Deep Learning
CS60010

Abir Das
Assistant Professor
Computer Science and Engineering Department
Indian Institute of Technology Kharagpur

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

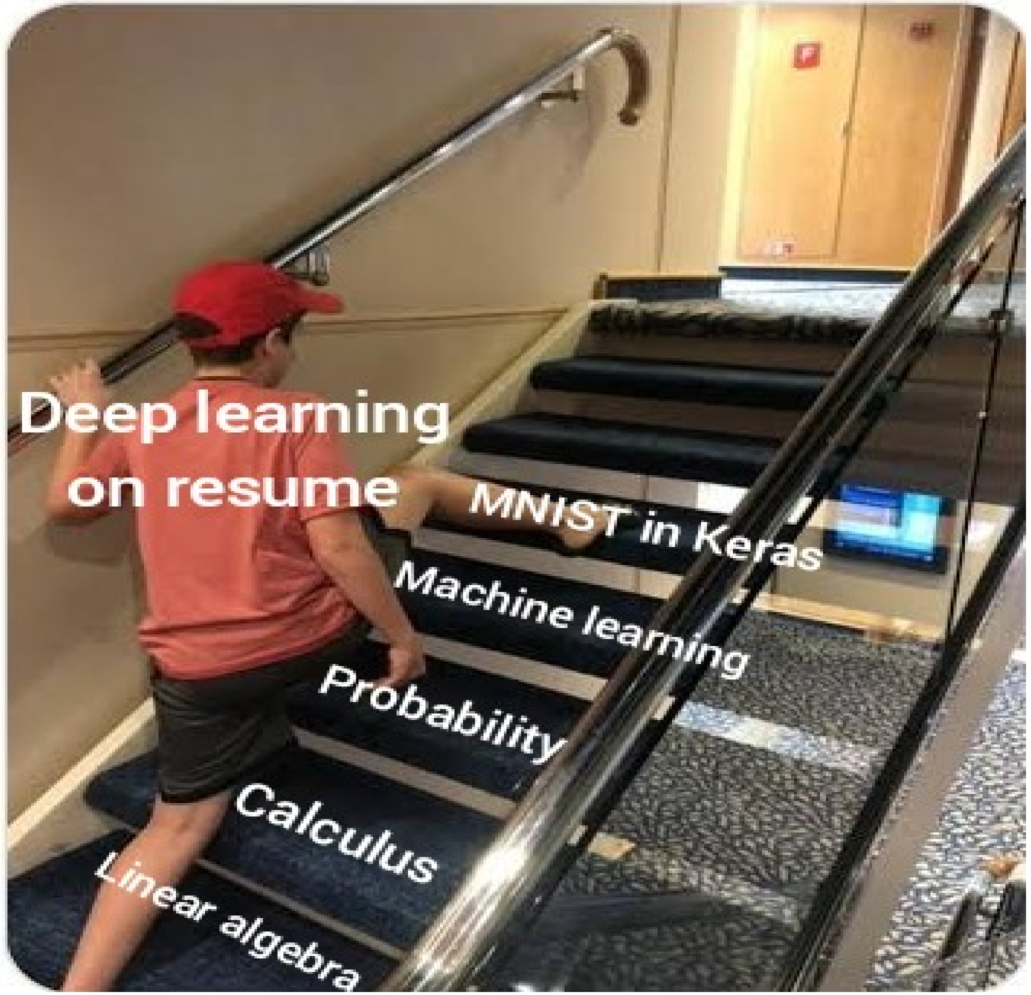
Announcements

- Class Test 1 is graded.
- Solutions are also uploaded.
- Login to the moodle server where you gave the exam. In gradebooks you will get to see the marks.
- Ignore the percentages. Just the marks obtained will be taken to compute your grades at the end. The percentages shown here are meaningless.

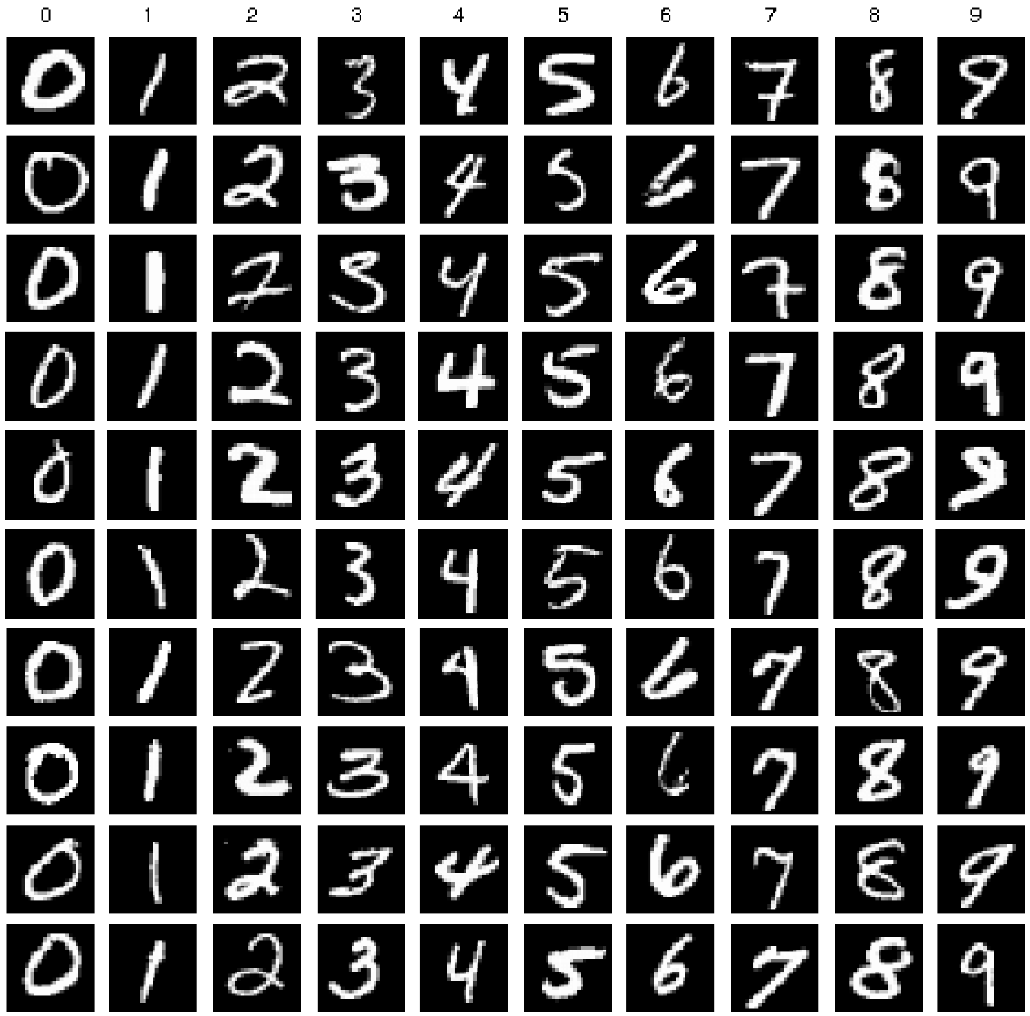
Agenda

The Building Blocks of Convolutional Neural Networks/CNNs/ConvNets

Importance of MNIST



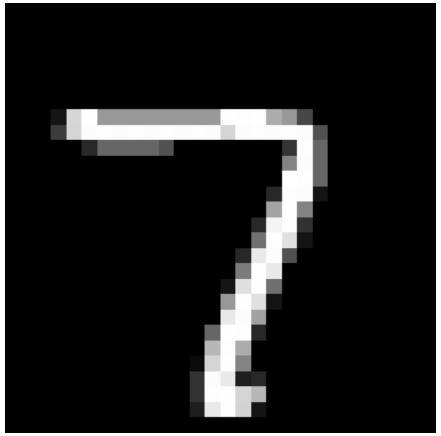
MNIST

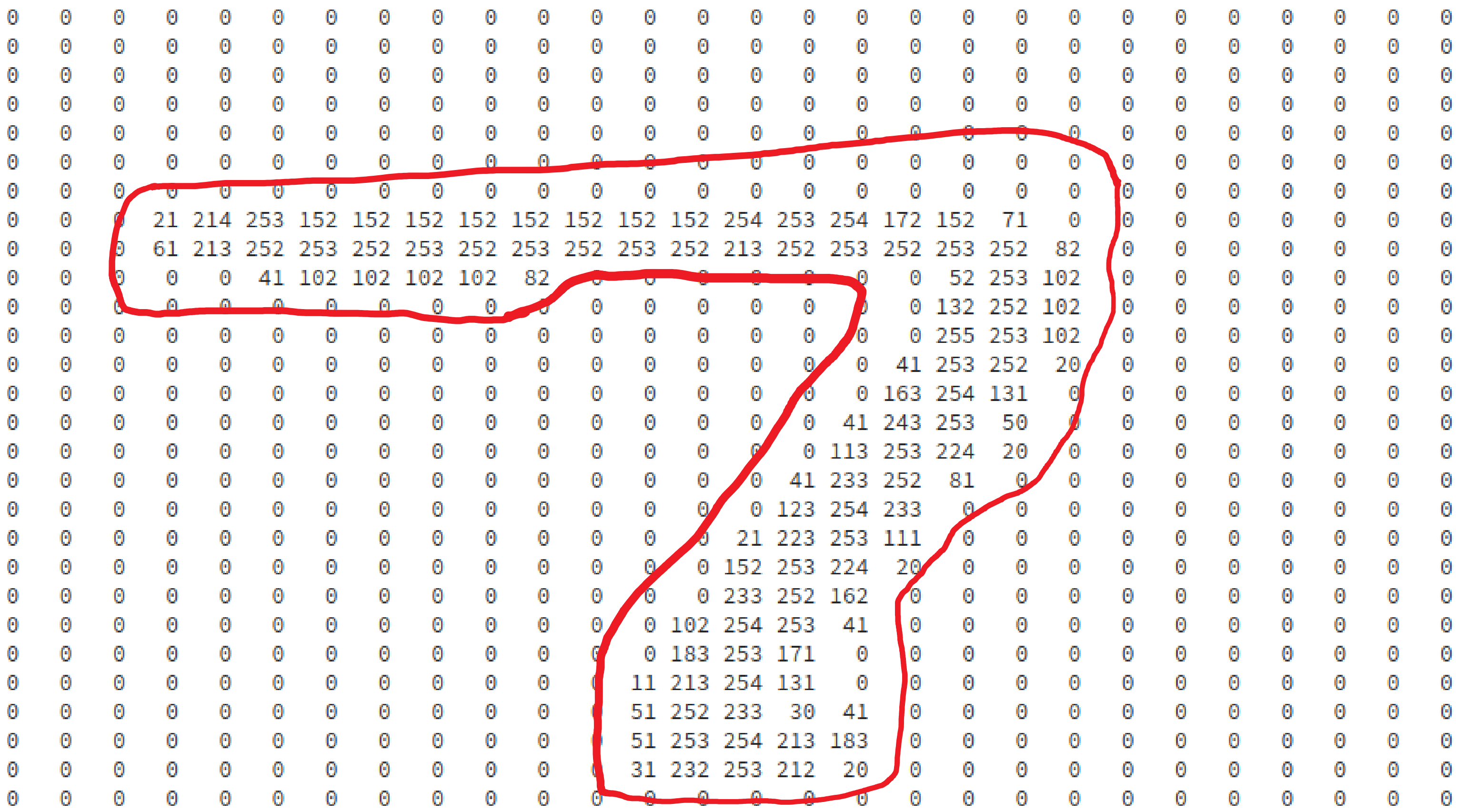


- database of handwritten digits
- 10 classes
  - Training set 60,000 images
  - Test set of 10,000 images
- Greyscale images of size 28x28
- Often treated as `Hello World’ for any ML/DL practioner

