# CS231n Lecture 5

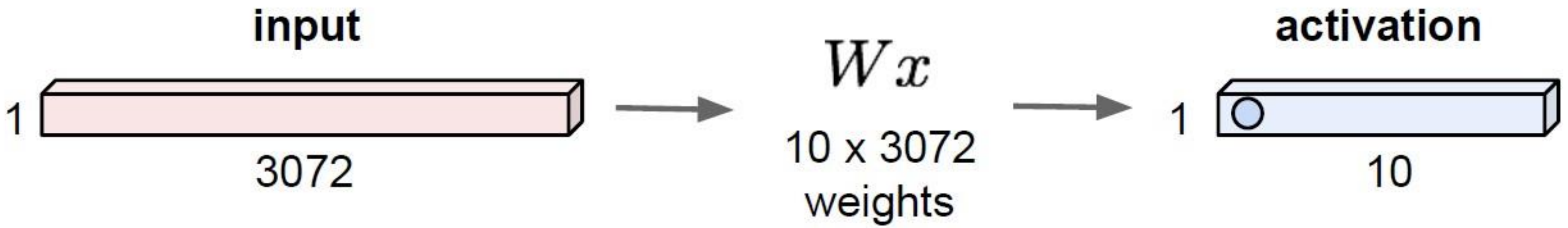
- BOAZ 10기 박성현
- BOAZ 11기 김태희
- BOAZ 11기 홍지민

