Convolutional Neural Network

Hung-yi Lee

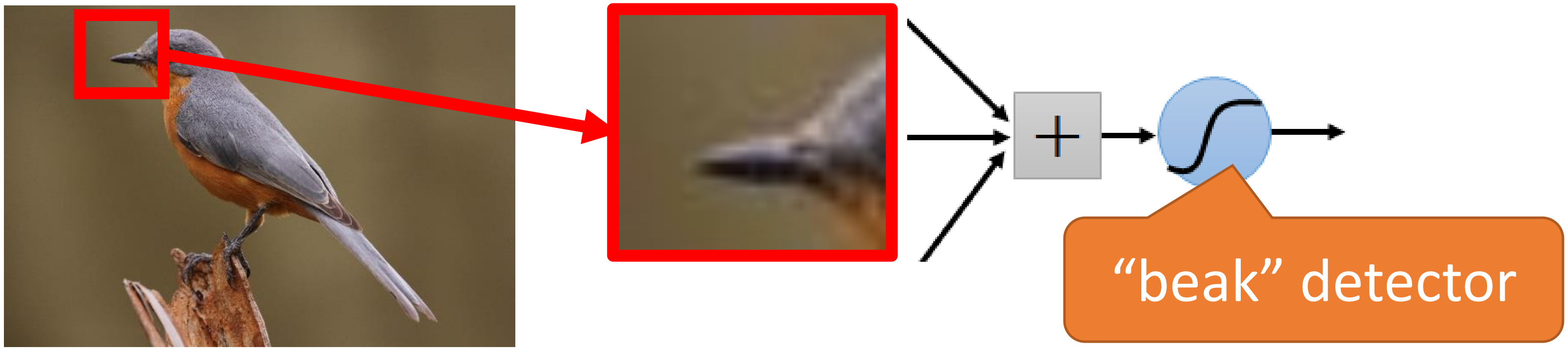
Can the network be simplified by considering the properties of images?

Why CNN for Image

- Some patterns are much smaller than the whole image

A neuron does not have to see the whole image to discover the pattern.

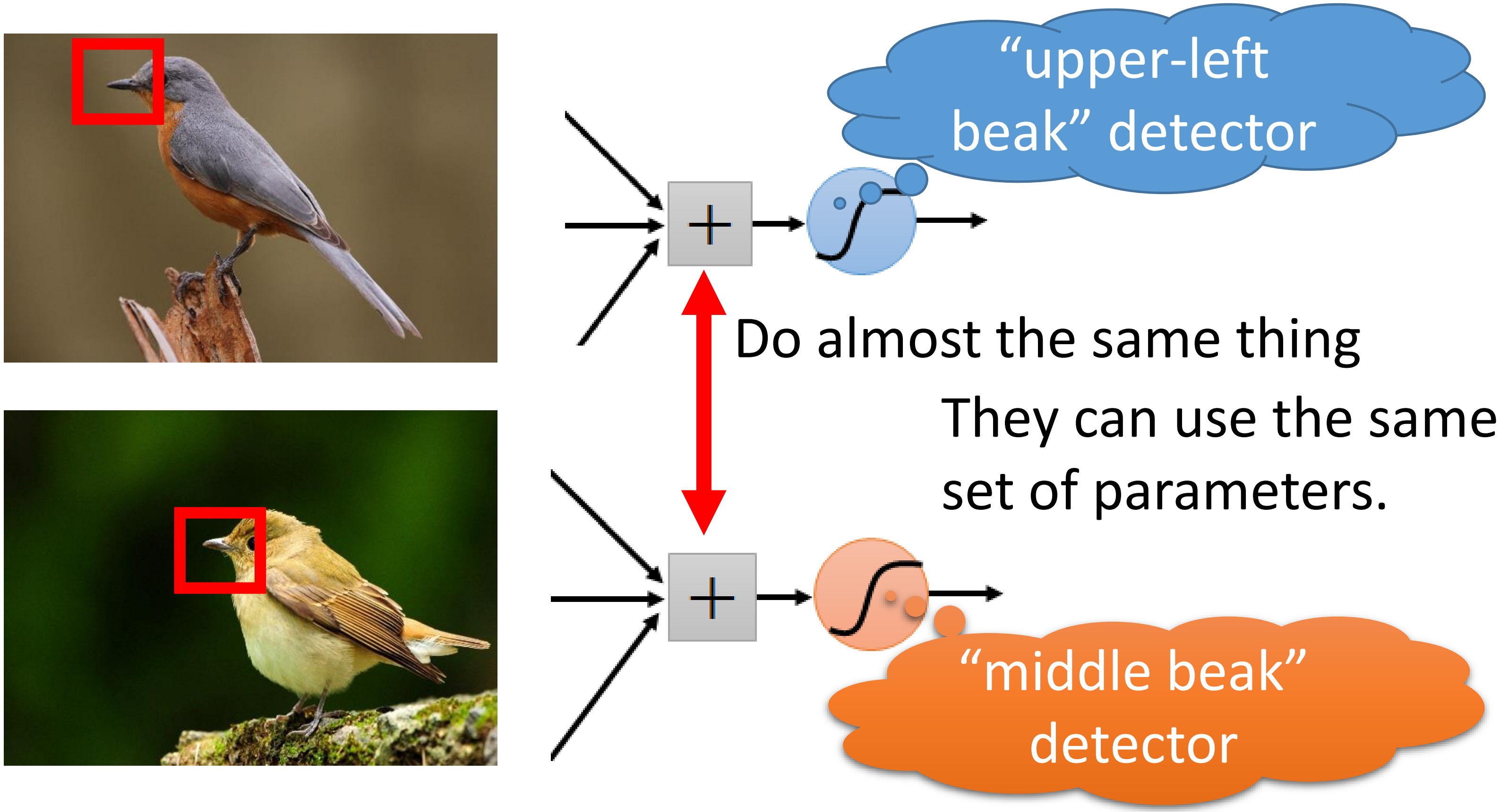
Connecting to small region with less parameters



“beak” detector

Why CNN for Image

- The same patterns appear in different regions.