Image taken from Researchgate

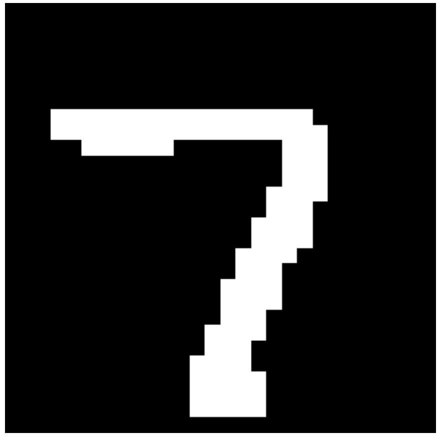
Example from MNIST





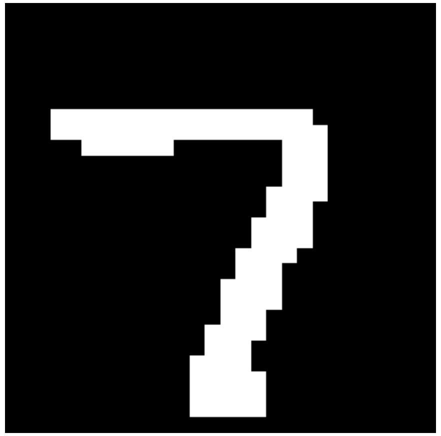
```
   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
   0   0   0  21 214 253 152 152 152 152 152 152 152 152 254 253 254 172 152  71   0   0   0   0   0   0   0   0
   0   0   0  61 213 252 253 252 253 252 253 252 253 252 213 252 253 252 253 252  82   0   0   0   0   0   0   0
   0   0   0   0   0  41 102 102 102 102  82   0   0   0   0   0   0   0  52 253 102   0   0   0   0   0   0   0
   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0 132 252 102   0   0   0   0   0   0   0
   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0 255 253 102   0   0   0   0   0   0   0
   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0  41 253 252  20   0   0   0   0   0   0   0
   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0 163 254 131   0   0   0   0   0   0   0   0
   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0  41 243 253  50   0   0   0   0   0   0   0   0
   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0 113 253 224  20   0   0   0   0   0   0   0   0
   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0  41 233 252  81   0   0   0   0   0   0   0   0   0
   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0 123 254 233   0   0   0   0   0   0   0   0   0   0
   0   0   0   0   0   0   0   0   0   0   0   0   0   0  21 223 253 111   0   0   0   0   0   0   0   0   0   0
   0   0   0   0   0   0   0   0   0   0   0   0   0   0 152 253 224  20   0   0   0   0   0   0   0   0   0   0
   0   0   0   0   0   0   0   0   0   0   0   0   0   0 233 252 162   0   0   0   0   0   0   0   0   0   0   0
   0   0   0   0   0   0   0   0   0   0   0   0   0 102 254 253  41   0   0   0   0   0   0   0   0   0   0   0
   0   0   0   0   0   0   0   0   0   0   0   0   0 183 253 171   0   0   0   0   0   0   0   0   0   0   0   0
   0   0   0   0   0   0   0   0   0   0   0   0  11 213 254 131   0   0   0   0   0   0   0   0   0   0   0   0
   0   0   0   0   0   0   0   0   0   0   0   0  51 252 233  30  41   0   0   0   0   0   0   0   0   0   0   0
   0   0   0   0   0   0   0   0   0   0   0   0  51 253 254 213 183   0   0   0   0   0   0   0   0   0   0   0
   0   0   0   0   0   0   0   0   0   0   0   0  31 232 253 212  20   0   0   0   0   0   0   0   0   0   0   0
   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
```

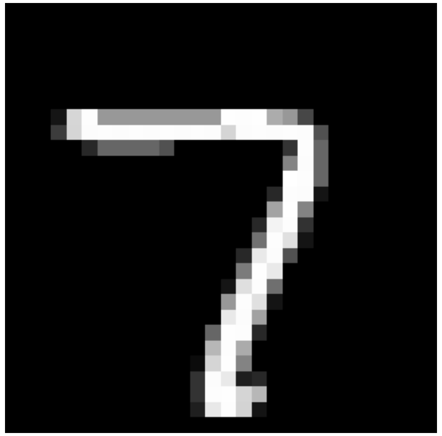
7 Like Window?

