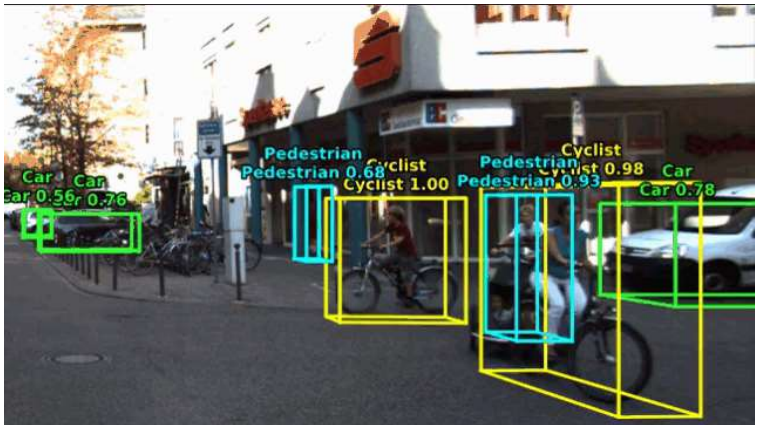
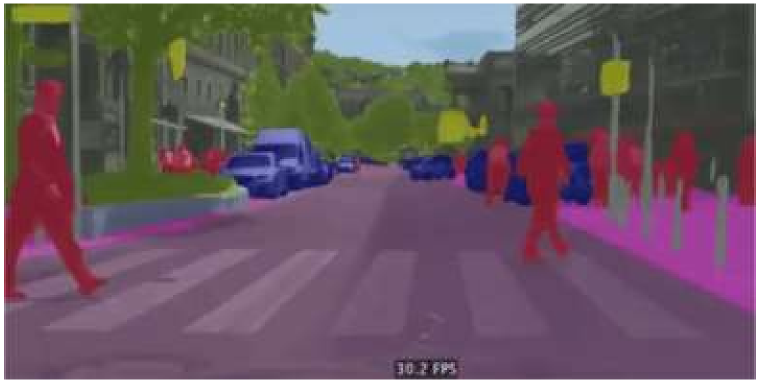
# ConvNets For Self-Driving Cars



