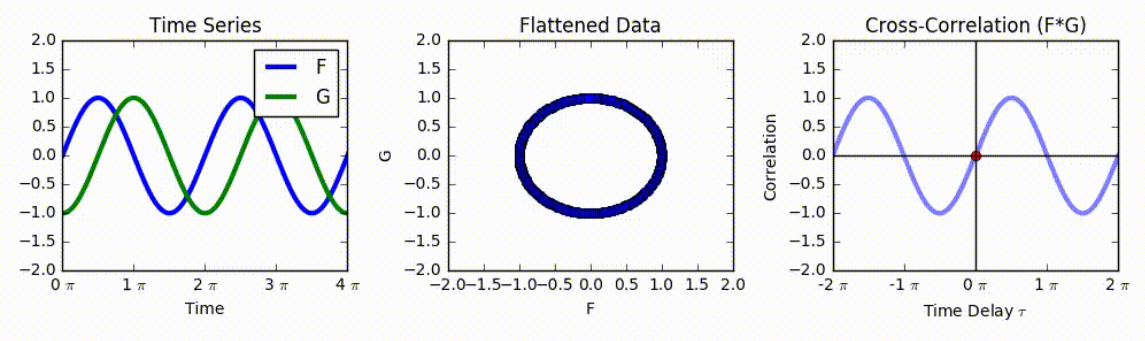
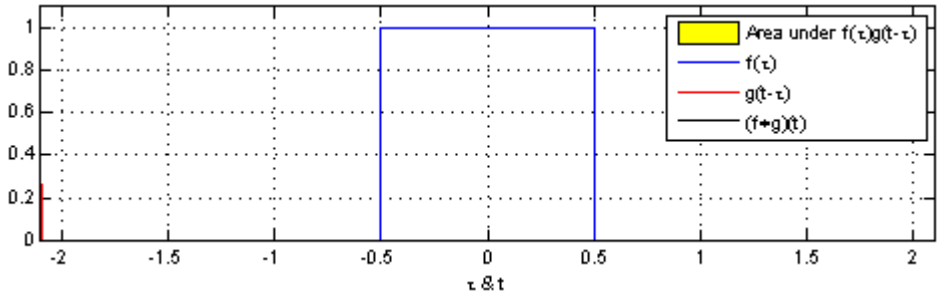
# Introduction to Convolutional Neural Networks

Dai Bui

# Intuition of Convolution

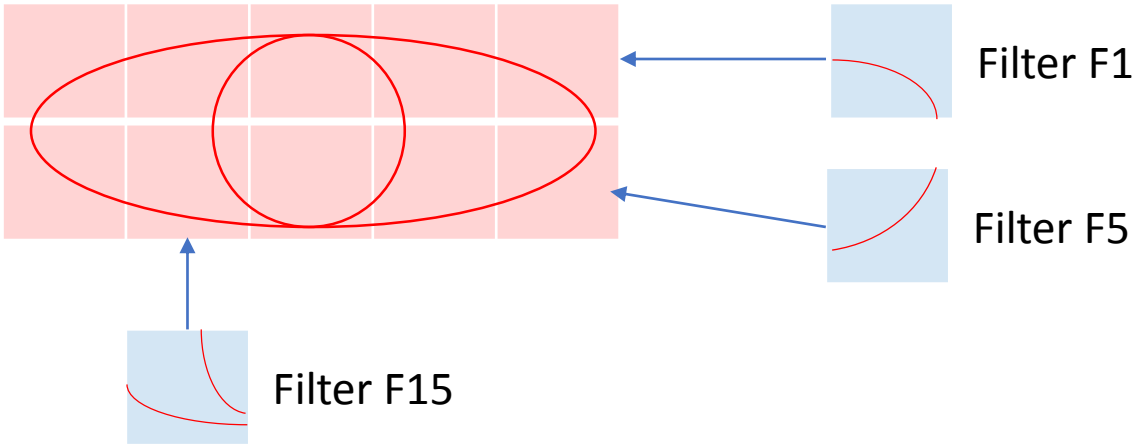
- Convolution, similar to cross correlation of two signals is high when they look similar