“upper-left beak” detector

Do almost the same thing

They can use the same set of parameters.

“middle beak” detector

Why CNN for Image

- Subsampling the pixels will not change the object

bird

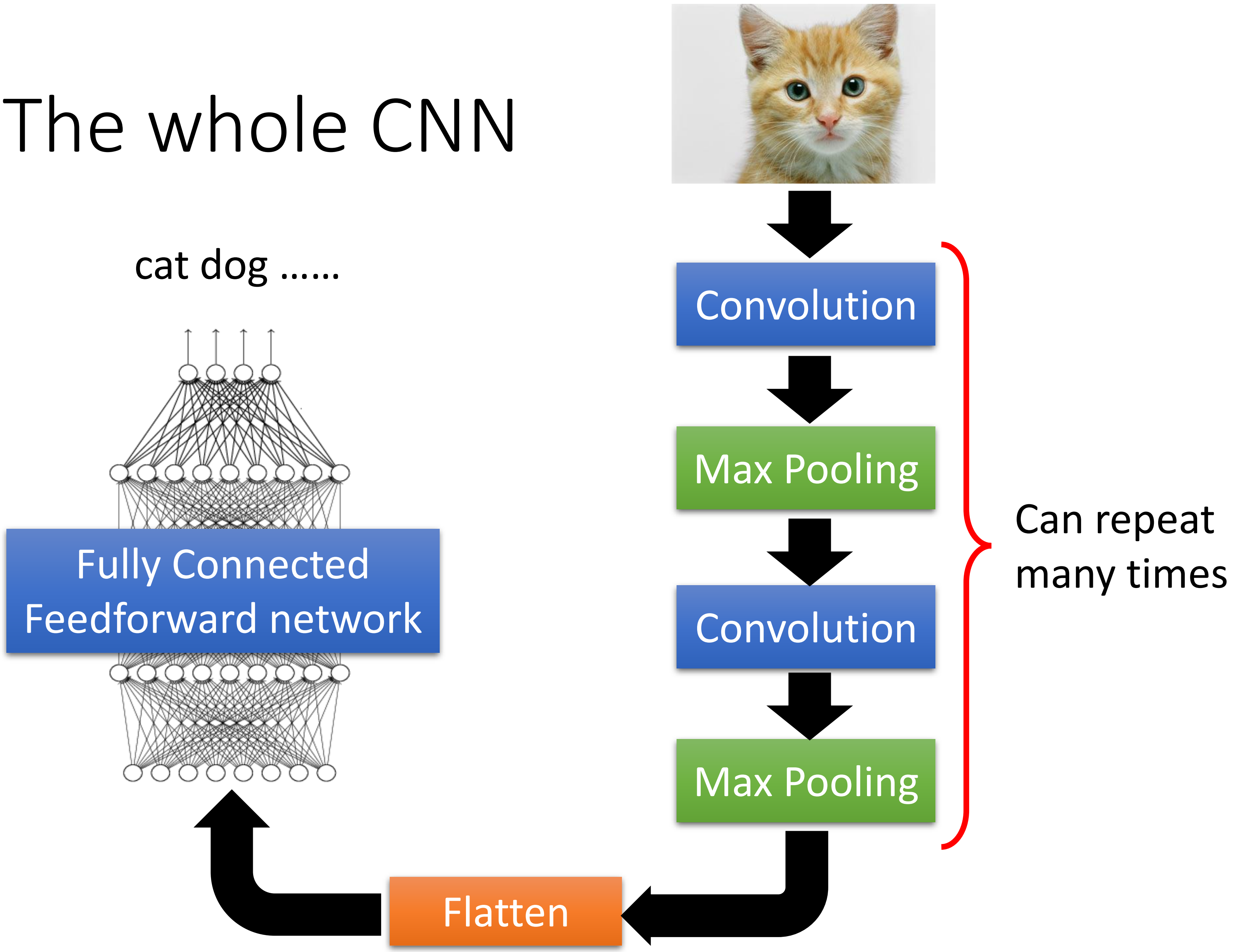
bird

subsampling

We can subsample the pixels to make image smaller

→ Less parameters for the network to process the image

The whole CNN



cat dog ……

Fully Connected Feedforward network

Convolution

Max Pooling

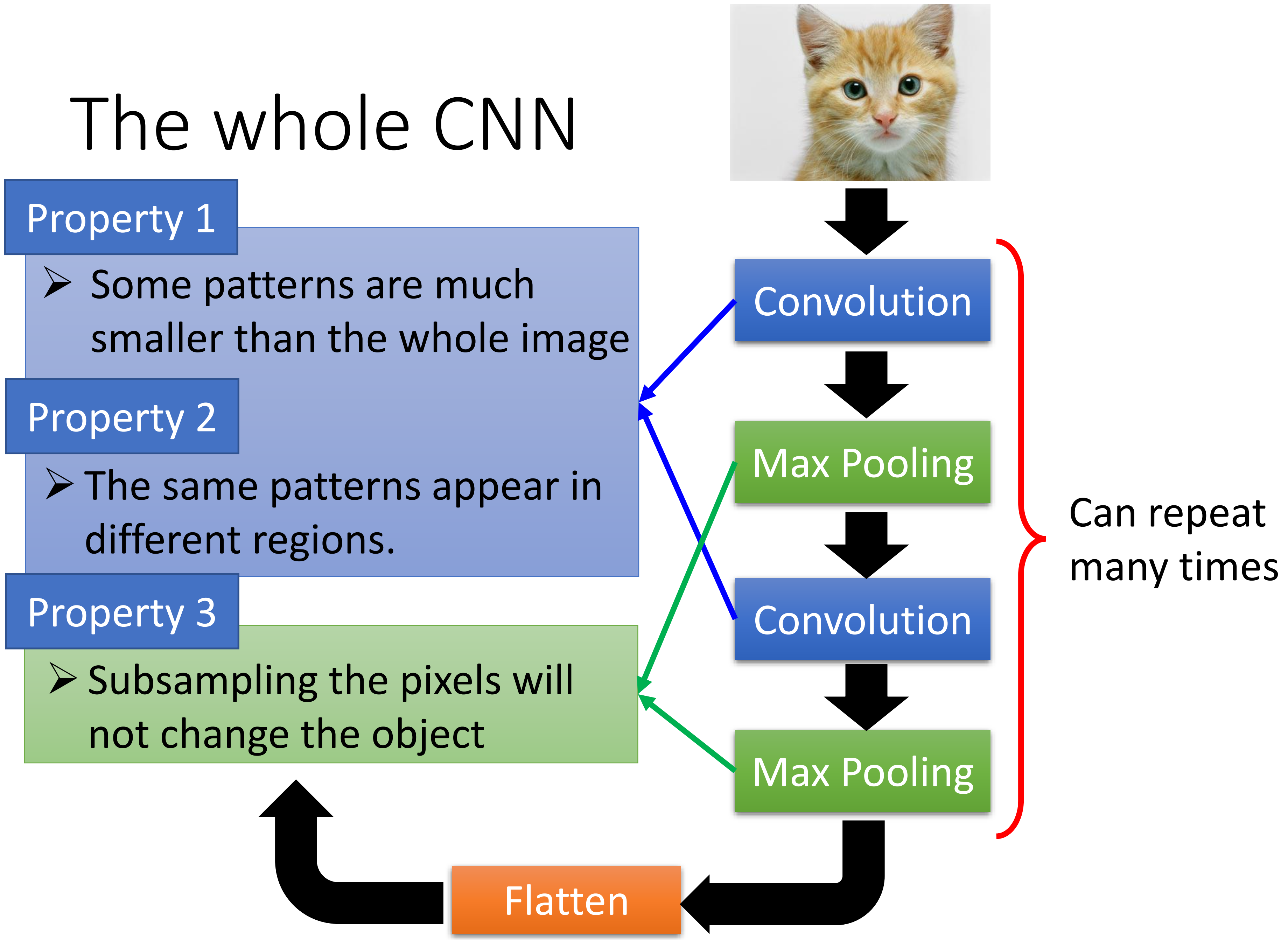
Convolution

Max Pooling

Can repeat many times

Flatten

