CNN Architecture

ML Instruction Team, Fall 2022

CE Department
Sharif University of Technology

Ali Sharifi-Zarchi
Behrooz Azarkhalili
Arian Amani
Hamidreza Yaghobi

Convolutional Neural Networks: History

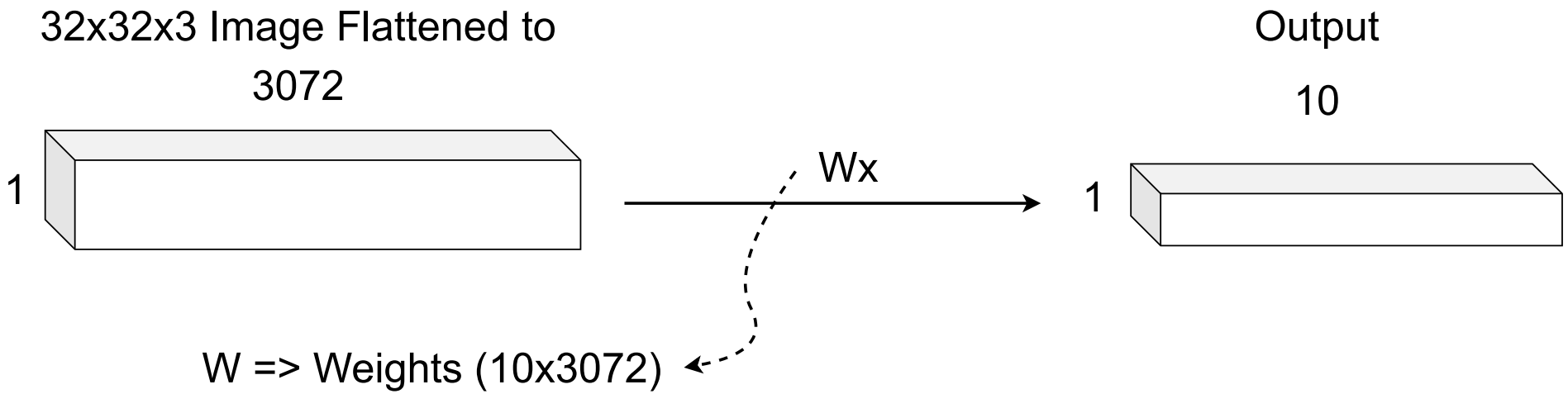
Biological Inspiration of Convolutional Neural Networks