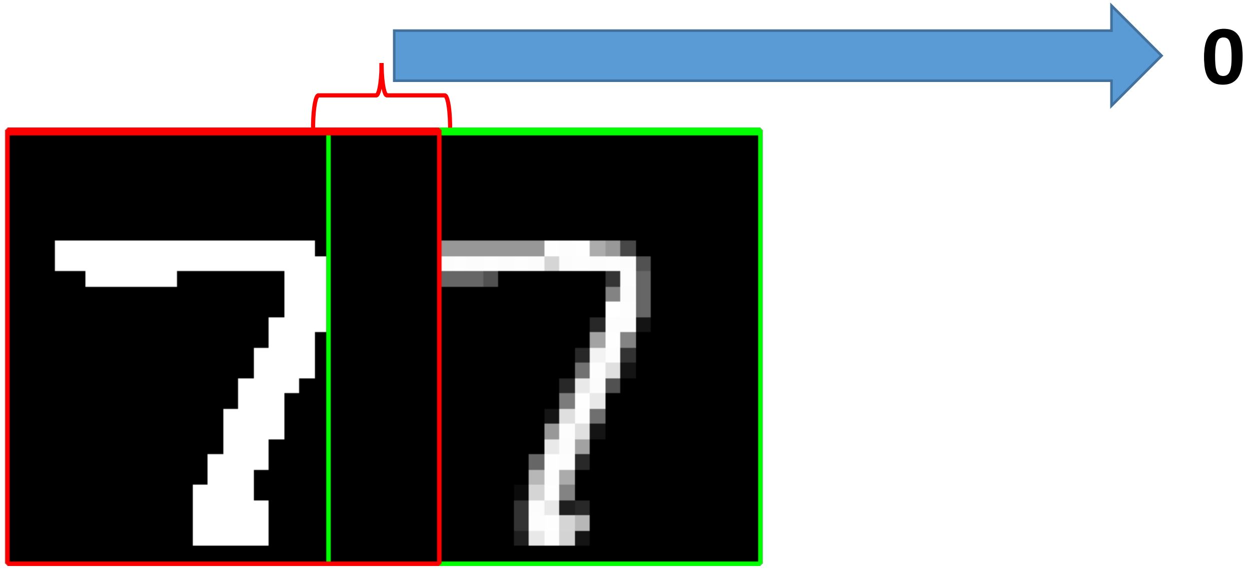


```
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0
0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0
0 0 0 0 0 1 1 1 1 1 1 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

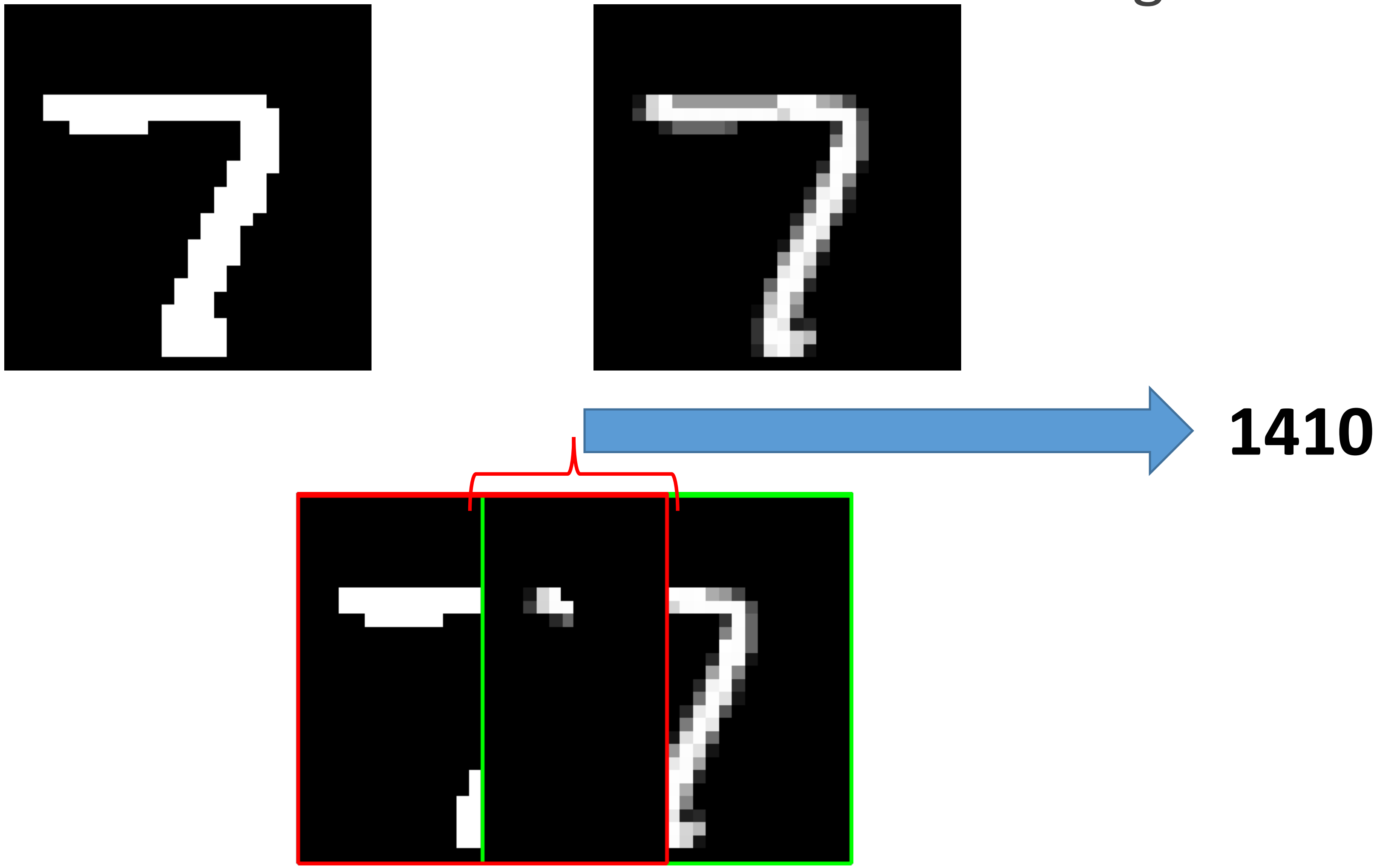
Filtering



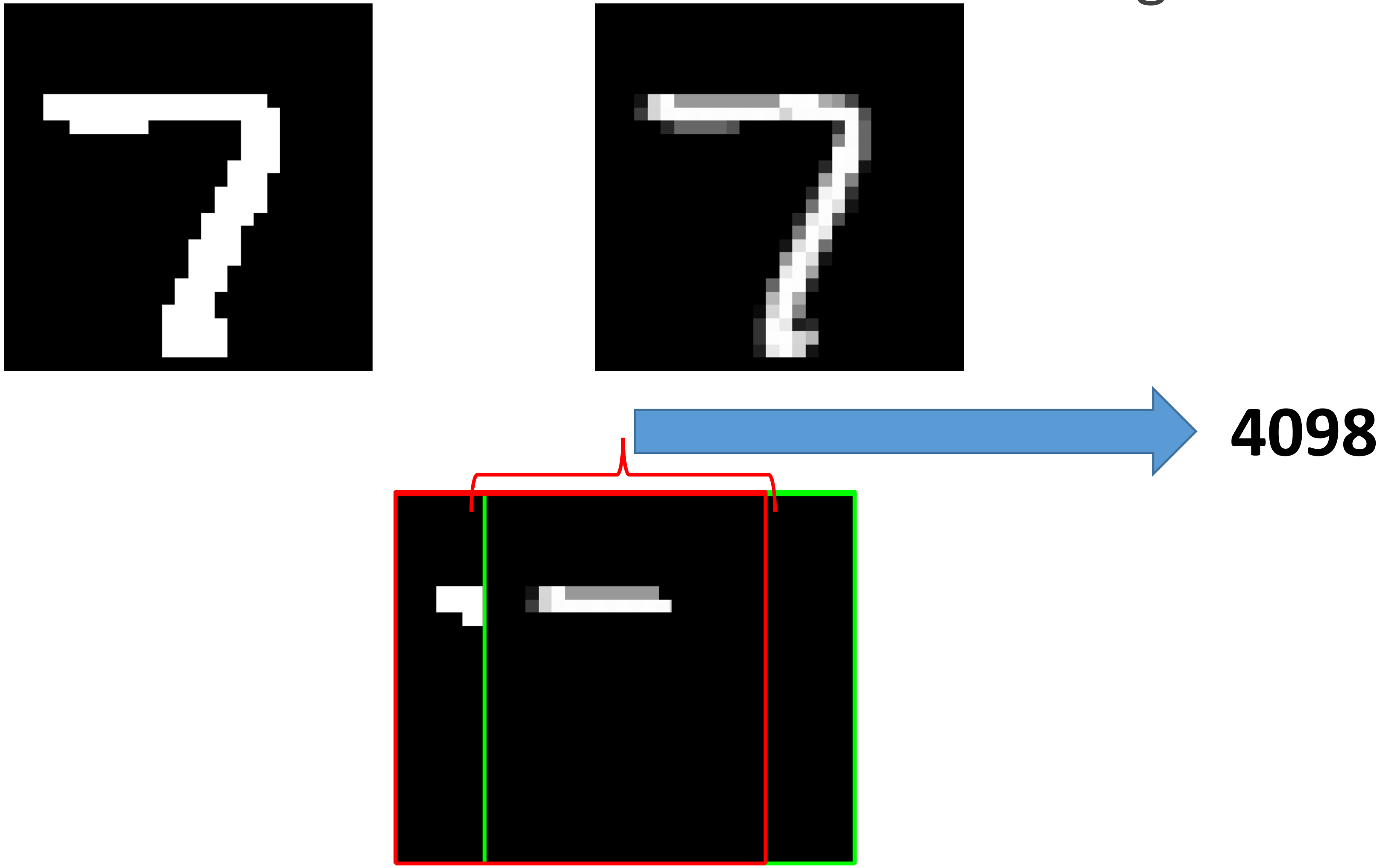




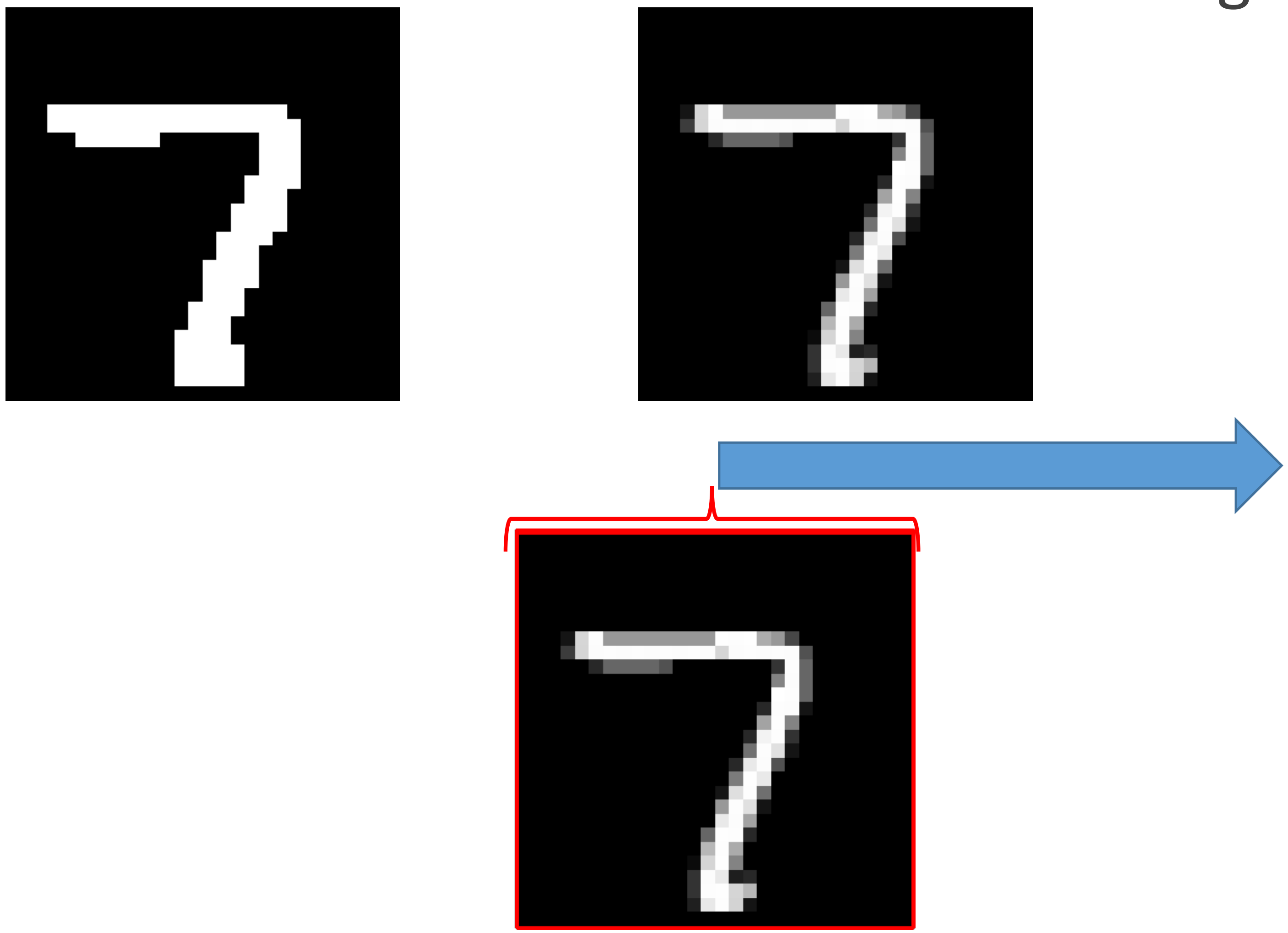
Filtering



Filtering

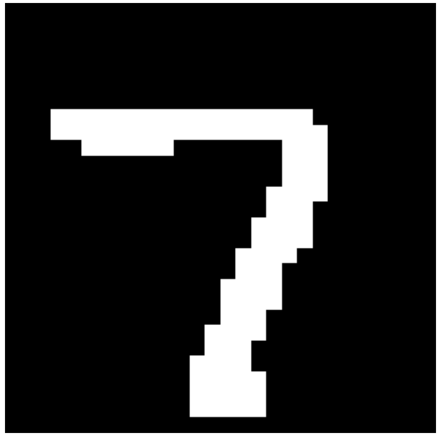


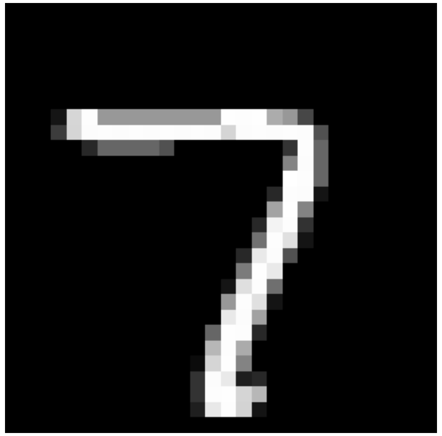
Filtering

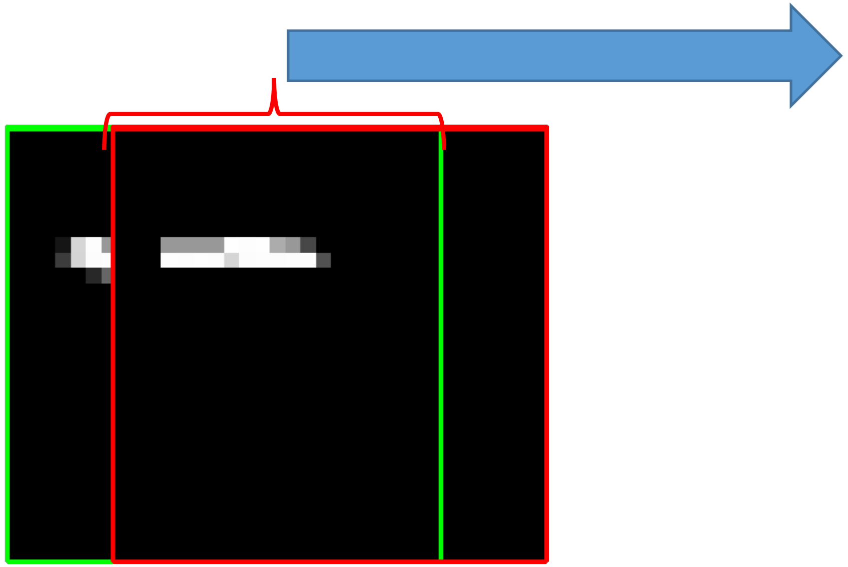


18604

Filtering

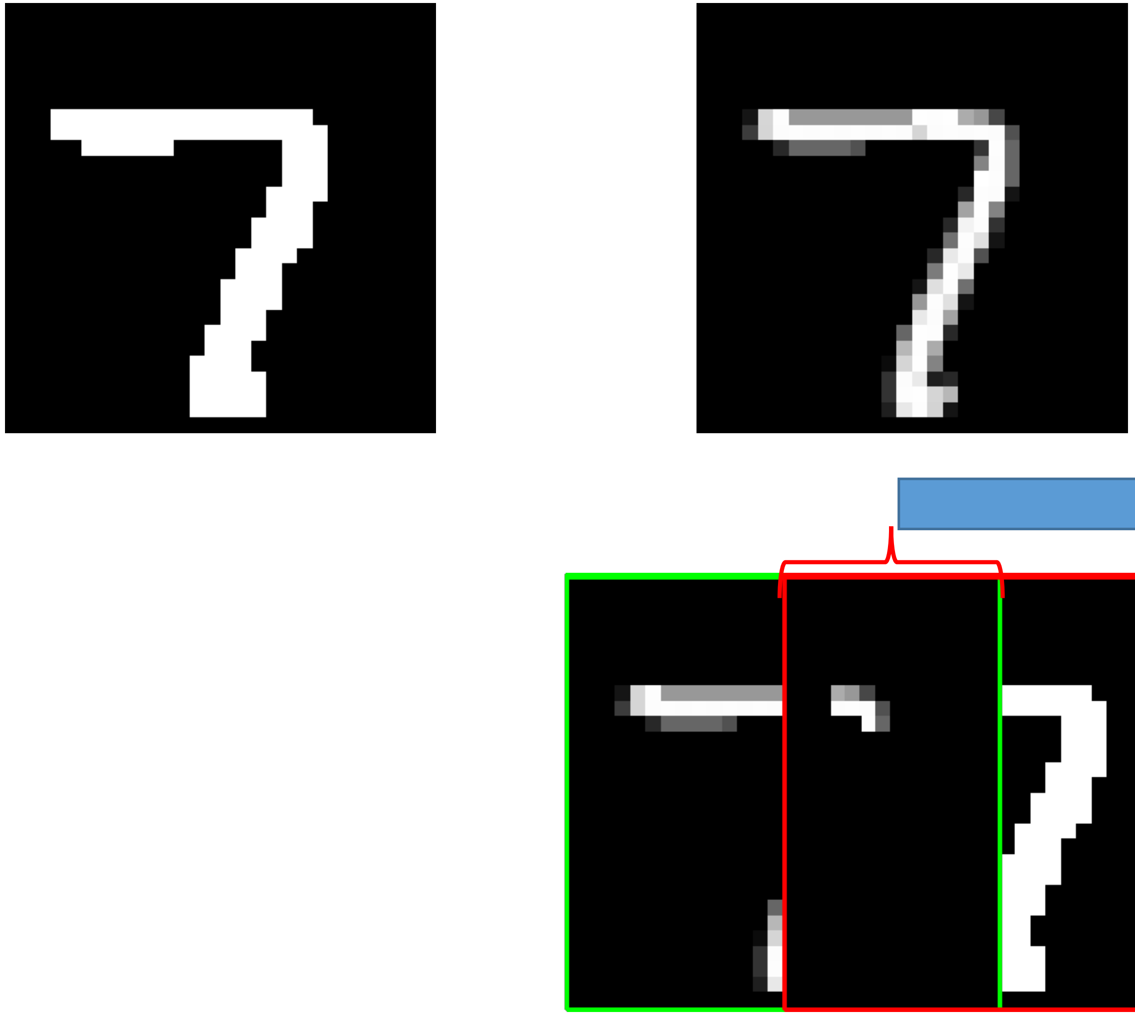




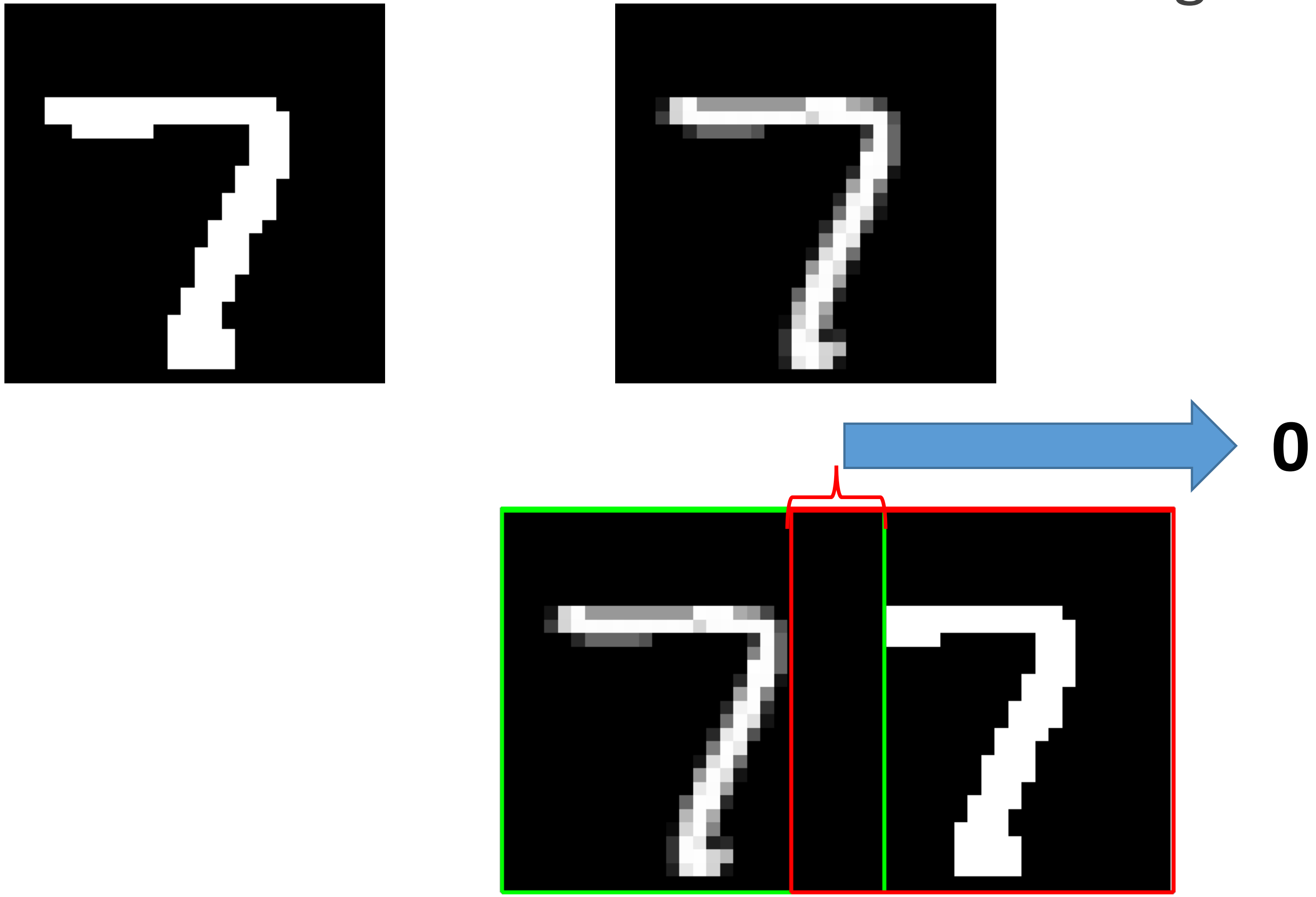


4331

Filtering