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



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

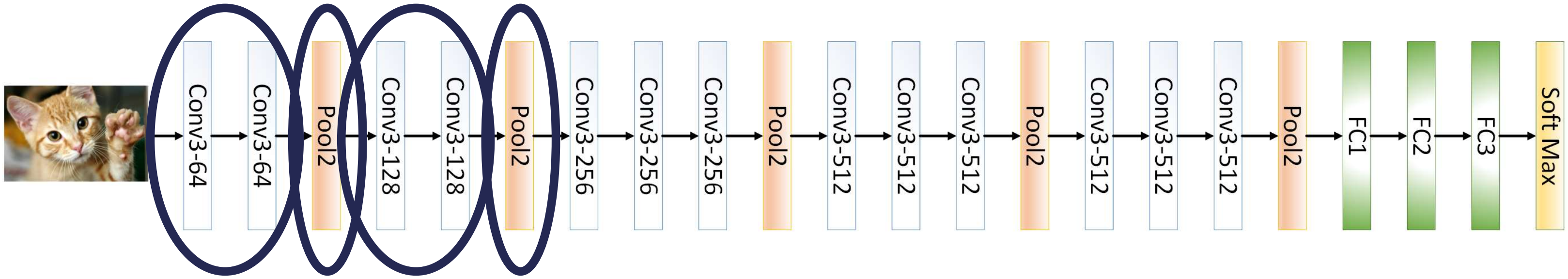
# Learning Objectives

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

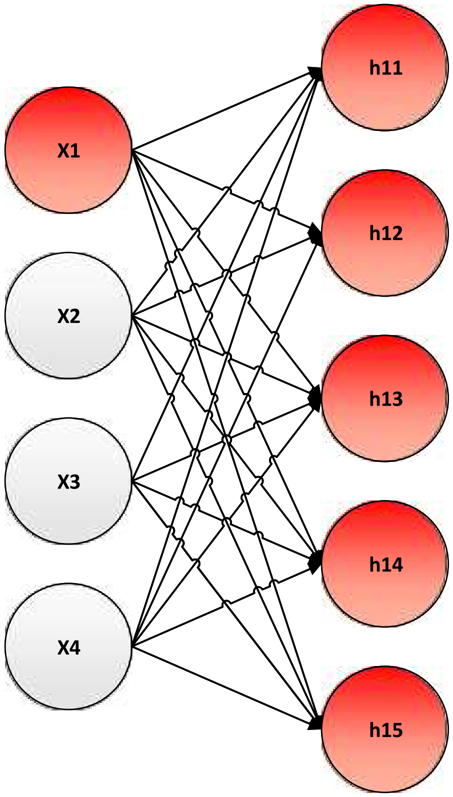
- Learn the advantages of using ConvNets over traditional neural networks for processing images

# 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**

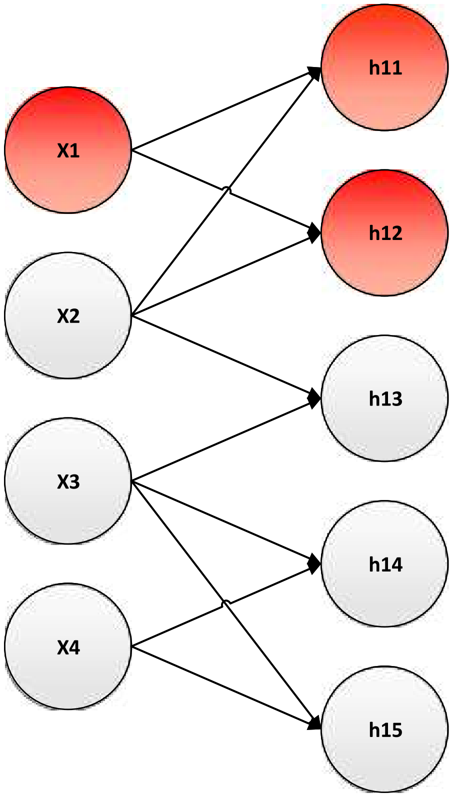


# Fully Connected VS Convolutional Layers



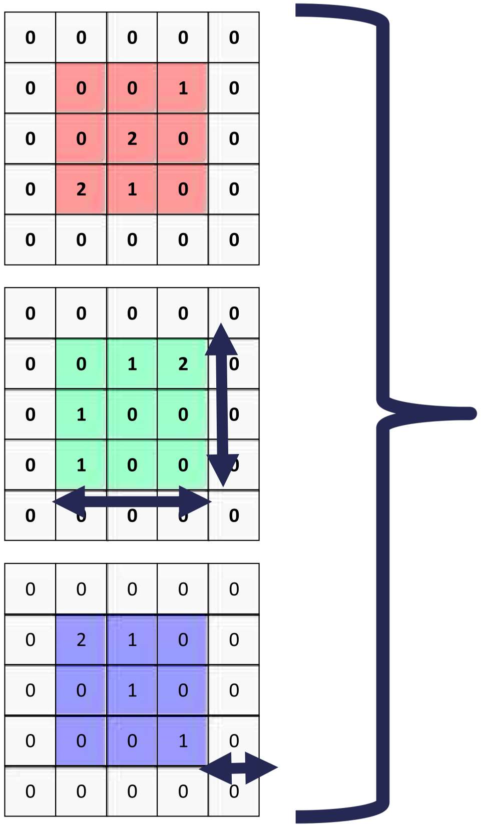
$$h_n = g(\boxed{W^T h_{n-1}} + b)$$

# Fully Connected VS Convolutional Layers



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

# Cross Correlation



|  |  |  |  |  |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 2 | 0 | 0 |
| 0 | 2 | 1 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 |