# Intuition for CNN for Pattern Recognition

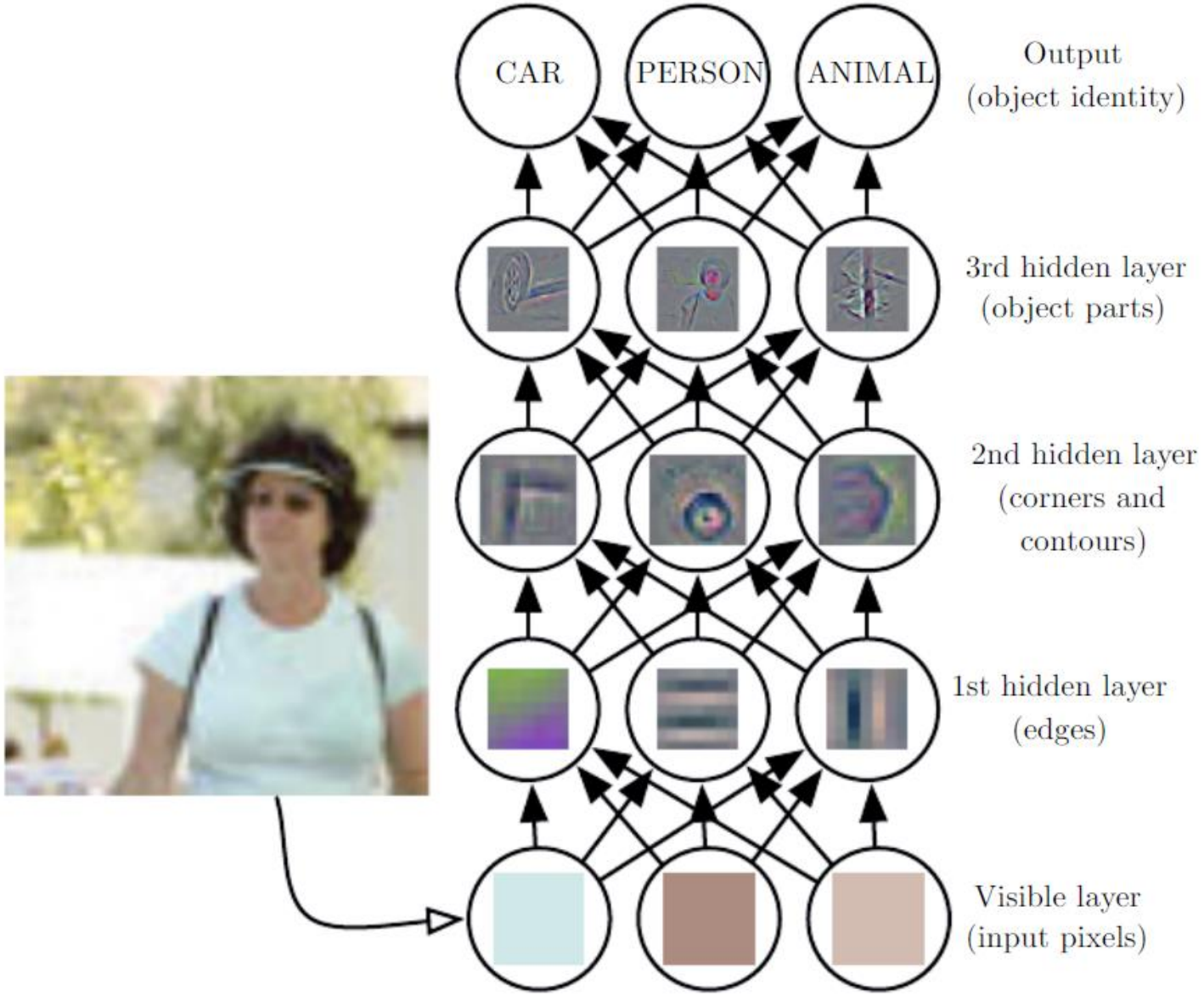
- How do we recognize such “eye”?
- We can divide the image into smaller pieces
- We will try to match (convolute) each piece with a pattern, called filter using convolution




| | | | | |
|---|---|---|---|---|
| **F23** | **F78** | **F45** | **F22** | **F1** |
| **F19** | **F15** | **F16** | **F14** | **F5** |

If we see this pattern -> This is an eye!

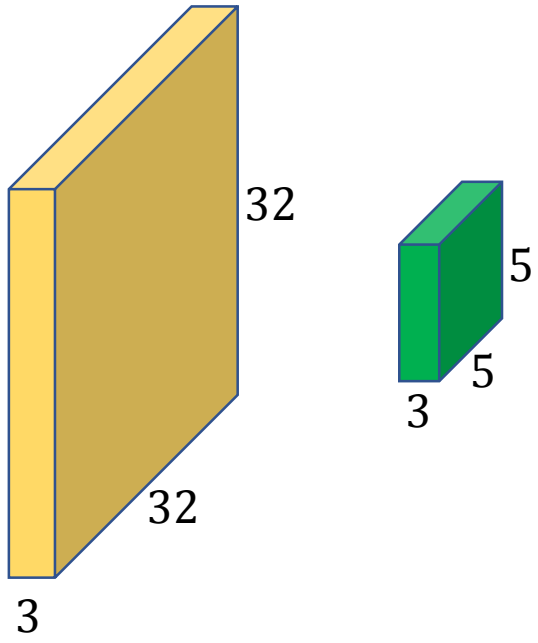
# ntuition for Pattern Recognition Using NN