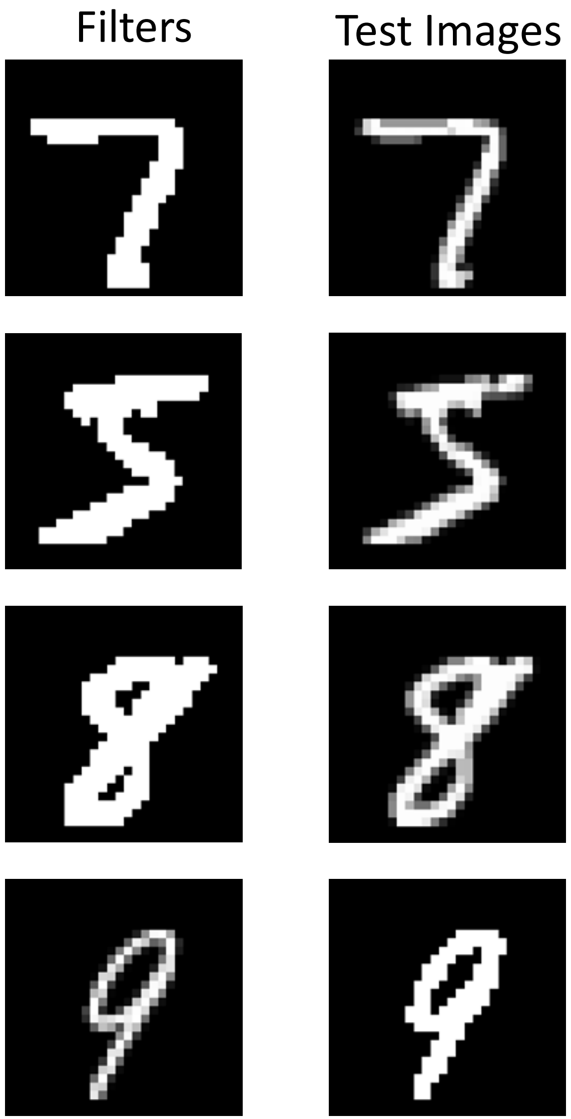


Filtering



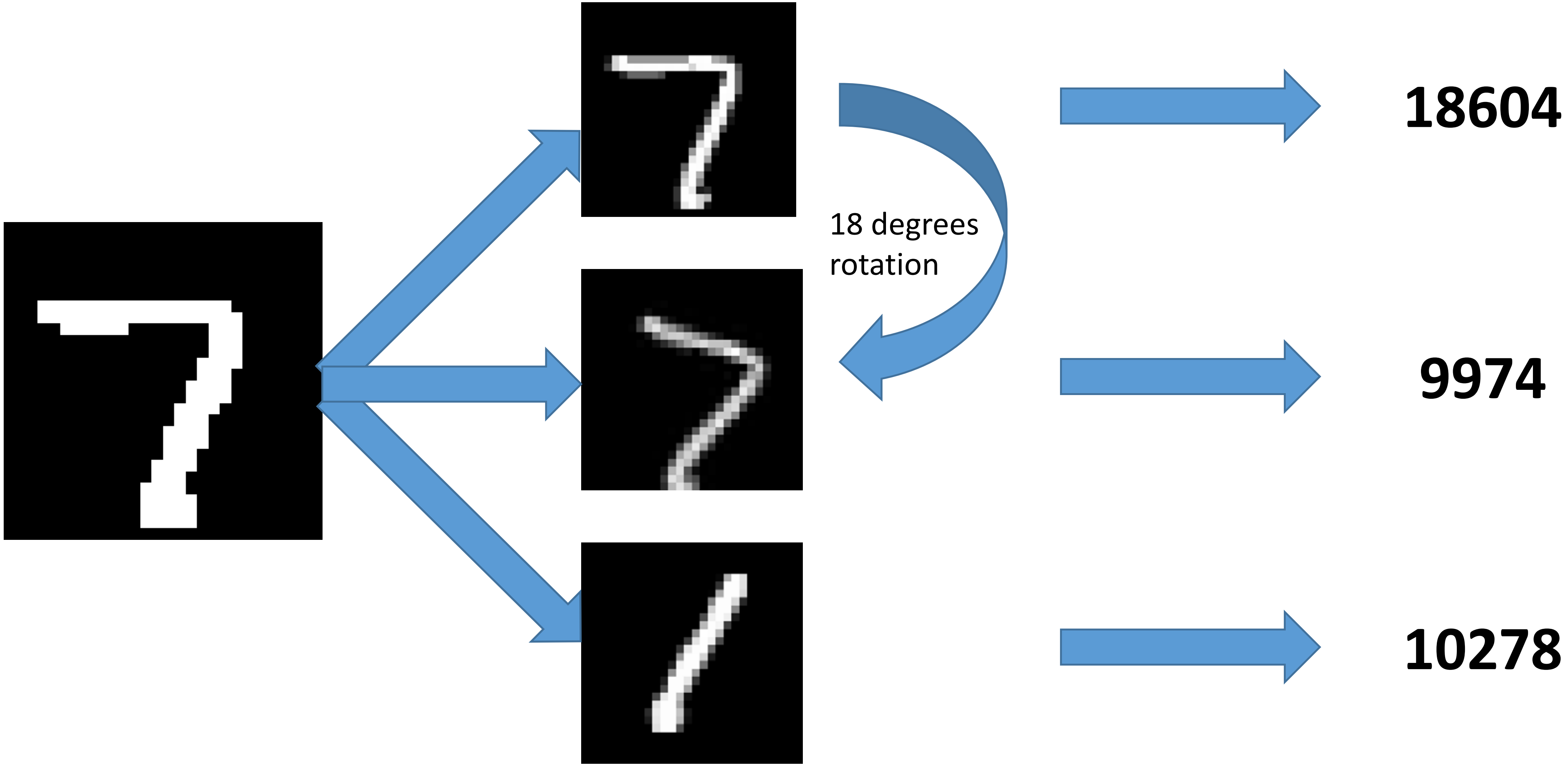
0

Classification by Matching Filters

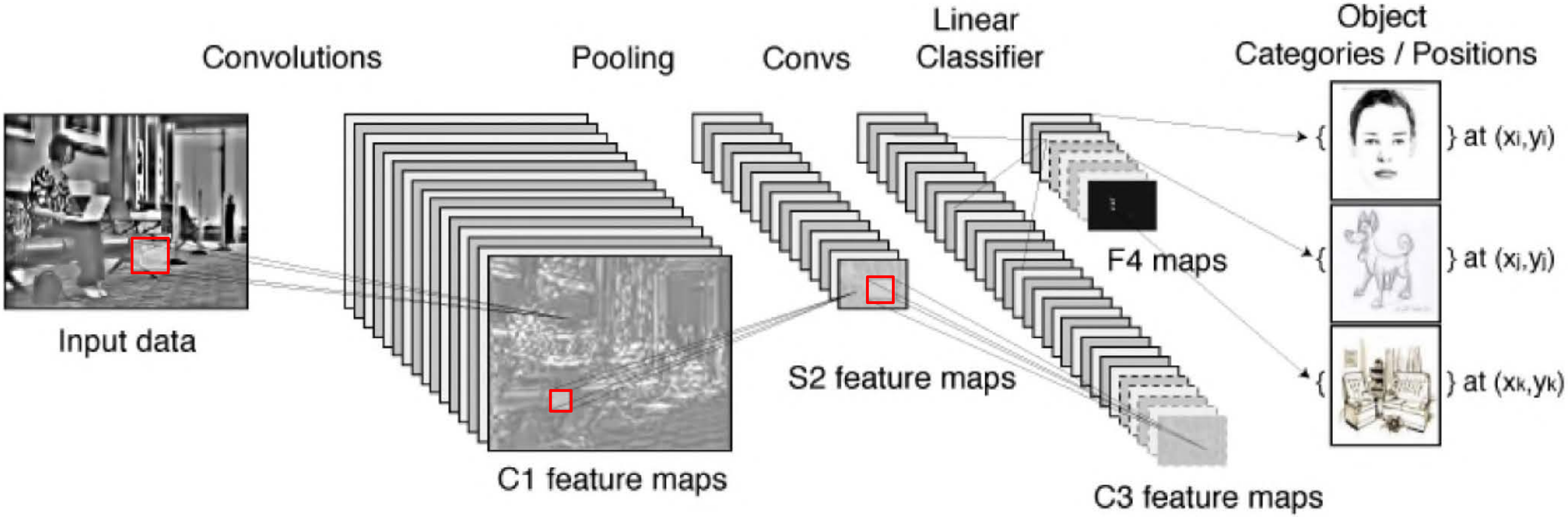


But what if the test image is a little
Rotated (or)
Skewed (or)
Zoomed out and so on

Effect of Slight Rotation

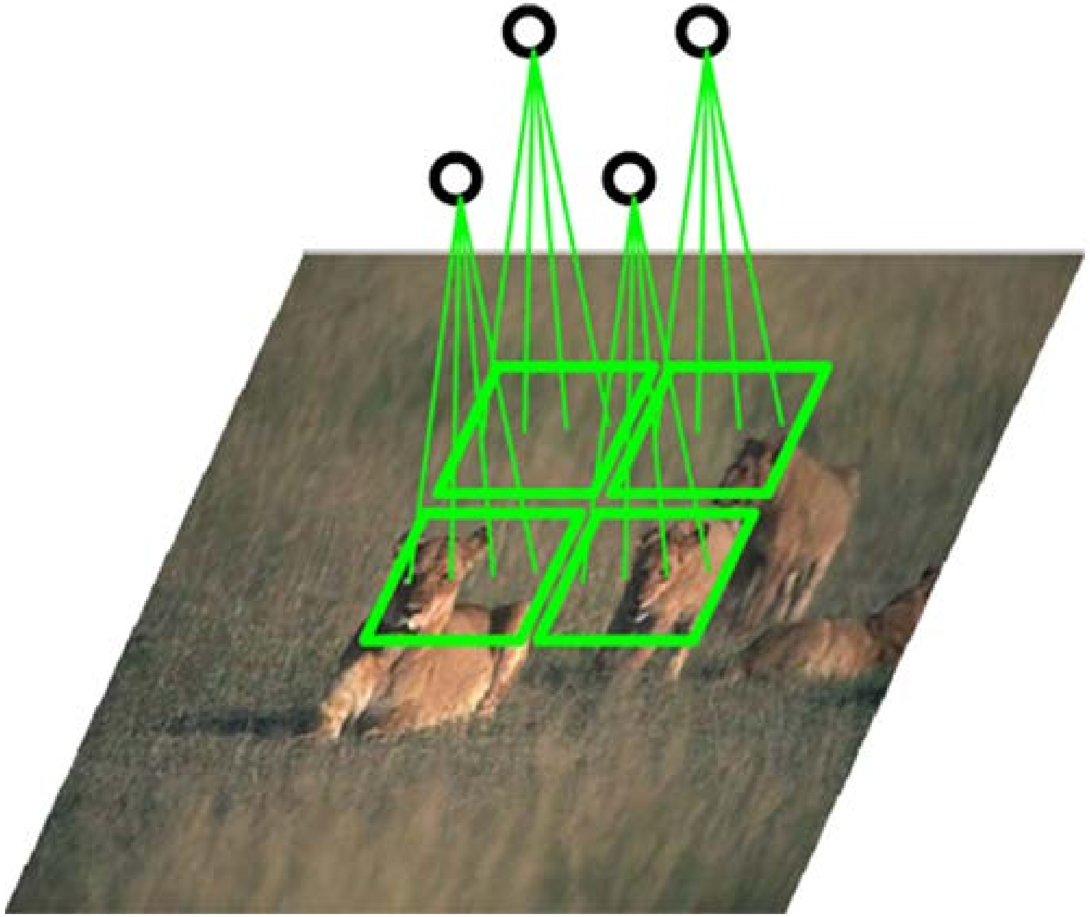


Convnets (Fukushima, LeCun, Hinton)



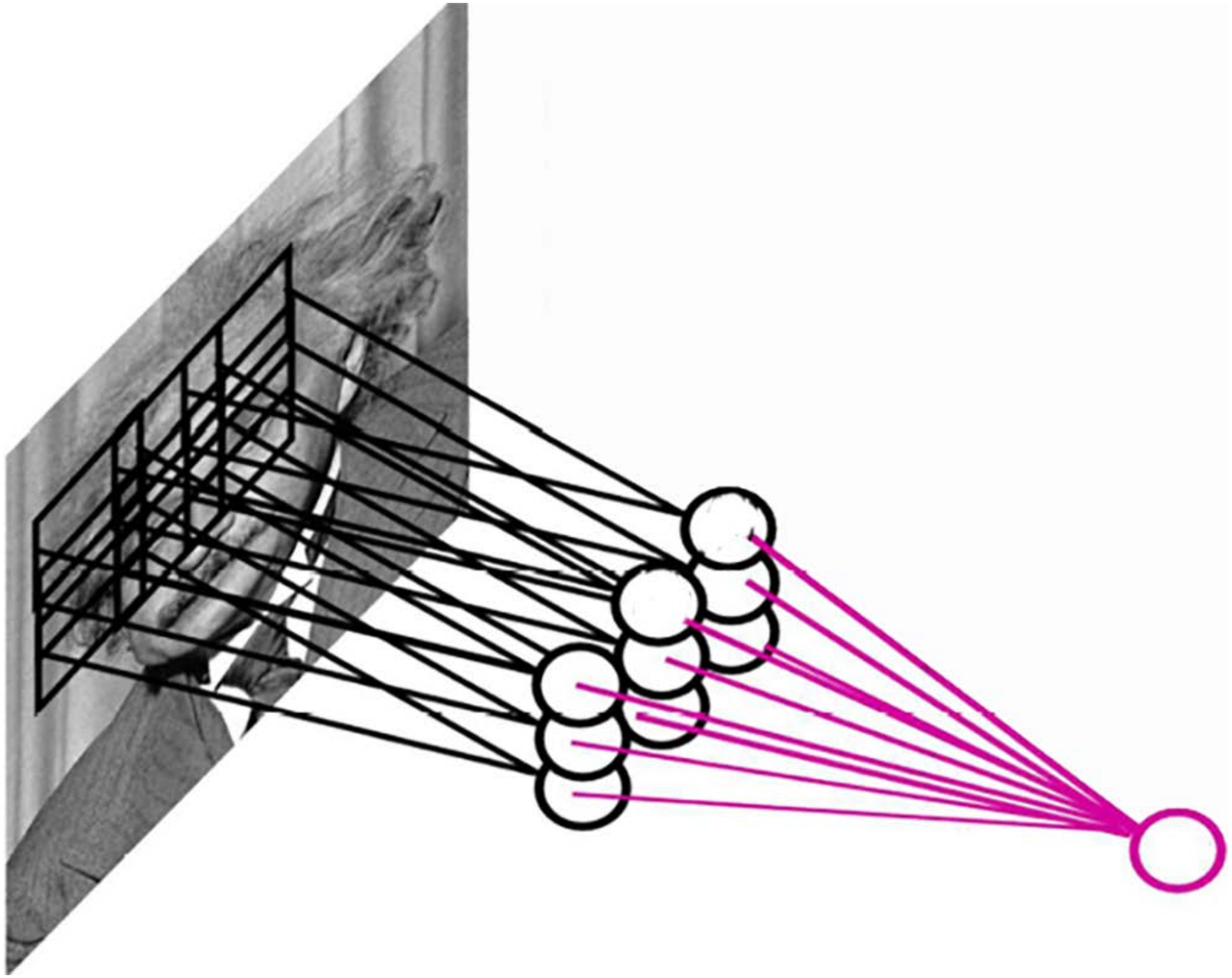
Slide courtesy: Nando de Freitas, University of Oxford

Convnets (Fukushima, LeCun, Hinton)



Slide courtesy: Nando de Freitas, University of Oxford

Pooling



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