The whole CNN



Property 1

- ➢ Some patterns are much smaller than the whole image

Property 2

- ➢ The same patterns appear in different regions.

Property 3

- ➢ Subsampling the pixels will not change the object

Convolution

Max Pooling

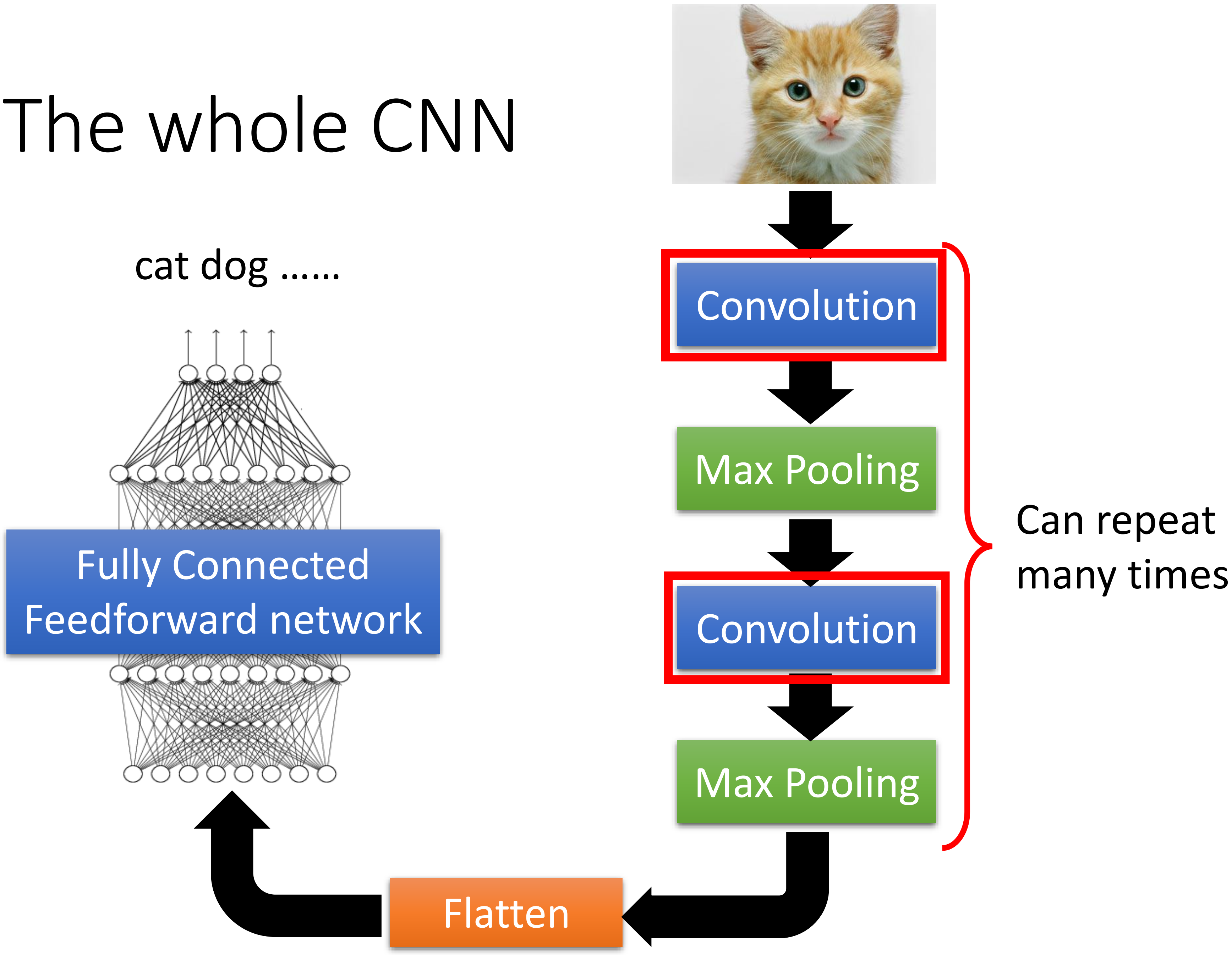
Convolution

Max Pooling

Can repeat many times

Flatten

The whole CNN



cat dog ……

Convolution

Max Pooling

Convolution

Max Pooling

Can repeat many times

Flatten

Fully Connected Feedforward network

CNN – Convolution

Those are the network parameters to be learned.