Hubel and Wiesel

- 1959
- 1961
- 1968

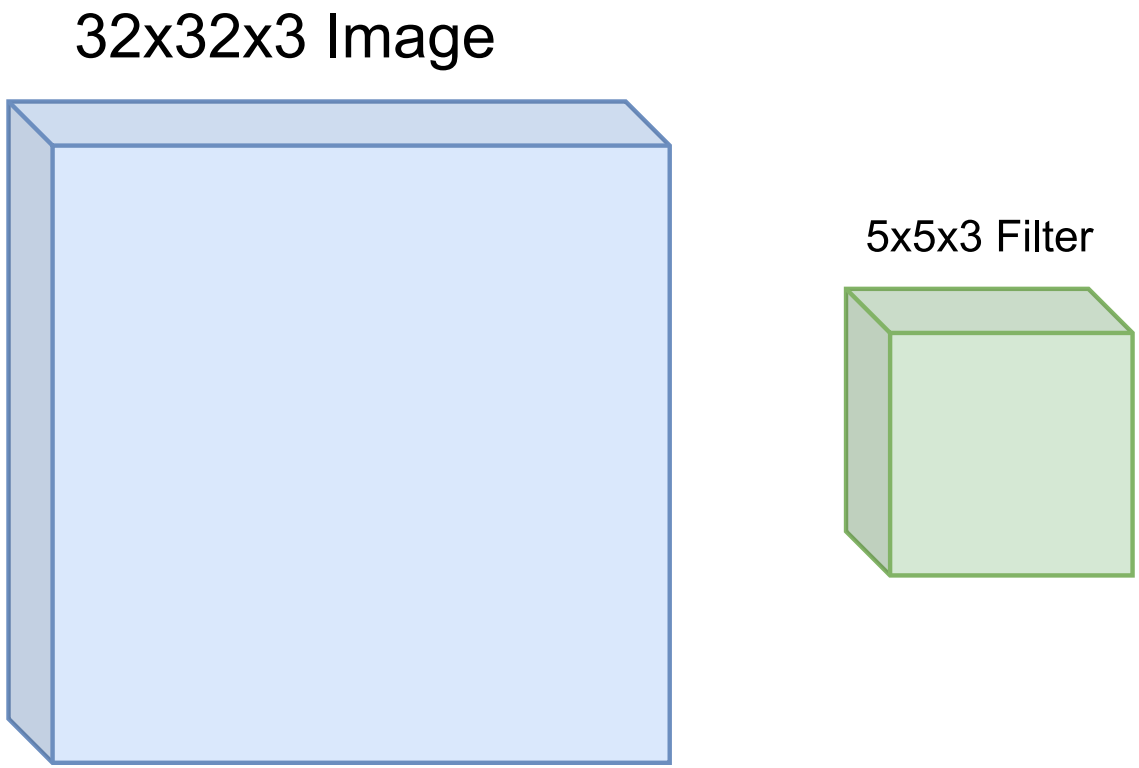
CNNs

What we've been using:
Fully Connected Layers



CNNs

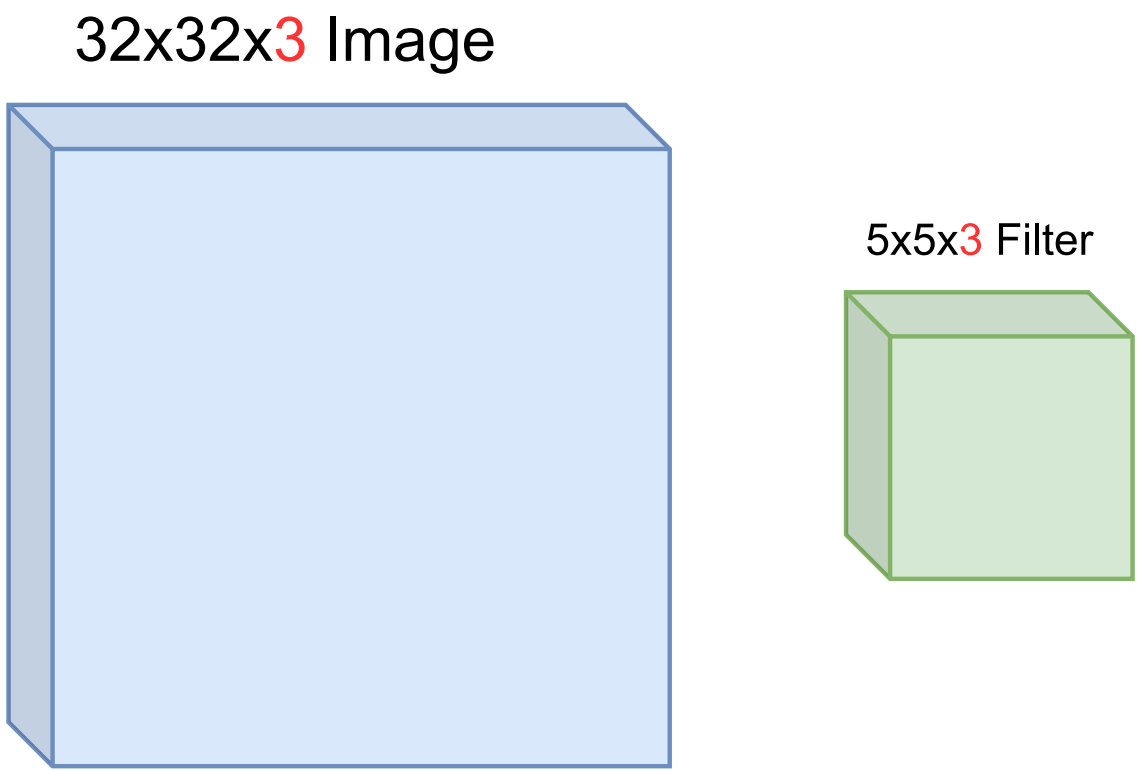
What we're going to learn:
Convolutional Layer



CNNs

Convolutional Layer

- Filters always extend the full depth of the input volume. (#Input channels == #Filter Channels)

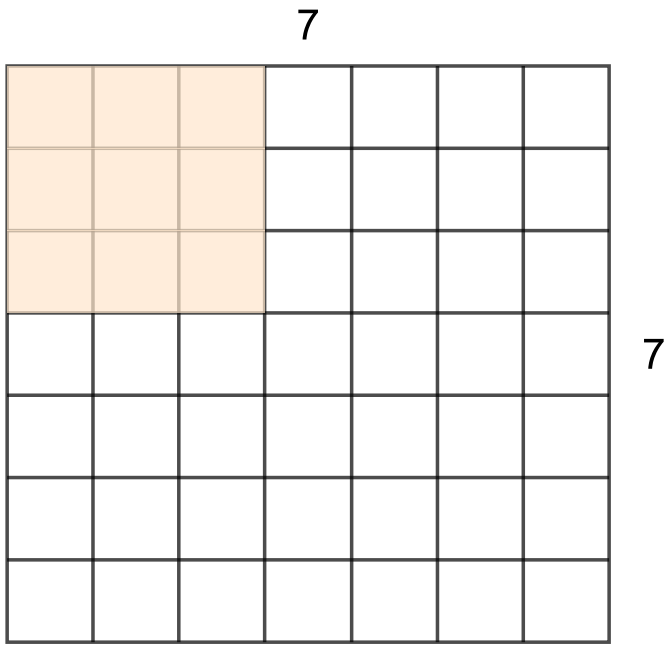


What is a Stride

The amount of movement between applications of the filter to the input image is referred to as the stride, and it is almost always symmetrical in height and width dimensions.