Image courtesy: Nando de Freitas, University of Oxford

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

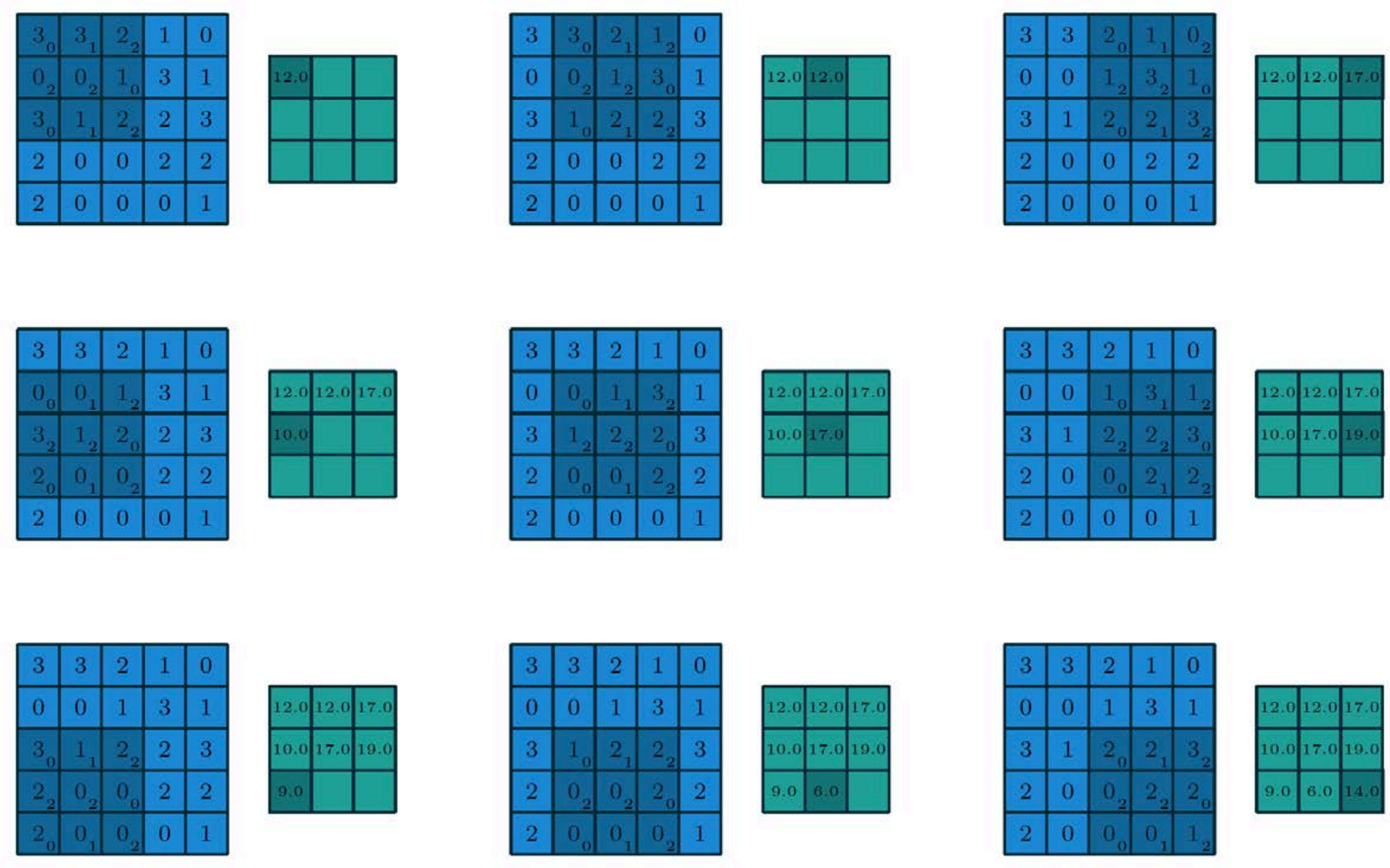


Image courtesy:Vincent Dumoulin

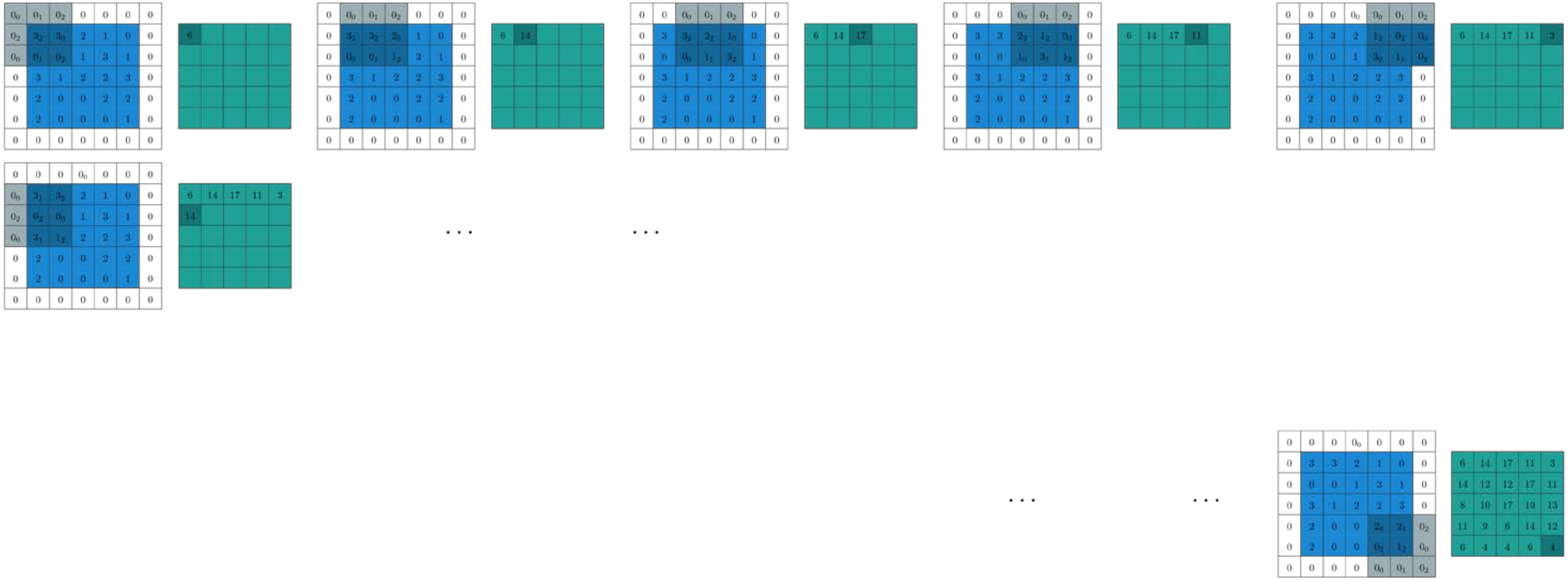
Convolution with Zero Padding

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



. . .   . . .

. . .   . . .

6	14	17	11	3
14	12	12	17	11
8	10	17	19	13
11	9	6	14	12
6	4	4	6	4

Convolution with Strides

Convolving a 3x3 kernel over a 5x5 input using 2x2 strides

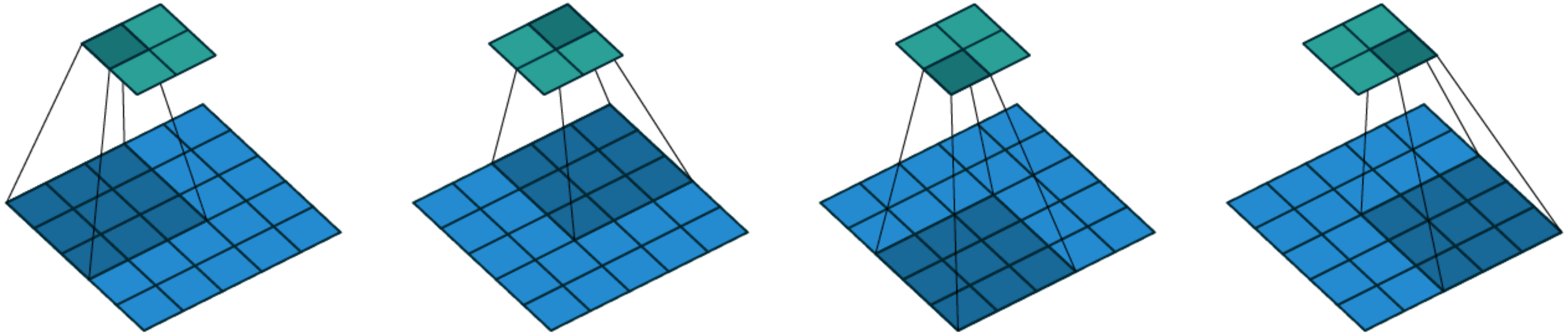


Image courtesy:Vincent Dumoulin

Convolution with Strides and Zero Padding

Convolving a 3x3 kernel over a 5 x5 input using 1 x1 strides and half padding