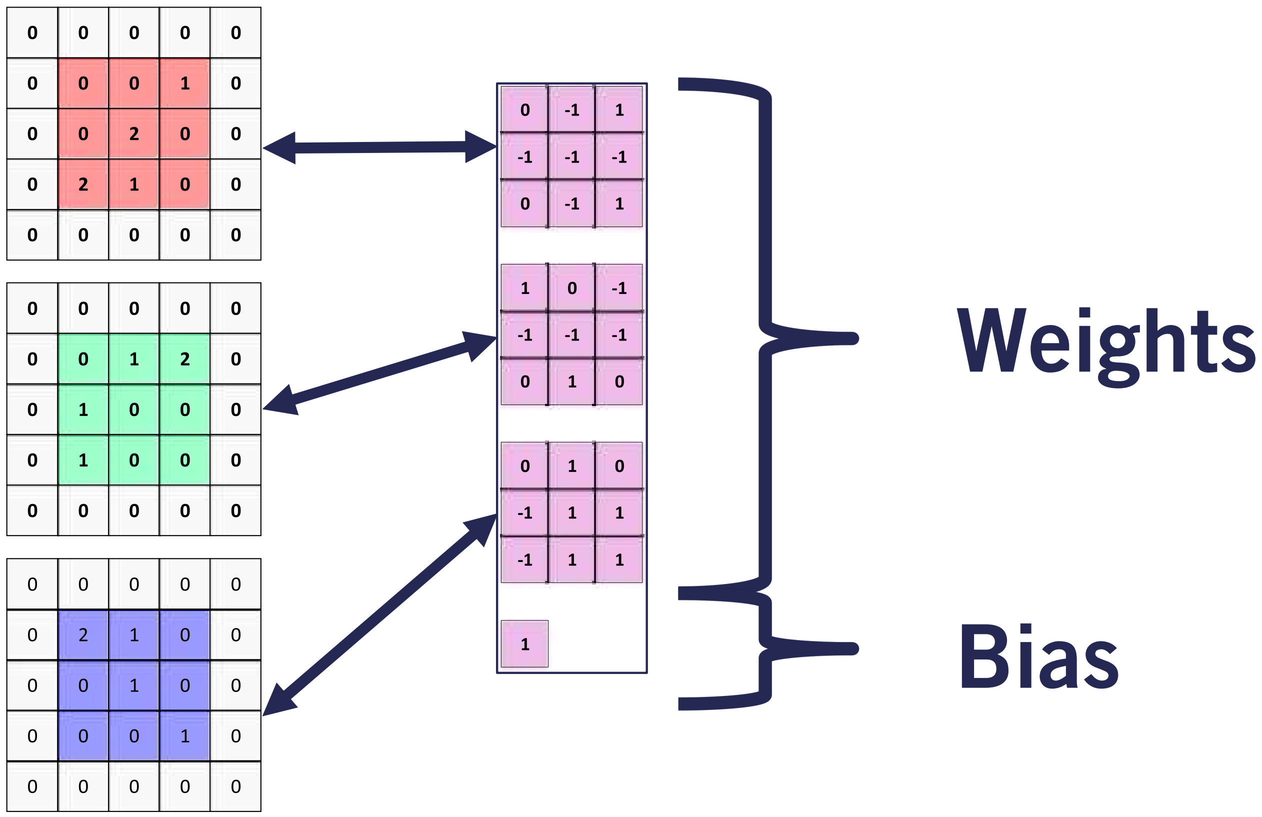
|  |  |  |  |  |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 2 | 0 |
| 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 |

|  |  |  |  |  |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 2 | 1 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 0 | 0 | 0 |

- **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



|  |  |  |  |  |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 2 | 0 | 0 |
| 0 | 2 | 1 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 |

|  |  |  |  |  |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 2 | 0 |
| 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 |

|  |  |  |  |  |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 2 | 1 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 0 | 0 | 0 |