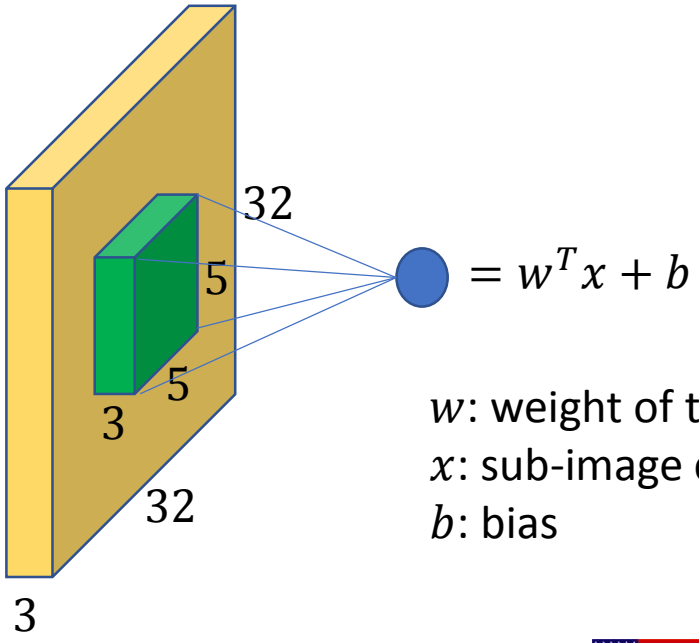

- If we want to detect if this is a person:
  - Detect the edges
  - From those edges, we form corners and complicated contours
  - Those corners and contours are combined to form objects, e.g., eyes, noses
  - Those objects then form a bigger objects, e.g, heads
  - Those bigger objects then form more bigger object, e.g., body
  - …
- Higher layers will combine smaller features from lower layers to recognize more complicated objects

# Convolution Layer

- 32x32x3 image is convolved with a 5x5x3 filter



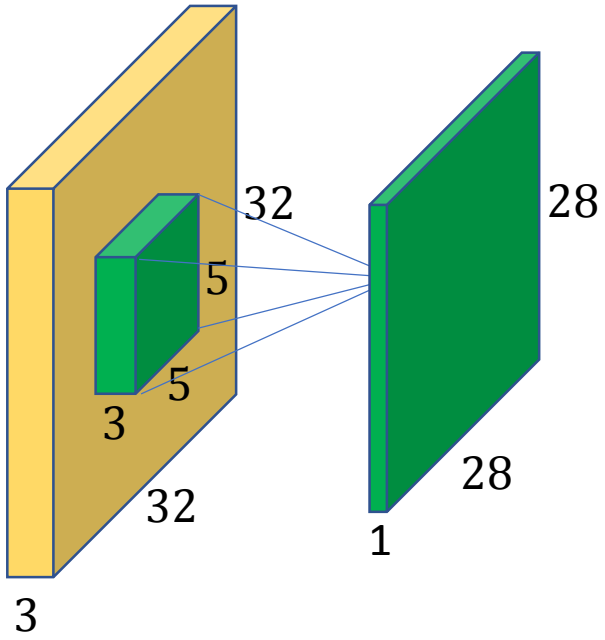

Convolution: Slide the filter over the image, at each point we compute the dot product of the filter with the region covered by the image




$w$: weight of the filter
$x$: sub-image covered by the filter
$b$: bias

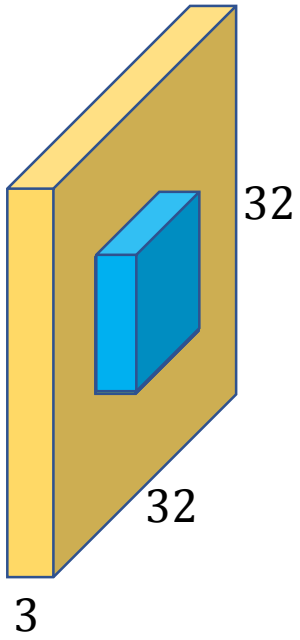
# Convolution Layer

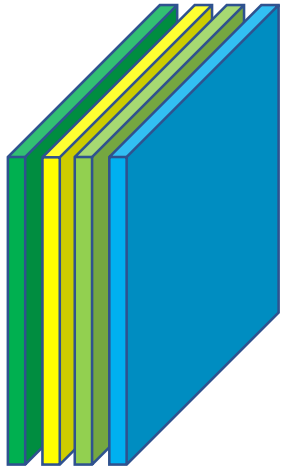
- Convolve (slide) the filter over the entire image, the resulting dot products at each location create a new matrix called activation map