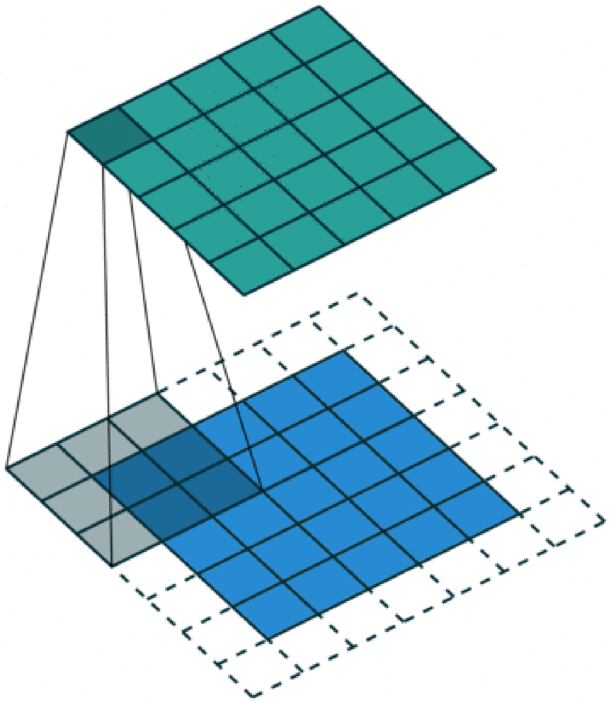


Image courtesy:Vincent Dumoulin

Inputs Generally have Multiple Channels

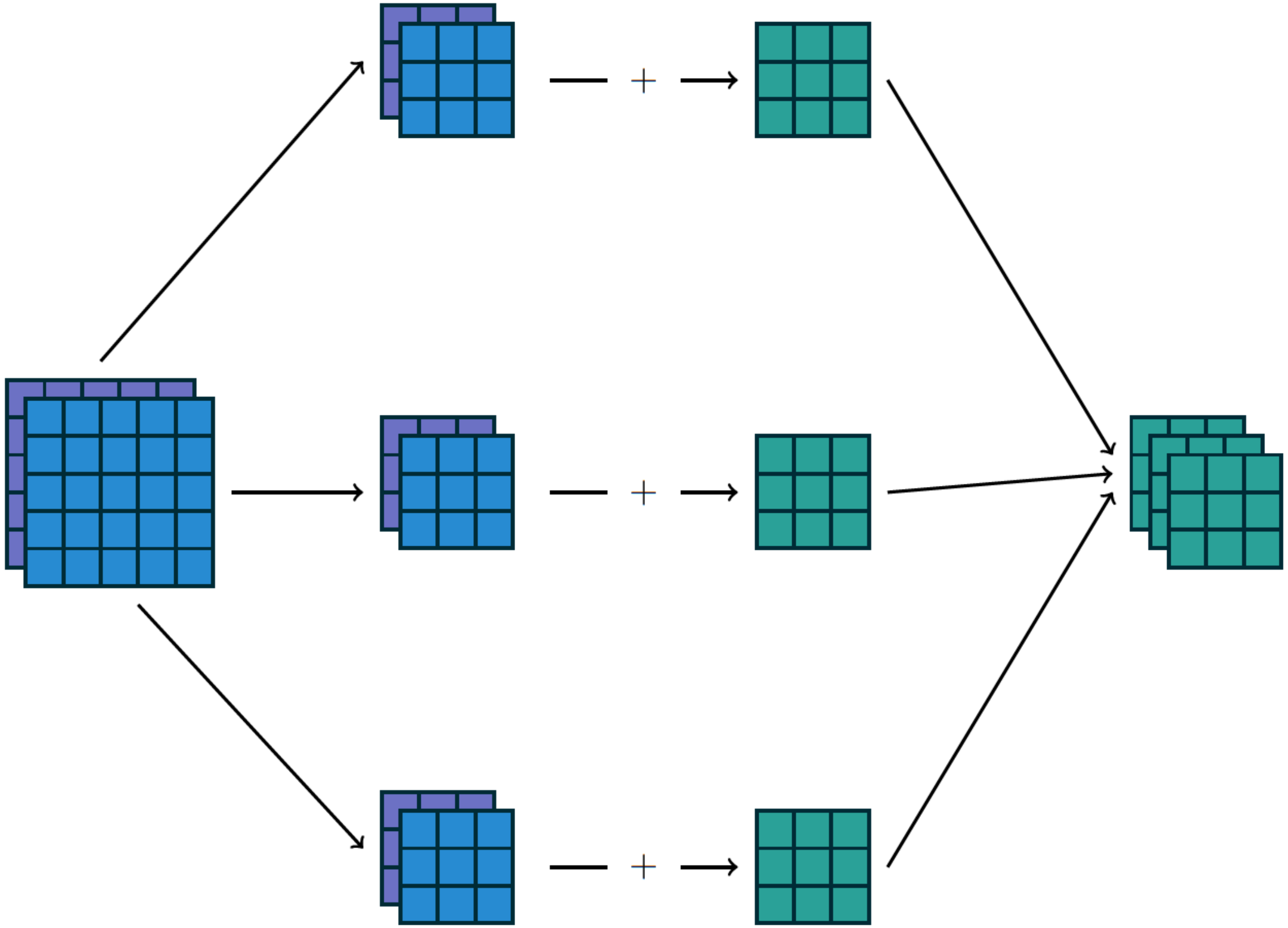


Image courtesy:Vincent Dumoulin

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 → This means image dimension becomes $w + 2p$

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

Note the box function

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

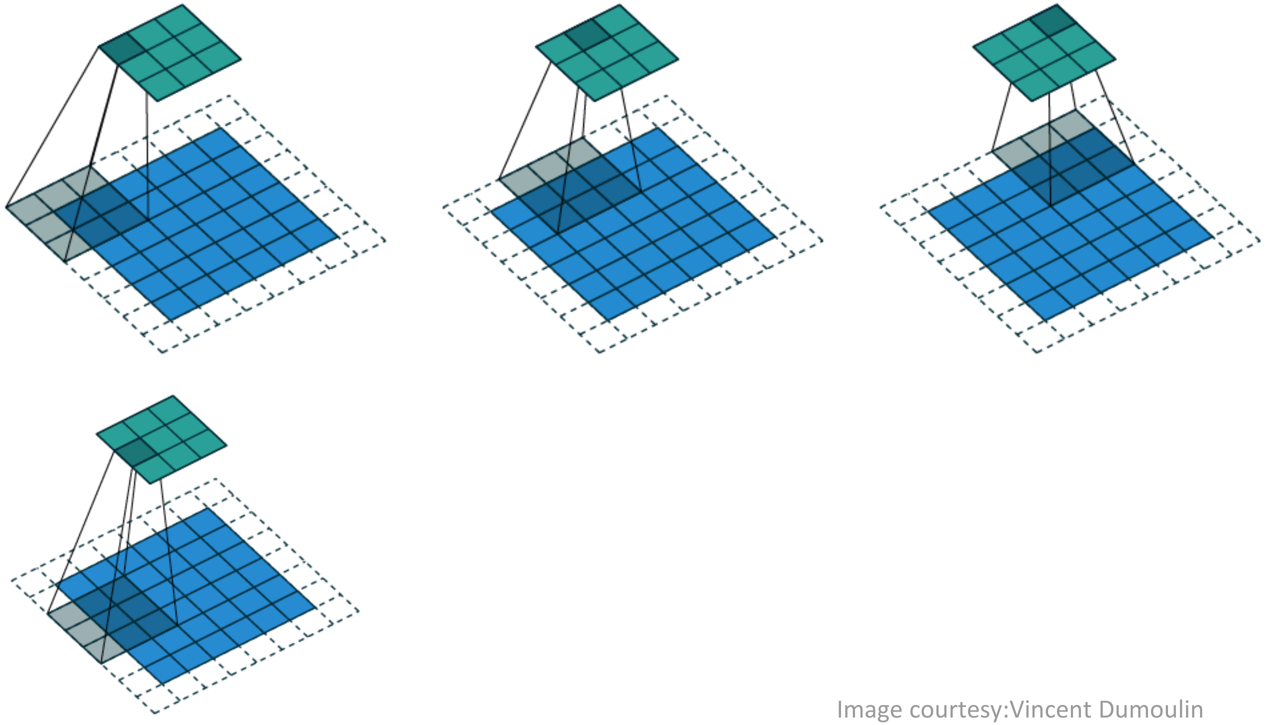
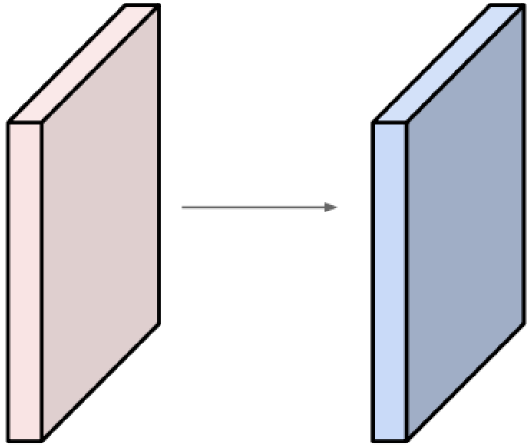


Image courtesy:Vincent Dumoulin

Convolution Arithmetic

Input volume: 32x32x3 [w, h, c].