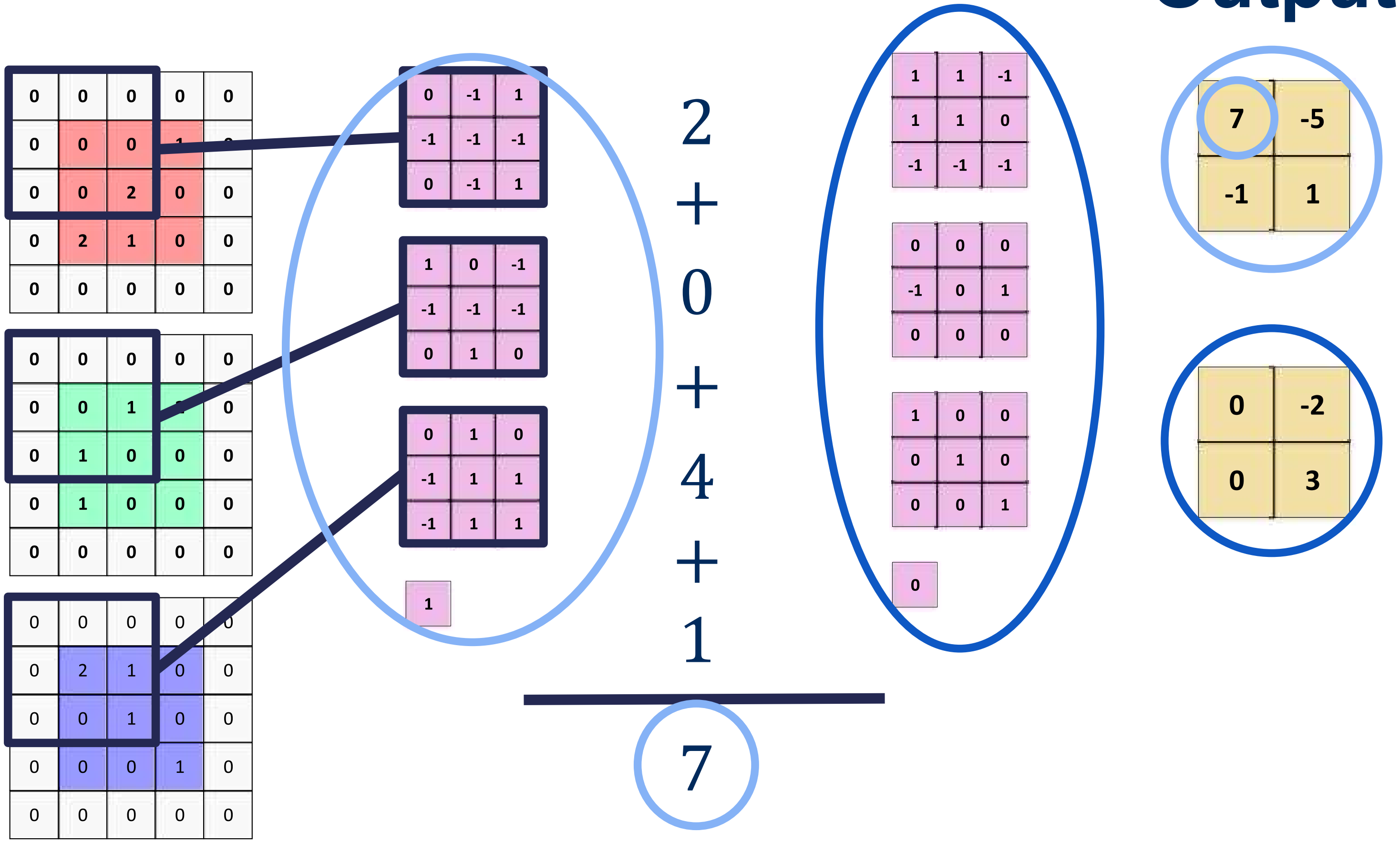
|  |  |  |
|---|---|---|
| 0 | -1 | 1 |
| -1 | -1 | -1 |
| 0 | -1 | 1 |