Closer look

- 7x7 input with 3x3 filter

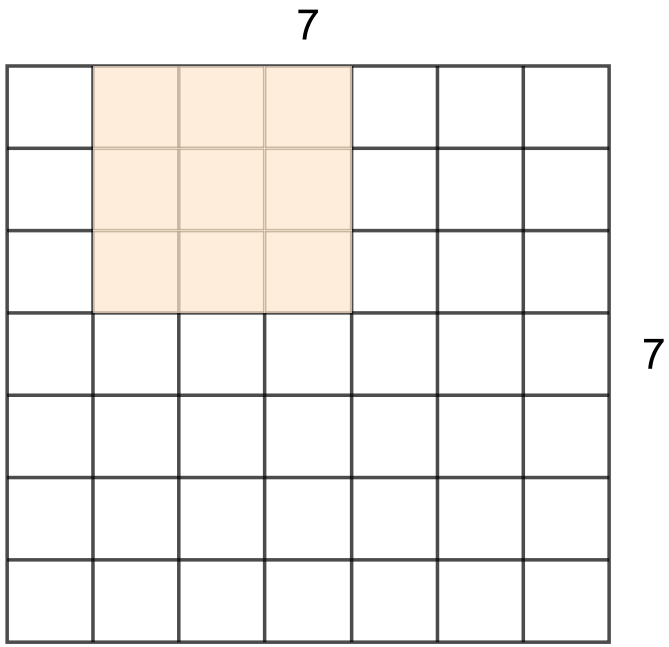


What is a Stride

The amount of movement between applications of the filter to the input image is referred to as the stride, and it is almost always symmetrical in height and width dimensions.

Closer look

- 7x7 input with 3x3 filter

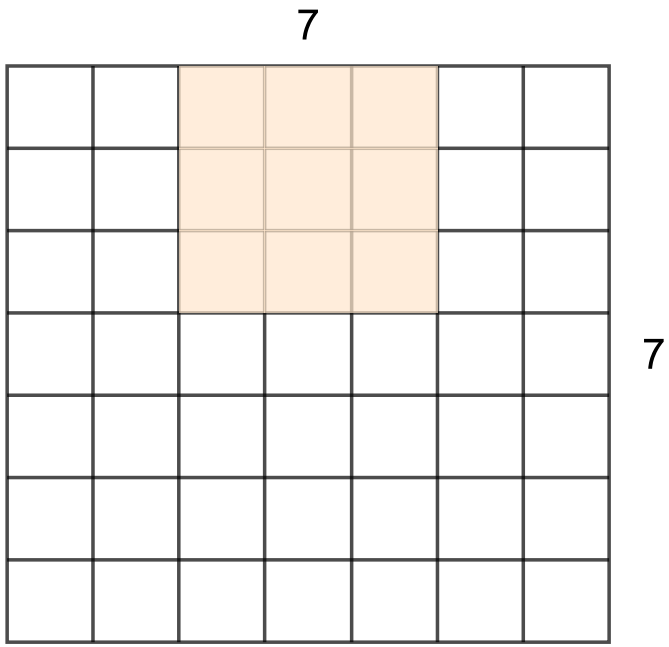


What is a Stride

The amount of movement between applications of the filter to the input image is referred to as the stride, and it is almost always symmetrical in height and width dimensions.

Closer look

- 7x7 input with 3x3 filter

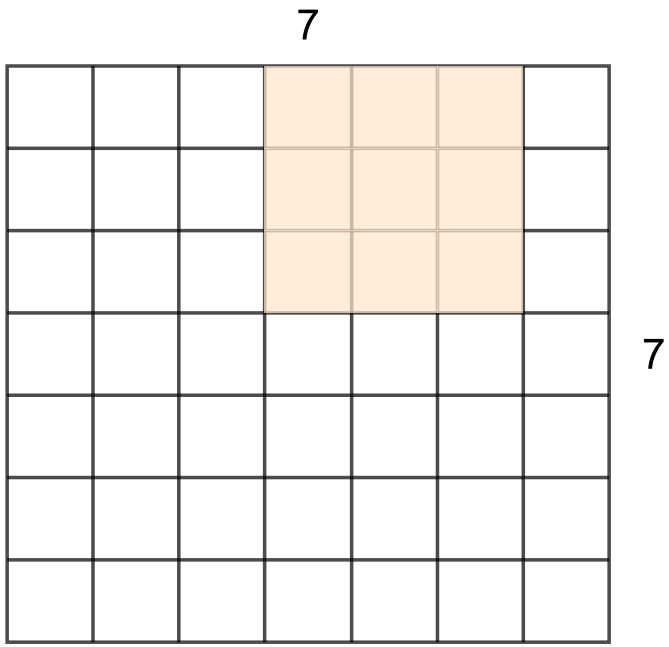


What is a Stride

The amount of movement between applications of the filter to the input image is referred to as the stride, and it is almost always symmetrical in height and width dimensions.

Closer look

- 7x7 input with 3x3 filter

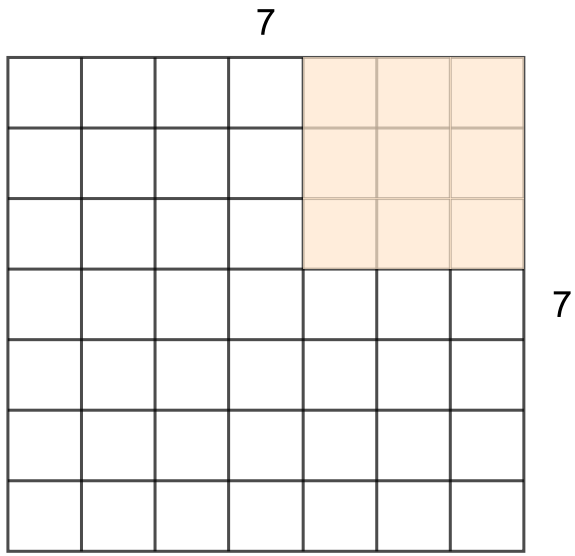


What is a Stride

The amount of movement between applications of the filter to the input image is referred to as the stride, and it is almost always symmetrical in height and width dimensions.