# Convolution Layer

- Each filter creates one activation map

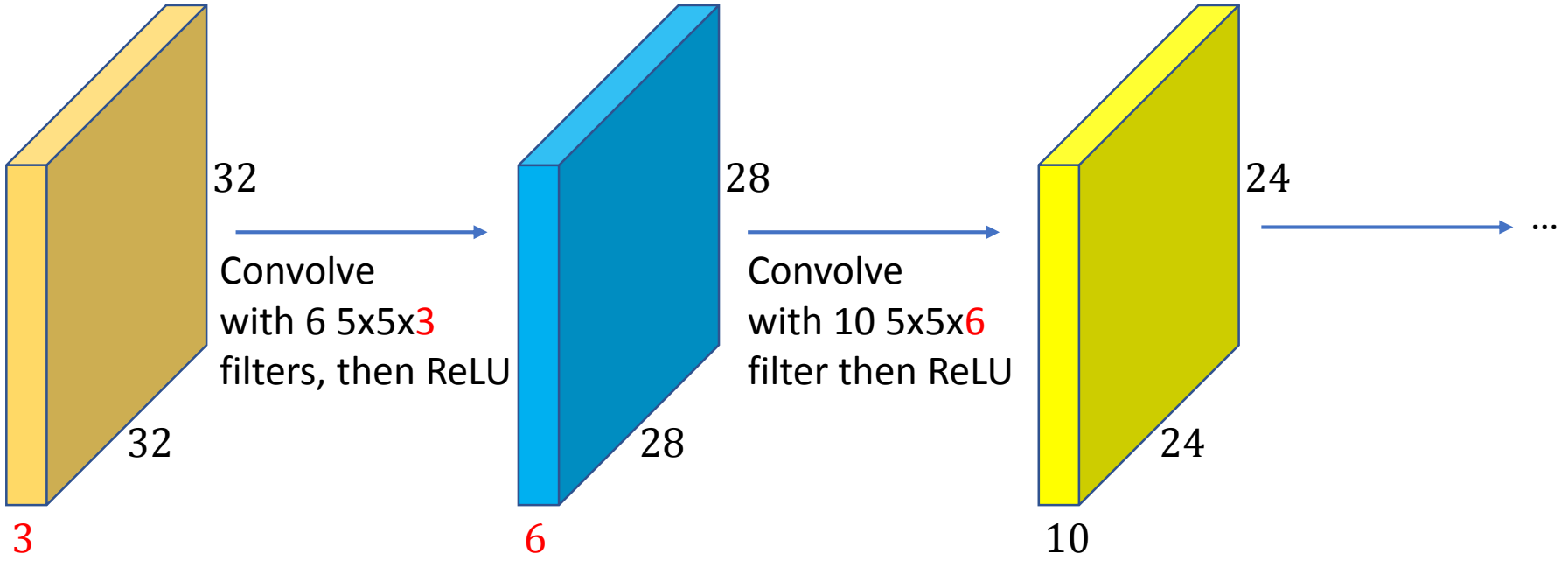


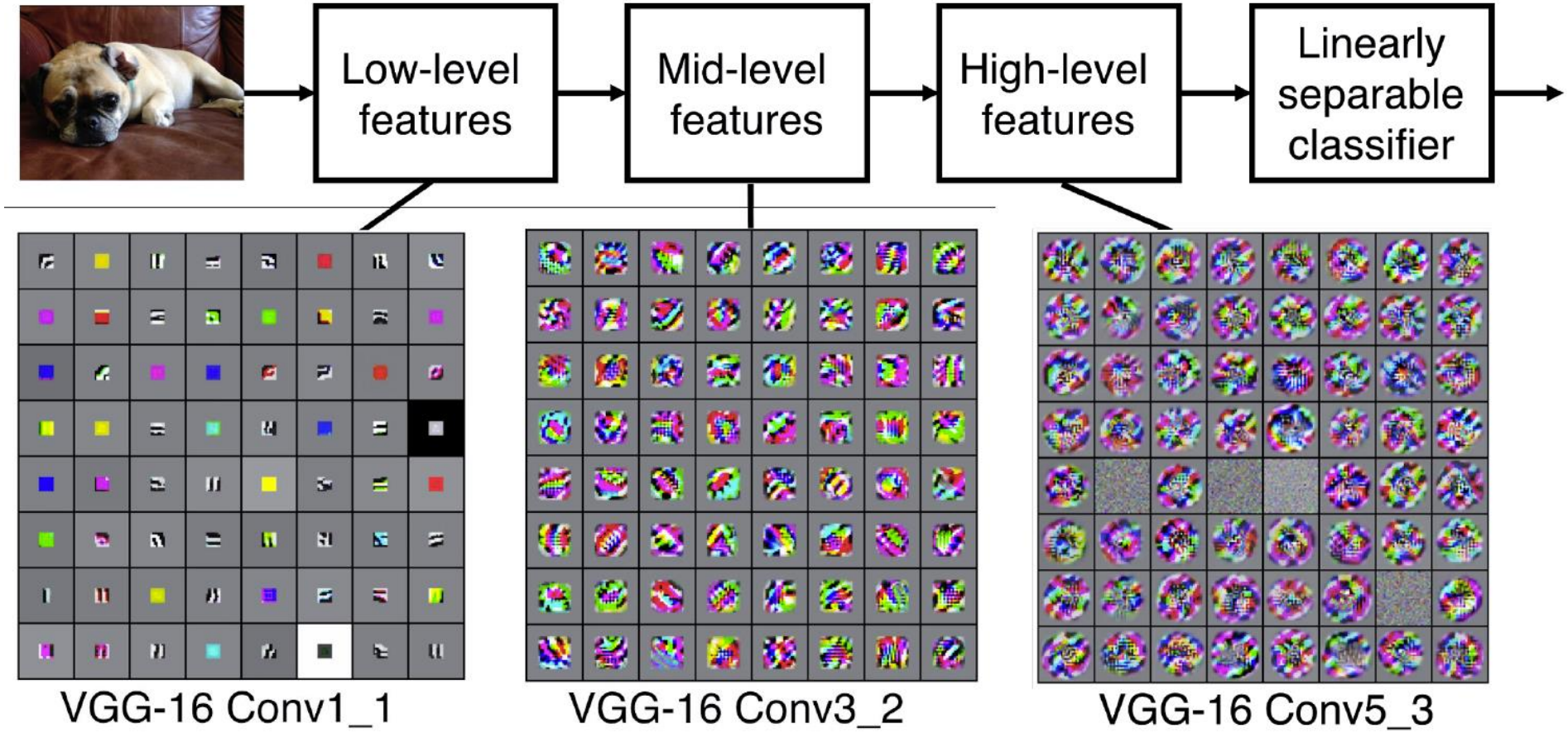
Newly formed activation maps are stacked up to create new image