BOAZ

01

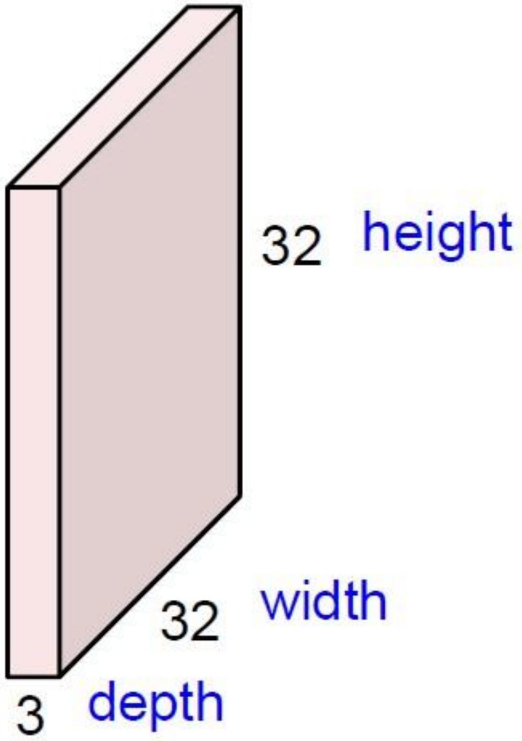
# Fully Connected Layer

32x32x3 image -> stretch to 3072 x 1

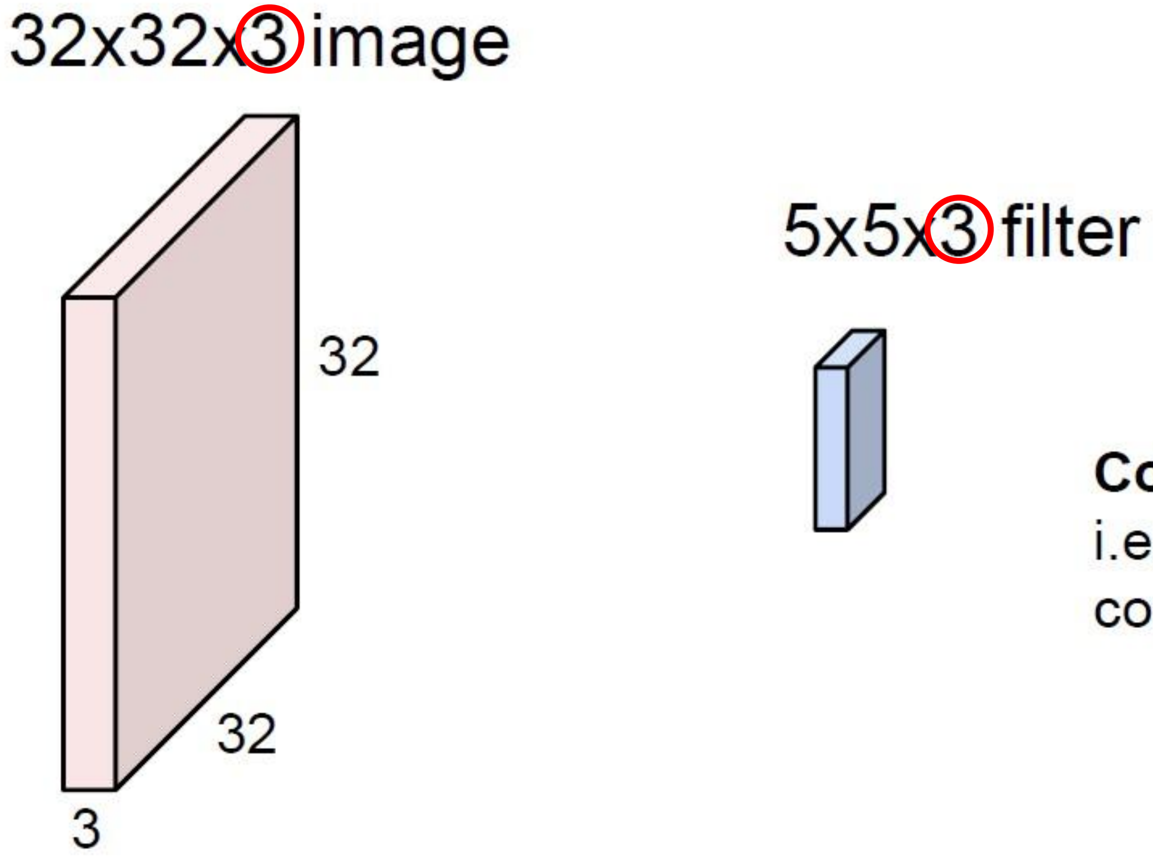


# 02 Convolution Layer

32x32x3 image -> preserve spatial structure

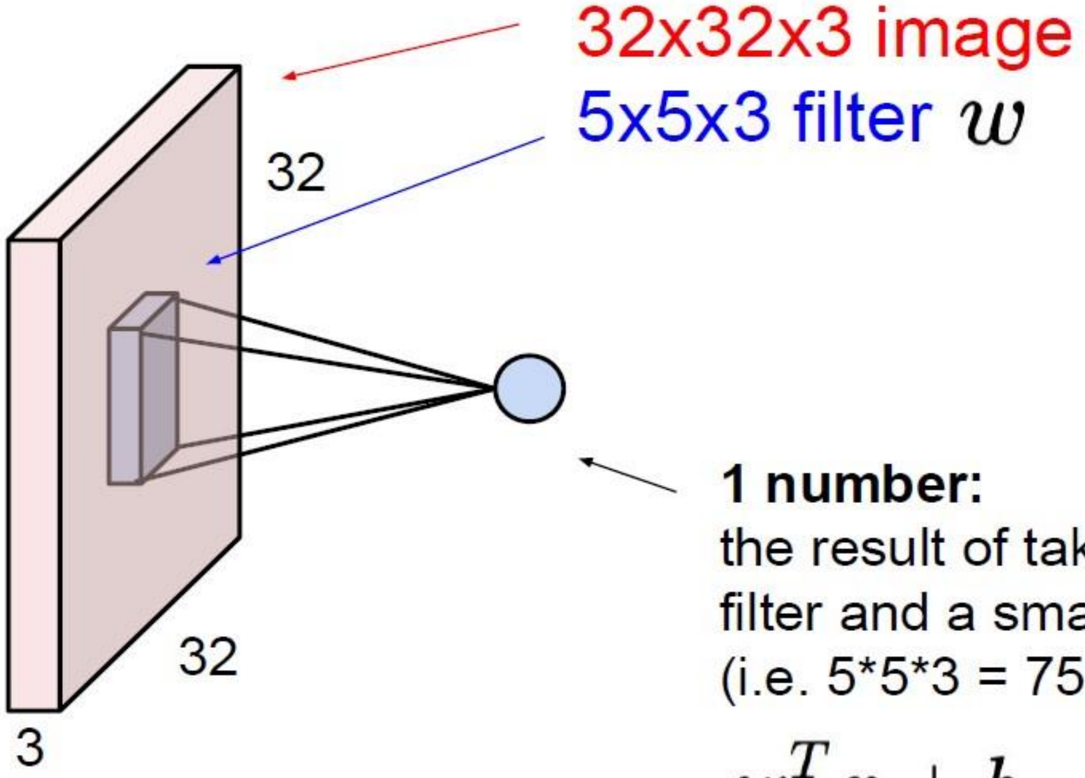


# Dot product & filter size



**Convolve** the filter with the image
i.e. “slide over the image spatially, computing dot products”

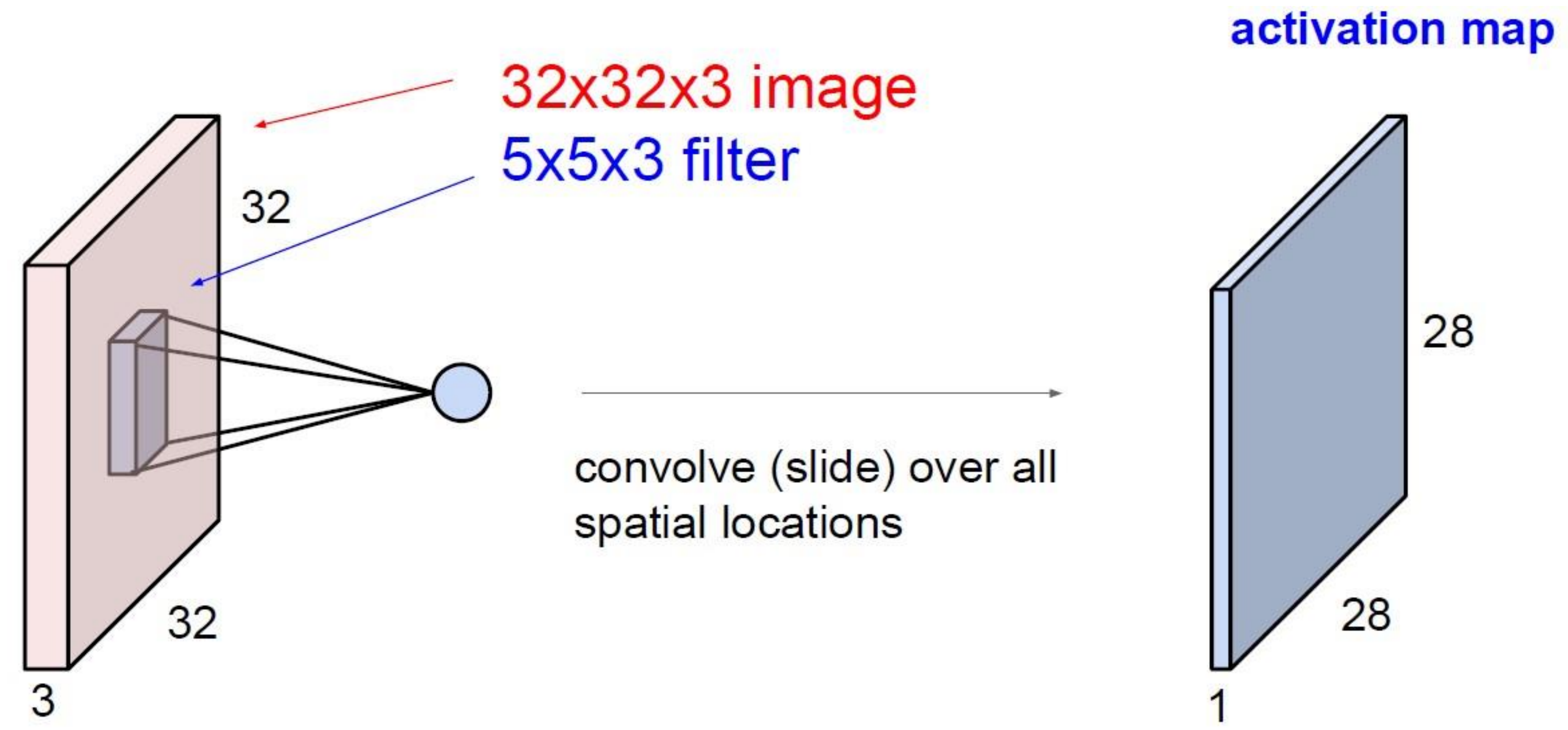
# 02 Convolution Layer



**1 number:**
the result of taking a dot product between the filter and a small 5x5x3 chunk of the image
(i.e. 5*5*3 = 75-dimensional dot product + bias)

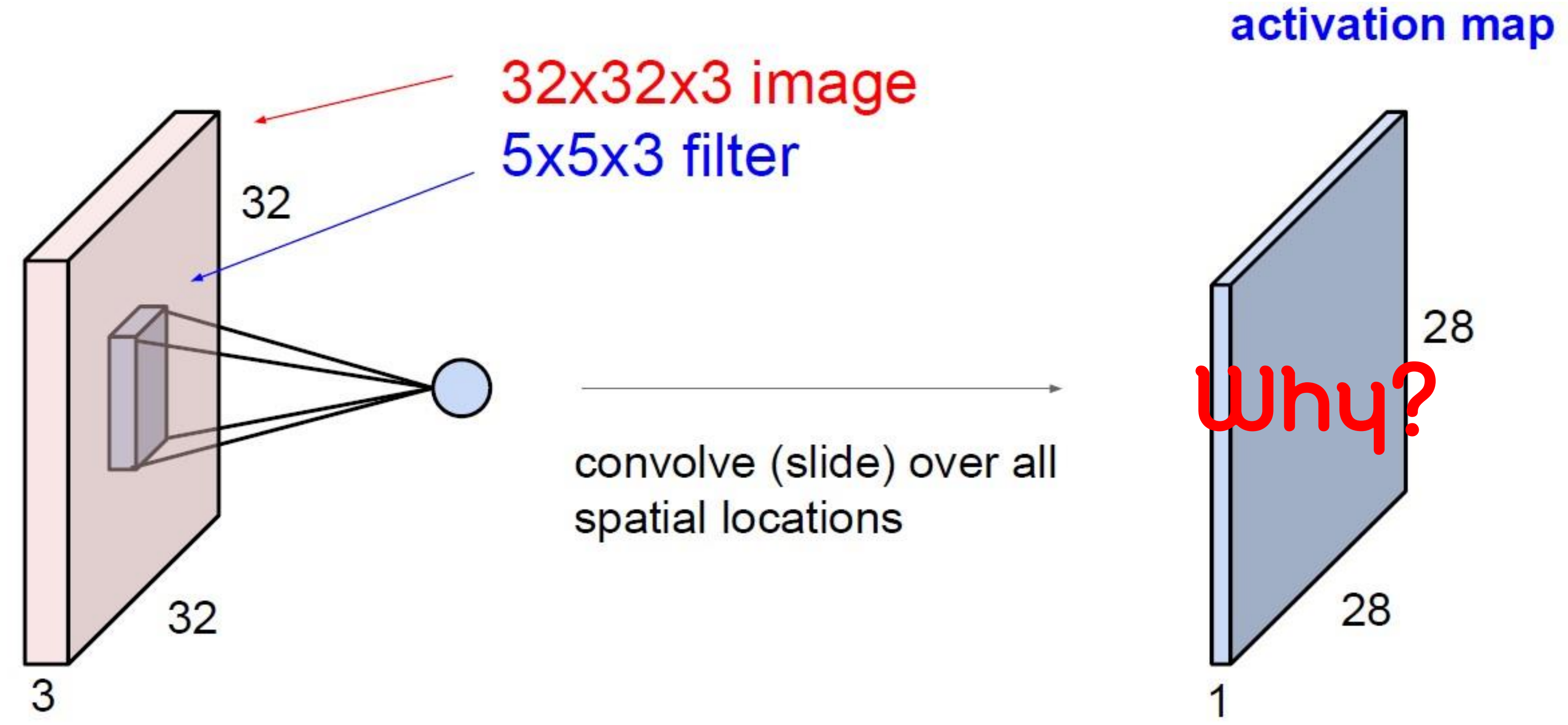
$$w^T x + b$$

# Convolution Layer



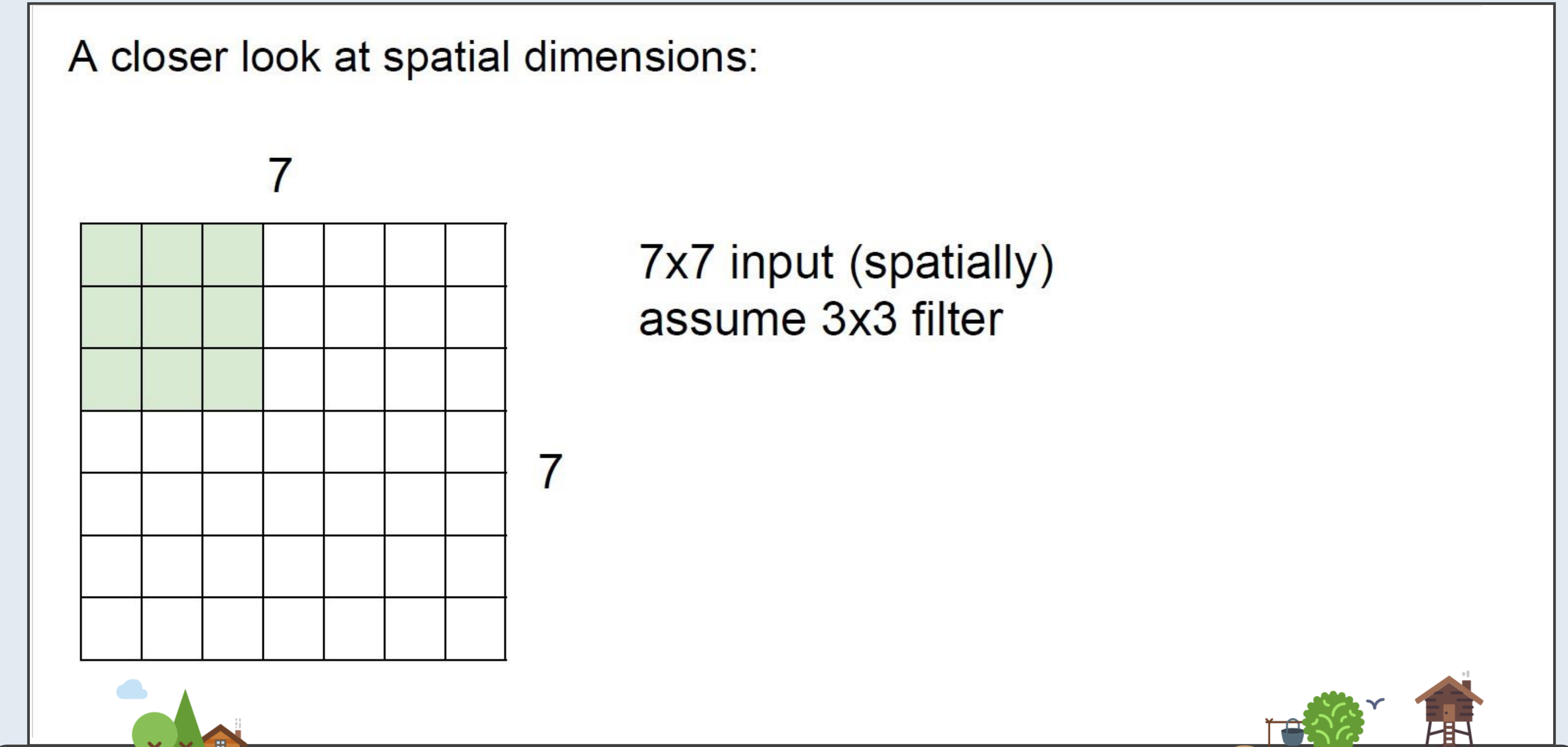
02

# Convolution Layer



03

# Filter computation



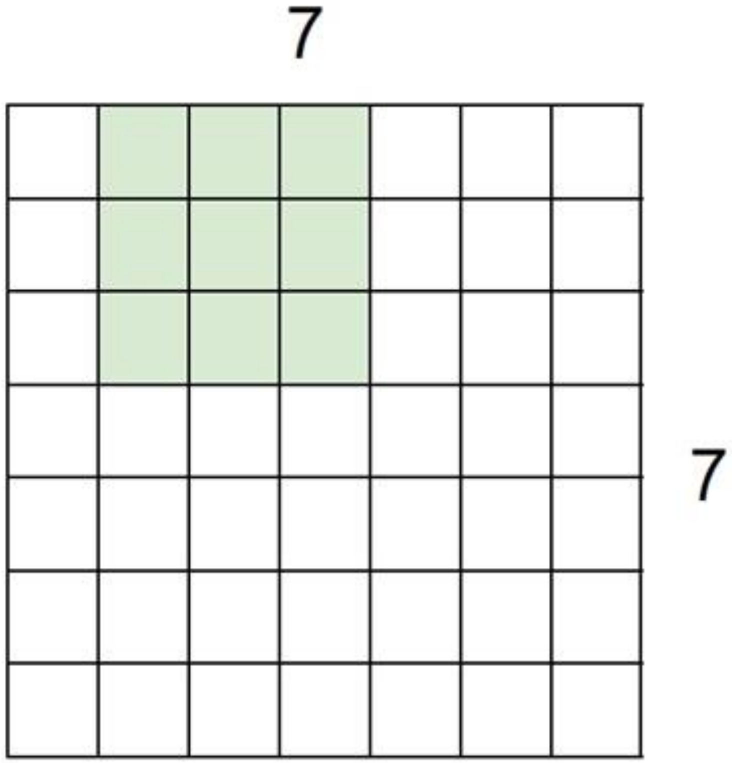
A closer look at spatial dimensions:

7x7 input (spatially)
assume 3x3 filter

03

# Filter computation

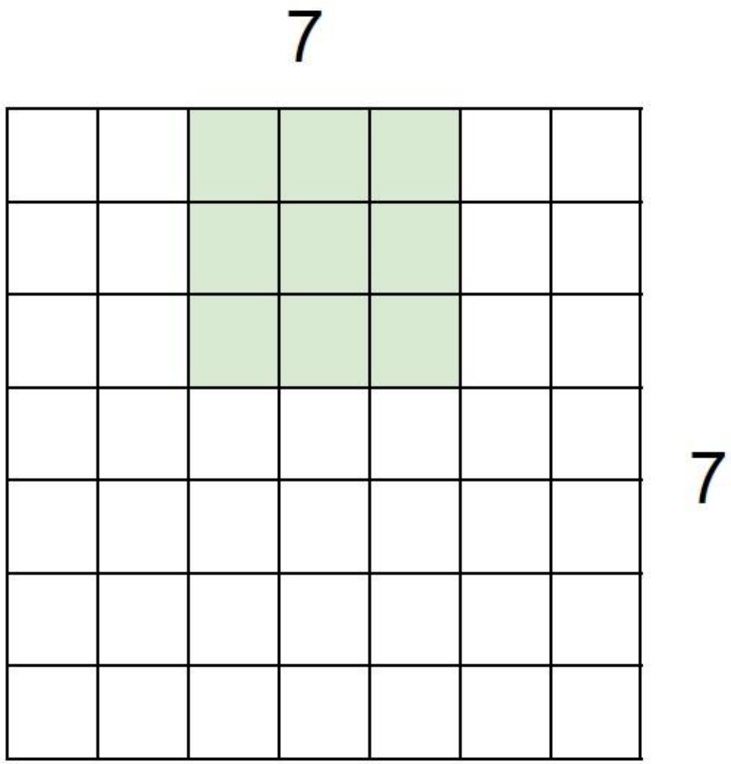
A closer look at spatial dimensions:



7x7 input (spatially)
assume 3x3 filter

# Filter computation

A closer look at spatial dimensions:

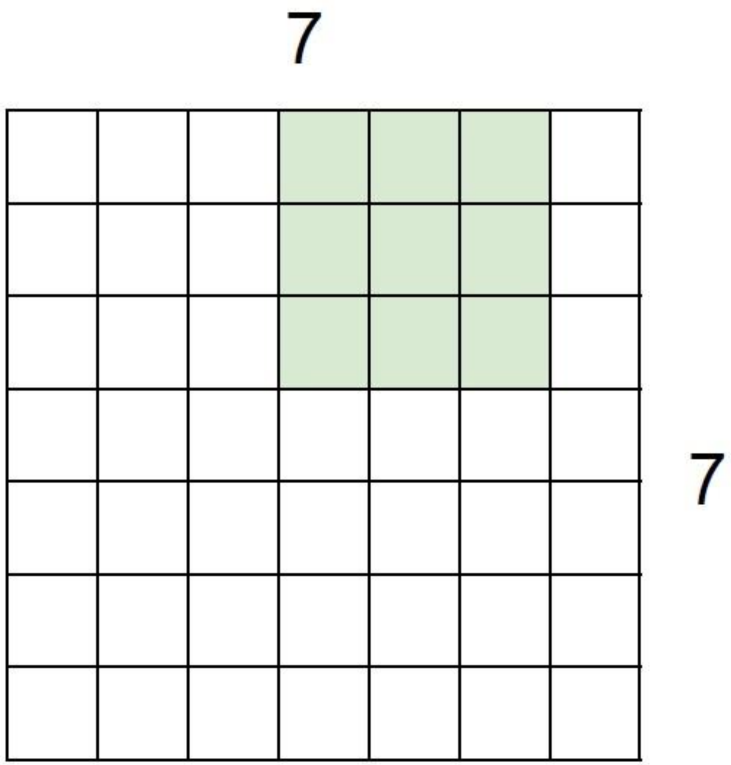


7x7 input (spatially)
assume 3x3 filter

03

# Filter computation

A closer look at spatial dimensions:

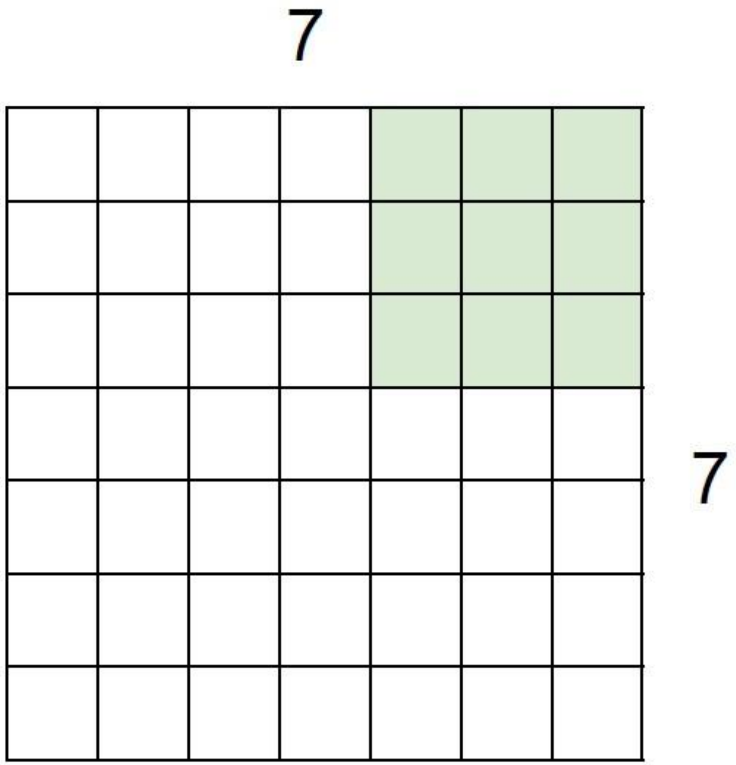


7x7 input (spatially)
assume 3x3 filter

03

# Filter computation

A closer look at spatial dimensions:

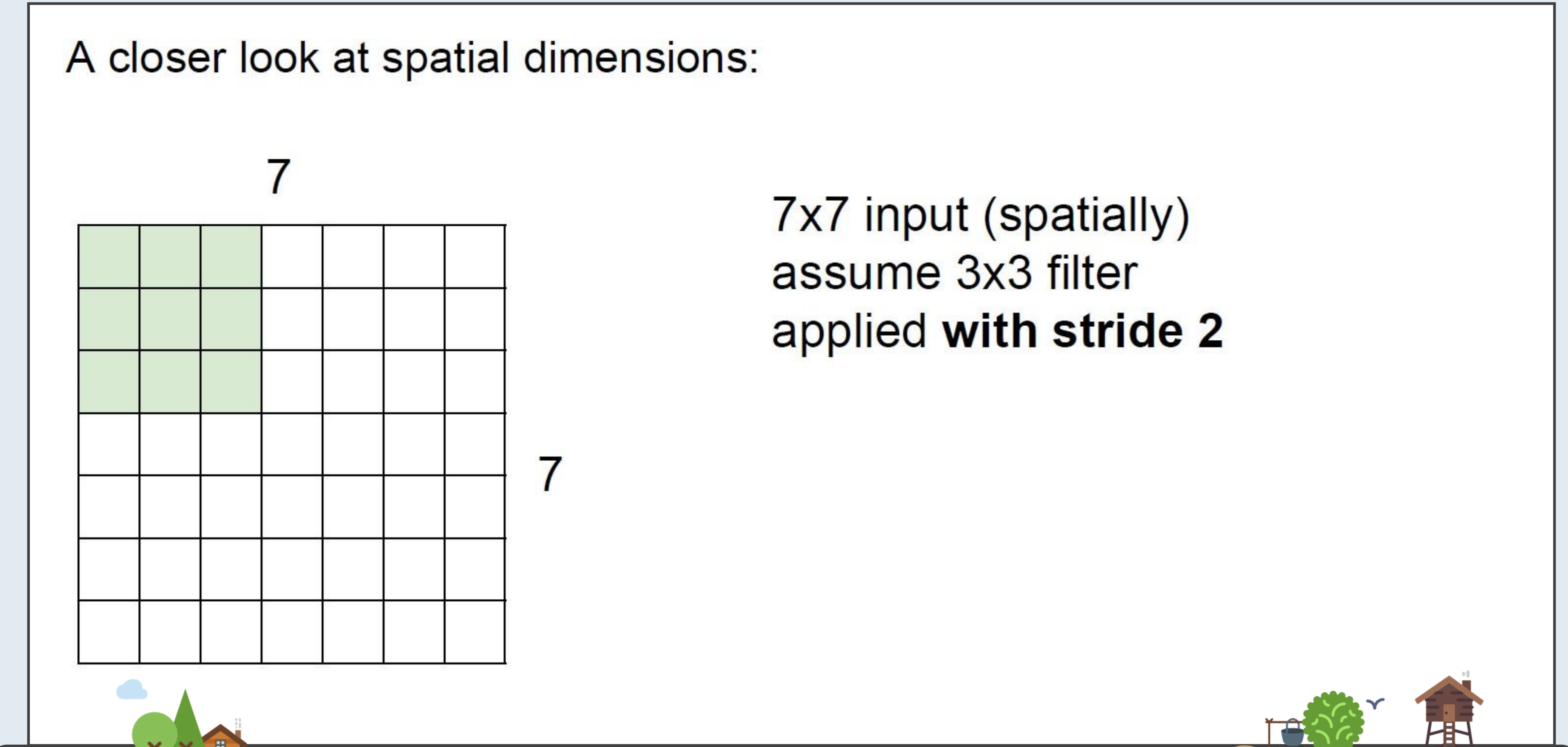


7x7 input (spatially)
assume 3x3 filter

**=> 5x5 output**

03

# Filter computation



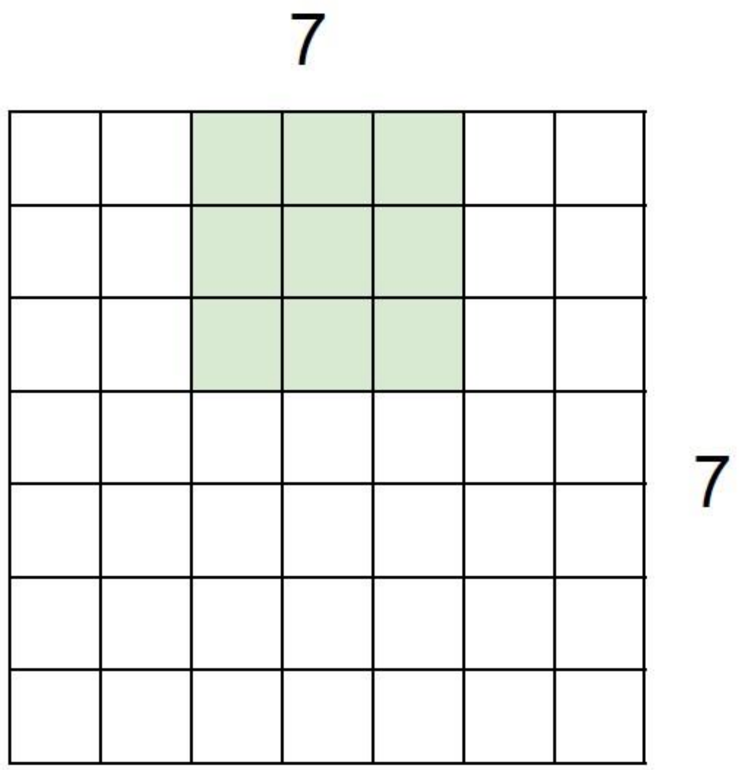
A closer look at spatial dimensions:

7x7 input (spatially)
assume 3x3 filter
applied **with stride 2**

03

# Filter computation

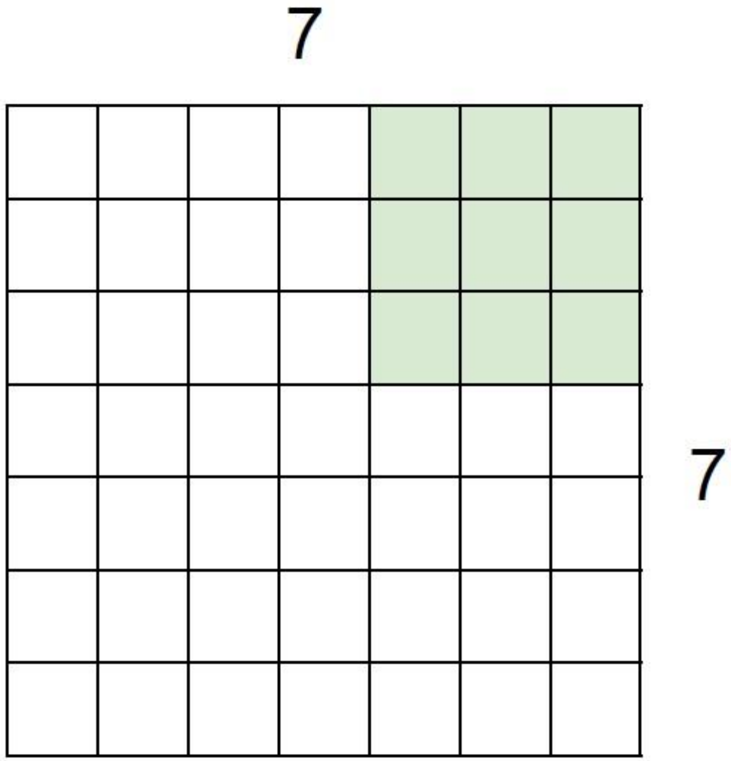
A closer look at spatial dimensions:



7x7 input (spatially)
assume 3x3 filter
applied **with stride 2**

# Filter computation

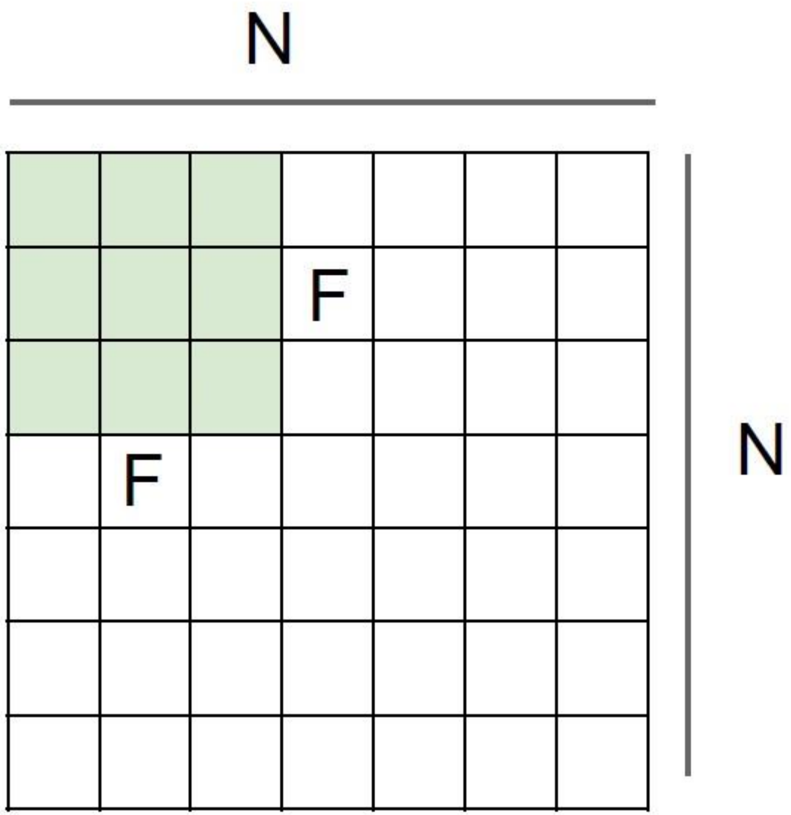
A closer look at spatial dimensions:



7x7 input (spatially)
assume 3x3 filter
applied **with stride 2**
**=> 3x3 output!**

03

# Filter computation



Output size:

**(N - F) / stride + 1**

e.g. N = 7, F = 3:

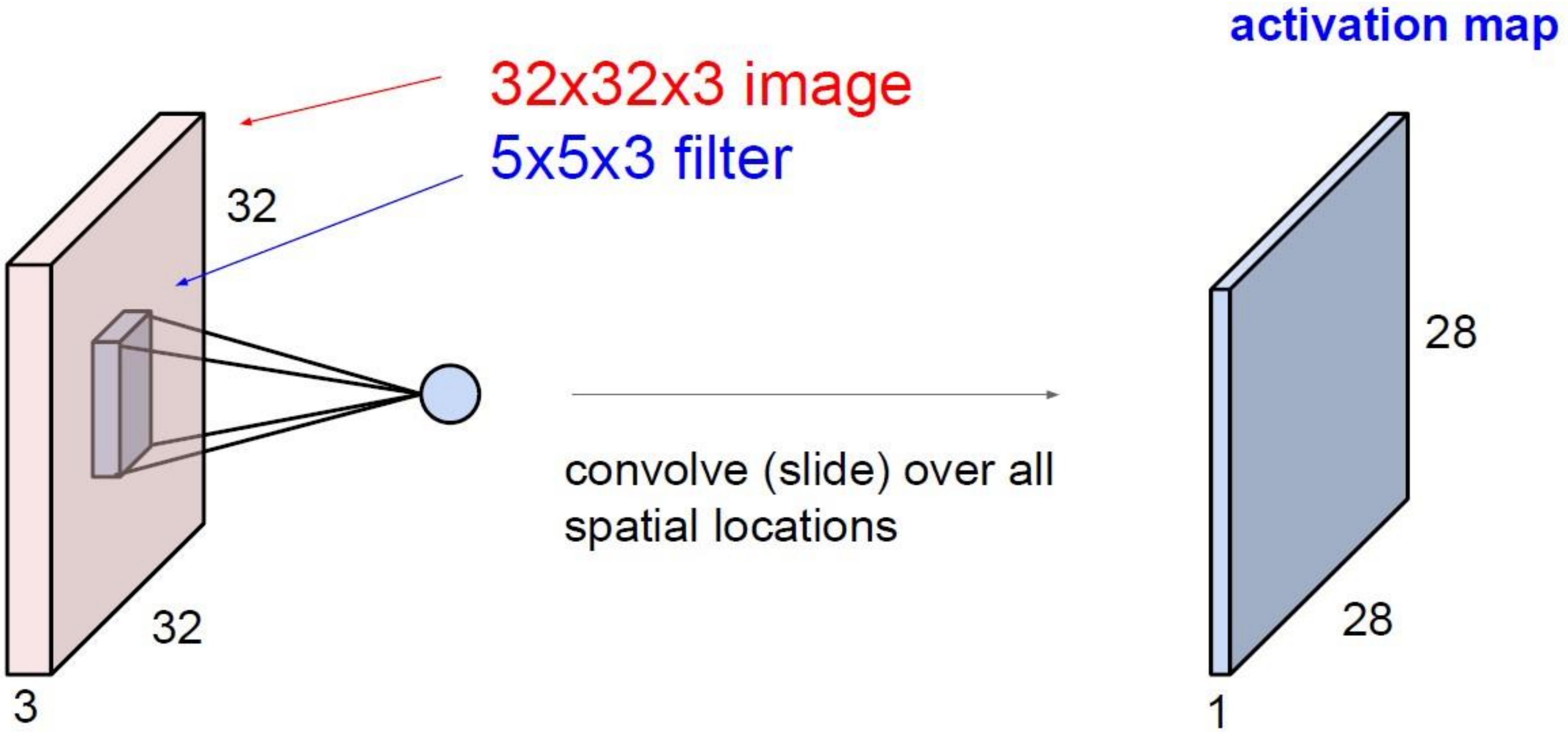
stride 1 => (7 - 3)/1 + 1 = 5

stride 2 => (7 - 3)/2 + 1 = 3

stride 3 => (7 - 3)/3 + 1 = 2.33 :\

04

# Convolution Layer computation



activation map

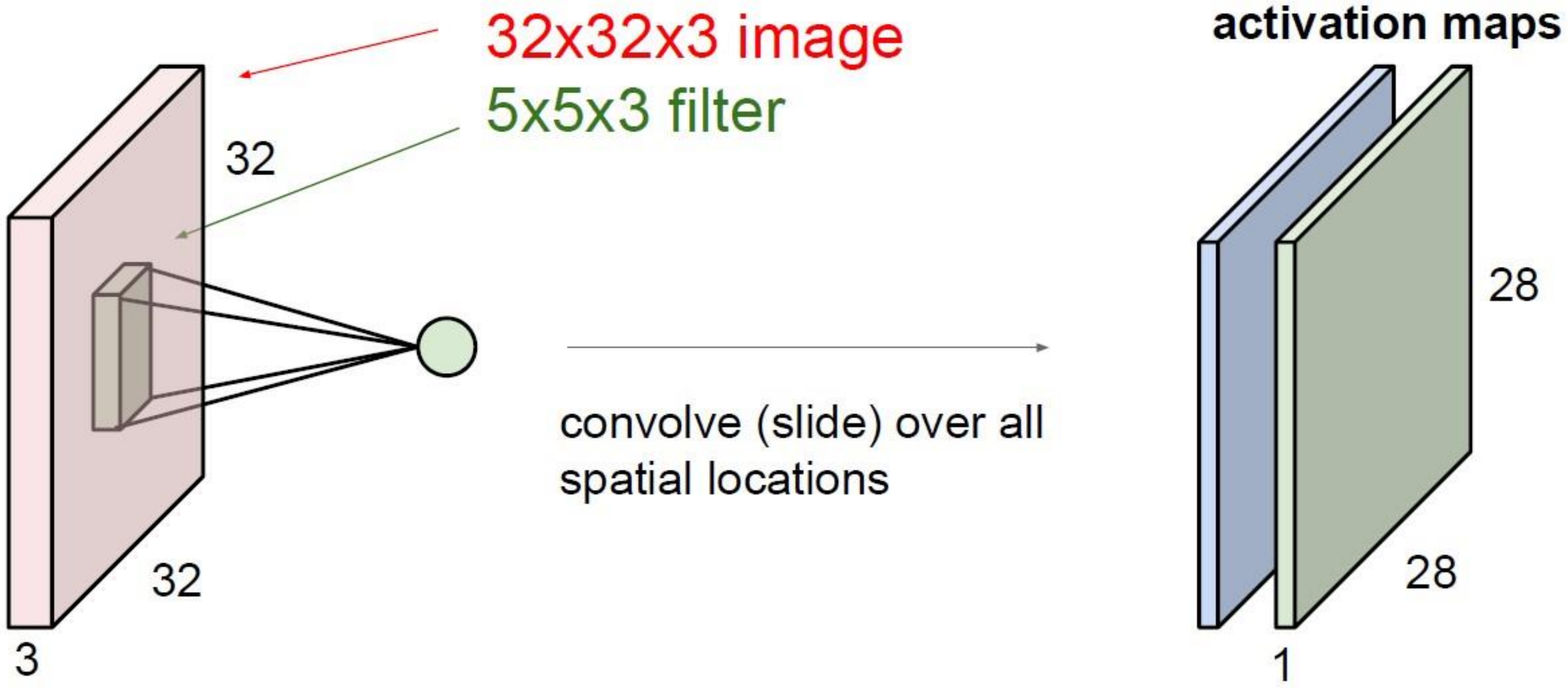
32x32x3 image

5x5x3 filter

convolve (slide) over all spatial locations

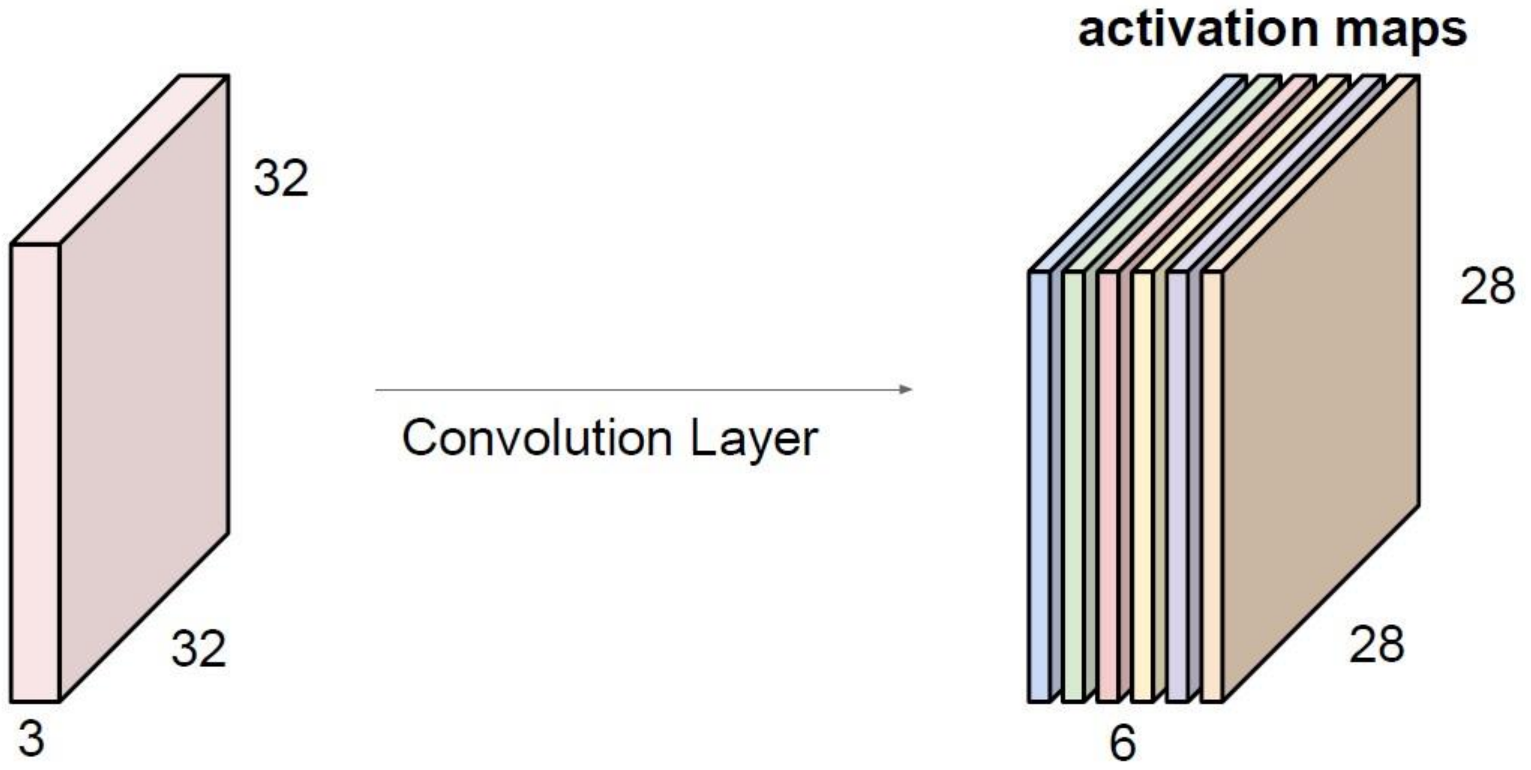
$(N - F) / \text{stride} + 1$ $(32-5)/1+1 = 28$

# Convolution Layer computation



# Convolution Layer computation

For example, if we had 6 5x5 filters, we'll get 6 separate activation maps:

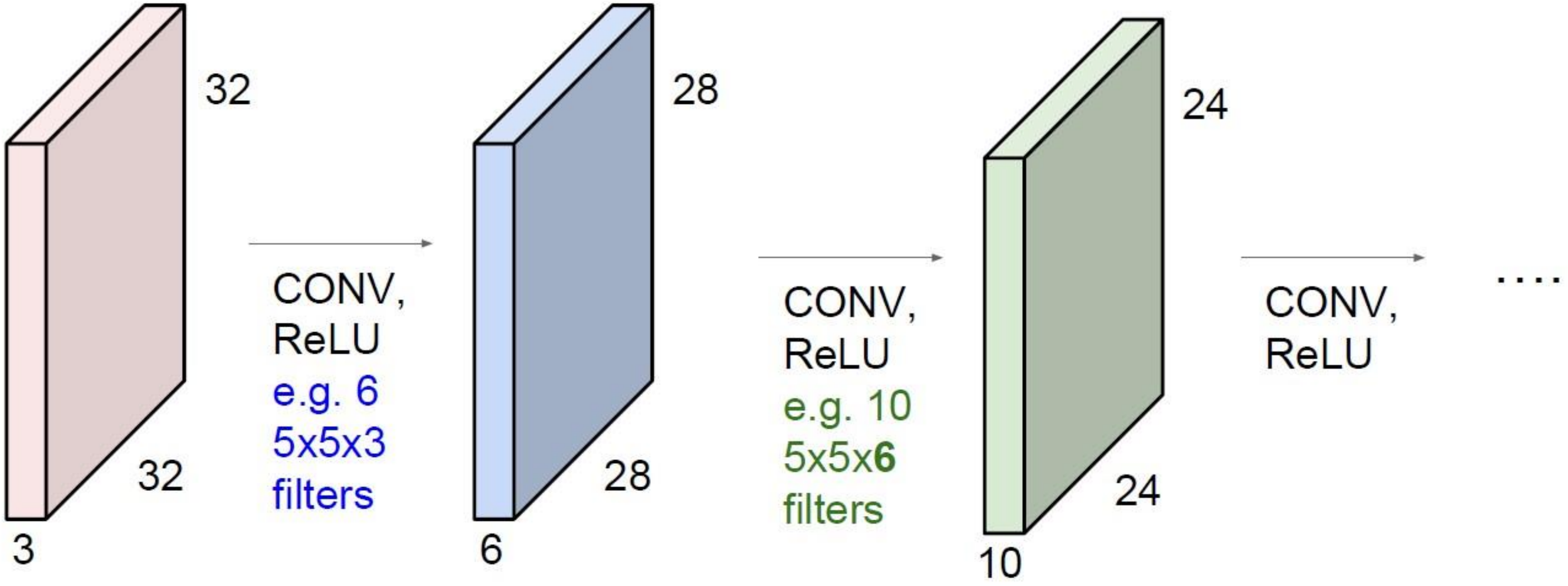


We stack these up to get a "new image" of size 28x28x6!