64 filters of size 3x3 [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]

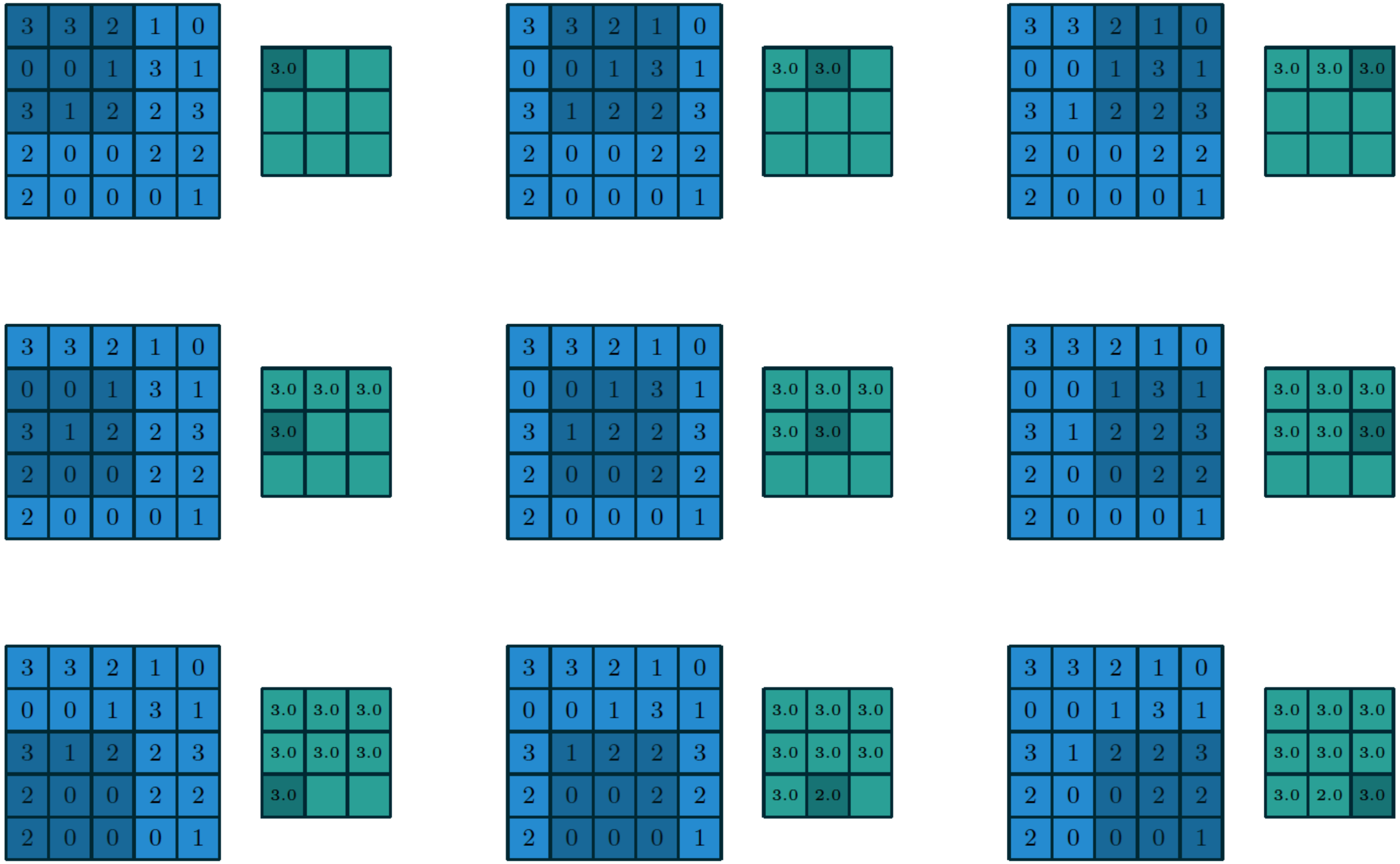
And What is the number of parameters in this convolution layer?

$64 \times 3 \times 3 \times 3$ [c_out, w, h, c_in] = 1728

Image courtesy: Andrej Karpathy

Pooling

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	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

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

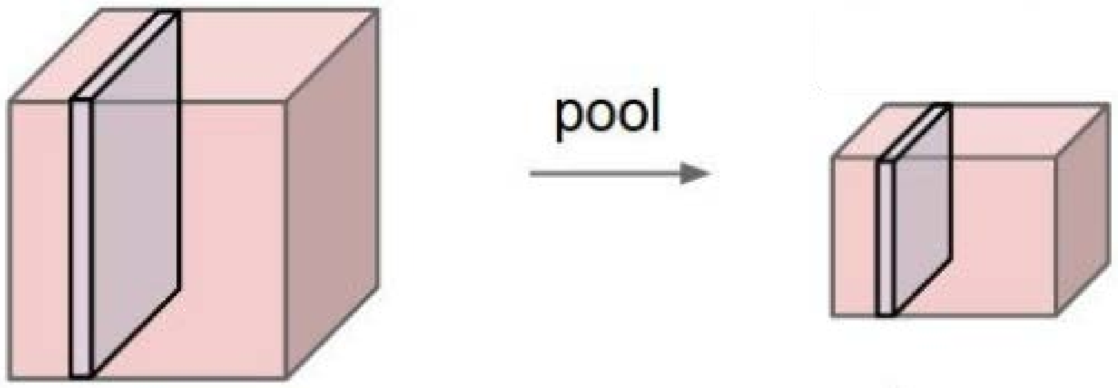
Output size = $\left\lfloor \frac{w-k}{s} \right\rfloor + 1$

Input volume: 32x32x3 [w, h, c].

Max-pooling kernel of size 2x2 [k, k] with stride 2 [s]

What is the output feature map size?