Closer look

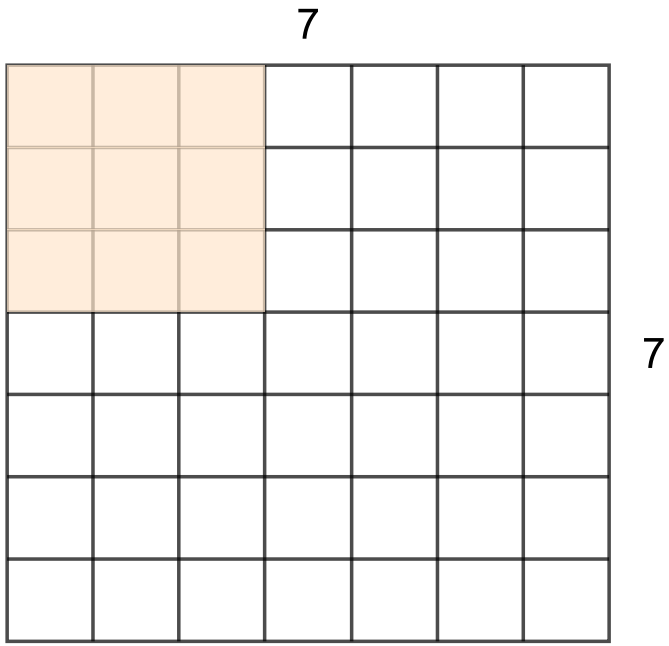
- 7x7 input with 3x3 filter
- This was a Stride 1 filter
- => Outputs 5x5



