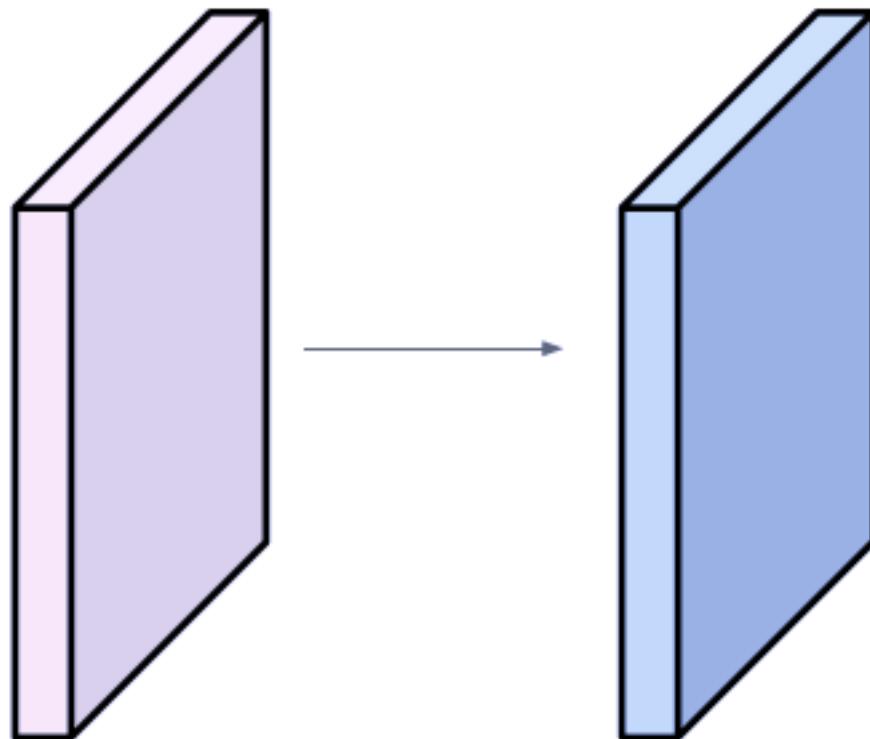
$$\left((32 - 5) / 1 \right)$$



UNIT 03. Convolution

◆ Example

- Input volume: $32 \times 32 \times 3$
- 10 5x5 filters with stride 1, pad 2
- Number of parameters in this layer?
- Each filter has $5 \times 5 \times 3 + 1 = 76$ params (+1 for bias)
 $\rightarrow 76 * 10 = 760$



