



# Deep Learning CS60010

**Abir Das**

Assistant Professor

Computer Science and Engineering Department  
Indian Institute of Technology Kharagpur

<http://cse.iitkgp.ac.in/~adas/>



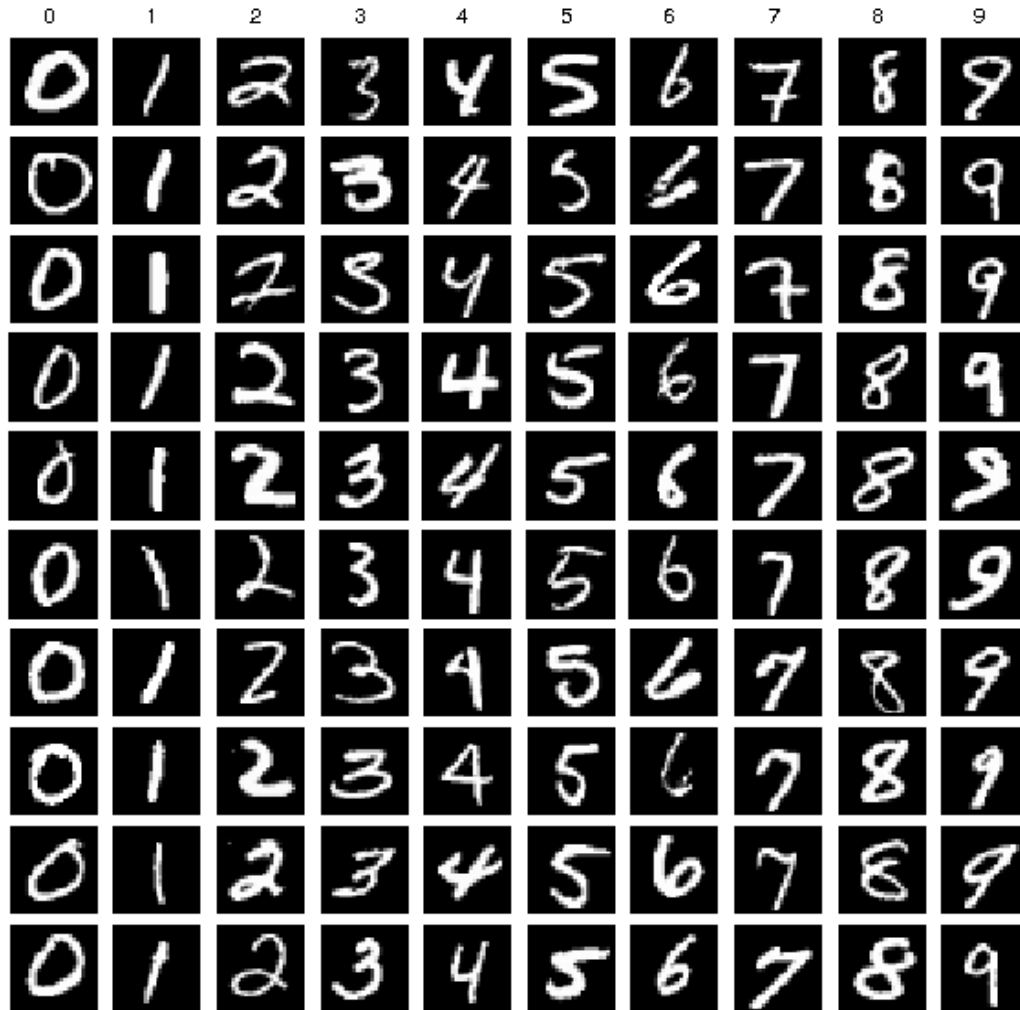
# Agenda

The Building Blocks of Convolutional Neural Networks/CNNs/ConvNets

# Importance of MNIST



# MNIST



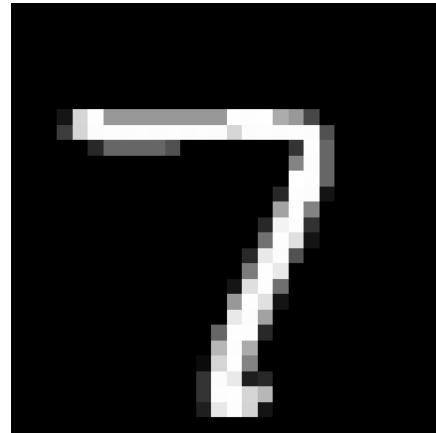
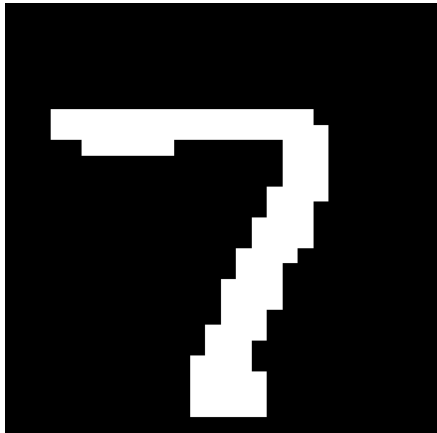
- database of handwritten digits
- 10 classes
  - Training set 60,000 images
  - Test set of 10,000 images
- Greyscale images of size 28x28
- Often treated as 'Hello World' for any ML/DL practitioner

Image taken from Researchgate

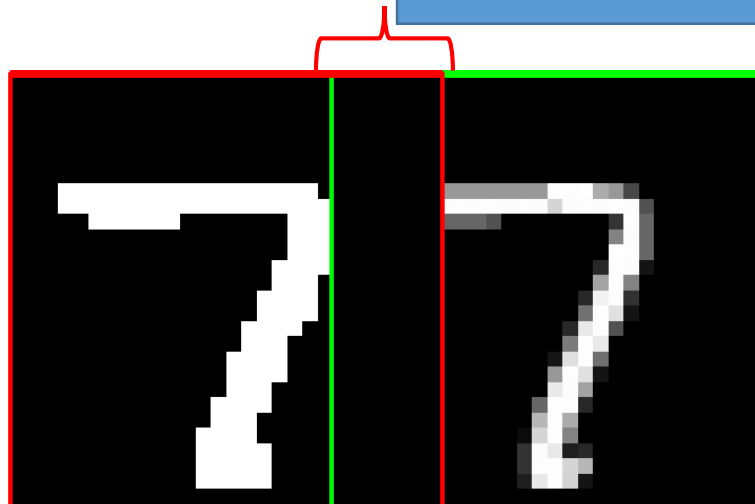
[illegible]

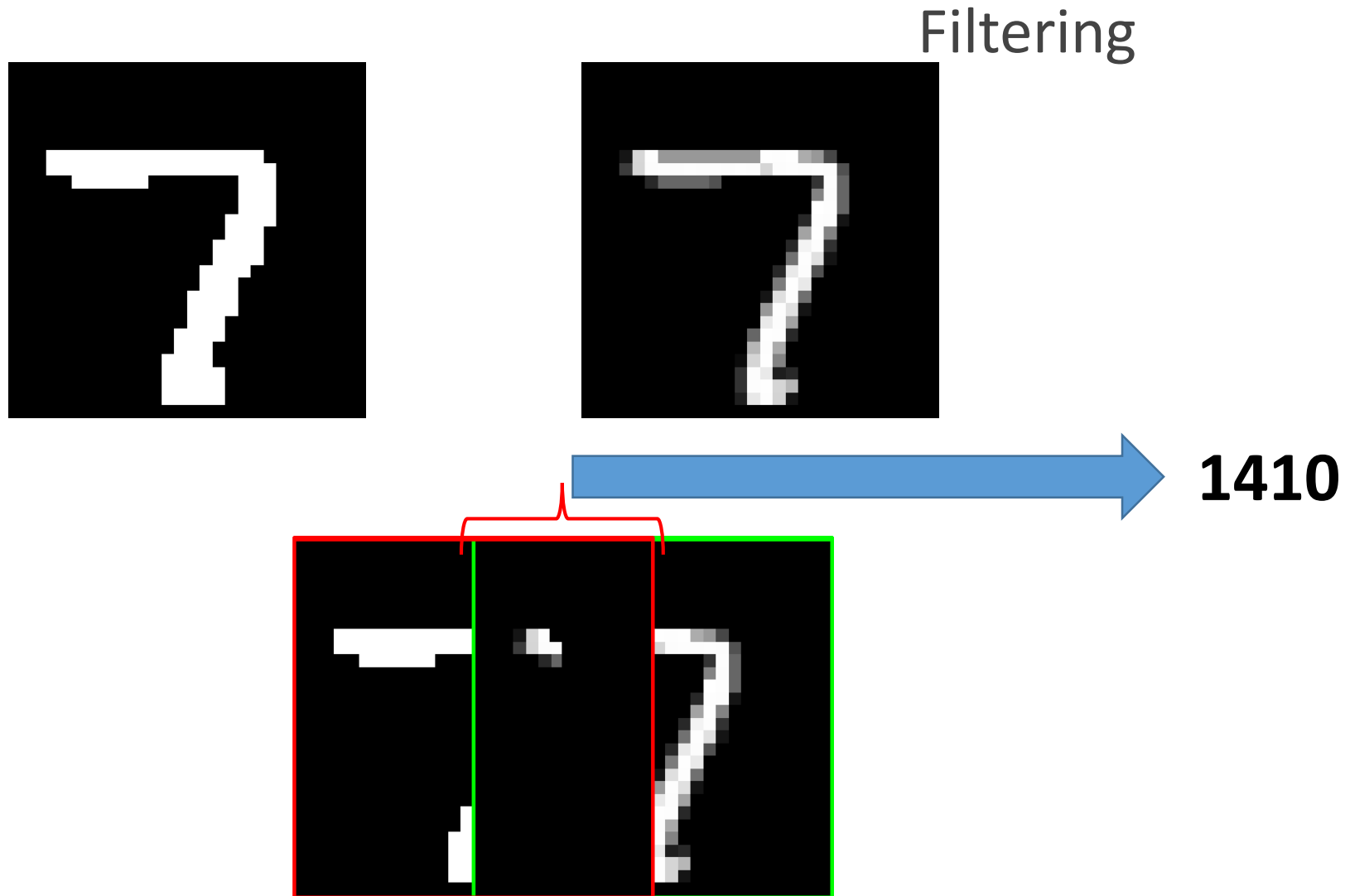
[illegible]