Stride

Now let's use Stride 2

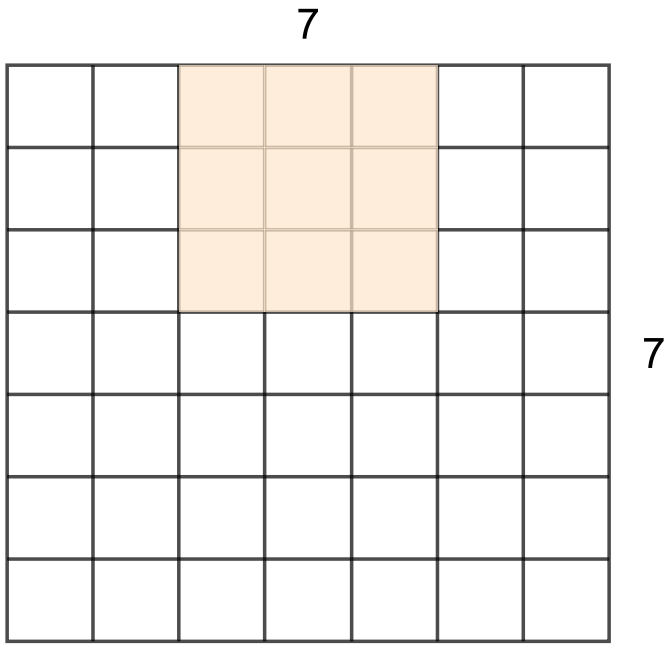
- 7x7 input with 3x3 filter



Stride

Now let's use Stride 2

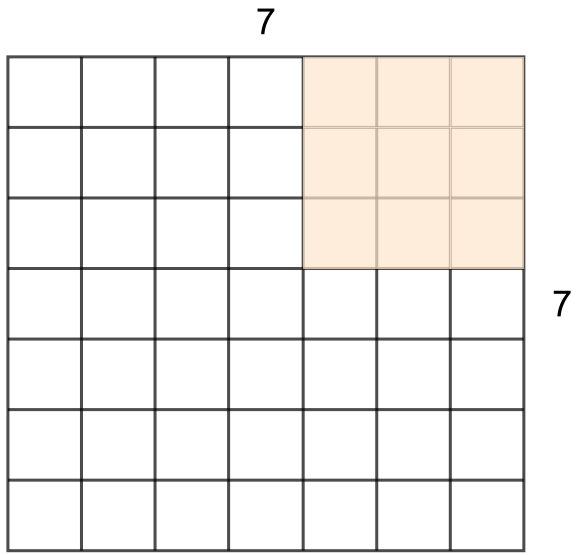
- 7x7 input with 3x3 filter



Stride

Now let's use Stride 2

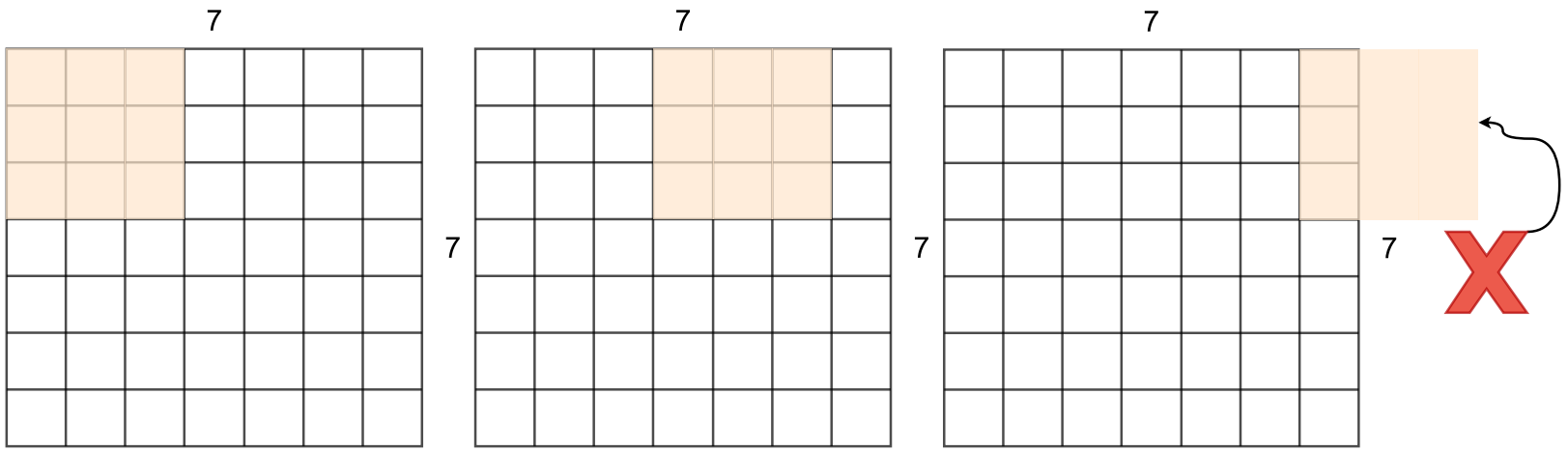
- 7x7 input with 3x3 filter
- This was a Stride 2 filter
- => Outputs 3x3



Stride

Stride 3?

- 7x7 input with 3x3 filter

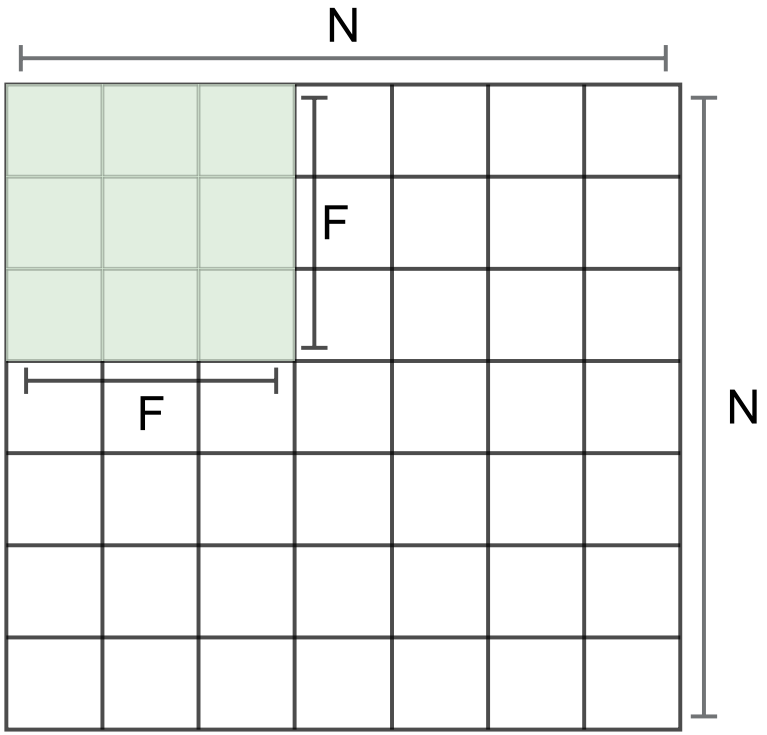


Stride

So 7x7 input with 3x3 filter and stride 3 doens't work!

Let't do the calculations:

$$OutputSize = (N - F)/Stride + 1$$

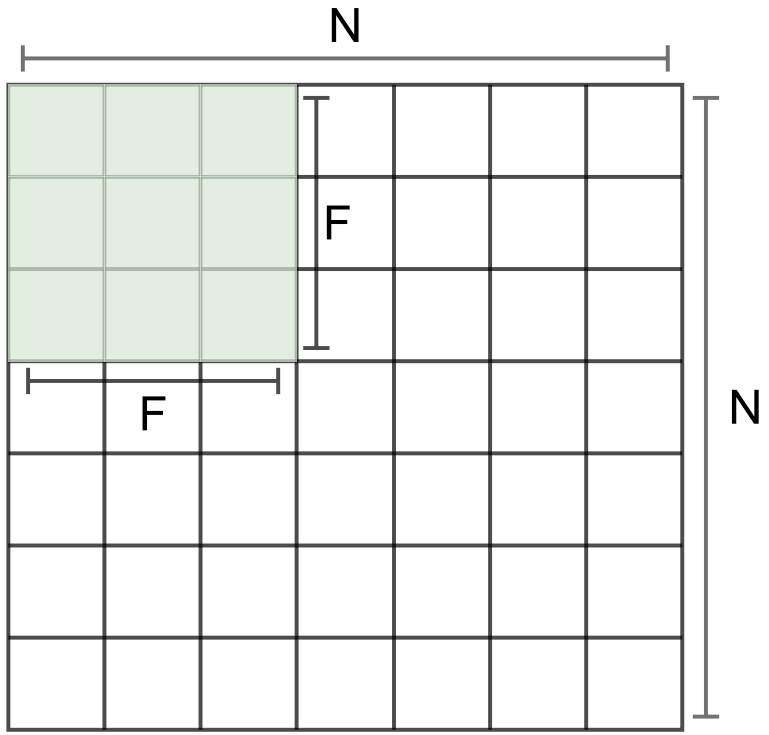


Stride

$$OutputSize = (N - F)/Stride + 1$$

N = 7, F = 3 =>

- Stride 1 ==> $(7 - 3)/1 + 1 = 5$
- Stride 2 ==> $(7 - 3)/2 + 1 = 3$
- Stride 3 ==> $(7 - 3)/3 + 1 = 2.33$:))



Strides

Do you see any problems?