05

# Convolution Network

**Preview:** ConvNet is a sequence of Convolution Layers, interspersed with activation functions



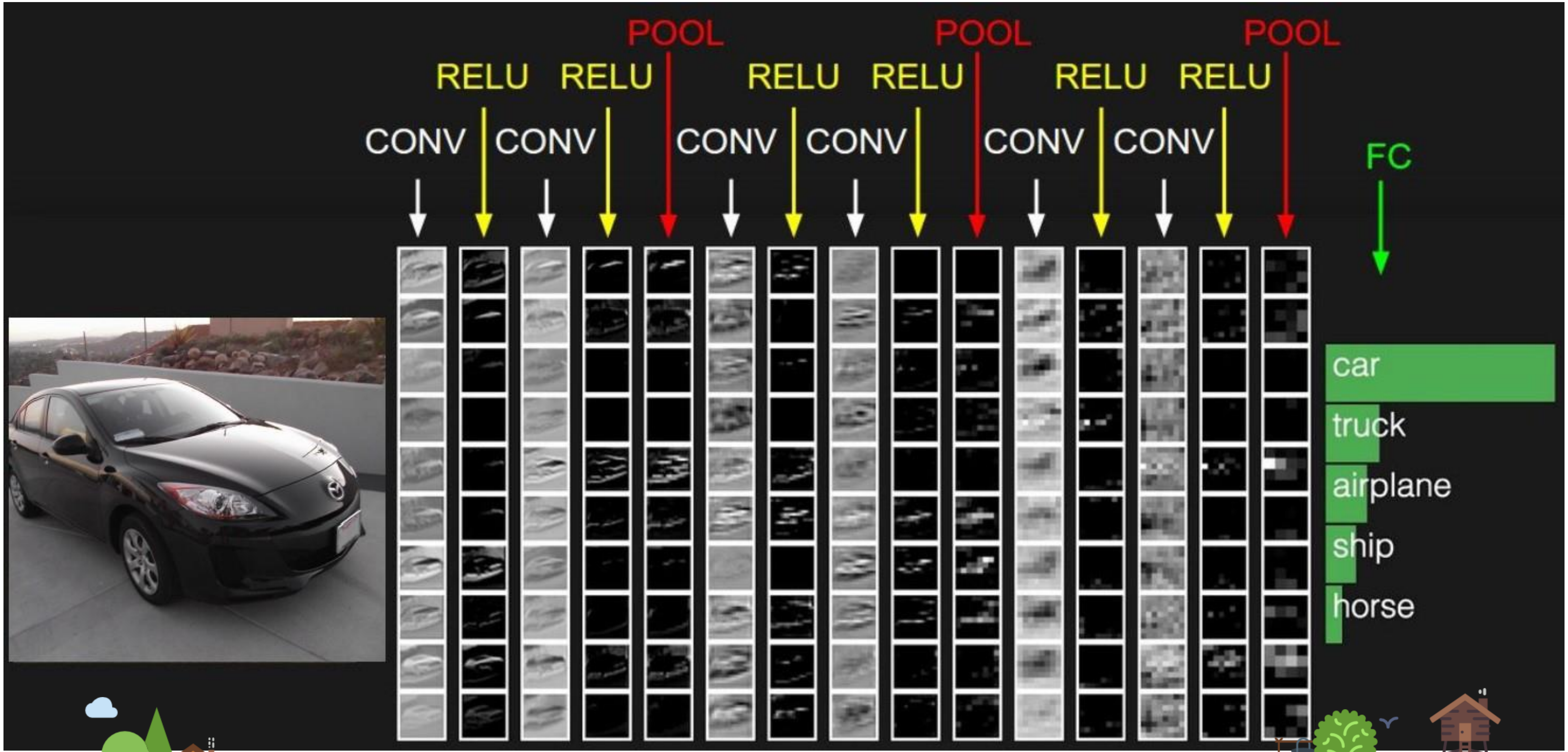
CONV, ReLU e.g. 6 5x5x3 filters

CONV, ReLU e.g. 10 5x5x**6** filters

CONV, ReLU

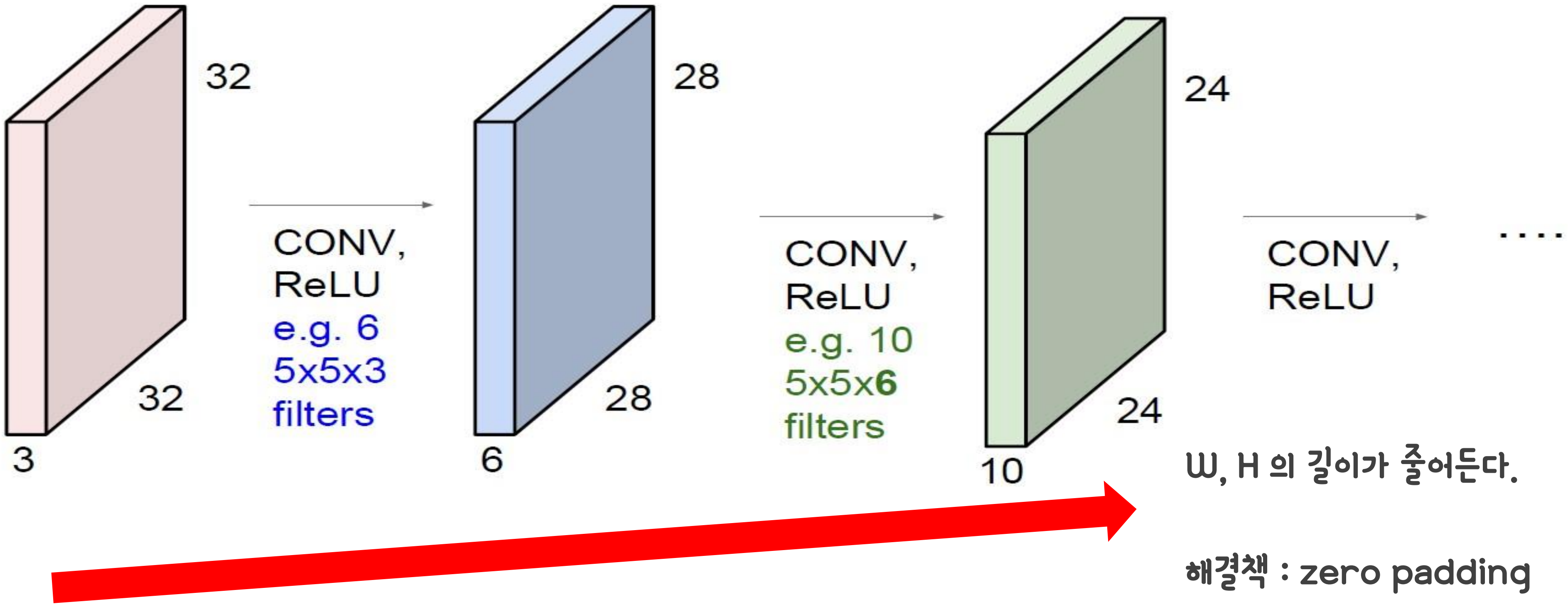
05

# Convolution Network



# 06 Zero padding

**Preview:** ConvNet is a sequence of Convolution Layers, interspersed with activation functions

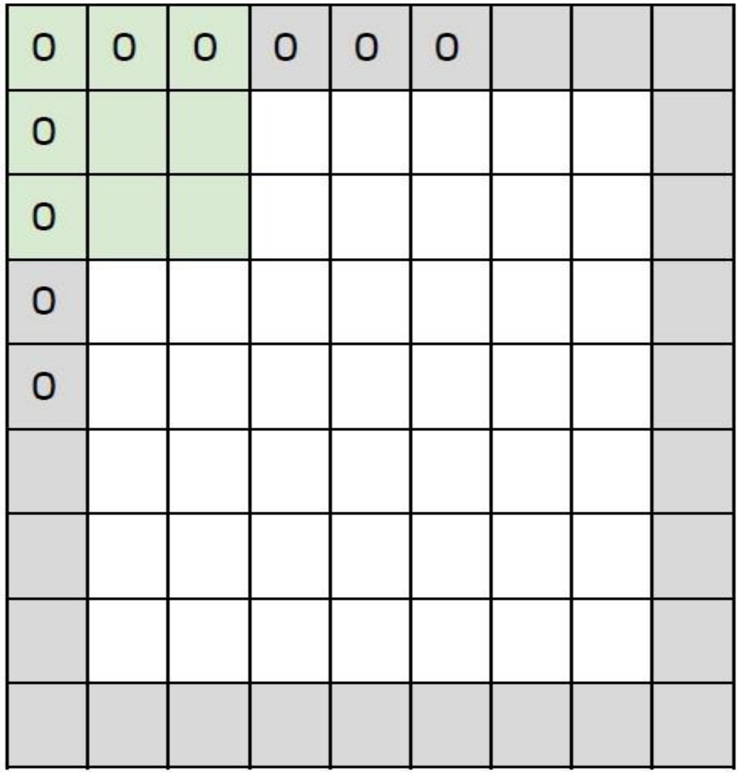


W, H 의 길이가 줄어든다.