Filtering

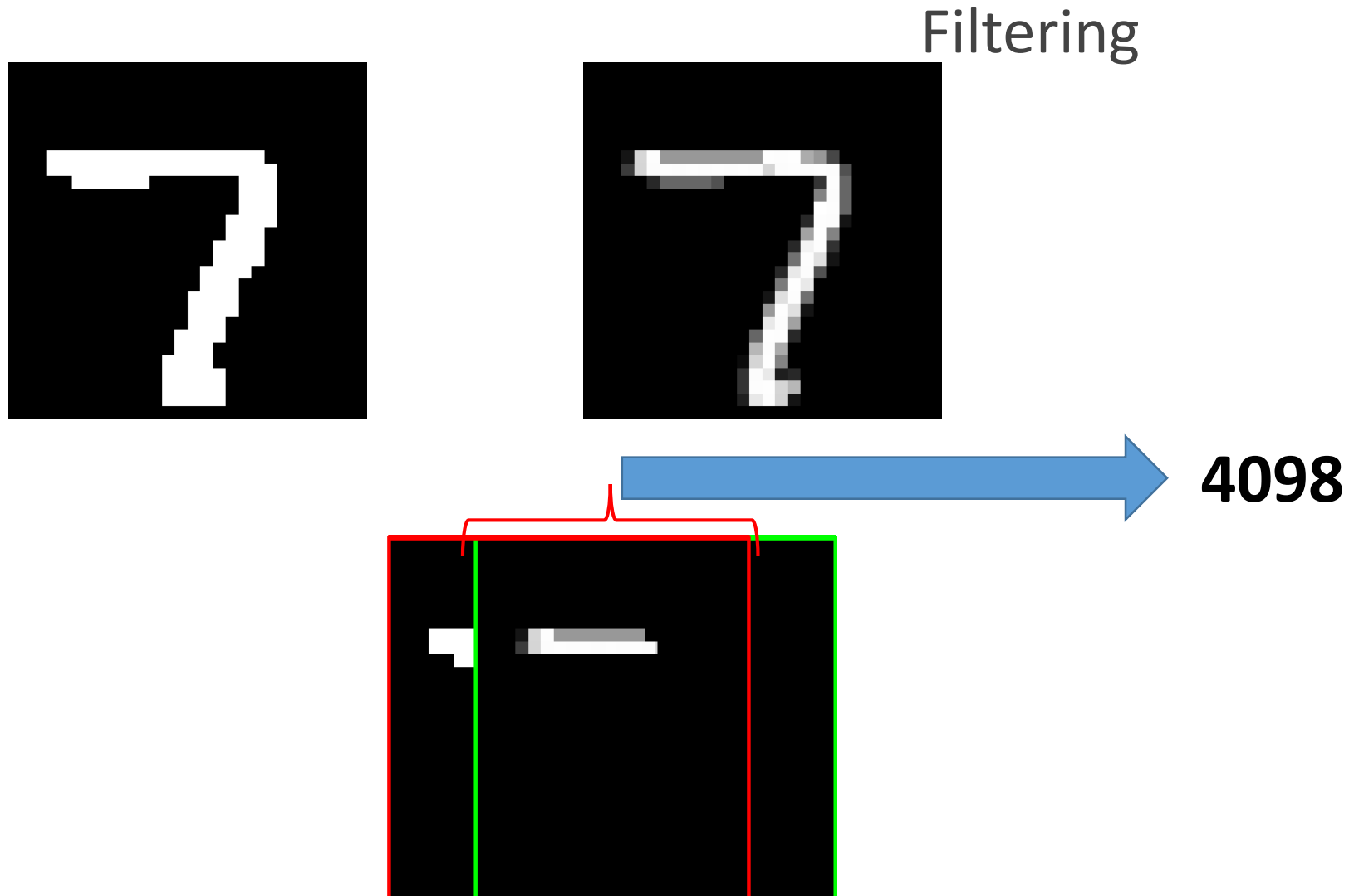


0

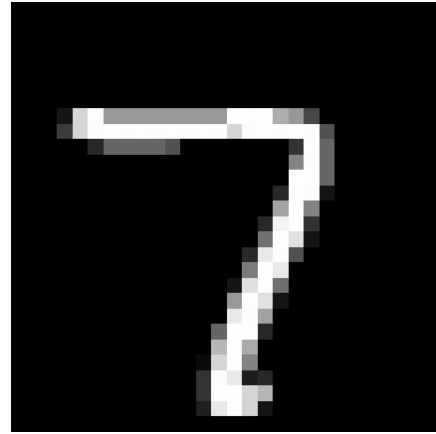
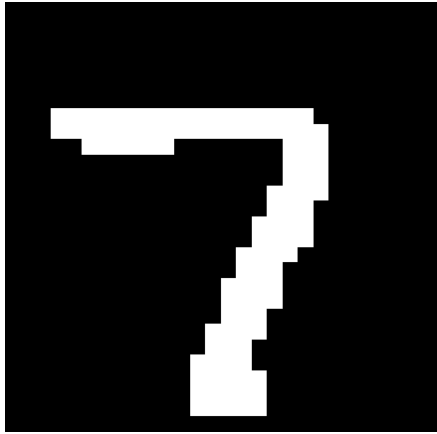




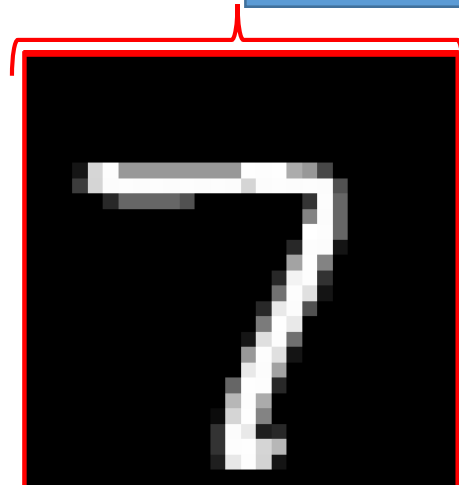




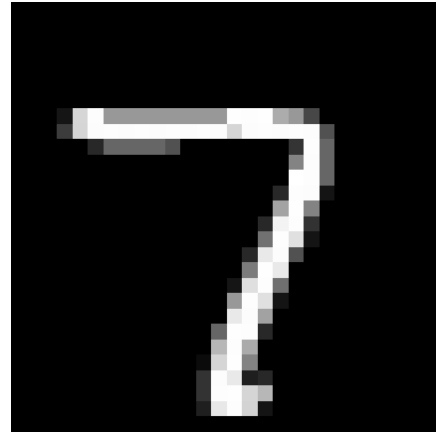
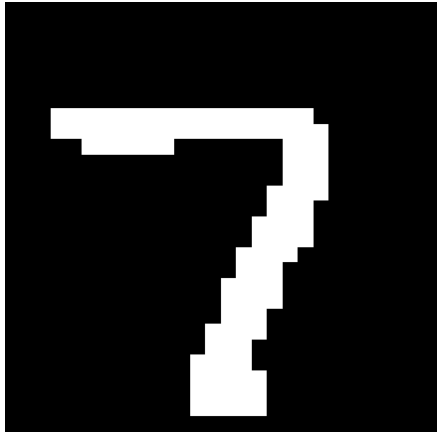
Filtering