Strides

- 1. Borders don't get enough attention.

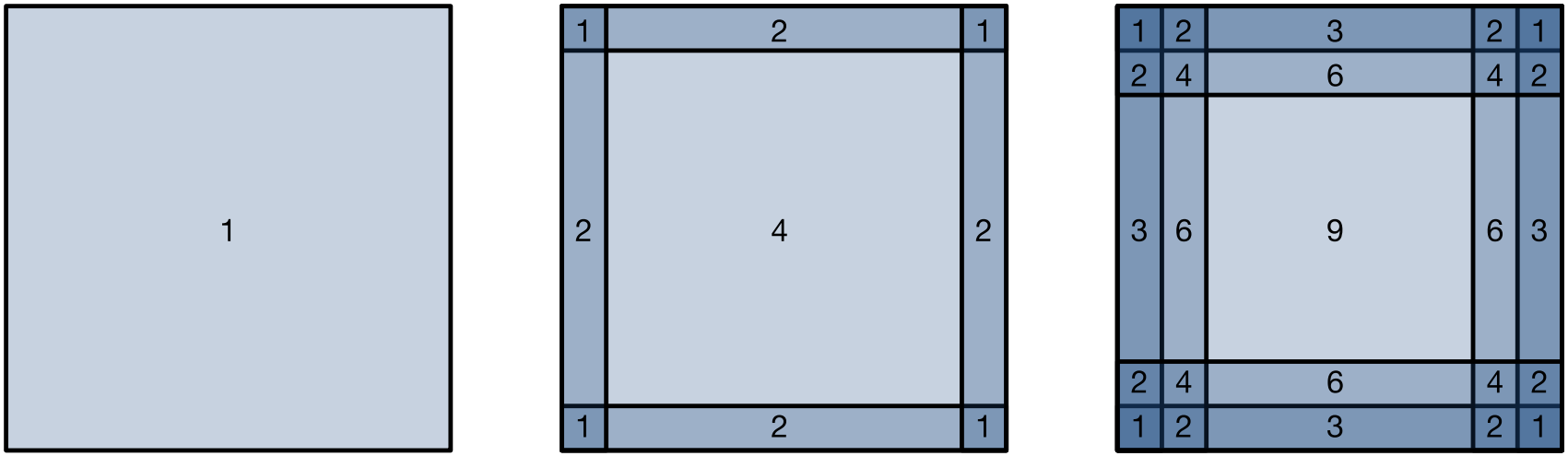


Figure: Dive into Deep Learning, Fig. 7.3.1: Pixel utilization for convolutions of size 1×1, 2×2, and 3×3 respectively.

Strides

- 1. Borders don't get enough attention.
- 2. Outputs shrink!