# Example: ConvNet Architecture

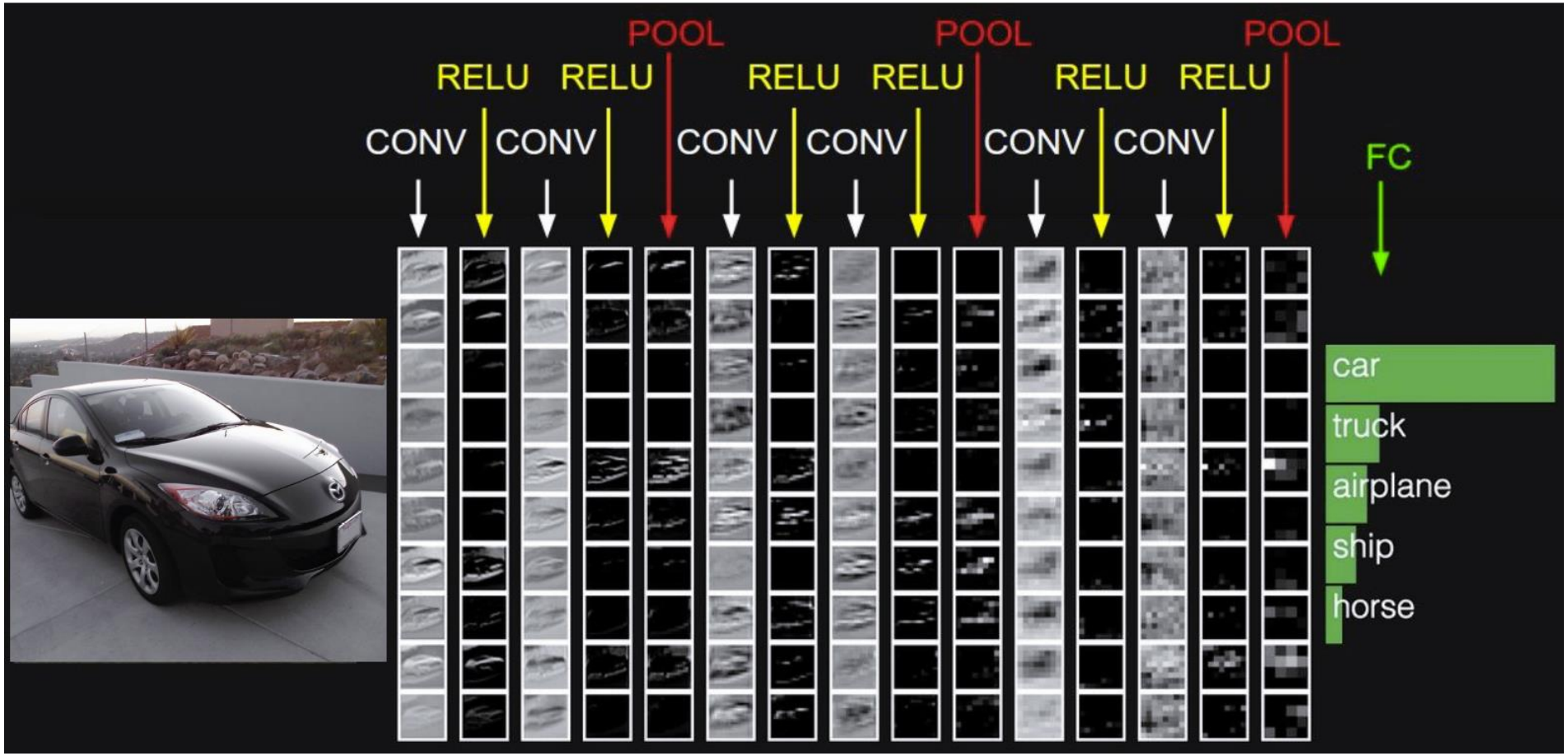
- A sequence of convolution layers



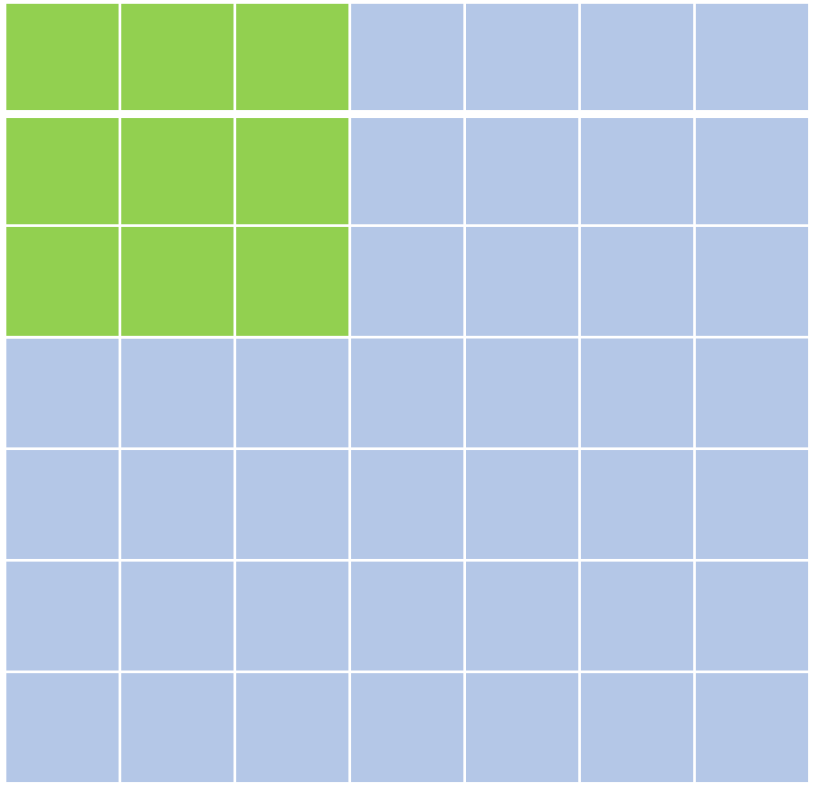
# VGG-16 Architecture



# Car Recognition Example

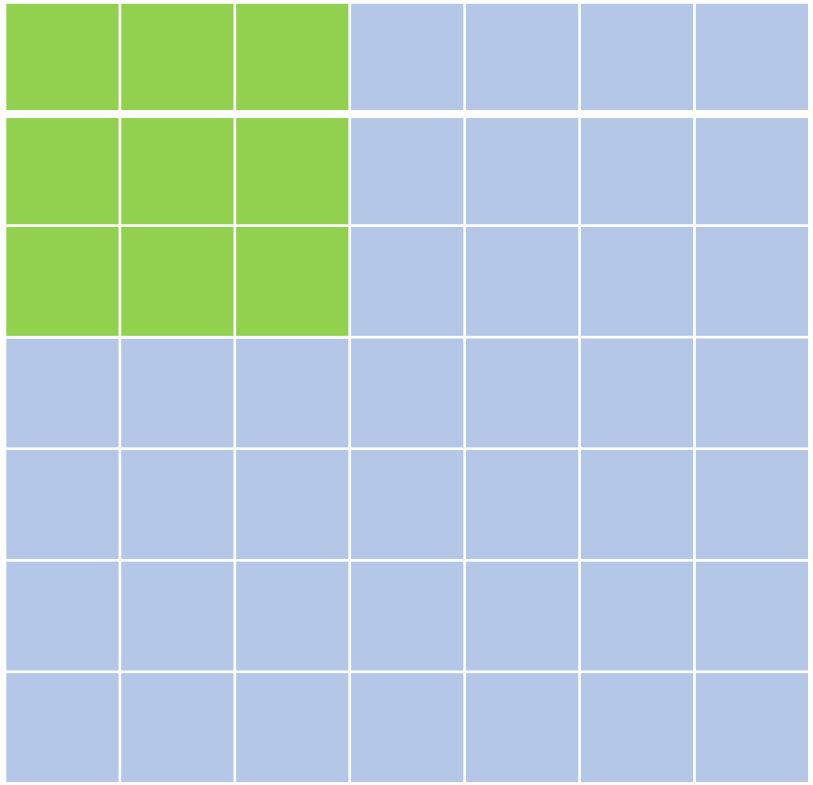


# Convolution in Details



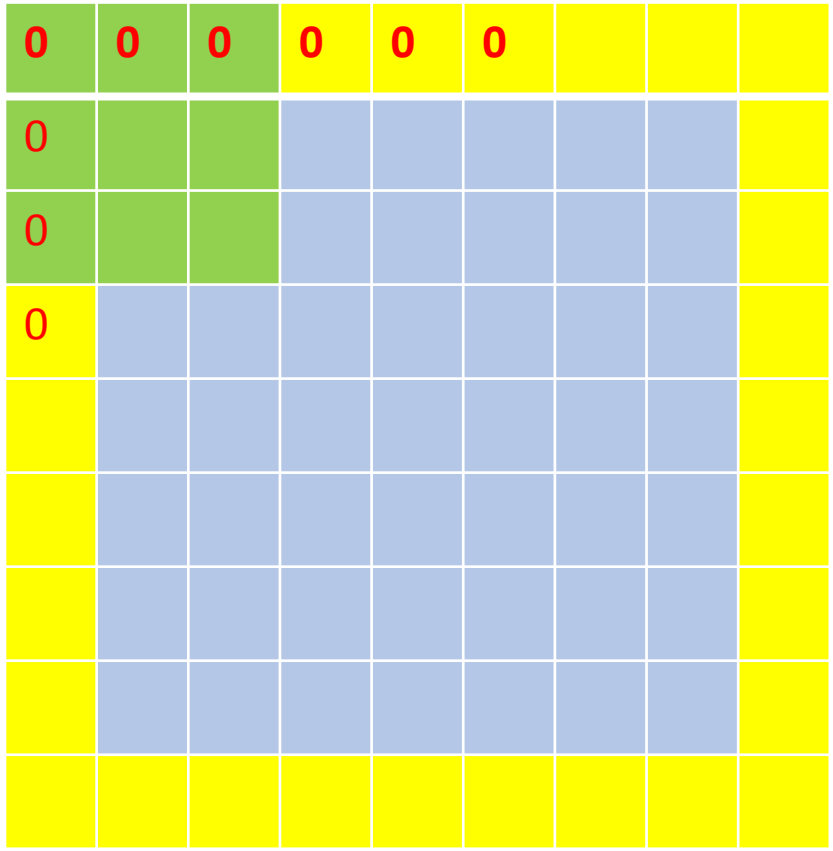
Convolution with stride 1

# Convolution in Details



Convolution with stride 2

# Padding




Insert 0 outside of the border of the image
7x7 image padded with one more pixel at the border
