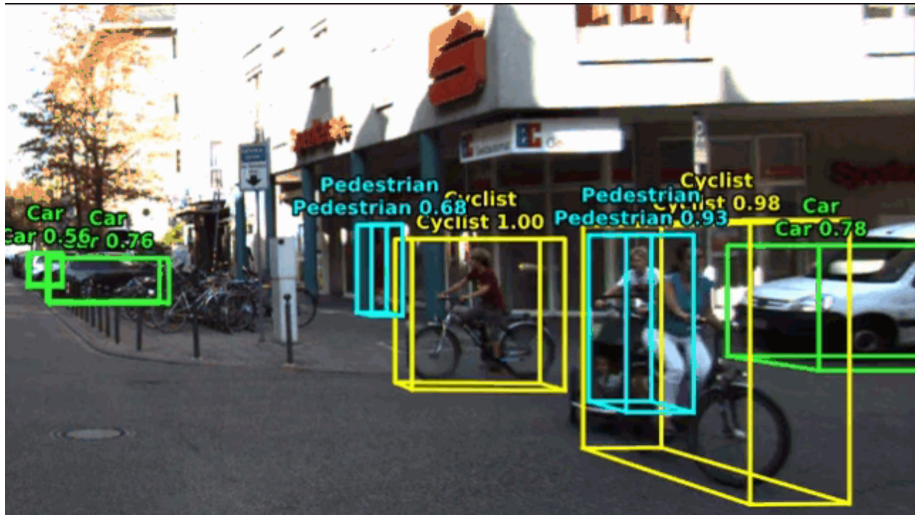
Convolutional Neural Networks

Course 3, Module 3, Lesson 5

UNIVERSITY OF TORONTO
FACULTY OF APPLIED SCIENCE & ENGINEERING

ConvNets For Self-Driving Cars



Code at: https://github.com/kujason/avod

Code at: https://github.com/oandrienko/fast-semantic-segmentation

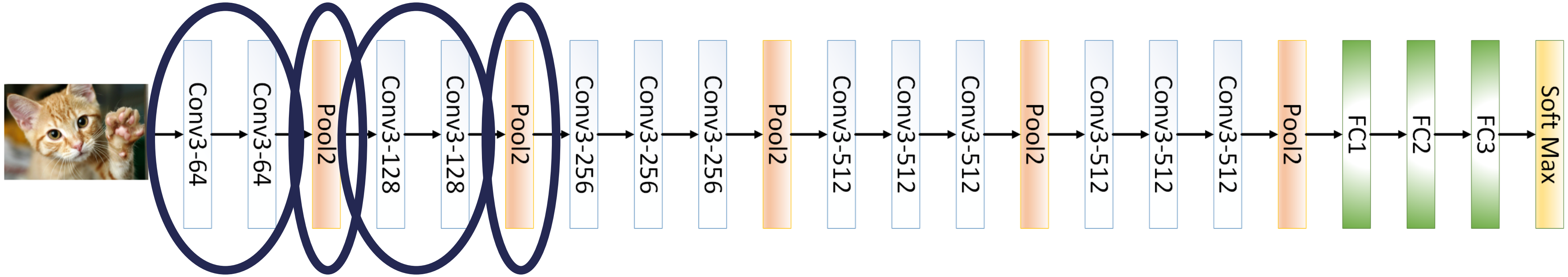
Learning Objectives

- Learn how a neural network can use cross-correlation in its hidden layers instead of general matrix multiplication, to form ConvNets