|  |  |  |
|---|---|---|
| 1 | 0 | -1 |
| -1 | -1 | -1 |
| 0 | 1 | 0 |

|  |  |  |
|---|---|---|
| 0 | 1 | 0 |
| -1 | 1 | 1 |
| -1 | 1 | 1 |

|  |
|---|
| 1 |

# Cross Correlation



|   |   |   |   |   |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 2 | 0 | 0 |
| 0 | 2 | 1 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 |

|   |   |   |   |   |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 2 | 0 |
| 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 |

|   |   |   |   |   |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 2 | 1 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 0 | 0 | 0 |

|   |   |   |
|---|---|---|
| 0 | -1 | 1 |
| -1 | -1 | -1 |
| 0 | -1 | 1 |

|   |   |   |
|---|---|---|
| 1 | 0 | -1 |
| -1 | -1 | -1 |
| 0 | 1 | 0 |

|   |   |   |
|---|---|---|
| 0 | 1 | 0 |
| -1 | 1 | 1 |
| -1 | 1 | 1 |

1

$$
\begin{array}{r}
2 \\
+\ 0 \\
+\ 4 \\
+\ 1 \\
\hline
7
\end{array}
$$

|   |   |   |
|---|---|---|
| 1 | 1 | -1 |
| 1 | 1 | 0 |
| -1 | -1 | -1 |

|   |   |   |
|---|---|---|
| 0 | 0 | 0 |
| -1 | 0 | 1 |
| 0 | 0 | 0 |

|   |   |   |
|---|---|---|
| 1 | 0 | 0 |
| 0 | 1 | 0 |
| 0 | 0 | 1 |

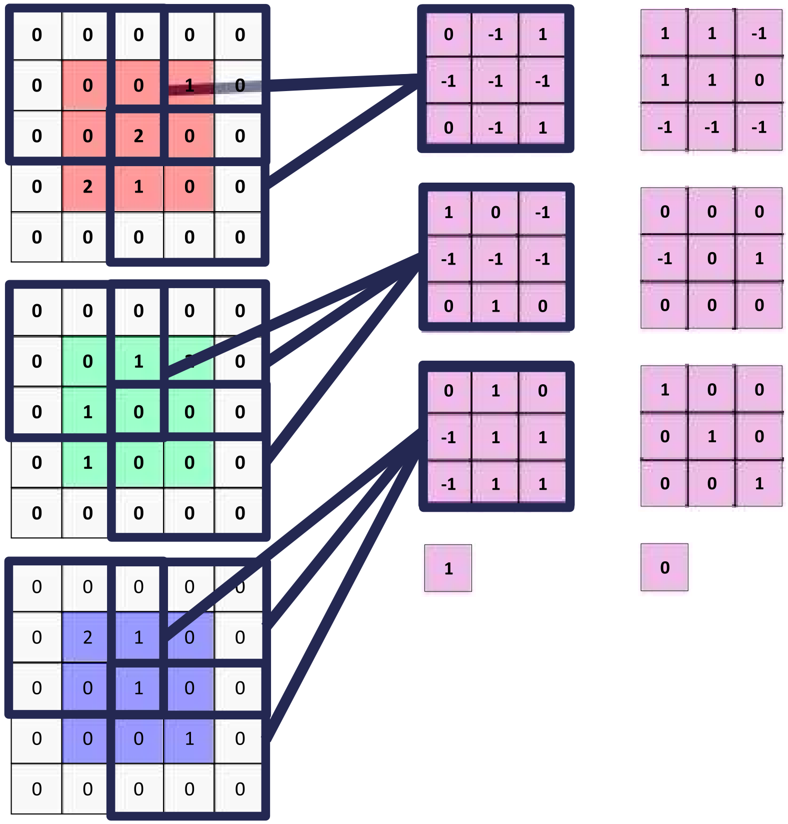
0

## Output

|   |   |
|---|---|
| 7 | -5 |
| -1 | 1 |

|   |   |
|---|---|
| 0 | -2 |
| 0 | 3 |

# Cross Correlation



| | | | | |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 2 | 0 | 0 |
| 0 | 2 | 1 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 |