1	0	0	0	0	1
0	1	0	0	1	0
0	0	1	1	0	0
1	0	0	0	1	0
0	1	0	0	1	0
0	0	1	0	1	0

6 x 6 image

1	-1	-1
-1	1	-1
-1	-1	1

Filter 1

Matrix

-1	1	-1
-1	1	-1
-1	1	-1

Filter 2

Matrix

⋮

Property 1

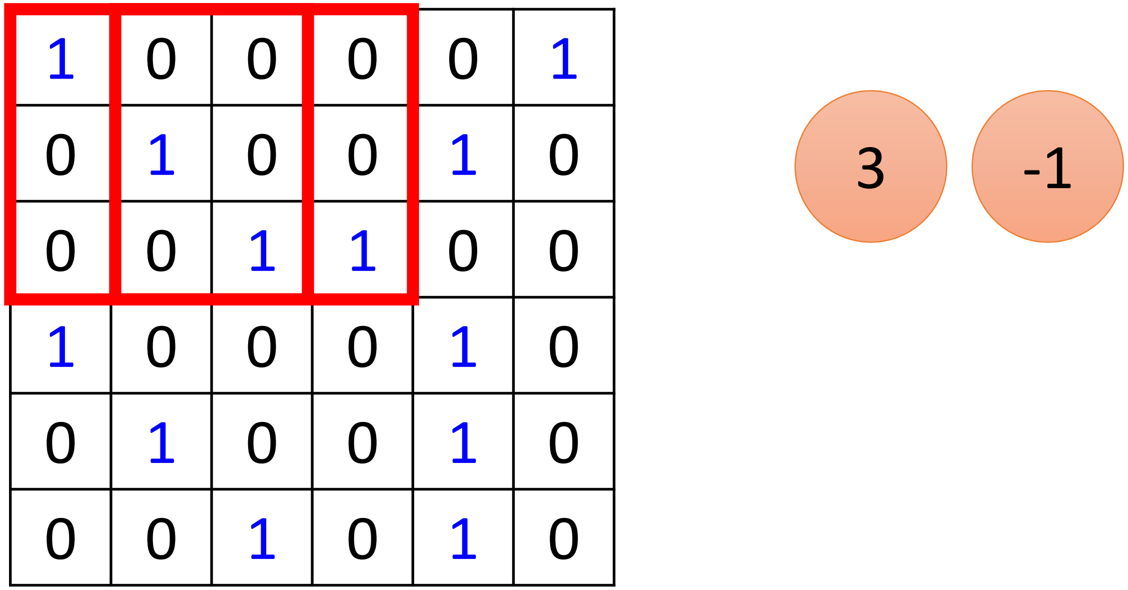
Each filter detects a small pattern (3 x 3).