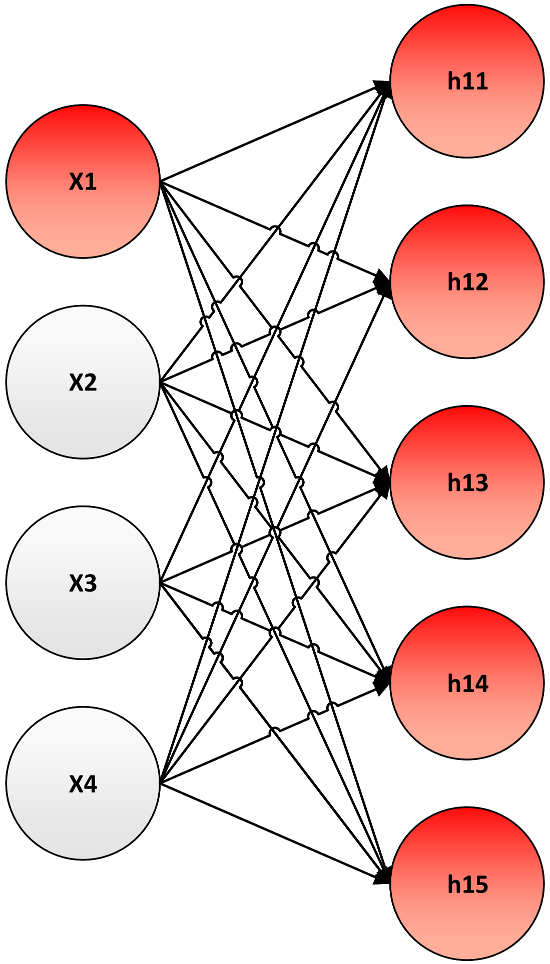
- Learn the advantages of using ConvNets over traditional neural networks for processing images

ConvNets

- Used for processing data defined on grid
- 1D time series data, 2D images, 3D videos
- Two major type of layers:
  1. **Convolution Layers**
  2. **Pooling Layers**
- **Example: VGG 16**