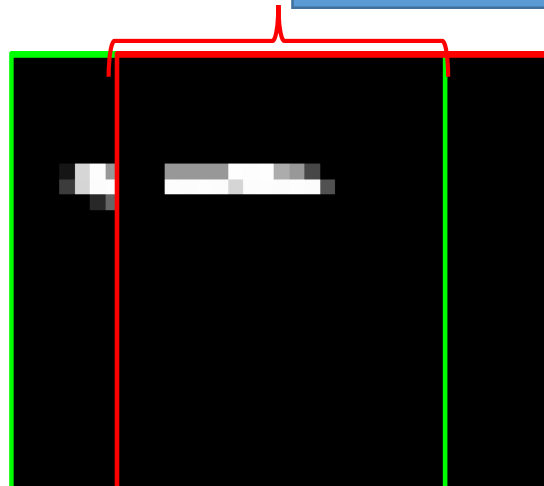
**18604**



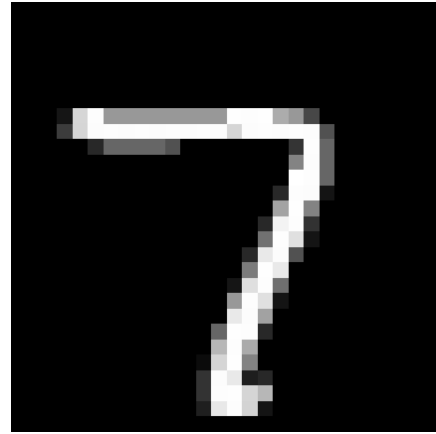
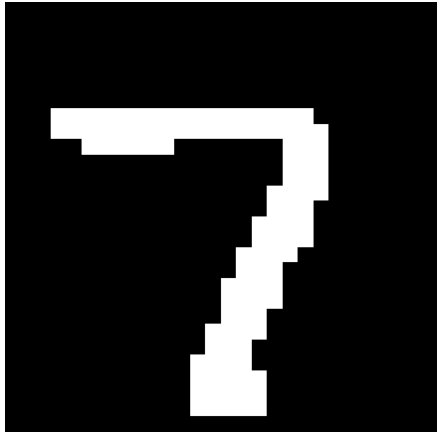
Filtering



4331



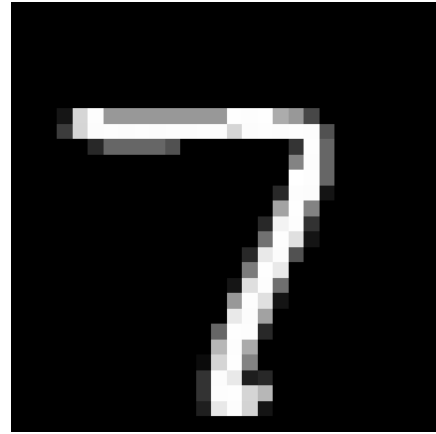
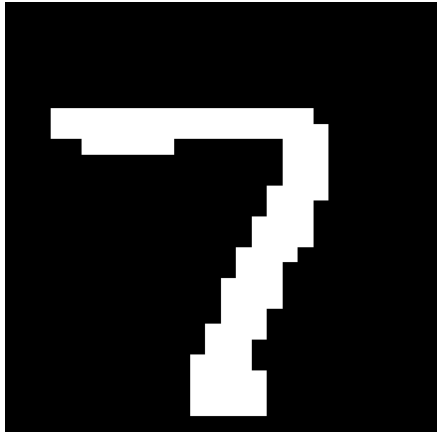
Filtering



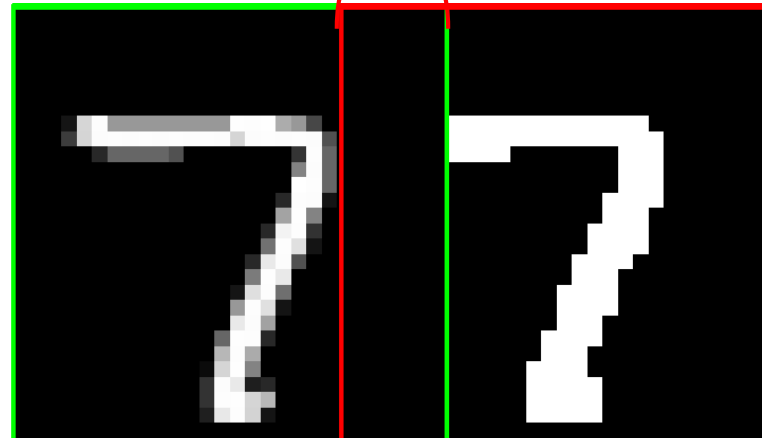
1589



Filtering

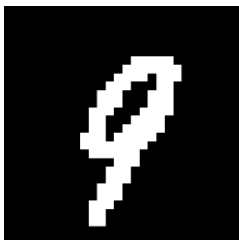
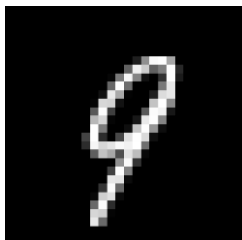
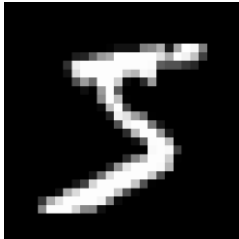
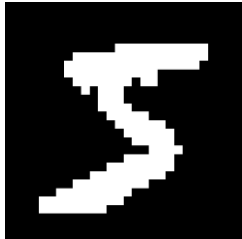
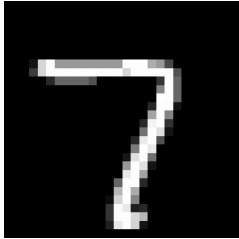
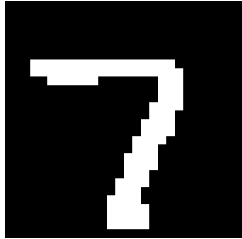


0



Filters

Test Images



## Classification by Matching Filters

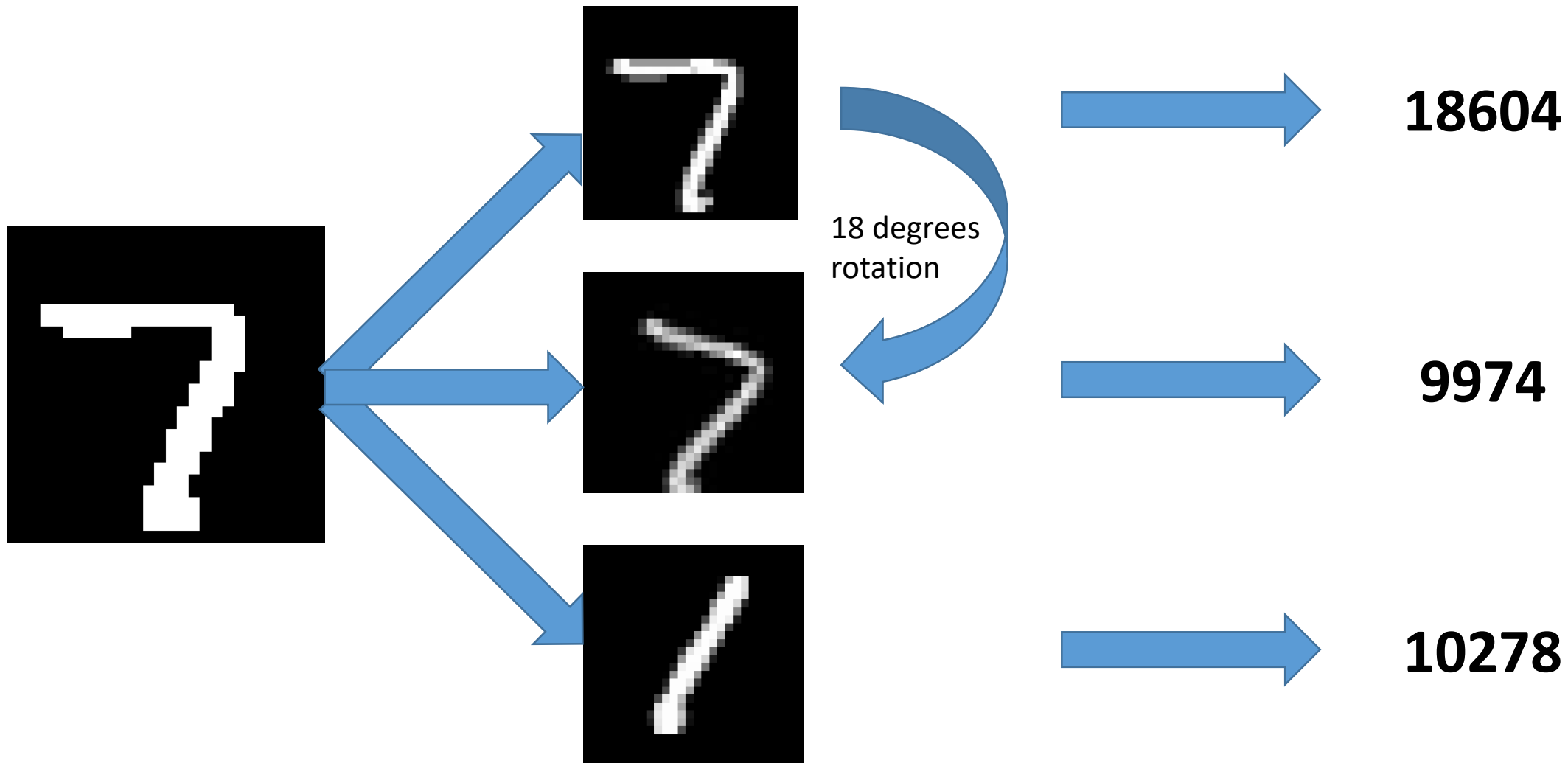
But what if the test image is a little

Rotated (or)