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



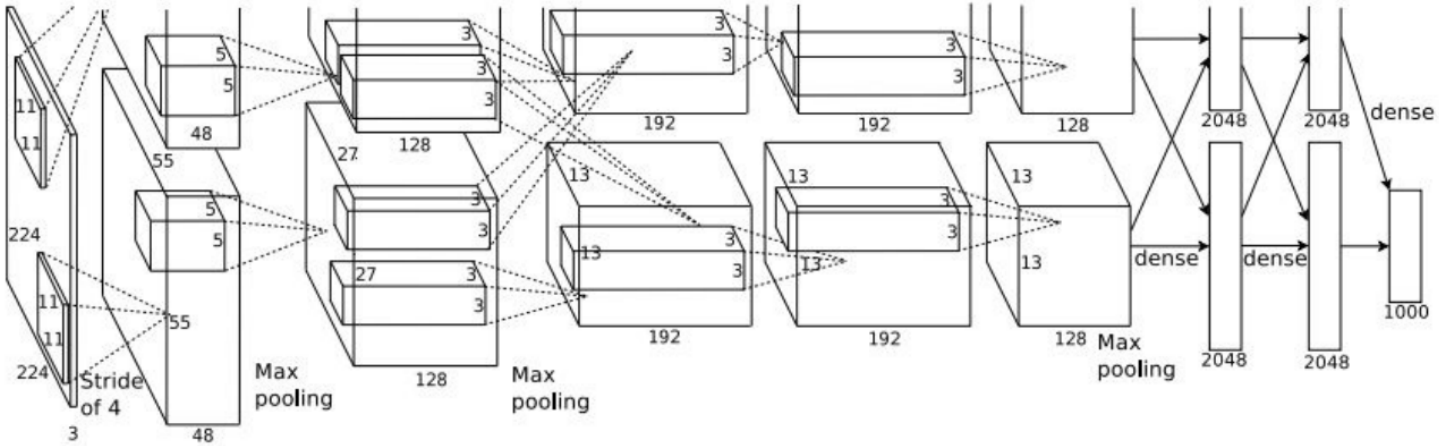
And What is the number of parameters in this pooling layer?

0

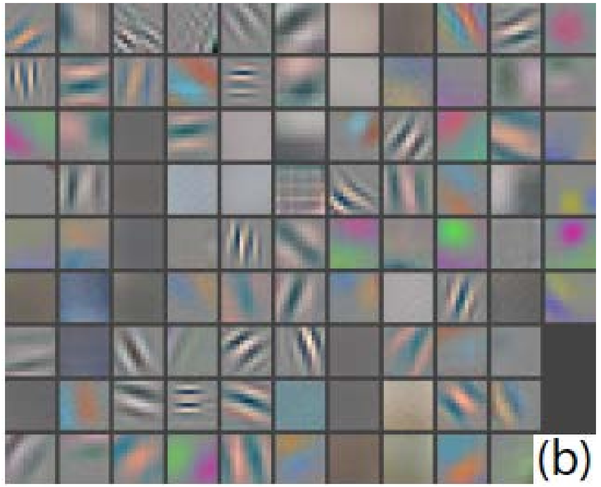
Image courtesy: Andrej Karpathy

Visualizations

AlexNet (2012)



First layer (CONV1): 96 11x11 filters



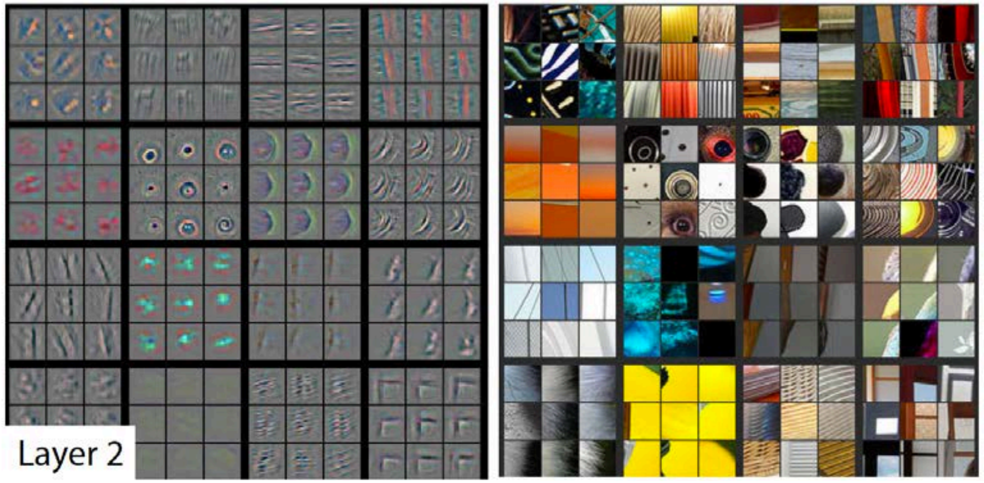
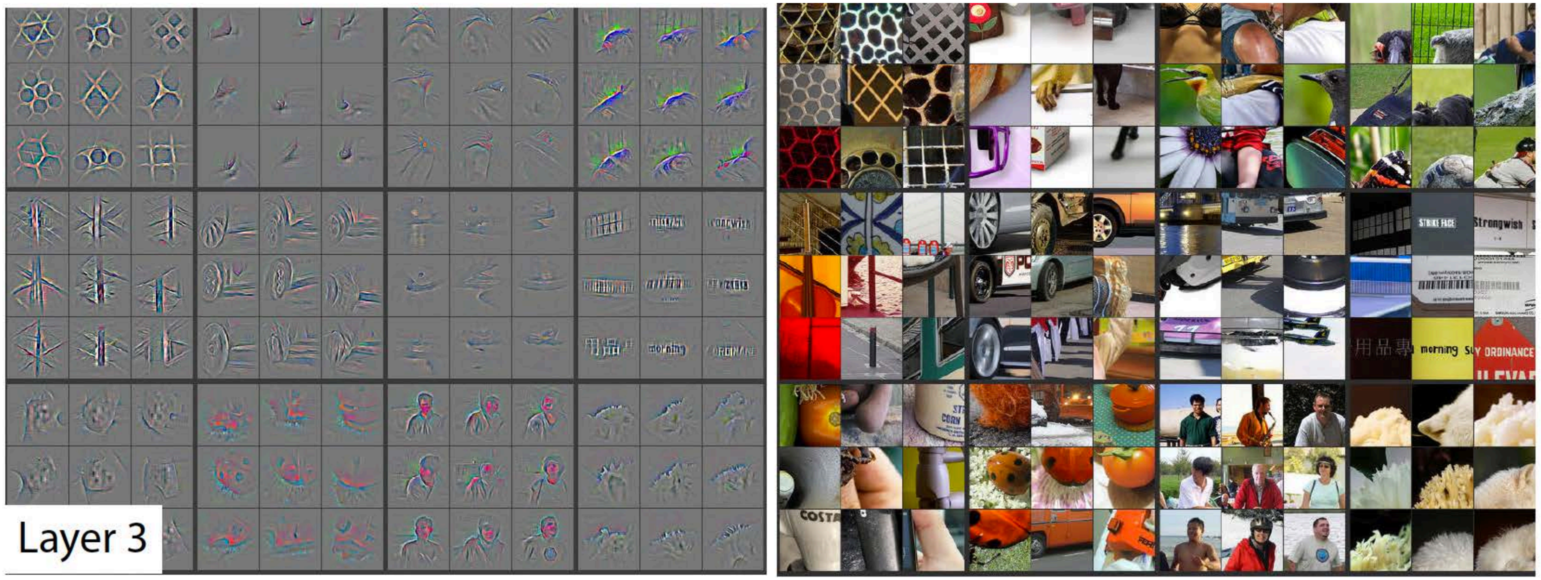


Image courtesy: Zeiler, Fergus, 2013

Visualizations



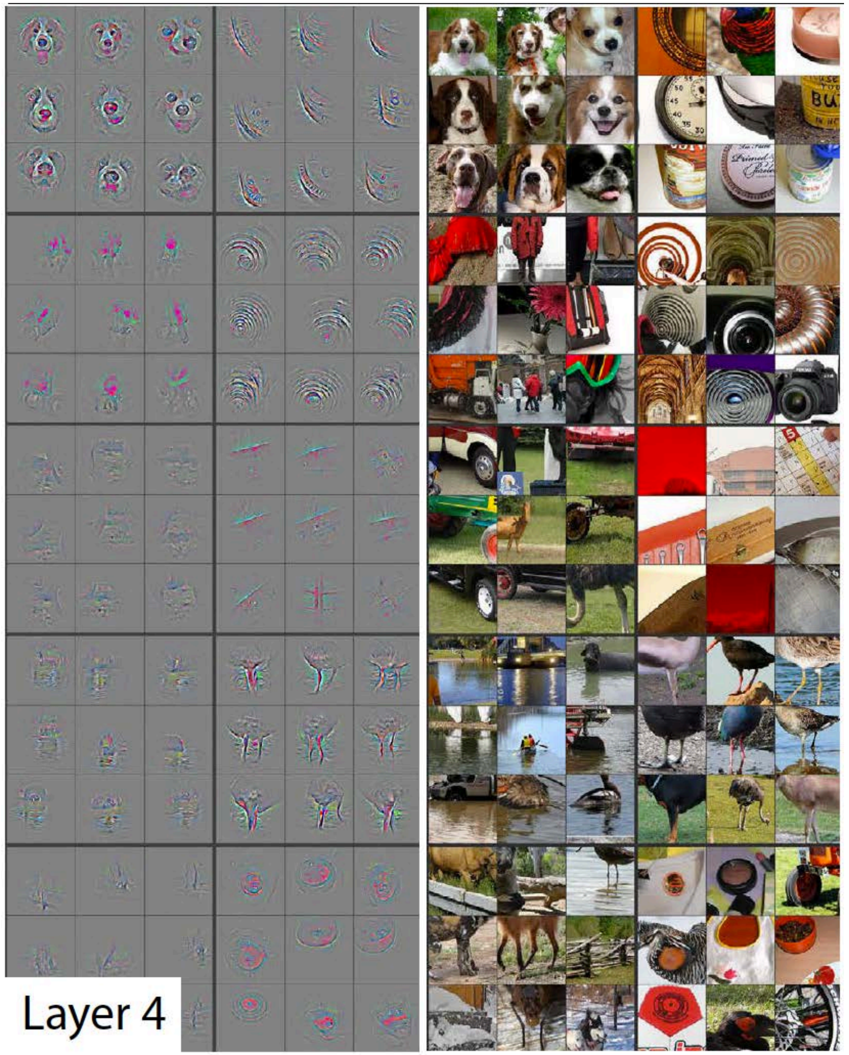
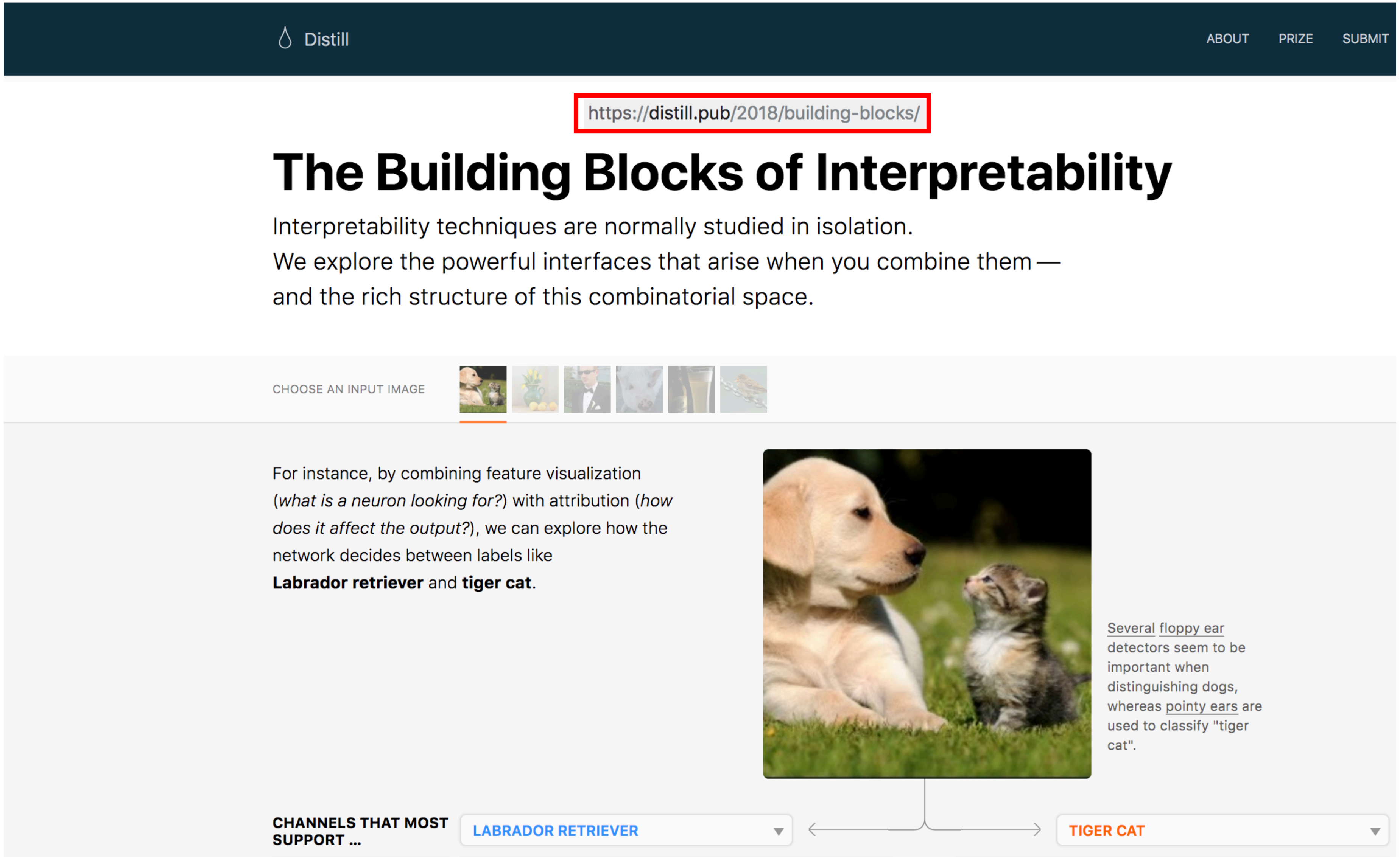


Image courtesy: Zeiler, Fergus, 2013



Distill ABOUT PRIZE SUBMIT

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 ...