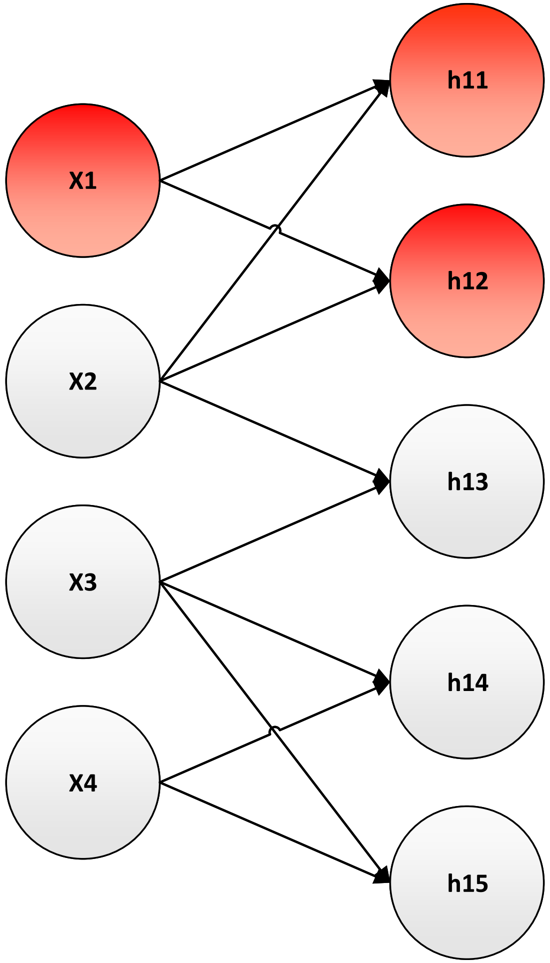


Fully Connected VS Convolutional Layers



$$h_n = g(W^T h_{n-1} + b)$$

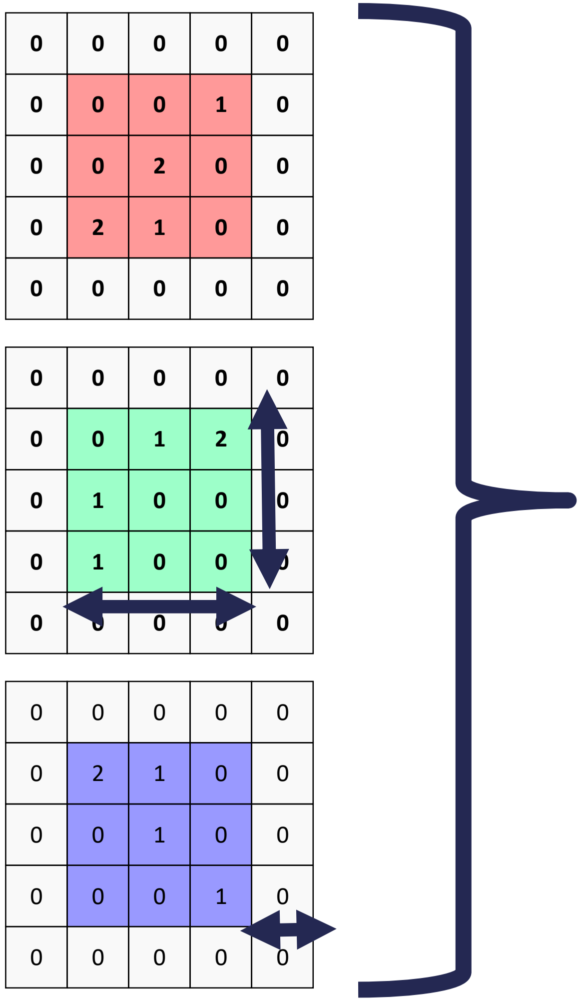