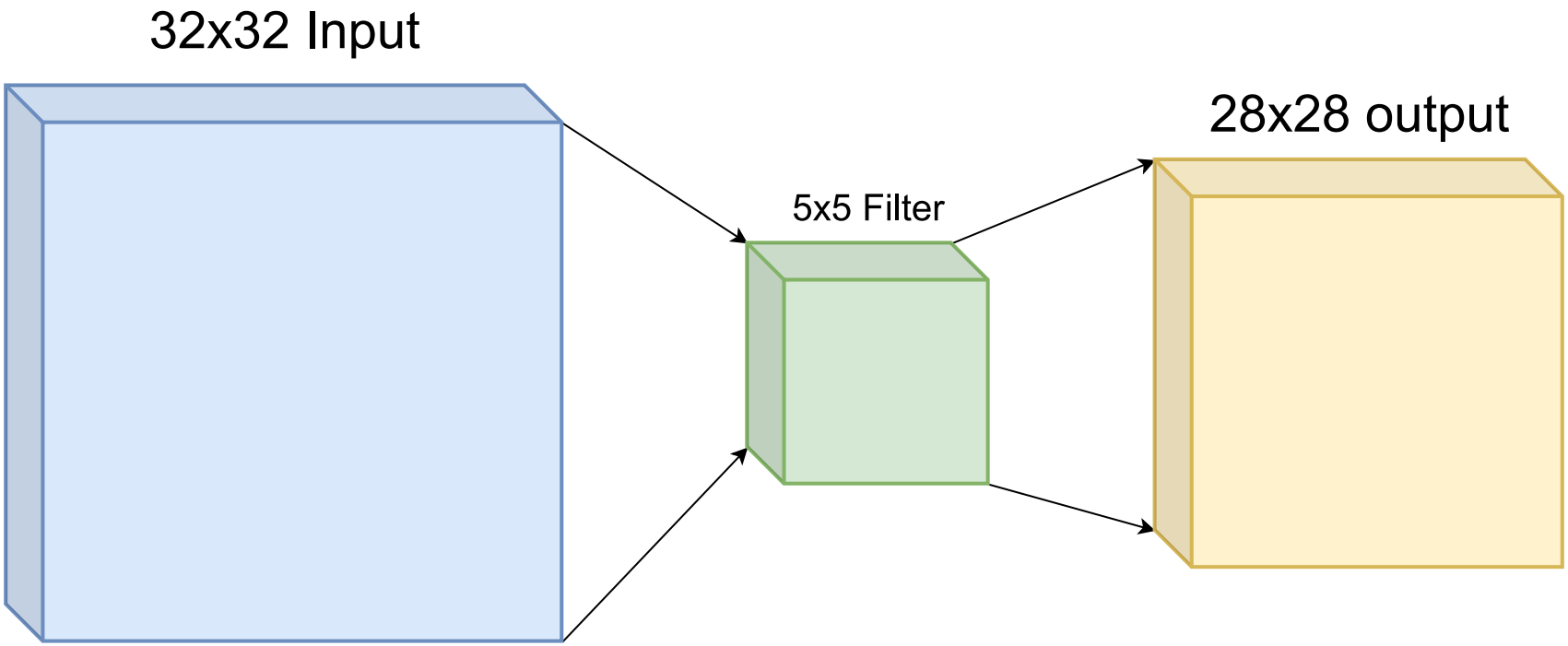


Figure: 32x32 input shrinks to 28x28 output. (information loss)

Strides

What is the solution?

Padding

Enters Padding

Padding

- We can use Padding to preserve the dimensionality
- It ensures that all pixels are used equally frequently

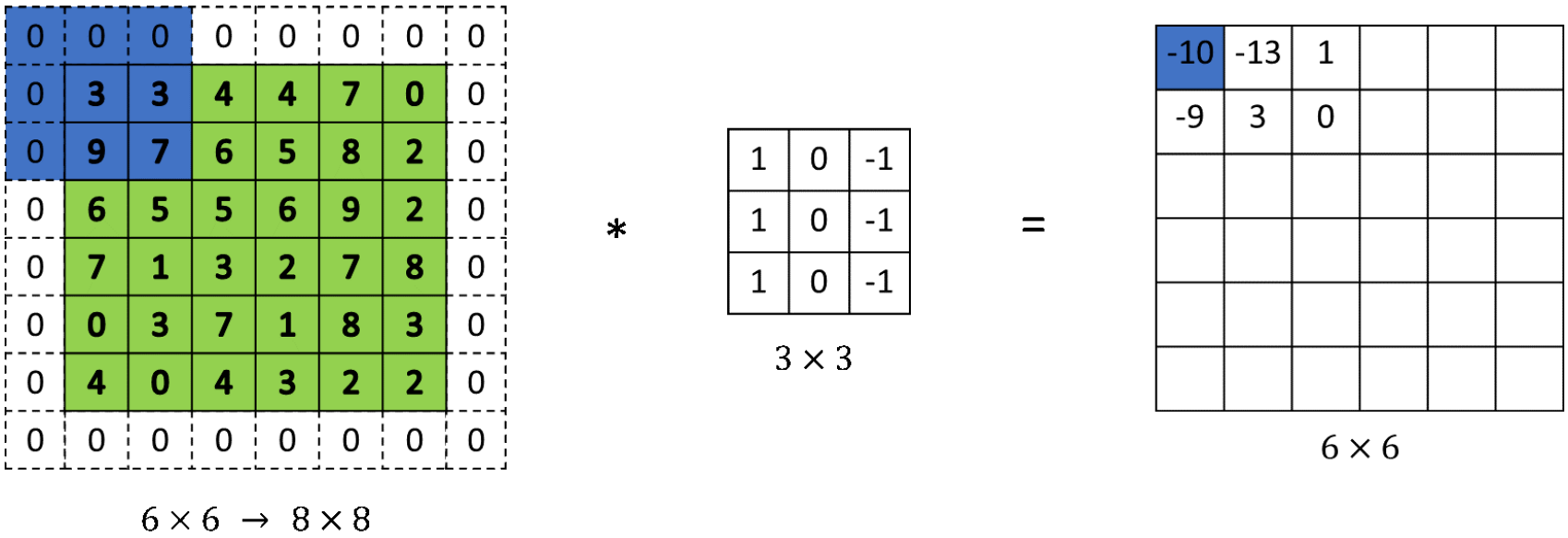


Figure: DataHacker.rs: CNN Padding - Applying padding of 1 before convolving with 3×3 filter

Padding

- It is common to use $P = (F - 1)/2$ with stride 1 to preserve the input size.