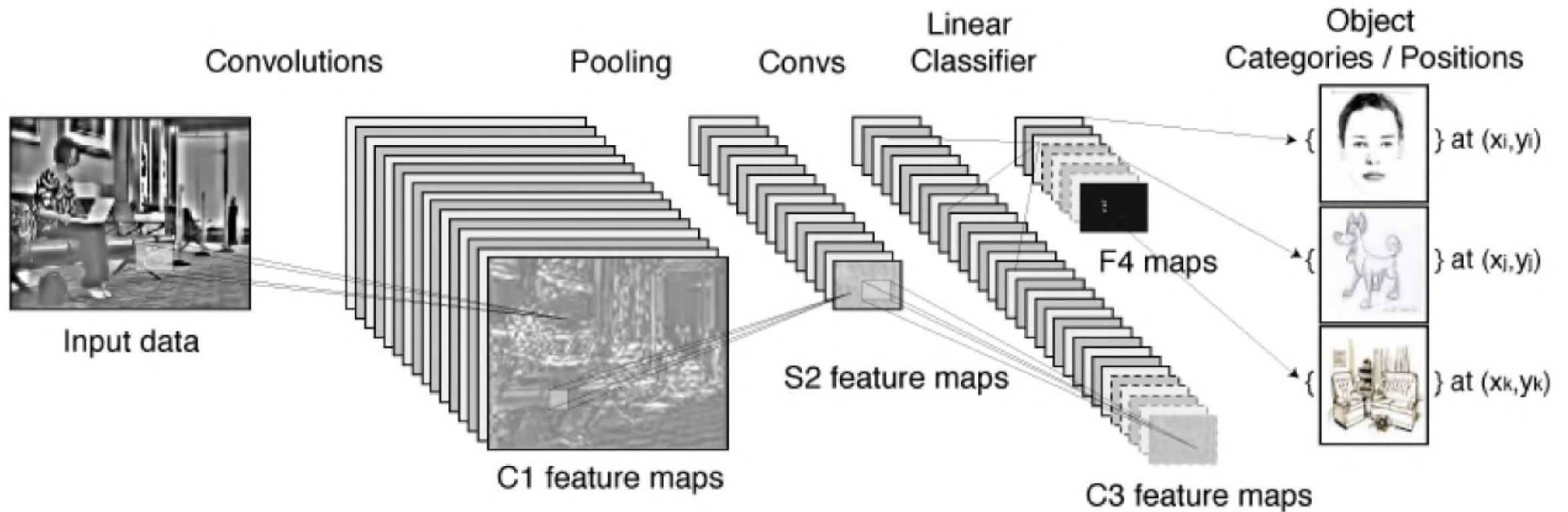
Skewed (or)