Fully Connected VS Convolutional Layers



Cross-correlation

$$h_n = g(W * h_{n-1} + b)$$

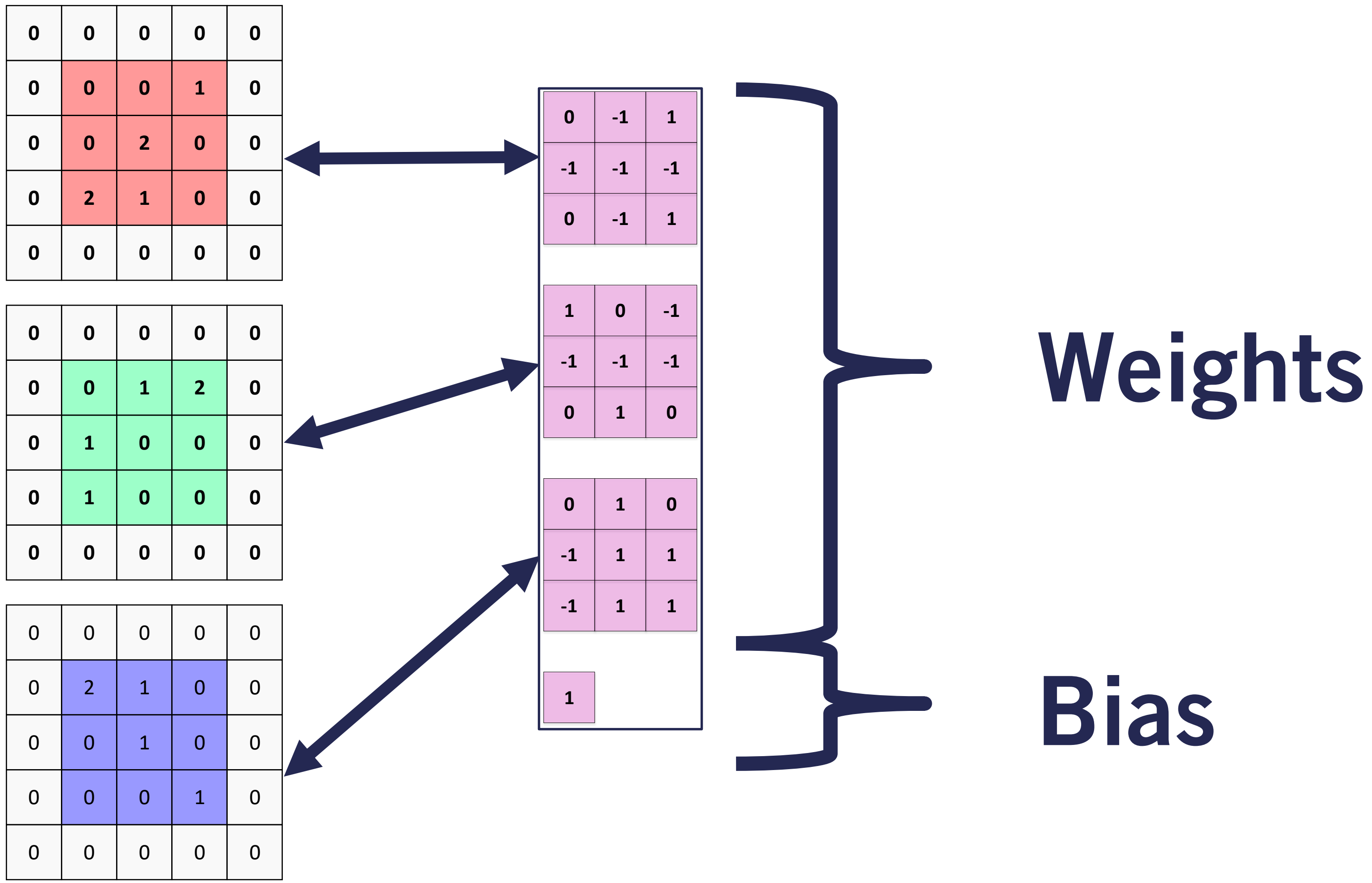
Cross Correlation



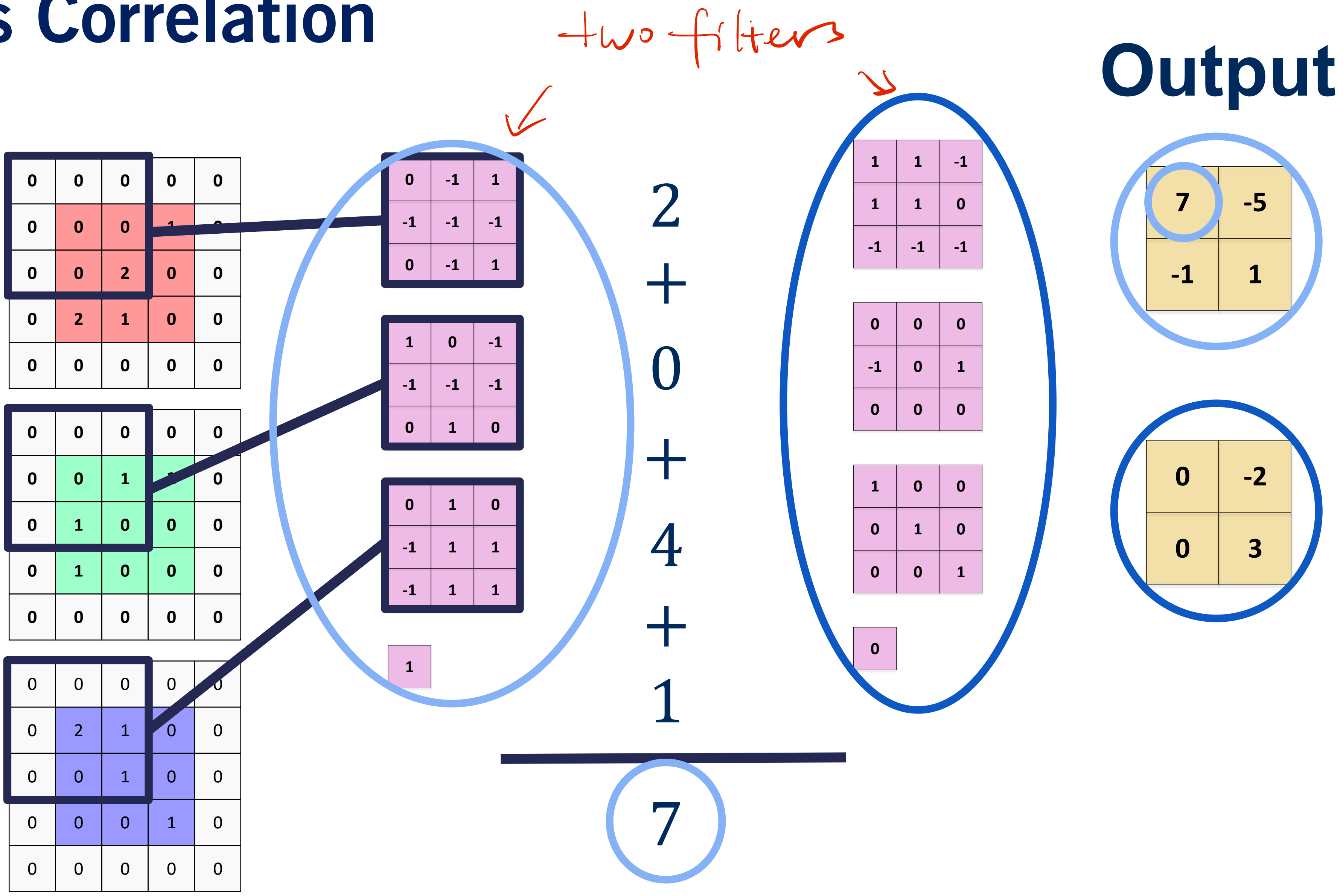
- **Width:** horizontal dimension of input volume — 3
- **Height**: vertical dimension of input volume — 3
- **Depth:** number of channels of input volume — 3