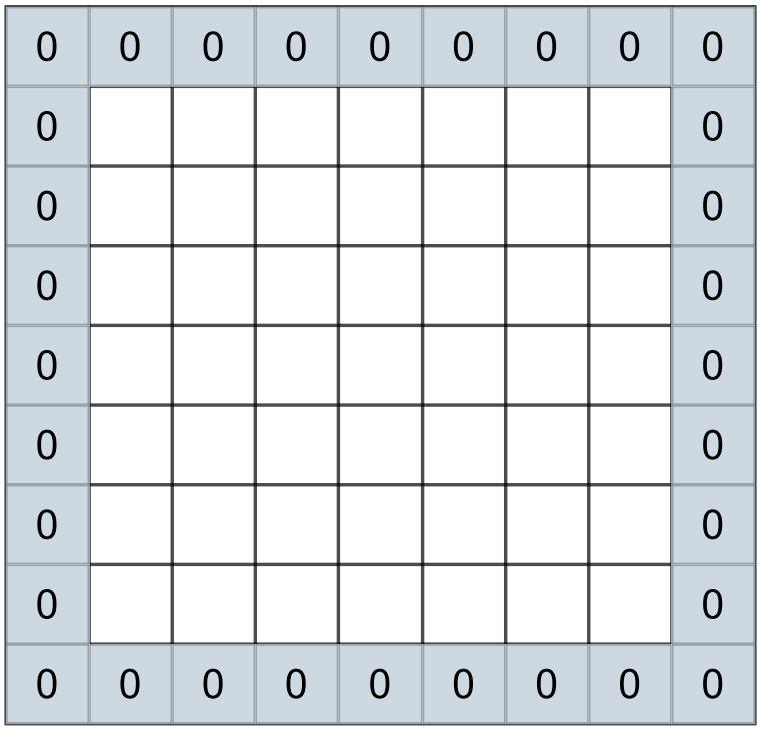


Figure: Applying zero-padding to a 7x7 input with padding 1

Padding

- It is common to use $P = (F - 1)/2$ with stride 1 to preserve the input size.
- $OutputSize = (N + 2P - F)/Stride + 1$

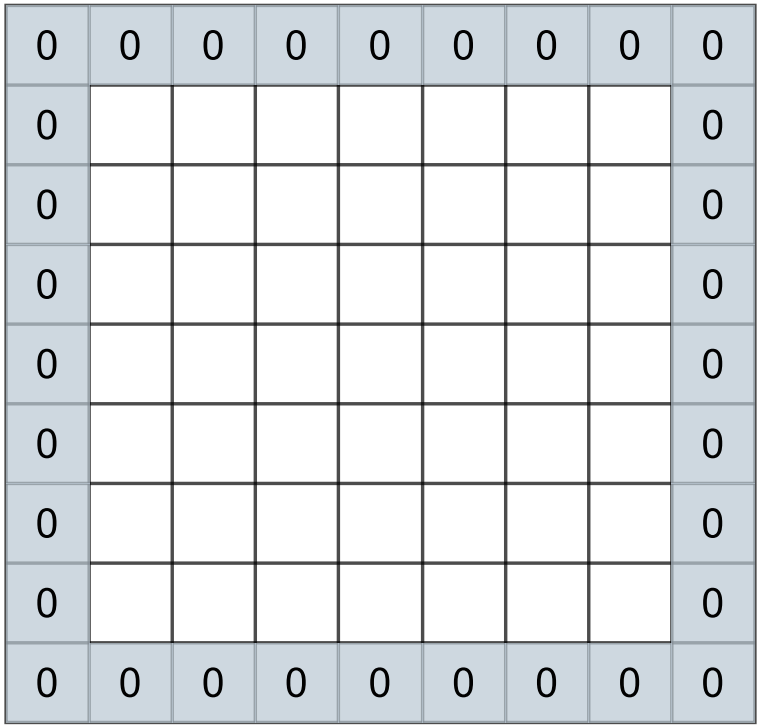


Figure: Applying zero-padding to a 7x7 input with padding 1

Padding

- It is common to use $P = (F-1)/2$ with stride 1 to preserve the input size.
- $OutputSize = (N + 2P - F)/Stride + 1$

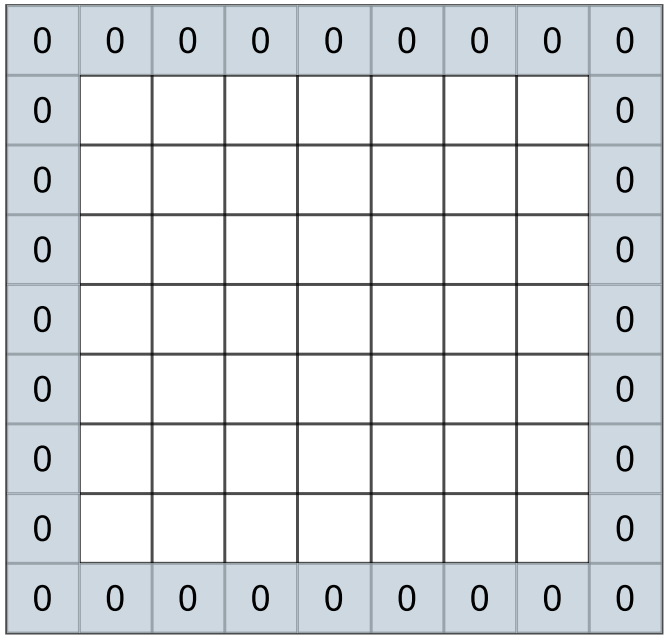


Figure: Applying zero-padding to a 7x7 input with padding 1

7x7 input, stride 1, P to preserve the dimentions?

- $F = 3 --> P = 1$
- $F = 5 --> P = 2$
- $F = 7 --> P = 3$

Channels, etc.

HI :))

Final Notes

Thank You!

Any Question?