Zoomed out and so on

## Effect of Slight Rotation

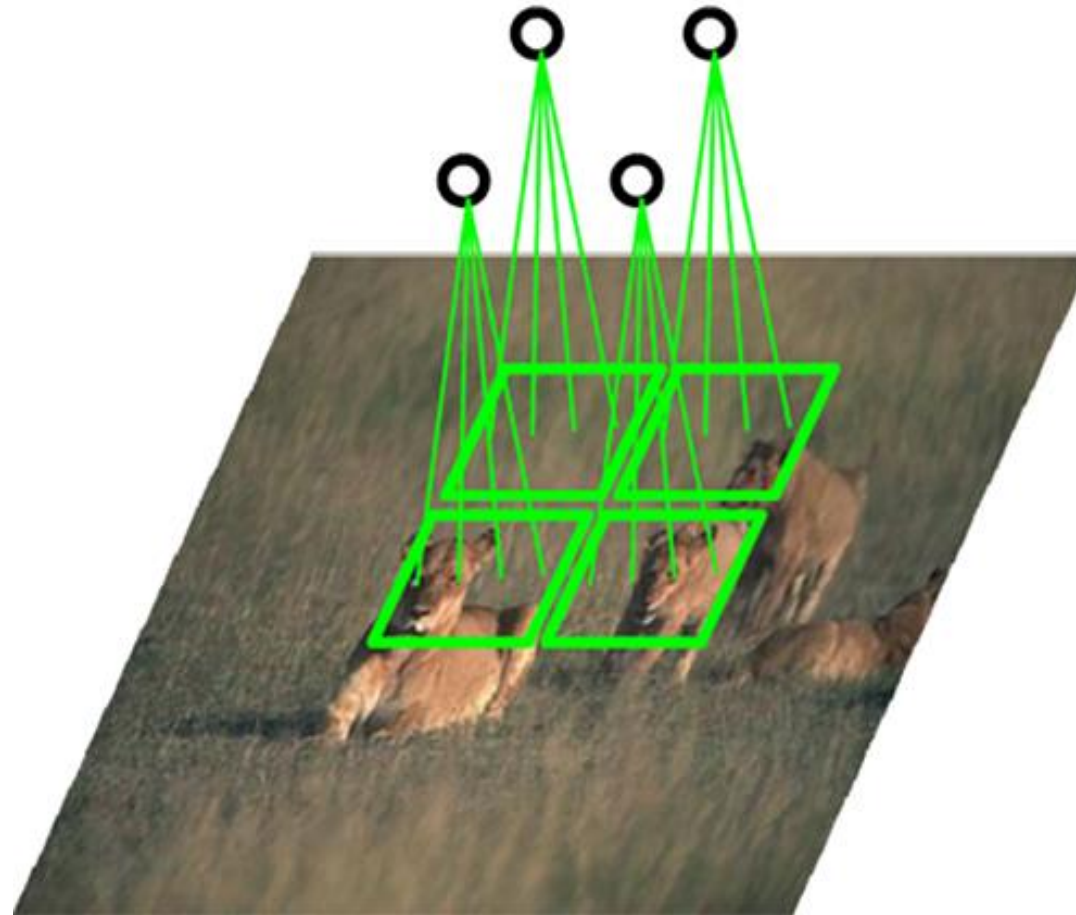


# Convnets (Fukushima, LeCun, Hinton)

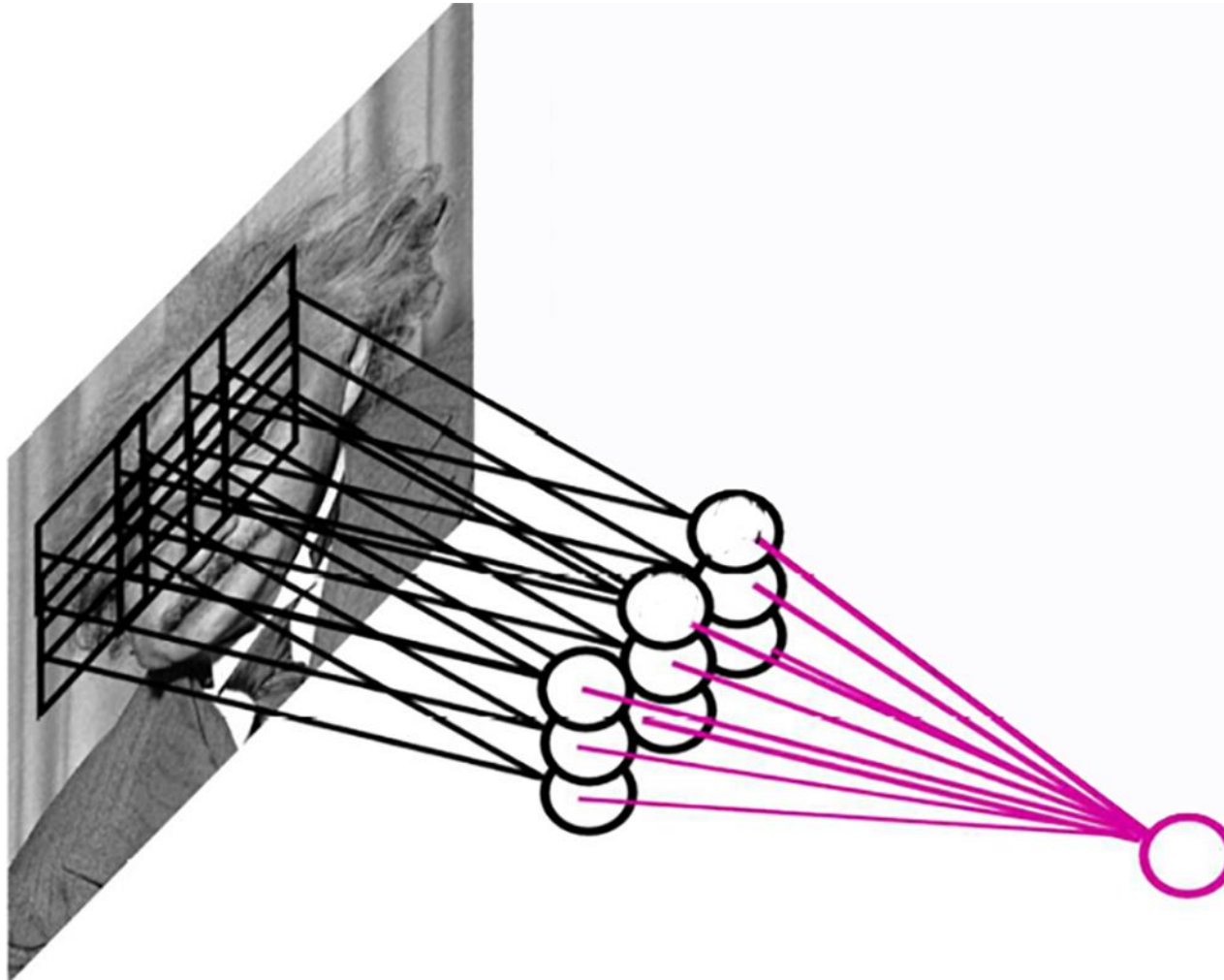




# Convnets (Fukushima, LeCun, Hinton)



# Pooling



By "pooling" (e.g., max or average) filter responses, we gain robustness to the exact spatial location of features.

# Convolution

Filter

0	1	2
2	2	0
0	1	2

Image

3	3	2	1	0
0	0	1	3	1
3	1	2	2	3
2	0	0	2	2
2	0	0	0	1

3 <sub>0</sub>	3 <sub>1</sub>	2 <sub>2</sub>	1	0
0 <sub>2</sub>	0 <sub>2</sub>	1 <sub>0</sub>	3	1
3 <sub>0</sub>	1 <sub>1</sub>	2 <sub>2</sub>	2	3
2	0	0	2	2
2	0	0	0	1

12.0		

3	3 <sub>0</sub>	2 <sub>1</sub>	1 <sub>2</sub>	0
0	0 <sub>2</sub>	1 <sub>2</sub>	3 <sub>0</sub>	1
3	1 <sub>0</sub>	2 <sub>1</sub>	2 <sub>2</sub>	3
2	0	0	2	2
2	0	0	0	1

12.0	12.0	

3	3	2 <sub>0</sub>	1 <sub>1</sub>	0 <sub>2</sub>
0	0	1 <sub>2</sub>	3 <sub>2</sub>	1 <sub>0</sub>
3	1	2 <sub>0</sub>	2 <sub>1</sub>	3 <sub>2</sub>
2	0	0	2	2
2	0	0	0	1

12.0	12.0	17.0

3	3	2	1	0
0 <sub>0</sub>	0 <sub>1</sub>	1 <sub>2</sub>	3	1
3 <sub>2</sub>	1 <sub>2</sub>	2 <sub>0</sub>	2	3
2 <sub>0</sub>	0 <sub>1</sub>	0 <sub>2</sub>	2	2
2	0	0	0	1

12.0	12.0	17.0
10.0		

3	3	2	1	0
0	0 <sub>0</sub>	1 <sub>1</sub>	3 <sub>2</sub>	1
3	1 <sub>2</sub>	2 <sub>2</sub>	2 <sub>0</sub>	3
2	0 <sub>0</sub>	0 <sub>1</sub>	2 <sub>2</sub>	2
2	0	0	0	1

12.0	12.0	17.0
10.0	17.0	

3	3	2	1	0
0	0	1 <sub>0</sub>	3 <sub>1</sub>	1 <sub>2</sub>
3	1	2 <sub>2</sub>	2 <sub>2</sub>	3 <sub>0</sub>
2	0	0 <sub>0</sub>	2 <sub>1</sub>	2 <sub>2</sub>
2	0	0	0	1

12.0	12.0	17.0
10.0	17.0	19.0

3	3	2	1	0
0	0	1	3	1
3 <sub>0</sub>	1 <sub>1</sub>	2 <sub>2</sub>	2	3
2 <sub>2</sub>	0 <sub>2</sub>	0 <sub>0</sub>	2	2
2 <sub>0</sub>	0 <sub>1</sub>	0 <sub>2</sub>	0	1

12.0	12.0	17.0
10.0	17.0	19.0
9.0		

3	3	2	1	0
0	0	1	3	1
3	1 <sub>0</sub>	2 <sub>1</sub>	2 <sub>2</sub>	3
2	0 <sub>2</sub>	0 <sub>2</sub>	2 <sub>0</sub>	2
2	0 <sub>0</sub>	0 <sub>1</sub>	0 <sub>2</sub>	1

12.0	12.0	17.0
10.0	17.0	19.0
9.0	6.0	

3	3	2	1	0
0	0	1	3	1
3	1	2 <sub>0</sub>	2 <sub>1</sub>	3 <sub>2</sub>
2	0	0 <sub>2</sub>	2 <sub>2</sub>	2 <sub>0</sub>
2	0	0 <sub>0</sub>	0 <sub>1</sub>	1 <sub>2</sub>