CNN – Convolution

1	-1	-1
-1	1	-1
-1	-1	1

Filter 1

stride=1



1	0	0	0	0	1
0	1	0	0	1	0
0	0	1	1	0	0
1	0	0	0	1	0
0	1	0	0	1	0
0	0	1	0	1	0

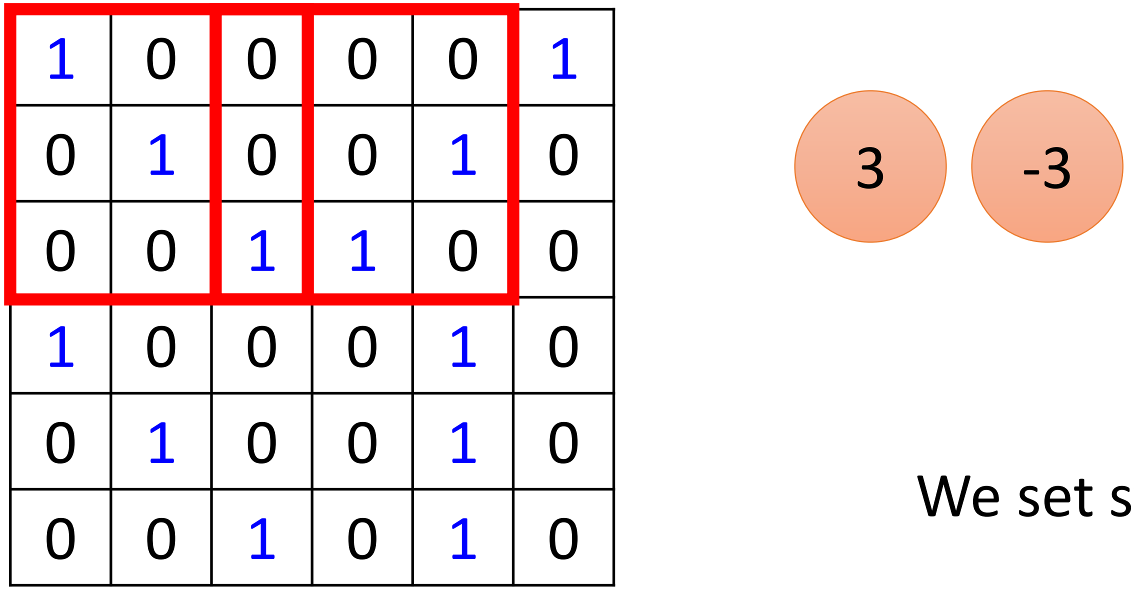
6 x 6 image

CNN – Convolution

1	-1	-1
-1	1	-1
-1	-1	1

Filter 1

If stride=2

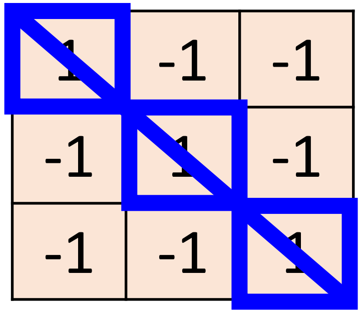


1	0	0	0	0	1
0	1	0	0	1	0
0	0	1	1	0	0
1	0	0	0	1	0
0	1	0	0	1	0
0	0	1	0	1	0

6 x 6 image

We set stride=1 below

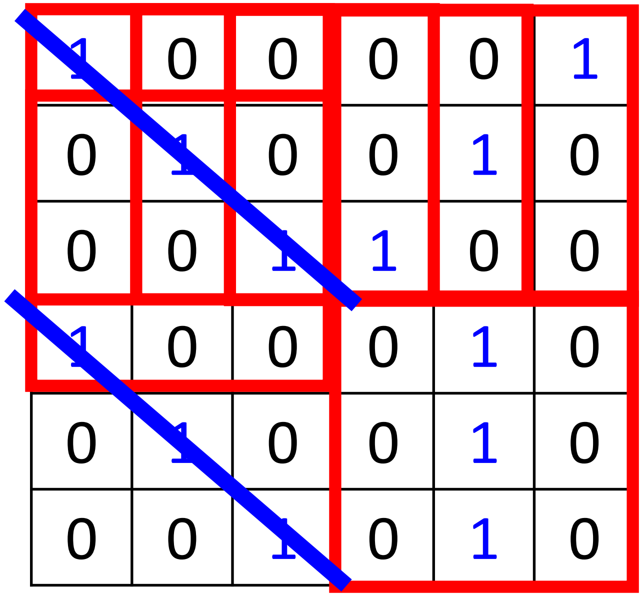
CNN – Convolution



1	-1	-1
-1	1	-1
-1	-1	1