then convolved with one 3x3 filter will produce 7x7 output

# Convolution Example

Input Volume (+pad 1) (7x7x3)

x[:,:,0]

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

x[:,:,1]

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

x[:,:,2]

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

Filter W0 (3x3x3)

w0[:,:,0]

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

w0[:,:,1]

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

w0[:,:,2]

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

Bias b0 (1x1x1)

b0[:,:,0]

| |
|---|
| 1 |

Filter W1 (3x3x3)

w1[:,:,0]

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

w1[:,:,1]

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

w1[:,:,2]

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

Bias b1 (1x1x1)

b1[:,:,0]

| |
|---|
| 0 |

Output Volume (3x3x2)

o[:,:,0]

| | | |
|---|---|---|
| -1 | -2 | -7 |
| 5 | -4 | -6 |
| 1 | 2 | 1 |

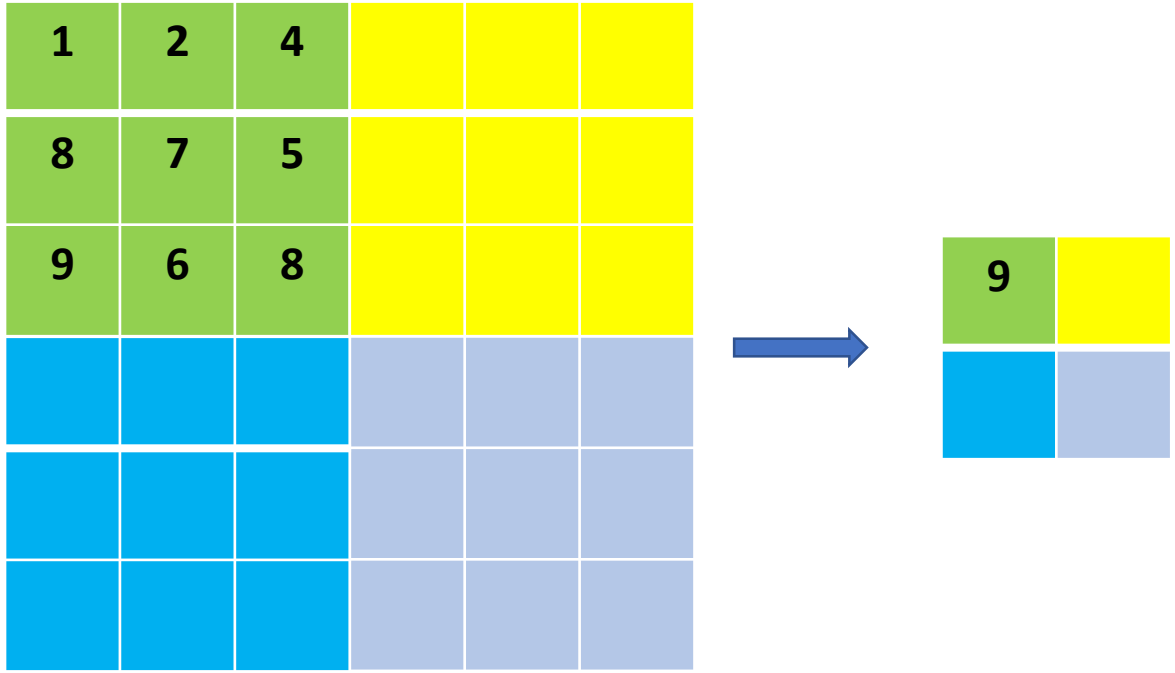
o[:,:,1]