12.0	12.0	17.0
10.0	17.0	19.0
9.0	6.0	14.0

Image courtesy: Vincent Dumoulin

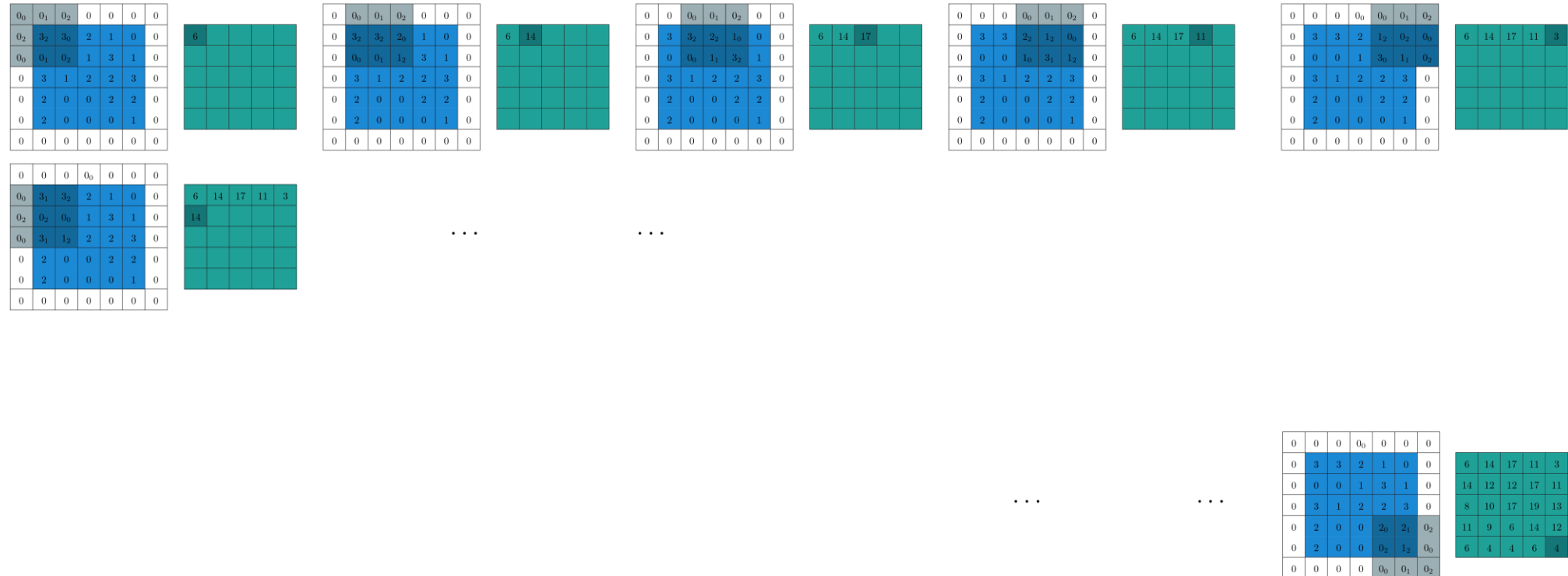
# Convolution with Zero Padding

Filter

0	1	2
2	2	0
0	1	2

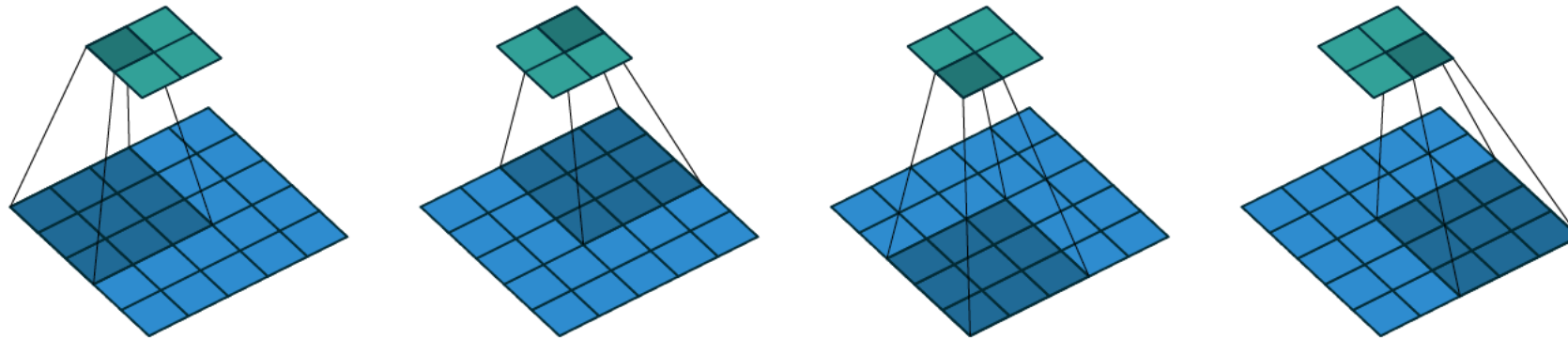
Image

3	3	2	1	0
0	0	1	3	1
3	1	2	2	3
2	0	0	2	2
2	0	0	0	1



# Convolution with Strides

Convolving a  $3 \times 3$  kernel over a  $5 \times 5$  input using  $2 \times 2$  strides



# Convolution with Strides and Zero Padding

Convolving a  $3 \times 3$  kernel over a  $5 \times 5$  input using  $1 \times 1$  strides and half padding

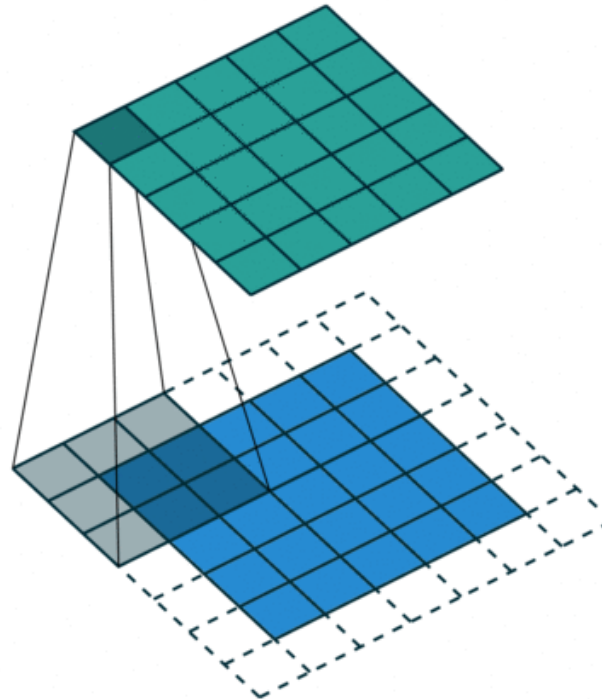


Image courtesy: Vincent Dumoulin

# Inputs Generally have Multiple Channels

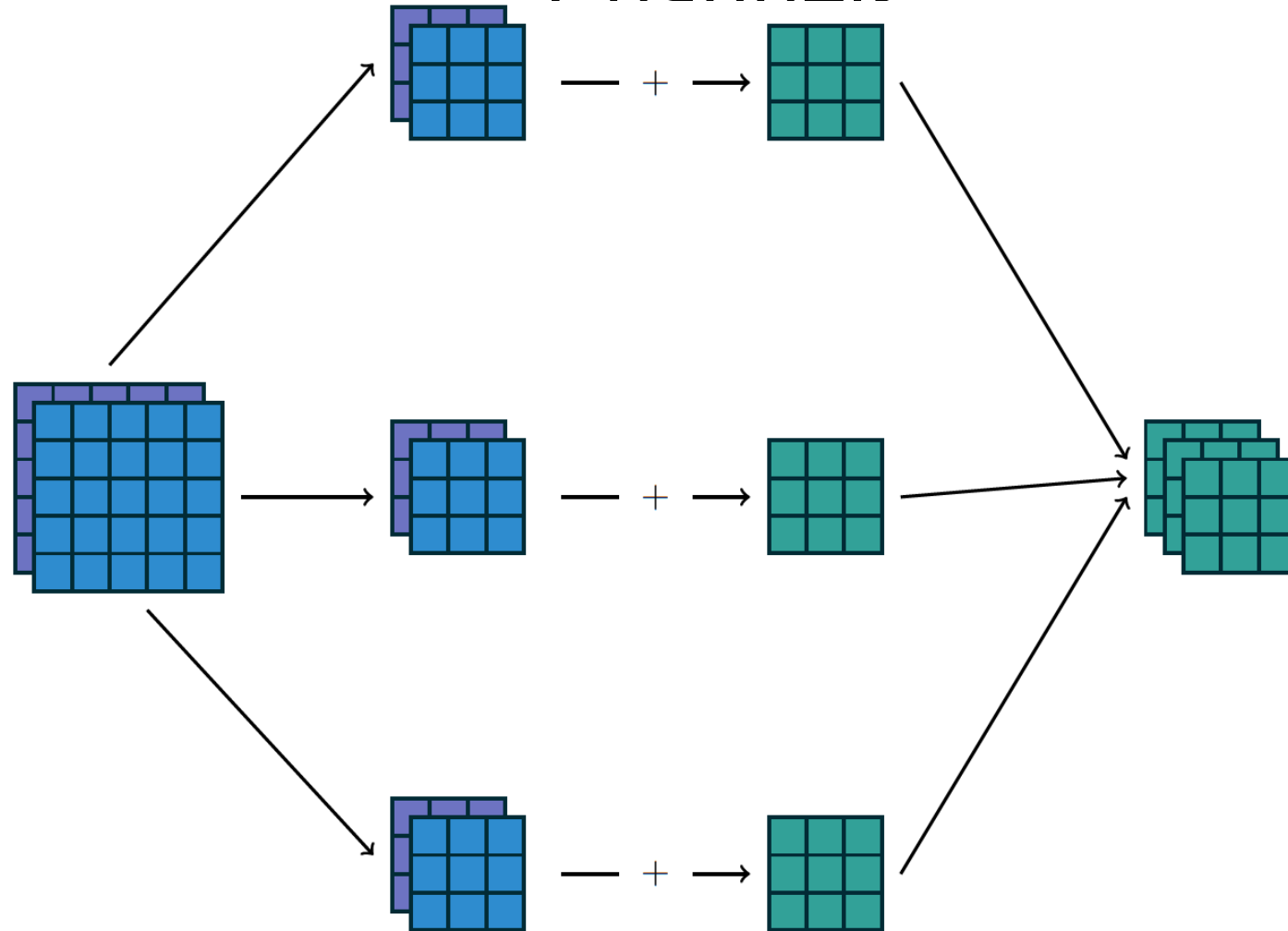


Image courtesy: Vincent Dumoulin

# Convolution Arithmetic

(For simplicity we are assuming **square** image and filter/kernel)