해결책 : zero padding

06

# Zero padding

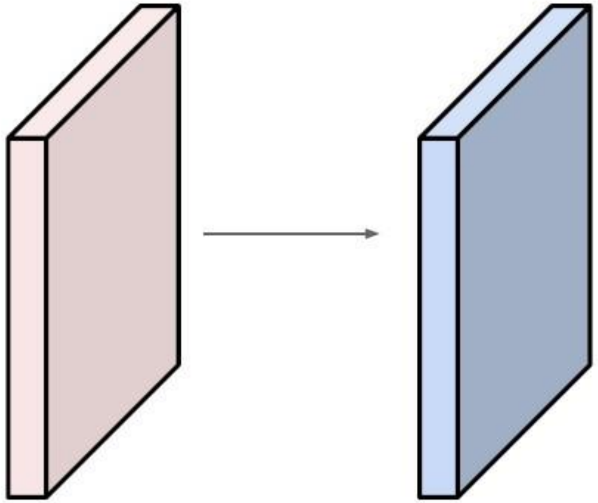


| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | | | |
| 0 | | | | | | | | |
| 0 | | | | | | | | |
| 0 | | | | | | | | |
| 0 | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |

- Activation map size 조절
- 이 후 연산은 동일

# 07 Example

Examples time:



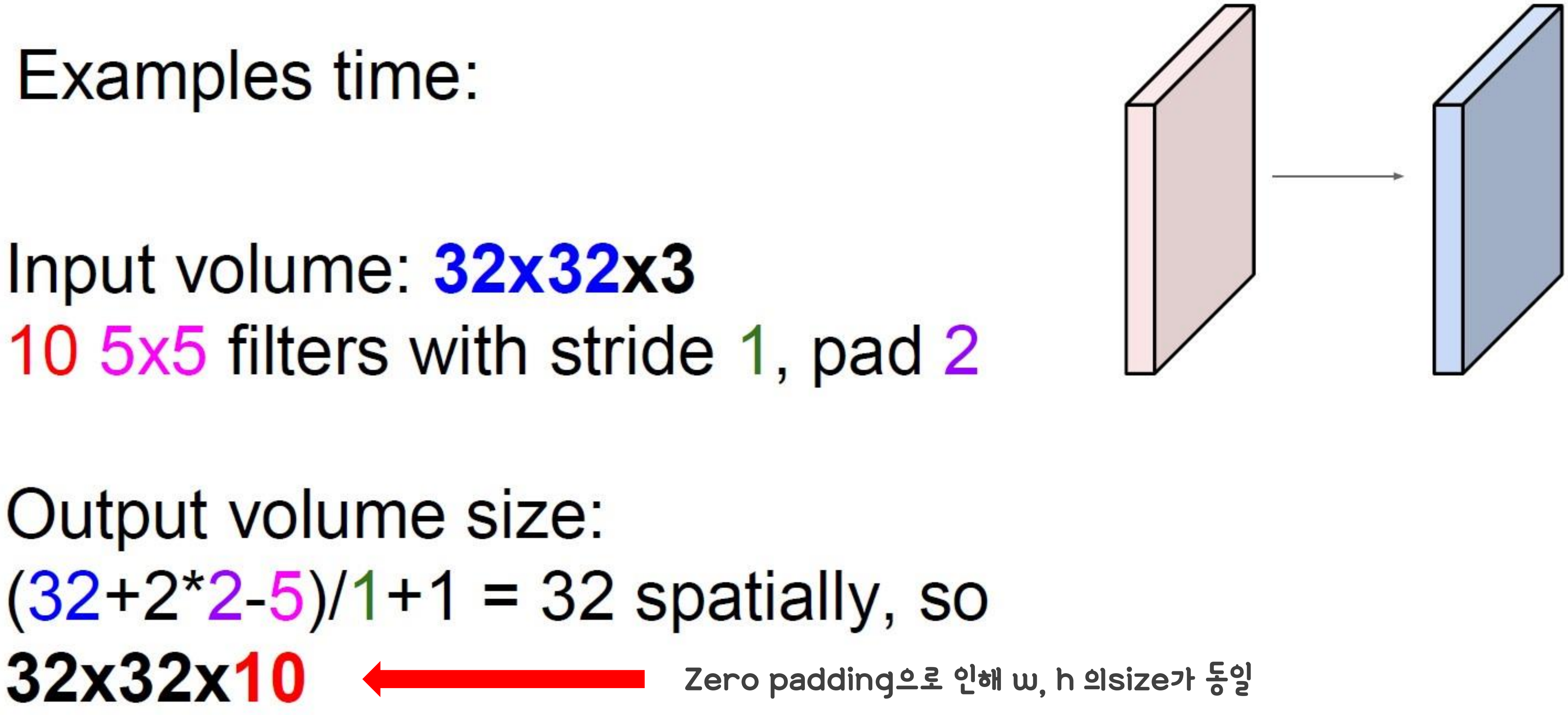
Input volume: **32x32x3**

10 5x5 filters with stride 1, pad 2

Output volume size: ?

07

# Example



Examples time:

Input volume: **32x32x3**

10 5x5 filters with stride 1, pad 2

Output volume size:

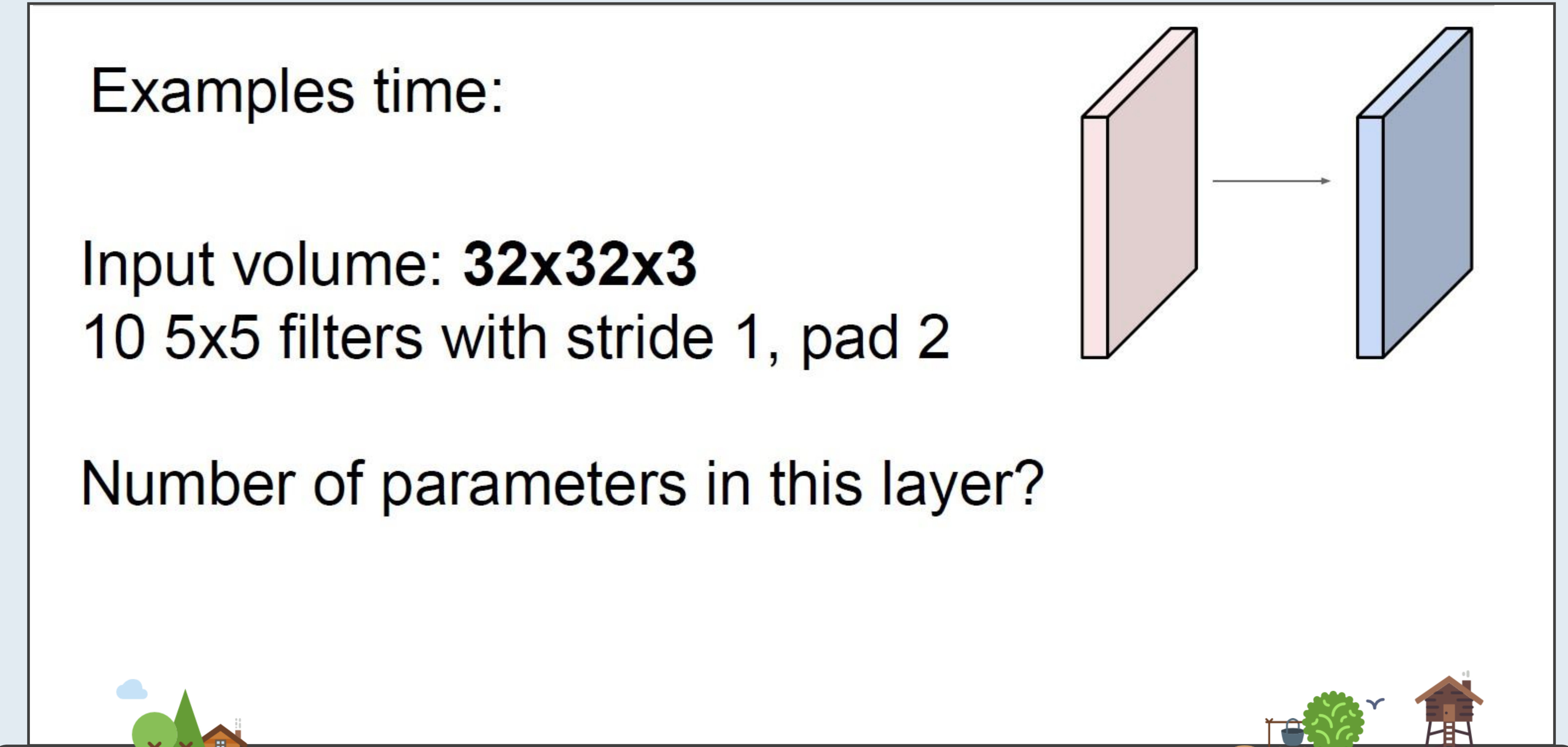
(32+2*2-5)/1+1 = 32 spatially, so

**32x32x10**

Zero padding으로 인해 w, h 의size가 동일

07

# Example

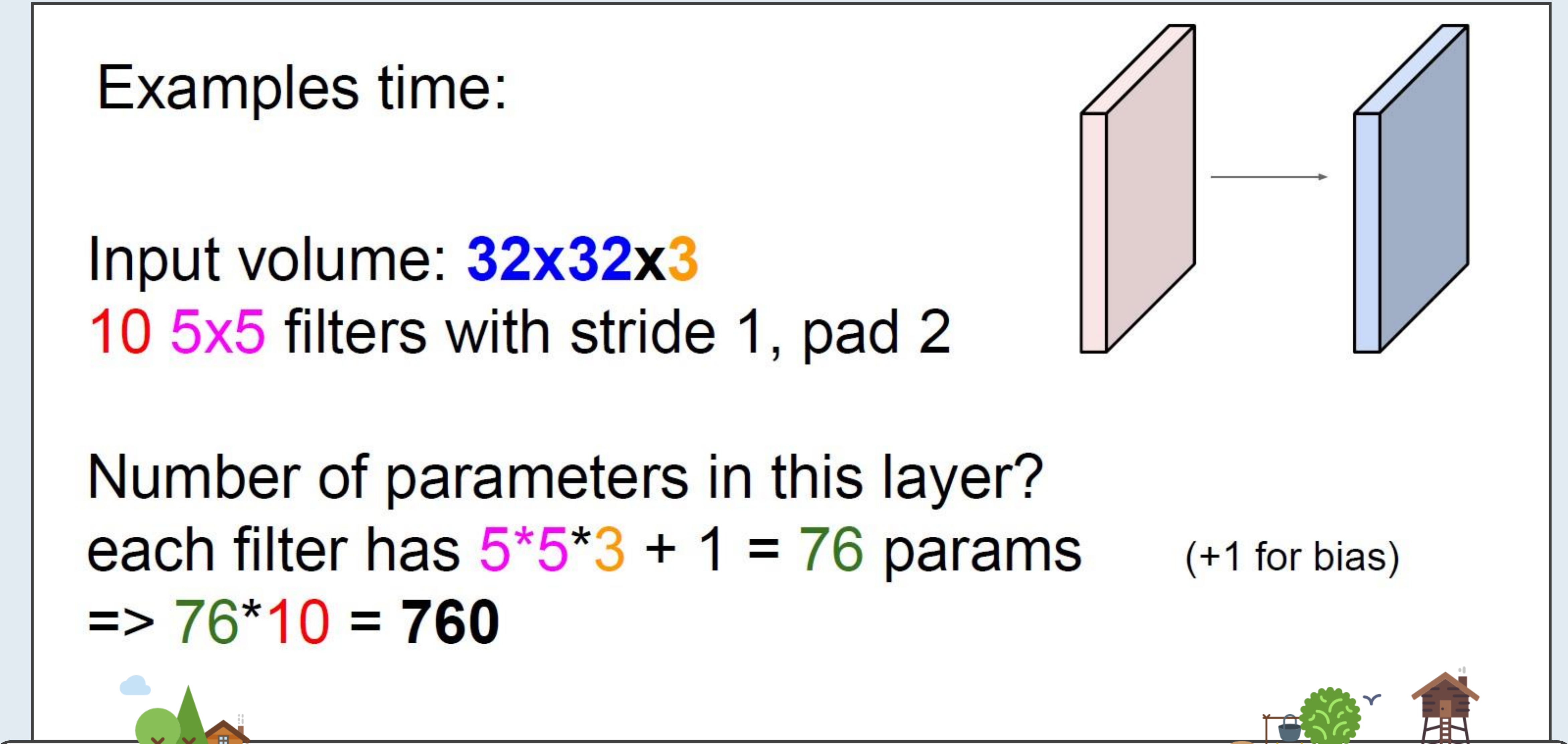


Examples time:

Input volume: **32x32x3**
10 5x5 filters with stride 1, pad 2

Number of parameters in this layer?