| | | |
|---|---|---|
| -2 | 0 | 8 |
| -2 | -3 | -4 |
| -4 | -4 | 0 |

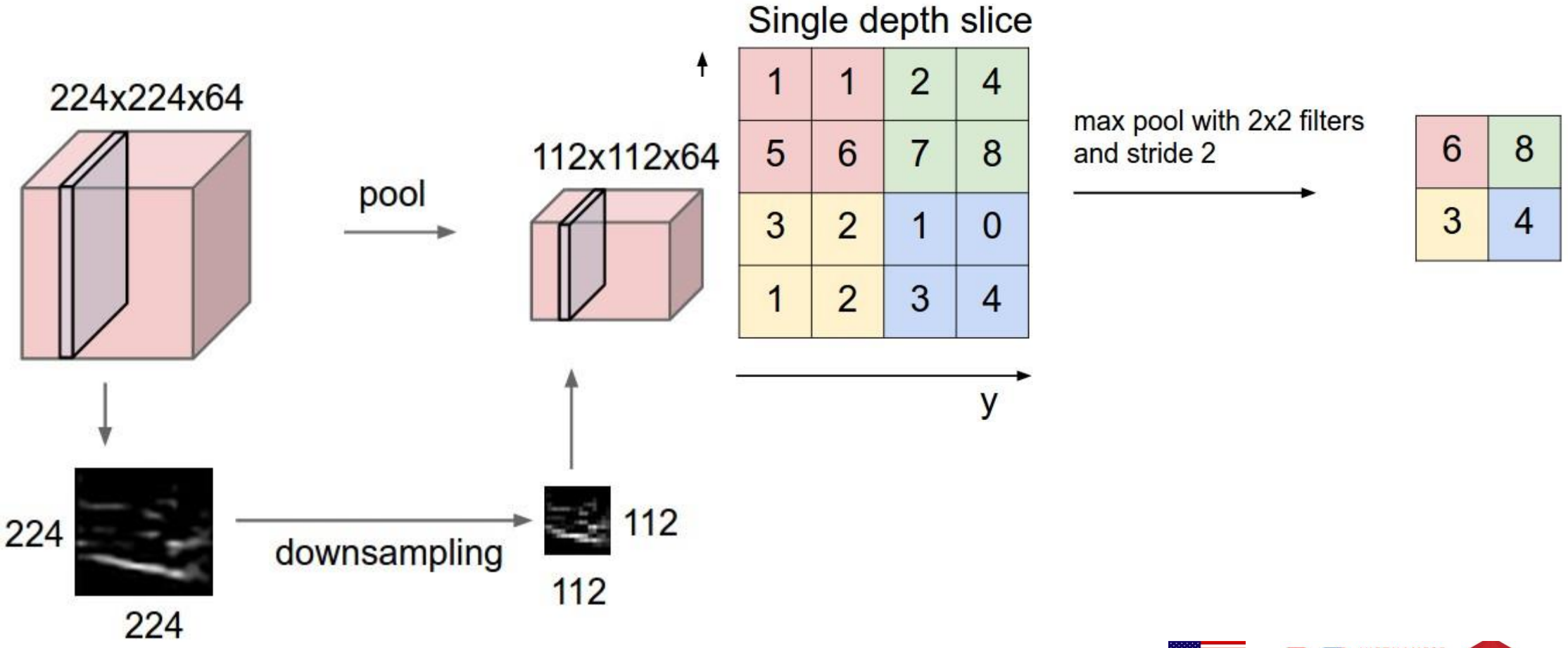
toggle movement

# Pooling

- What happens if we have too many layers and the activation maps are big?
  - We want to scale it down
- How do we scale it down?
  - For each submatrix, say 3x3 as in this case, we replace it with a single value
- Max pooling: submatrix is replace with its maximum value, in this case is 9
- Average pooling: submatrix is replace with its average value, in this case is 7



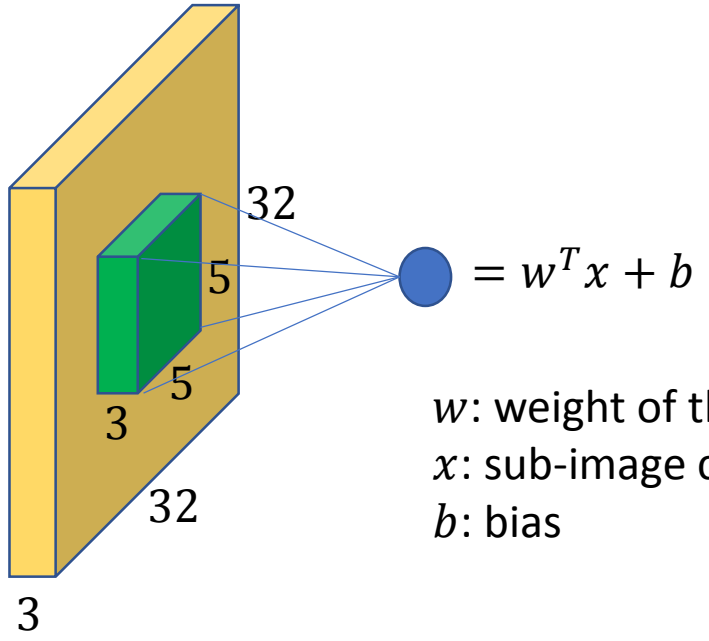
# Pooling


224x224x64
pool
112x112x64
Single depth slice
1 1 2 4
5 6 7 8
3 2 1 0
1 2 3 4
y
max pool with 2x2 filters
and stride 2
6 8
3 4
224
224
downsampling
112
112

# Convolution Questions

- What happens if we expand $w$ so that it has the same size as the image? (effectively, we use a fully connected network for convolution)
  - It is very expensive to do the computation when the image is large
- Should we expand $w$ so that it has the same size as the image?
  - Perhaps it is not necessary because image pixels are more related to their nearby pixels

Convolution: Slide the filter over the image, at each point we compute the dot product of the filter with the region covered by the image




$w$: weight of the filter
$x$: sub-image covered by the filter
$b$: bias

# CNN Practice