- **Padding size:** essential to retain shape! — 1

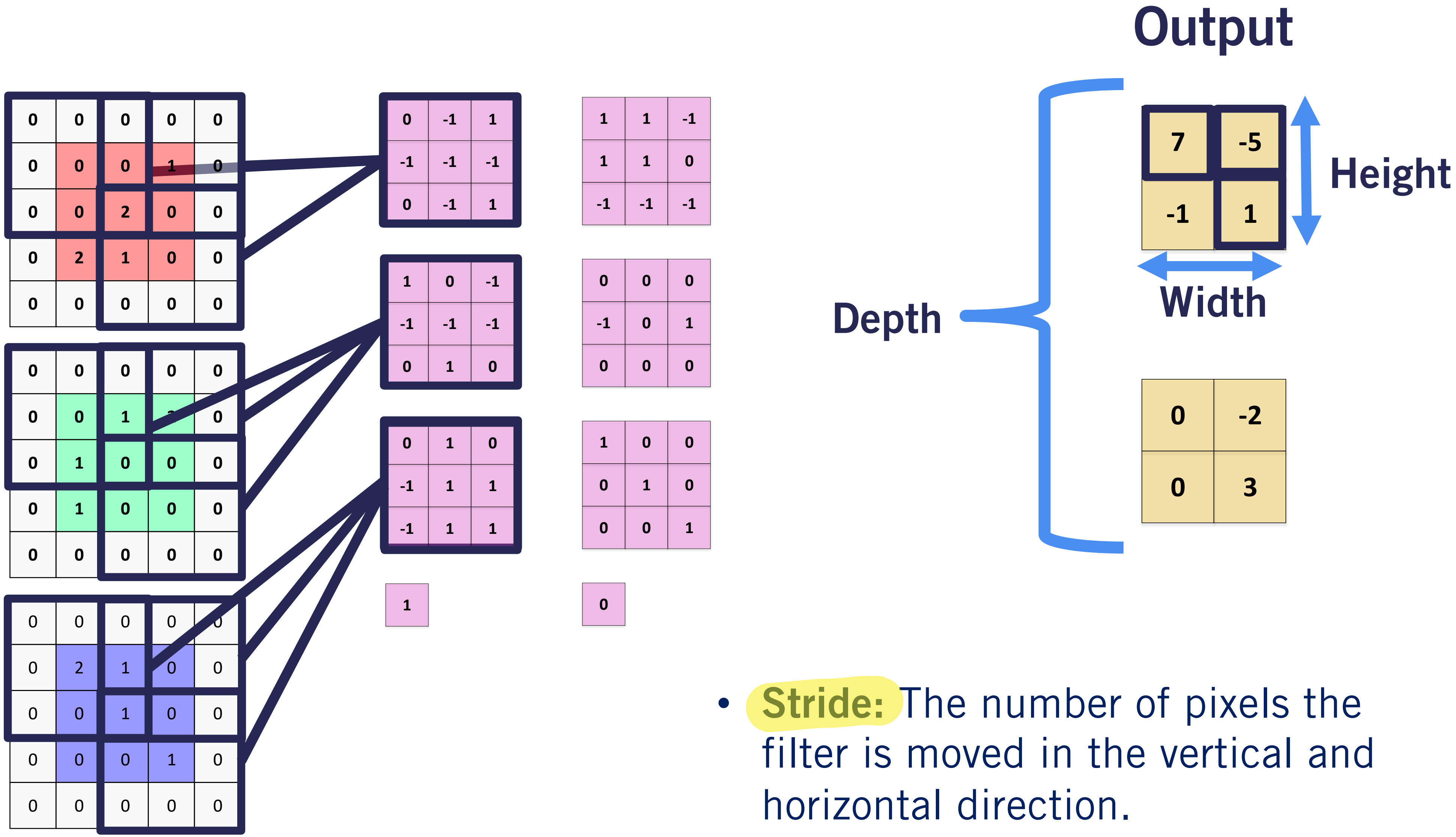
Cross Correlation



Cross Correlation



Cross Correlation



- **Stride:** The number of pixels the filter is moved in the vertical and horizontal direction.

Output Volume Shape

- Filters are size $\boldsymbol{m}\times\boldsymbol{m}$
- Number of filters = $\boldsymbol{K}$
- Stride = $\boldsymbol{S}$, Padding = $\boldsymbol{P}$

$$\boldsymbol{W_{out}} = \frac{\boldsymbol{W_{in}} - m + 2\times P}{S} + 1$$

$$\boldsymbol{H_{out}} = \frac{\boldsymbol{H_{in}} - m + 2\times P}{S} + 1$$

$$\boldsymbol{D_{out}} = K$$

Pooling Layers: Max Pooling

Output invariant to small translation of the input

$$\max(21, 8, 12, 19) = 21$$

21	8	8	12
12	19	9	7
8	10	4	3
18	9	10	9

21	12
18	10

Output Volume Shape

- Pool size $\boldsymbol{n}\times\boldsymbol{n}$
- Stride = $\boldsymbol{S}$

$$\boldsymbol{W_{out}} = \frac{\boldsymbol{W_{in}} - n}{S} + 1$$

$$\boldsymbol{H_{out}} = \frac{\boldsymbol{H_{in}} - n}{S} + 1$$

$$\boldsymbol{D_{out}} = \boldsymbol{D_{in}}$$

Pooling Layers: Max Pooling

$$\max(21, 8, 12, 19) = 21$$

8	21	8	8	12
9	12	19	9	7
4	8	10	4	3
10	18	9	10	9

21	19
18	10

c.f.

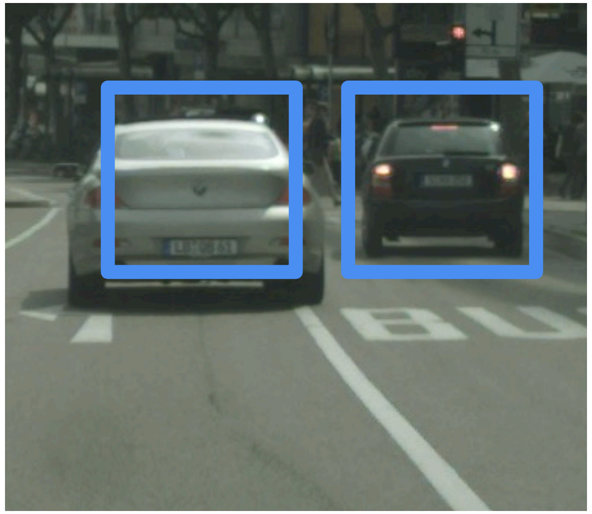
21	12
18	10

Advantages of ConvNets

- Convolutional neural networks are by design, a natural choice to process images

- Convolutional layers have **less parameters** than fully connected layers, reducing the chances of overfitting

- Convolutional layers use the same parameters to process every block of the image. Along with pooling layers, this leads to **translation invariance**, which is particularly important for image understanding



Summary

- ConvNets were one of the **first** neural network models to perform well at a time where other feedforward architectures failed

- ConvNets were one of the **first** neural network models to solve important commercial applications, such as handwritten digit recognition in the early 1990s [LeCun et. al.]

- **Next: 2D Object Detection**