Image width = image height =  $w$

Filter width = Filter height =  $k$

Stride =  $s$

Output size =  $\left\lfloor \frac{w-k}{s} \right\rfloor + 1$

Padding =  $p \rightarrow$  This means image dimension becomes  $w + 2p$

So, output size =  $\left\lfloor \frac{w+2p-k}{s} \right\rfloor + 1$

Note the box function

$$w = 6, p = 1, k = 3, s = 2$$

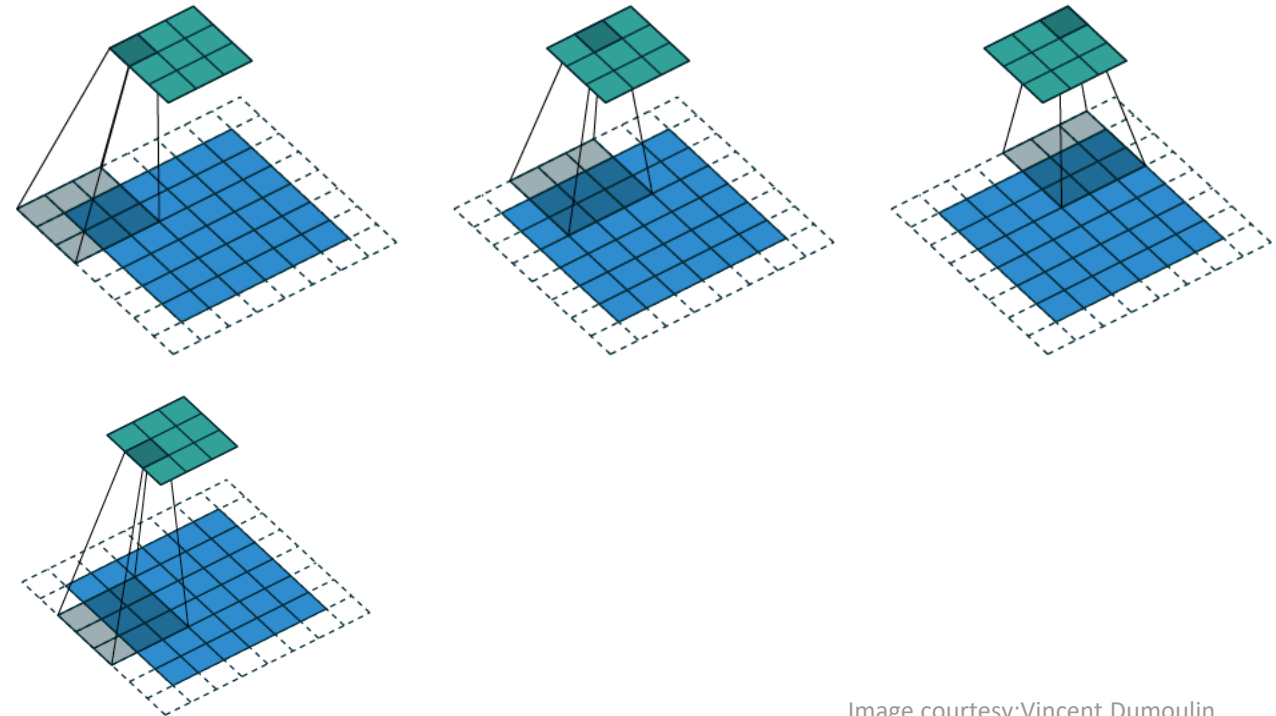
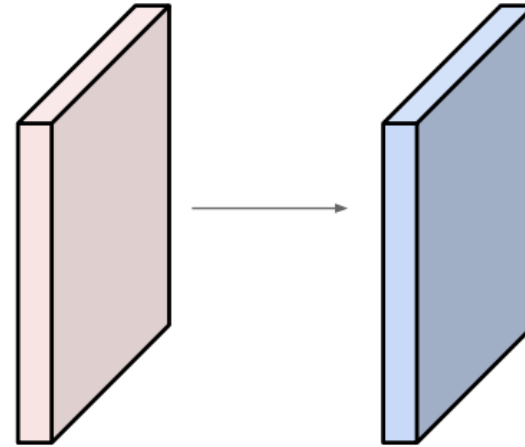


Image courtesy: Vincent Dumoulin



# Convolution Arithmetic

Input volume:  $32 \times 32 \times 3$  [w, h, c].  
64 filters of size  $3 \times 3$  [k, k] with  
stride 2 [s], pad 1 [p]



What is the output feature map size?

$$\left\lfloor \frac{32 + 2 * 1 - 3}{2} \right\rfloor + 1 = 16$$

So,  $16 \times 16 \times 64$  [w, h, c]

And What is the number of  
parameters in this convolution layer?

$$64 \times 3 \times 3 \times \textcolor{red}{3} [c\_out, w, h, c\_in] = 1728$$

# Pooling

3	3	2	1	0
0	0	1	3	1
3	1	2	2	3
2	0	0	2	2
2	0	0	0	1

3.0		

3	3	2	1	0
0	0	1	3	1
3	1	2	2	3
2	0	0	2	2
2	0	0	0	1

3.0	3.0	

3	3	2	1	0
0	0	1	3	1
3	1	2	2	3
2	0	0	2	2
2	0	0	0	1

3.0	3.0	3.0

3x3 max-pooling on  
5x5 input with  
1x1 stride

3	3	2	1	0
0	0	1	3	1
3	1	2	2	3
2	0	0	2	2
2	0	0	0	1

3.0	3.0	3.0
3.0		

3	3	2	1	0
0	0	1	3	1
3	1	2	2	3
2	0	0	2	2
2	0	0	0	1

3.0	3.0	3.0
3.0	3.0	

3	3	2	1	0
0	0	1	3	1
3	1	2	2	3
2	0	0	2	2
2	0	0	0	1

3.0	3.0	3.0
3.0	3.0	3.0

3	3	2	1	0
0	0	1	3	1
3	1	2	2	3
2	0	0	2	2
2	0	0	0	1

3.0	3.0	3.0
3.0	3.0	3.0
3.0		

3	3	2	1	0
0	0	1	3	1
3	1	2	2	3
2	0	0	2	2
2	0	0	0	1

3.0	3.0	3.0
3.0	3.0	3.0
3.0	2.0	

3	3	2	1	0
0	0	1	3	1
3	1	2	2	3
2	0	0	2	2
2	0	0	0	1

3.0	3.0	3.0
3.0	3.0	3.0
3.0	2.0	3.0

Image courtesy: Vincent Dumoulin

# Pooling Arithmetic

(For simplicity we are assuming **square** input and max pooling kernel)

Input width = Input height =  $w$

Filter width = Filter height =  $k$

Stride =  $s$

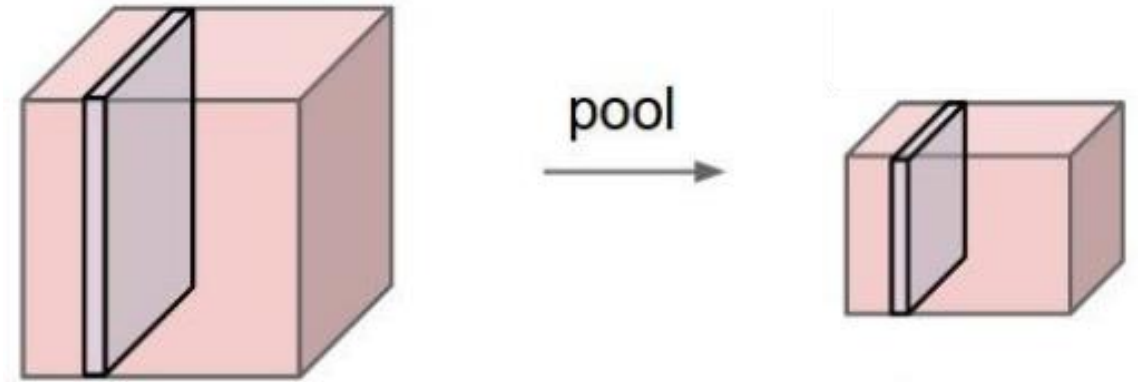
$$\text{Output size} = \left\lfloor \frac{w-k}{s} \right\rfloor + 1$$

Input volume:  $32 \times 32 \times 3$  [ $w, h, c$ ].

Max-pooling kernel of size  $2 \times 2$  [ $k, k$ ] with stride 2 [ $s$ ]

What is the output feature map size?

$$\left\lfloor \frac{32-2}{2} \right\rfloor + 1 = 16 \quad \text{So, } 16 \times 16 \times 3 \text{ [w, h, c]}$$

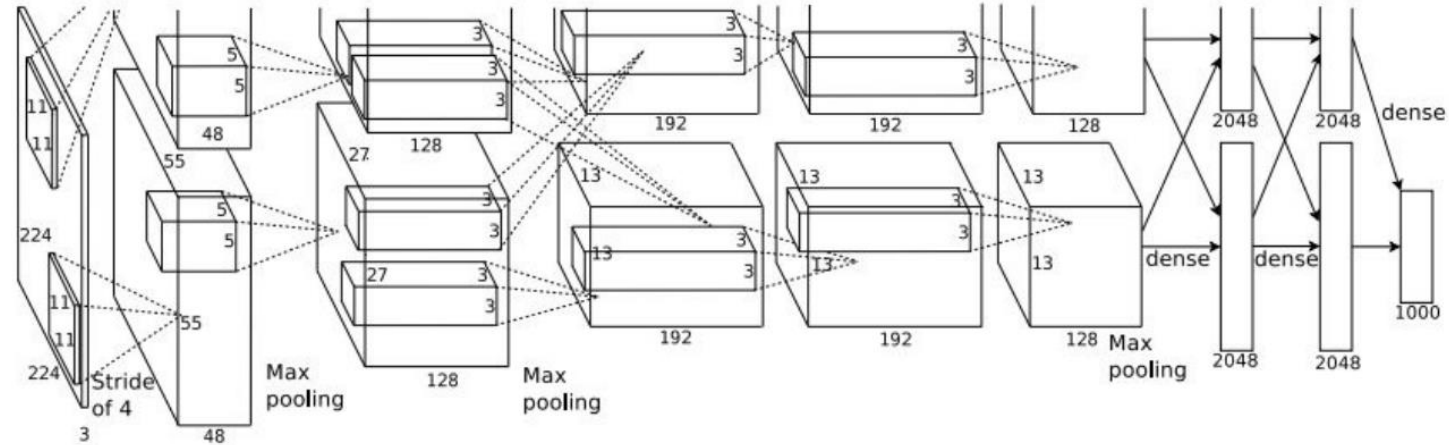


And What is the number of parameters in this pooling layer?