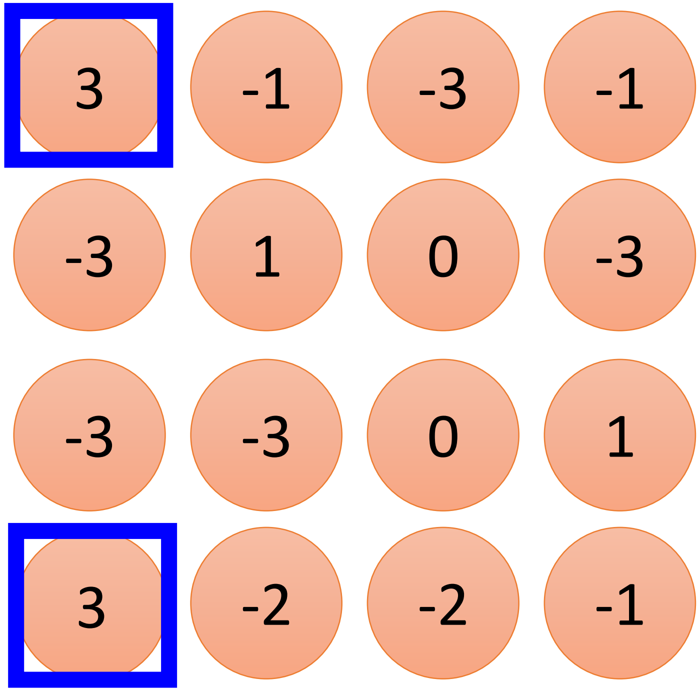
Filter 1

stride=1



1	0	0	0	0	1
0	1	0	0	1	0
0	0	1	1	0	0
1	0	0	0	1	0
0	1	0	0	1	0
0	0	1	0	1	0

6 x 6 image



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

Property 2

CNN – Convolution

-1	1	-1
-1	1	-1
-1	1	-1

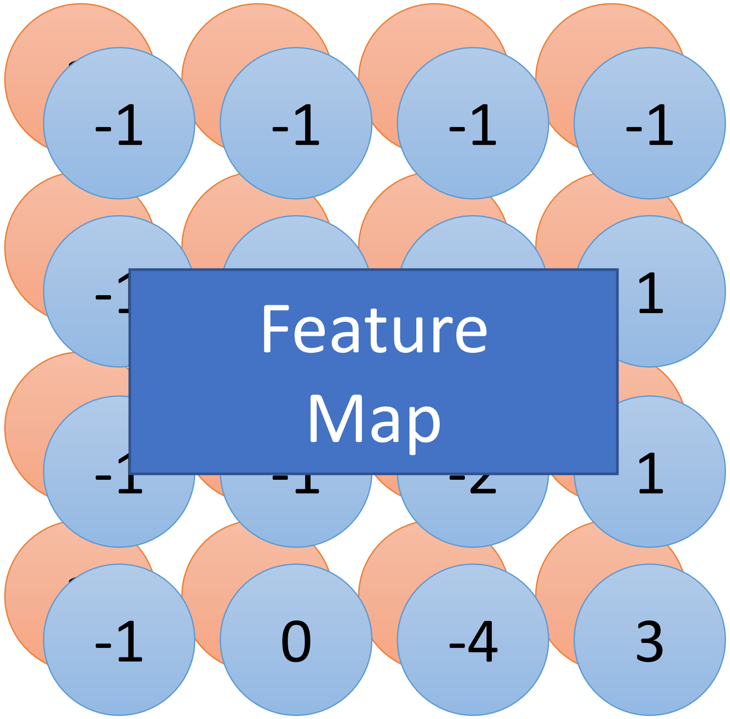
Filter 2

stride=1

1	0	0	0	0	1
0	1	0	0	1	0
0	0	1	1	0	0
1	0	0	0	1	0
0	1	0	0	1	0
0	0	1	0	1	0

6 x 6 image

Do the same process for every filter



4 x 4 image

CNN – Colorful image

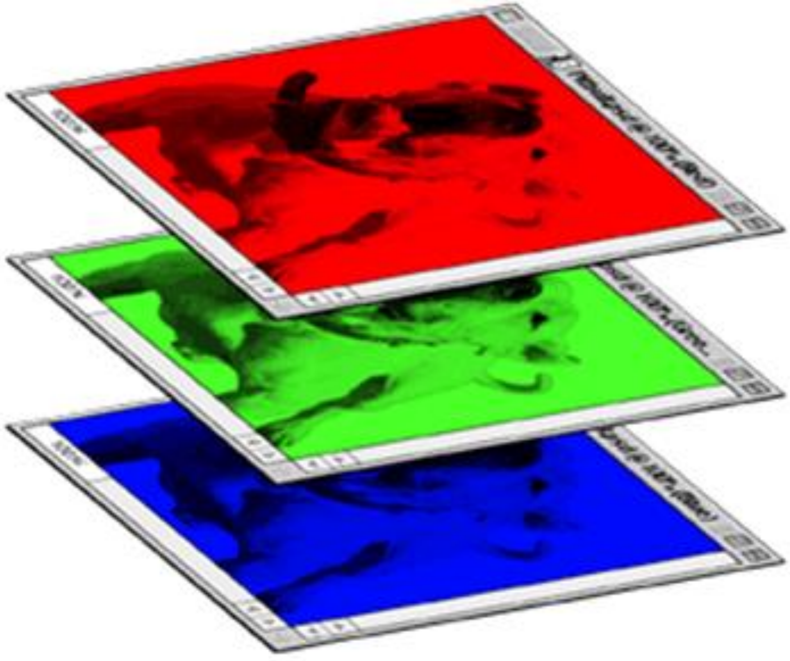
1	-1	-1