# 07 Example



Examples time:

Input volume: **32x32x3**

10 5x5 filters with stride 1, pad 2

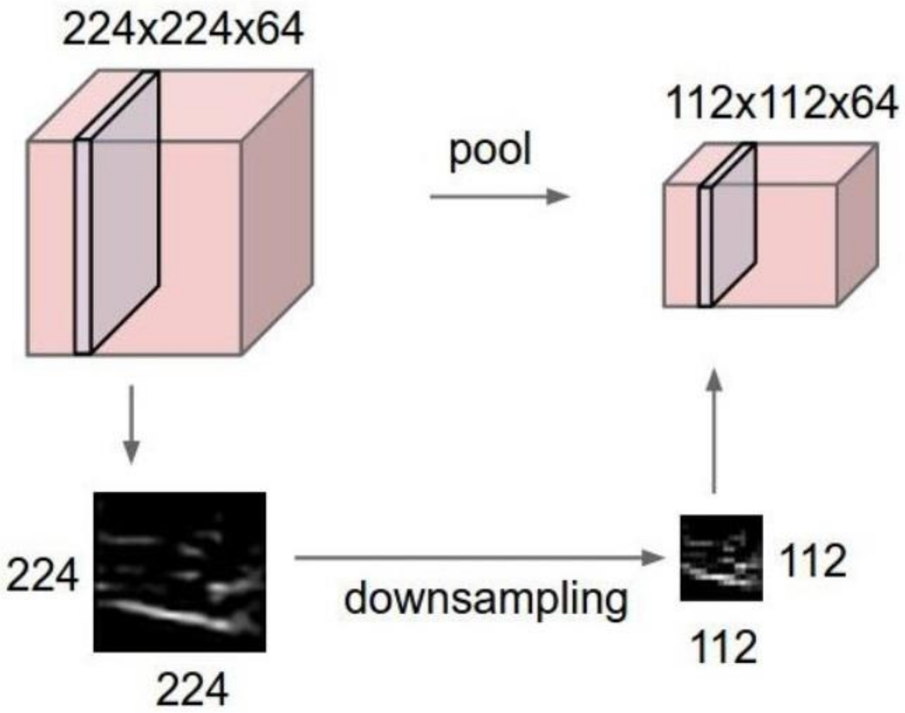
Number of parameters in this layer?

each filter has 5*5*3 + 1 = 76 params (+1 for bias)

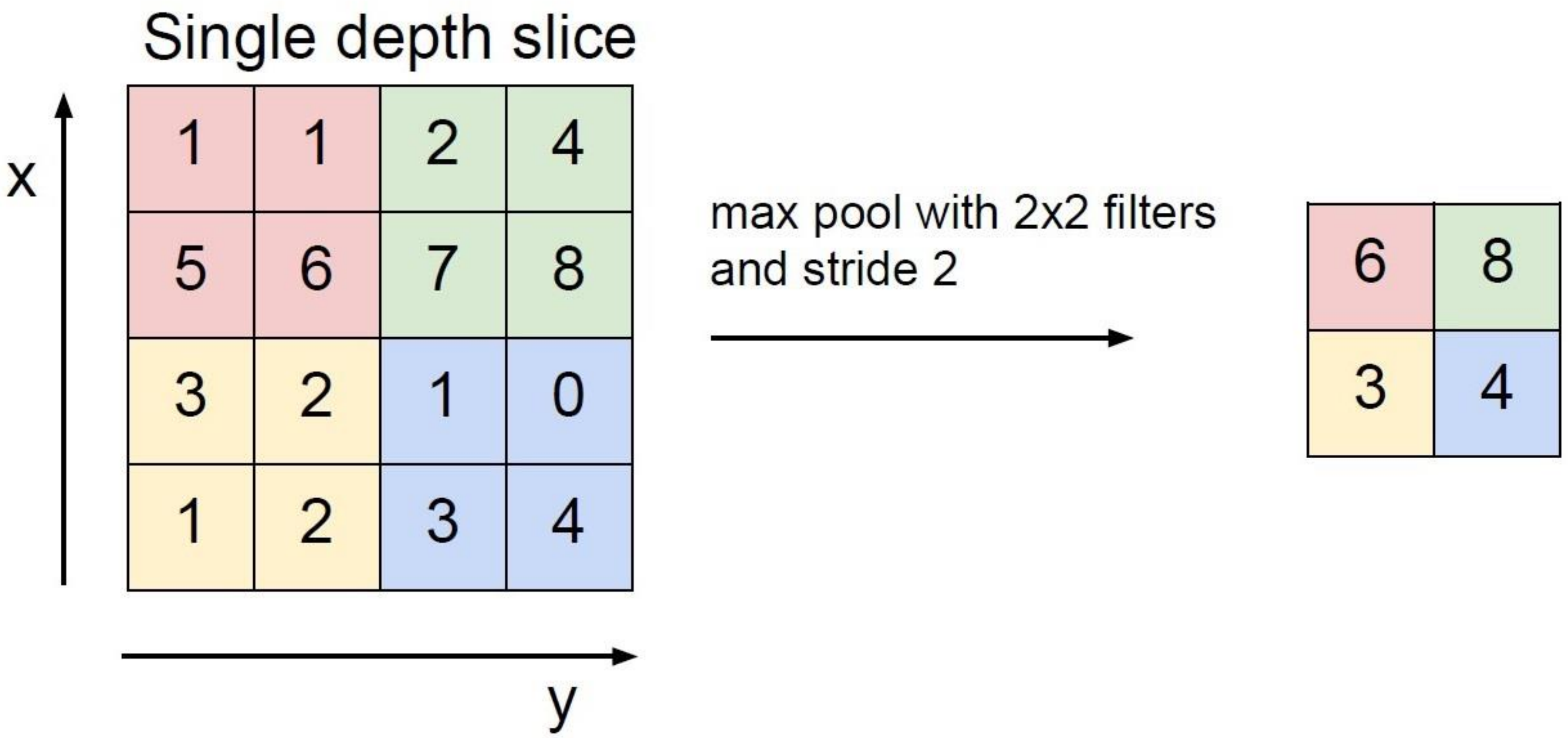
=> 76*10 = **760**

08

# Pooling layer



# 08 Max pooling



Single depth slice

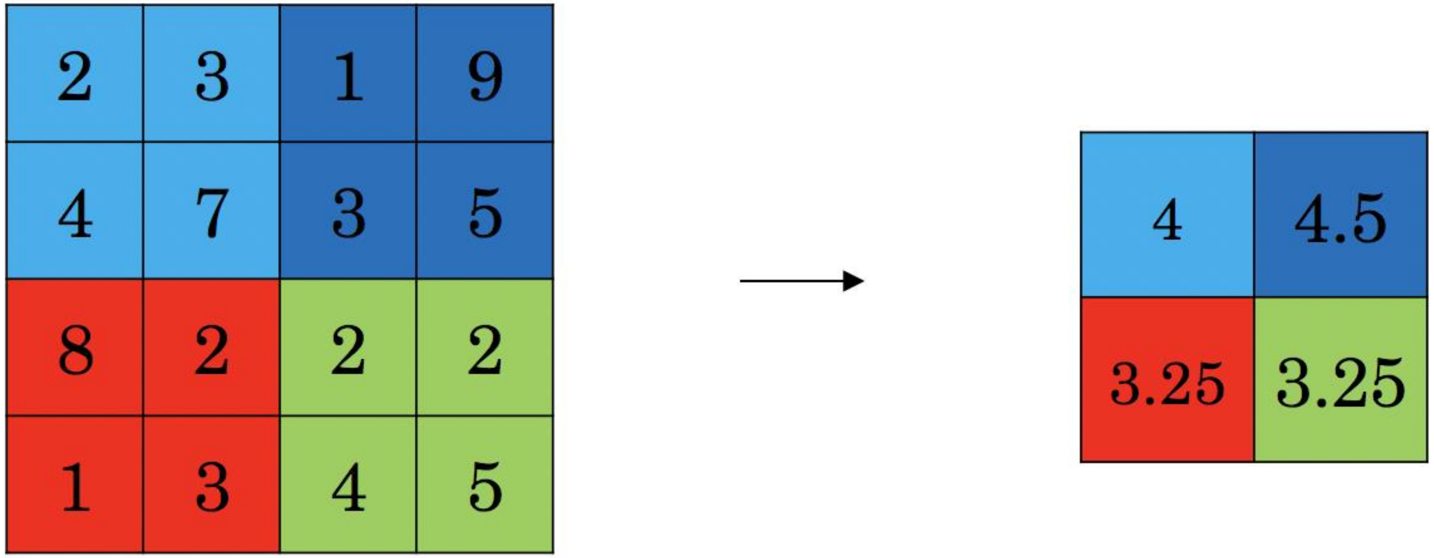
|  |  |  |  |
|---|---|---|---|
| 1 | 1 | 2 | 4 |
| 5 | 6 | 7 | 8 |
| 3 | 2 | 1 | 0 |
| 1 | 2 | 3 | 4 |

max pool with 2x2 filters and stride 2

|  |  |
|---|---|
| 6 | 8 |
| 3 | 4 |

08

# Average pooling



| | | | |
|---|---|---|---|
| 2 | 3 | 1 | 9 |
| 4 | 7 | 3 | 5 |
| 8 | 2 | 2 | 2 |
| 1 | 3 | 4 | 5 |

| | |
|---|---|
| 4 | 4.5 |
| 3.25 | 3.25 |

End

# 참고 자료

CS231n : http://cs231n.stanford.edu/syllabus.html