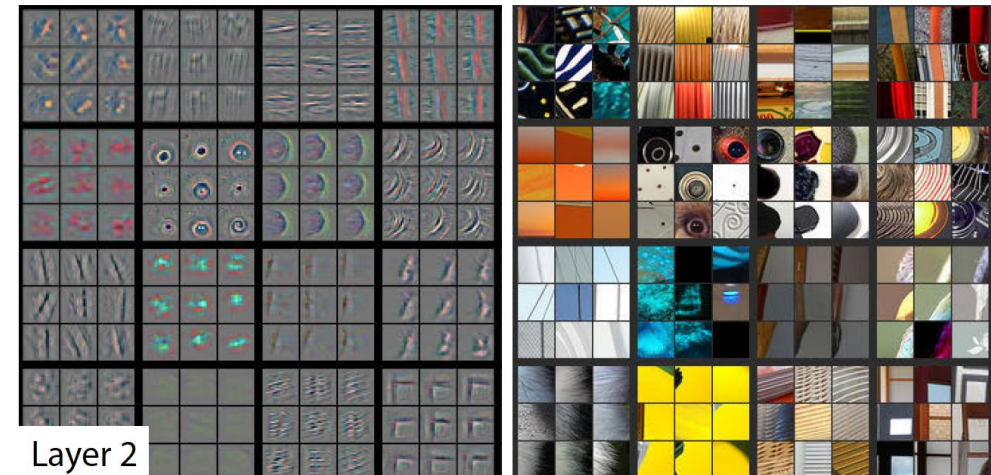
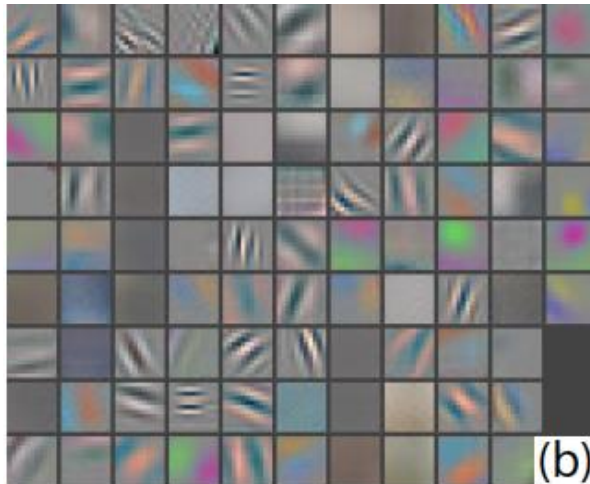
0

# Visualizations

AlexNet (2012)



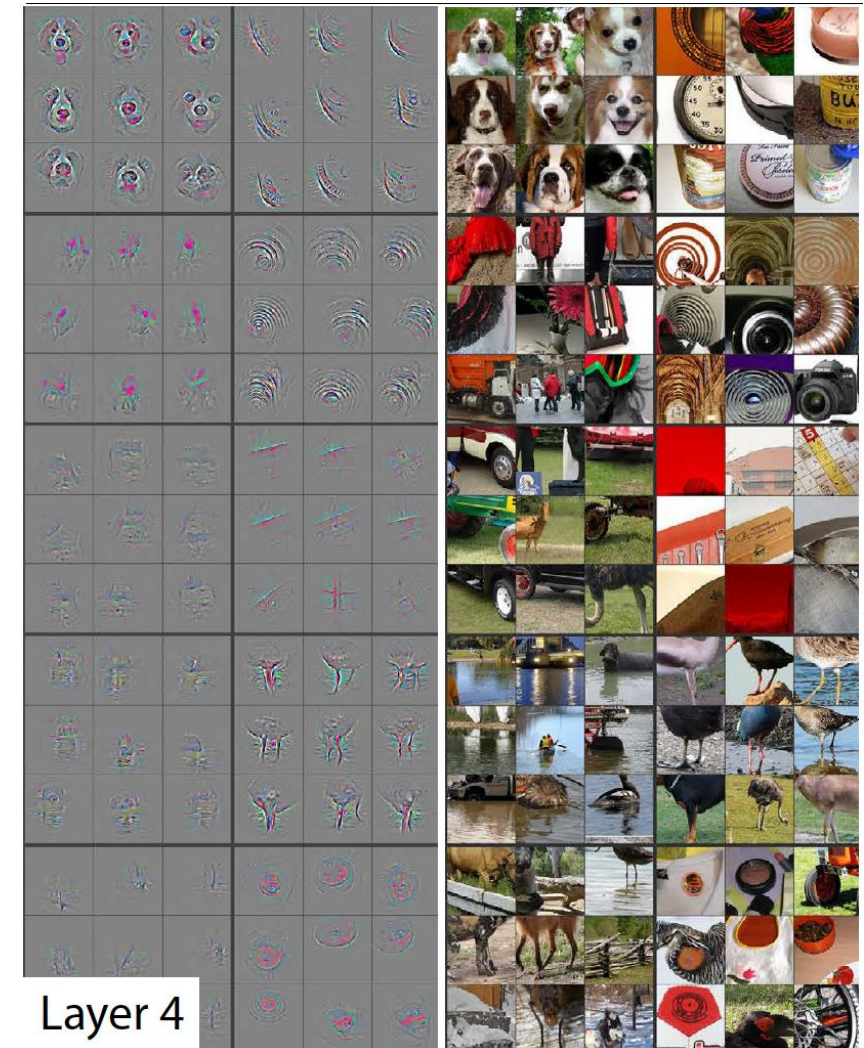
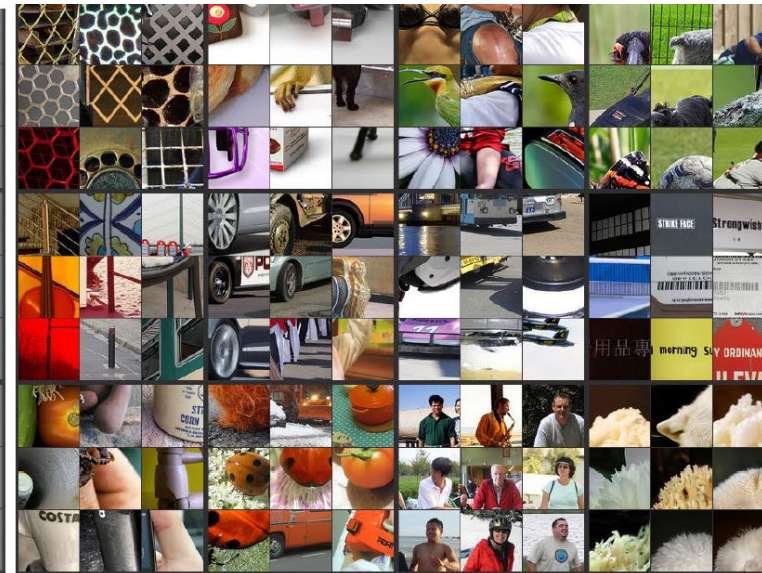
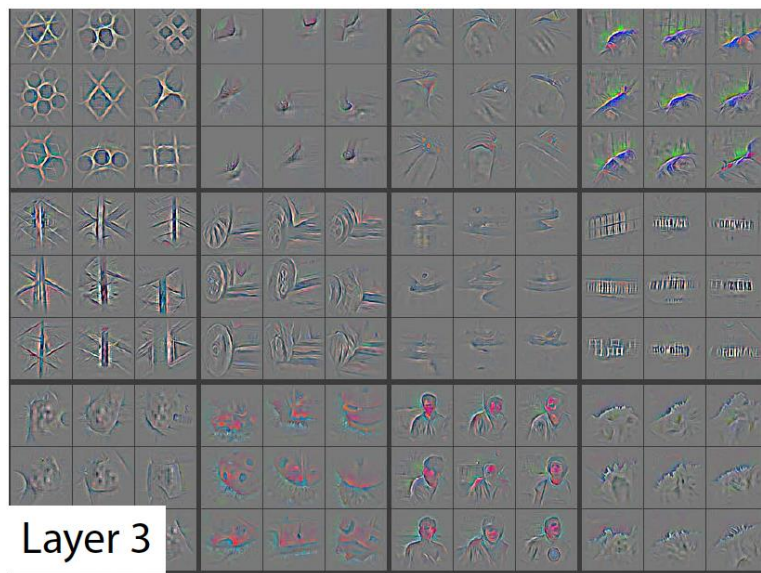
First layer (CONV1): 96 11x11 filters



Layer 2



# Visualizations



<https://distill.pub/2018/building-blocks/>

## The Building Blocks of Interpretability

Interpretability techniques are normally studied in isolation.

We explore the powerful interfaces that arise when you combine them —  
and the rich structure of this combinatorial space.

CHOOSE AN INPUT IMAGE



For instance, by combining feature visualization (*what is a neuron looking for?*) with attribution (*how does it affect the output?*), we can explore how the network decides between labels like **Labrador retriever** and **tiger cat**.



Several floppy ear detectors seem to be important when distinguishing dogs, whereas pointy ears are used to classify "tiger cat".

CHANNELS THAT MOST SUPPORT ...

LABRADOR RETRIEVER

TIGER CAT