| | | | | |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 |

| | | | | |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 2 | 1 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 0 | 0 | 0 |

| | | |
|---|---|---|
| 0 | -1 | 1 |
| -1 | -1 | -1 |
| 0 | -1 | 1 |

| | | |
|---|---|---|
| 1 | 0 | -1 |
| -1 | -1 | -1 |
| 0 | 1 | 0 |

| | | |
|---|---|---|
| 0 | 1 | 0 |
| -1 | 1 | 1 |
| -1 | 1 | 1 |

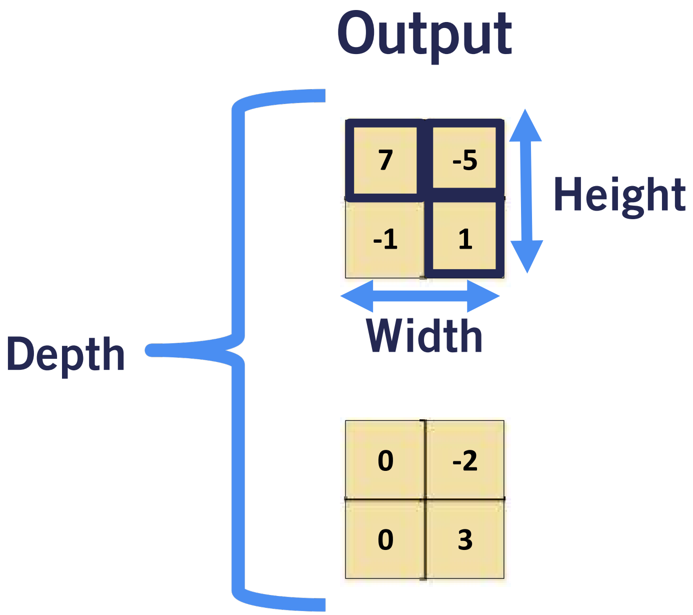
1

| | | |
|---|---|---|
| 1 | 1 | -1 |
| 1 | 1 | 0 |
| -1 | -1 | -1 |

| | | |
|---|---|---|
| 0 | 0 | 0 |
| -1 | 0 | 1 |
| 0 | 0 | 0 |

| | | |
|---|---|---|
| 1 | 0 | 0 |
| 0 | 1 | 0 |
| 0 | 0 | 1 |

0



Output

Height

Width

Depth

| | |
|---|---|
| 7 | -5 |
| -1 | 1 |

| | |
|---|---|
| 0 | -2 |
| 0 | 3 |

- **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

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

| | | | |
|---|---|---|---|
| 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 $\boldsymbol{n \times n}$
- Stride = $\boldsymbol{S}$

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

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

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

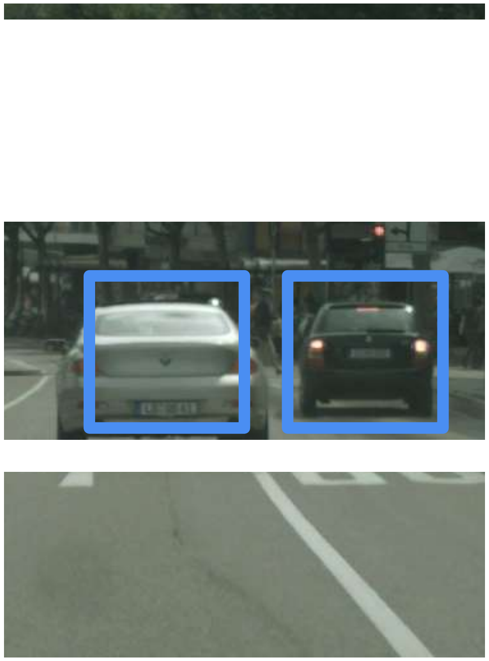
# Pooling Layers: Max Pooling

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

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

| | |
|---|---|
| 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