-1	1	-1
-1	-1	1

Filter 1

-1	1	-1
-1	1	-1
-1	1	-1

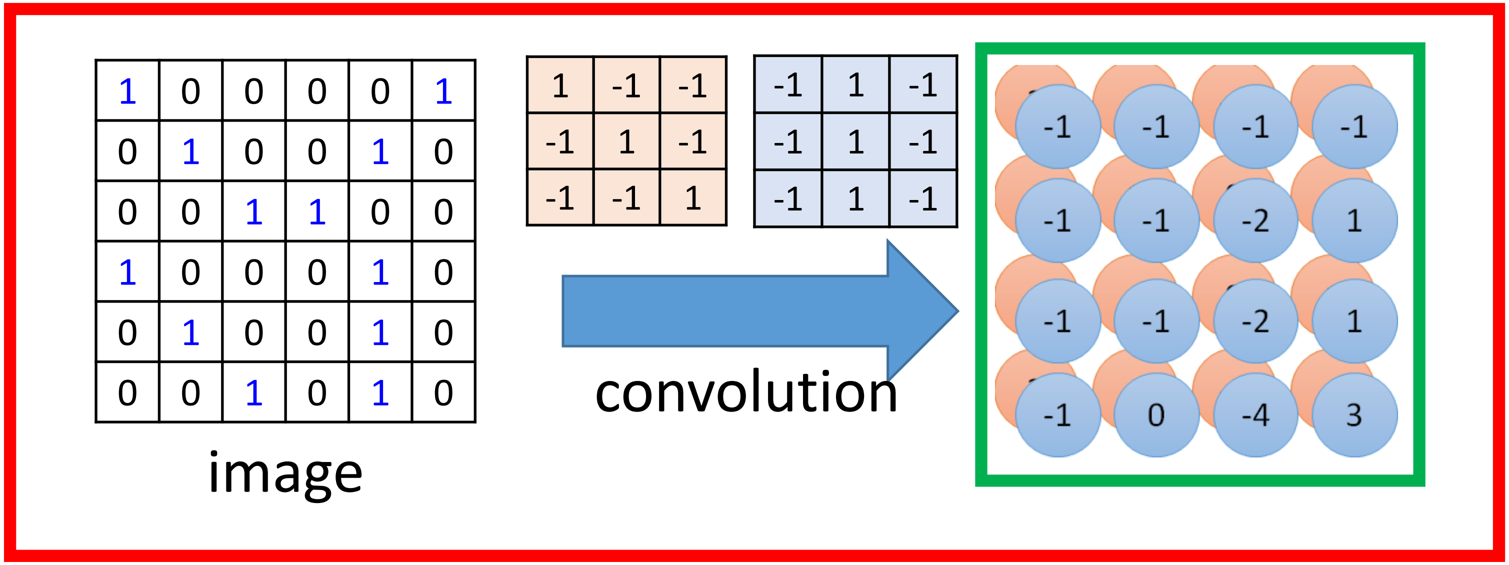
Filter 2

Colorful image



1	0	0	0	0	1
0	1	0	0	1	0
0	0	1	1	0	0
1	0	0	0	1	0
0	1	0	0	1	0
0	0	1	0	1	0

Convolution v.s. Fully Connected



1	0	0	0	0	1
0	1	0	0	1	0
0	0	1	1	0	0
1	0	0	0	1	0
0	1	0	0	1	0
0	0	1	0	1	0

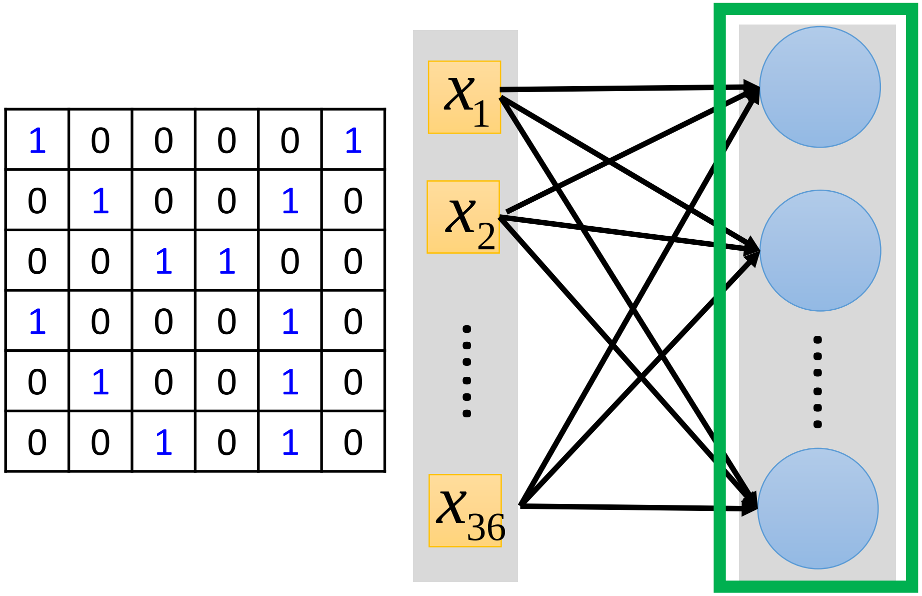
image

1	-1	-1
-1	1	-1
-1	-1	1

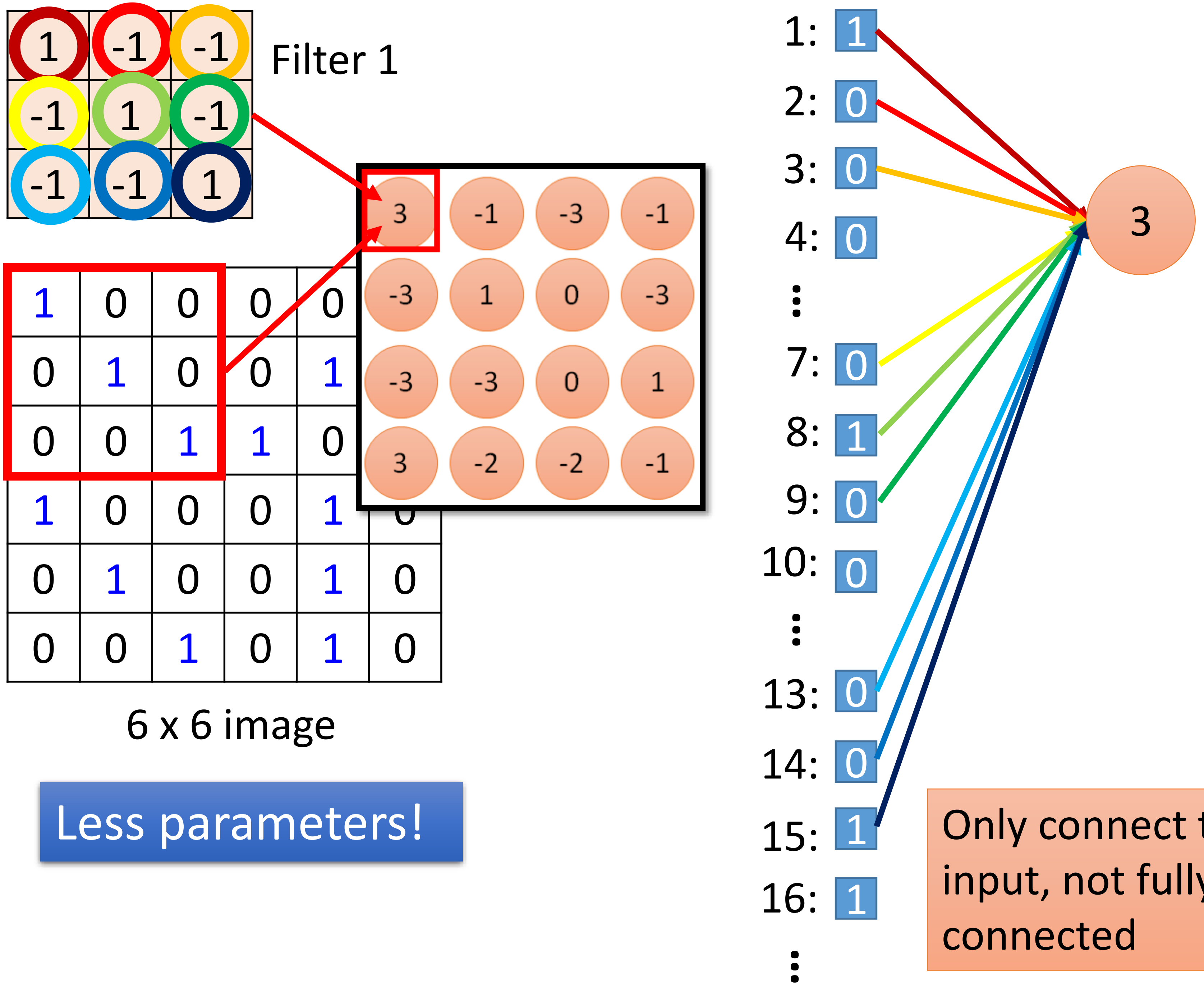
-1	1	-1
-1	1	-1
-1	1	-1

convolution

Fully-connected



1	0	0	0	0	1
0	1	0	0	1	0
0	0	1	1	0	0