Contents

Unit 01 | Intro

Unit 02 | Layers in CNN

Unit 03 | Convolution

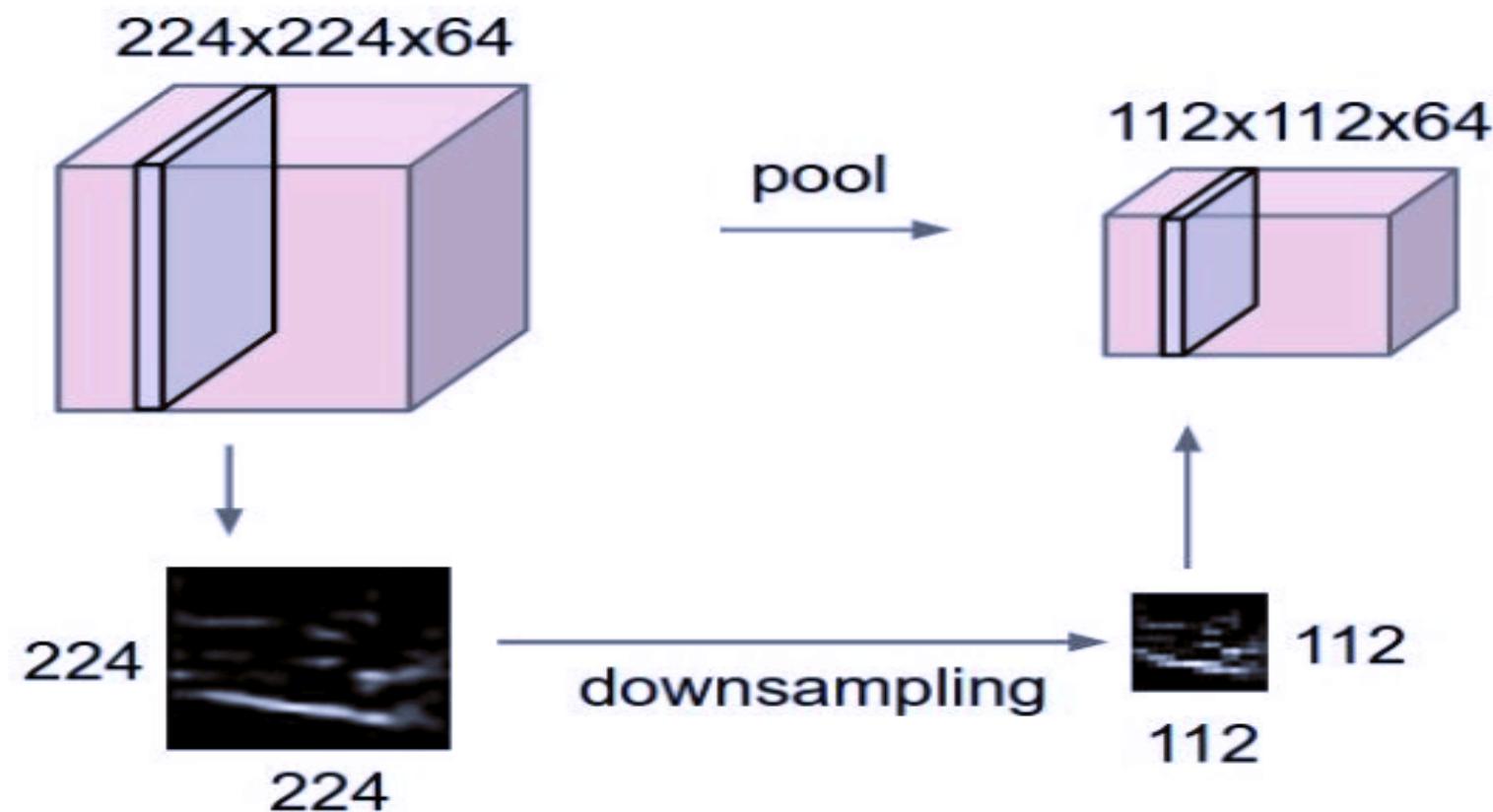
Unit 04 | Sub-Sampling

Unit 05 | Summary

UNIT 04. Sub-Sampling

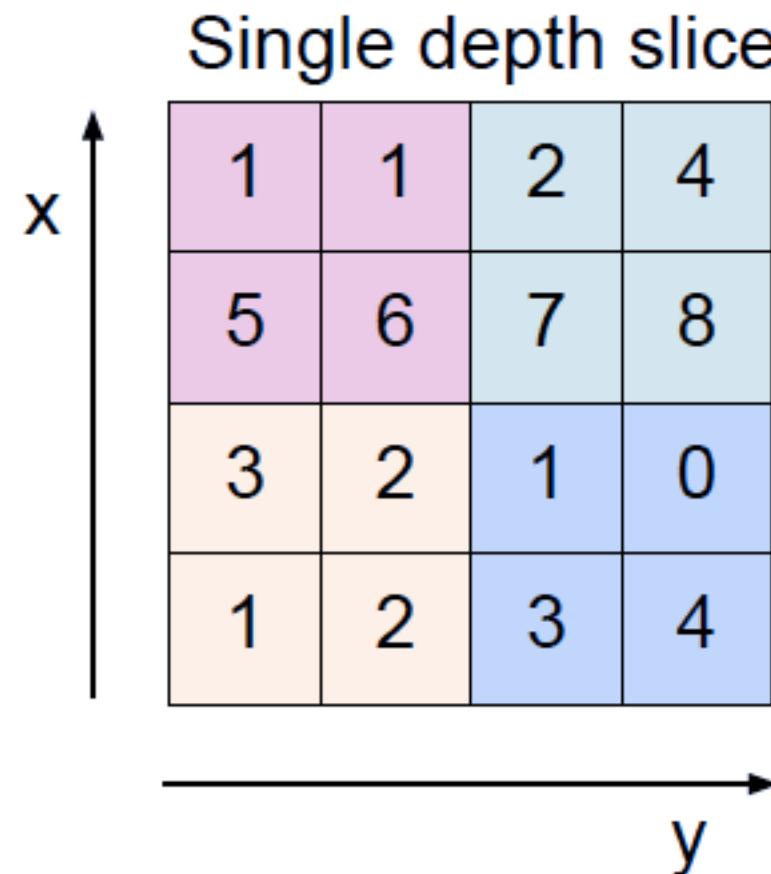
◆ Subsampling

- makes the representations smaller and more manageable
- operates over each activation map independently:



UNIT 04. Sub-Sampling

◆ Max-pooling



max pool with 2x2 filters
and stride 2

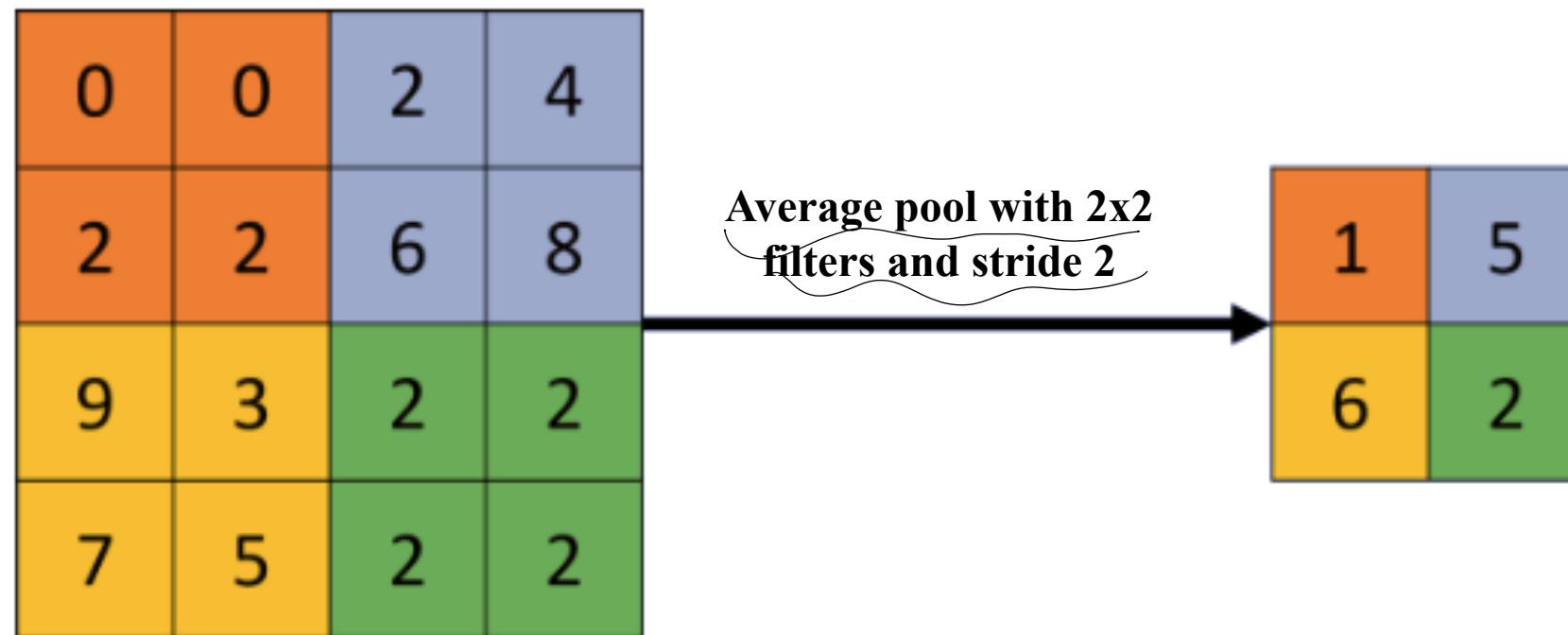
6	8
3	4

stride 2 means pooling 2x2

UNIT 04. Sub-Sampling

◆ Average Pooling

Single depth slice



Contents

Unit 01 | Intro

Unit 02 | Layers in CNN

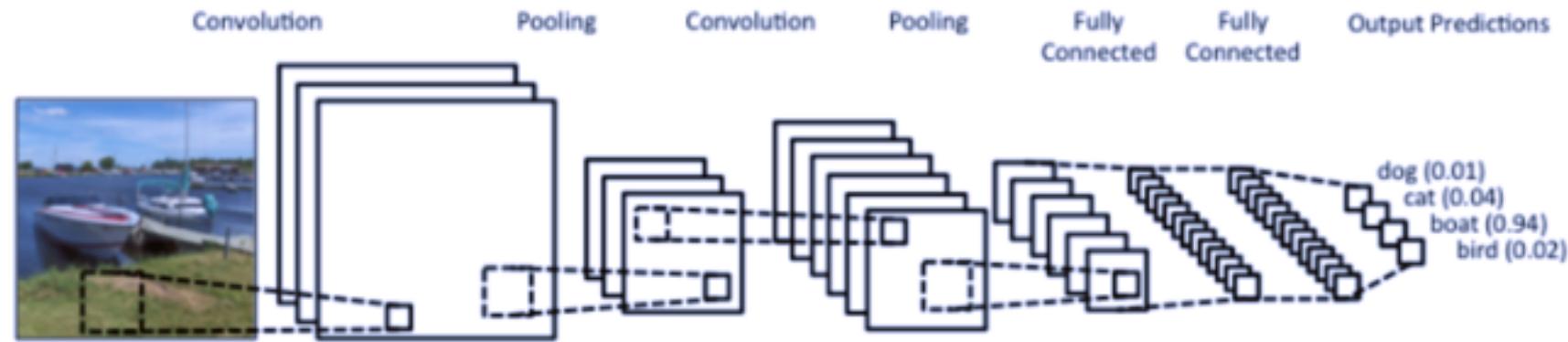
Unit 03 | Convolution

Unit 04 | Sub-Sampling

Unit 05 | Summary

UNIT 05. Summary

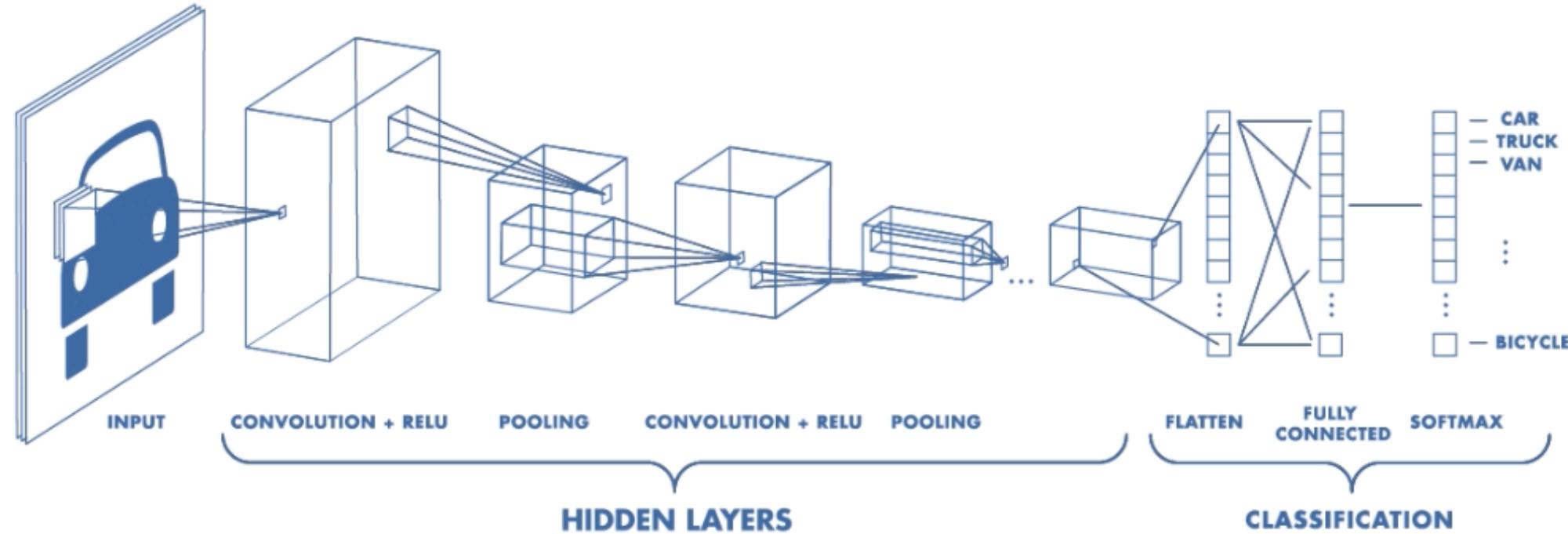
◆ Layers in CNN



- CNN = **Convolutions** followed by **subsampling** and **fully connected layer**
- **Feature Extraction** <- Convolution + subsampling layers
- **Classifier** <- Fully connected layer

UNIT 02. Layers in CNN

◆ Layers in CNN



- This is Pretty much **everything** about the convolutional neural network
- **Convolution** + **Subsampling** + **Full Connection**

UNIT 02. Layers in CNN

◆ Layers in CNN

Why is so powerful?

- Local connectivity(receptive field)
- Shared Weights and Biases
- Compositionality

Q & A



Reference

<https://cs.stanford.edu/people/karpathy/convnetjs/demo/cifar10.html>

<http://cs231n.github.io/convolutional-networks/>

<https://medium.com/fbdevclagos/applications-of-deep-learning-4f725f9147ac>

imagenet-classification-with-deep-convolutional-neural-networks(Alex Krizhevsky
2012)

<https://laonple.blog.me/220643128255>