1	0	0	0	1	0
0	1	0	0	1	0
0	0	1	0	1	0



Filter 1

1	-1	-1
-1	1	-1
-1	-1	1

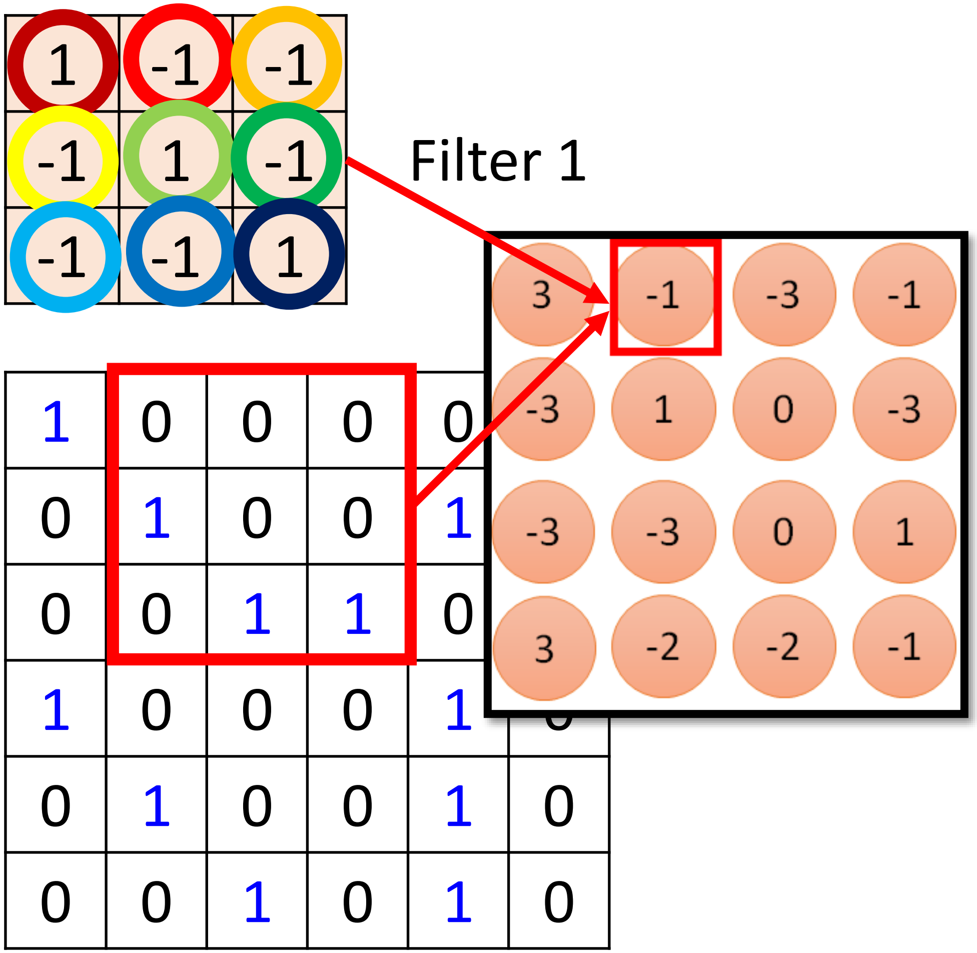
3	-1	-3	-1
-3	1	0	-3
-3	-3	0	1
3	-2	-2	-1

1	0	0	0	0	
0	1	0	0	1	
0	0	1	1	0	
1	0	0	0	1	0
0	1	0	0	1	0
0	0	1	0	1	0

6 x 6 image

Less parameters!

Only connect to 9 input, not fully connected



Filter 1

1	-1	-1
-1	1	-1
-1	-1	1

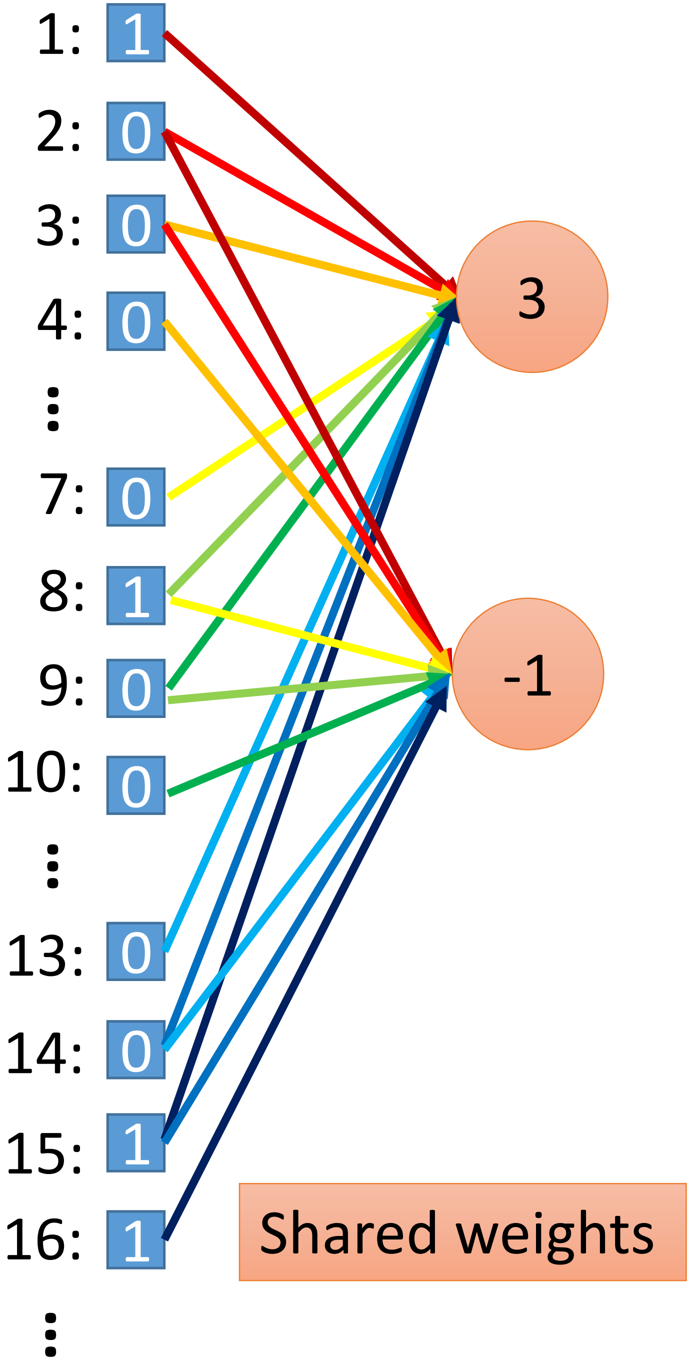
3	-1	-3	-1		
-3	1	0	-3		
-3	-3	0	1		
3	-2	-2	-1		

1	0	0	0	0	
0	1	0	0	1	
0	0	1	1	0	
1	0	0	0	1	0
0	1	0	0	1	0
0	0	1	0	1	0

6 x 6 image

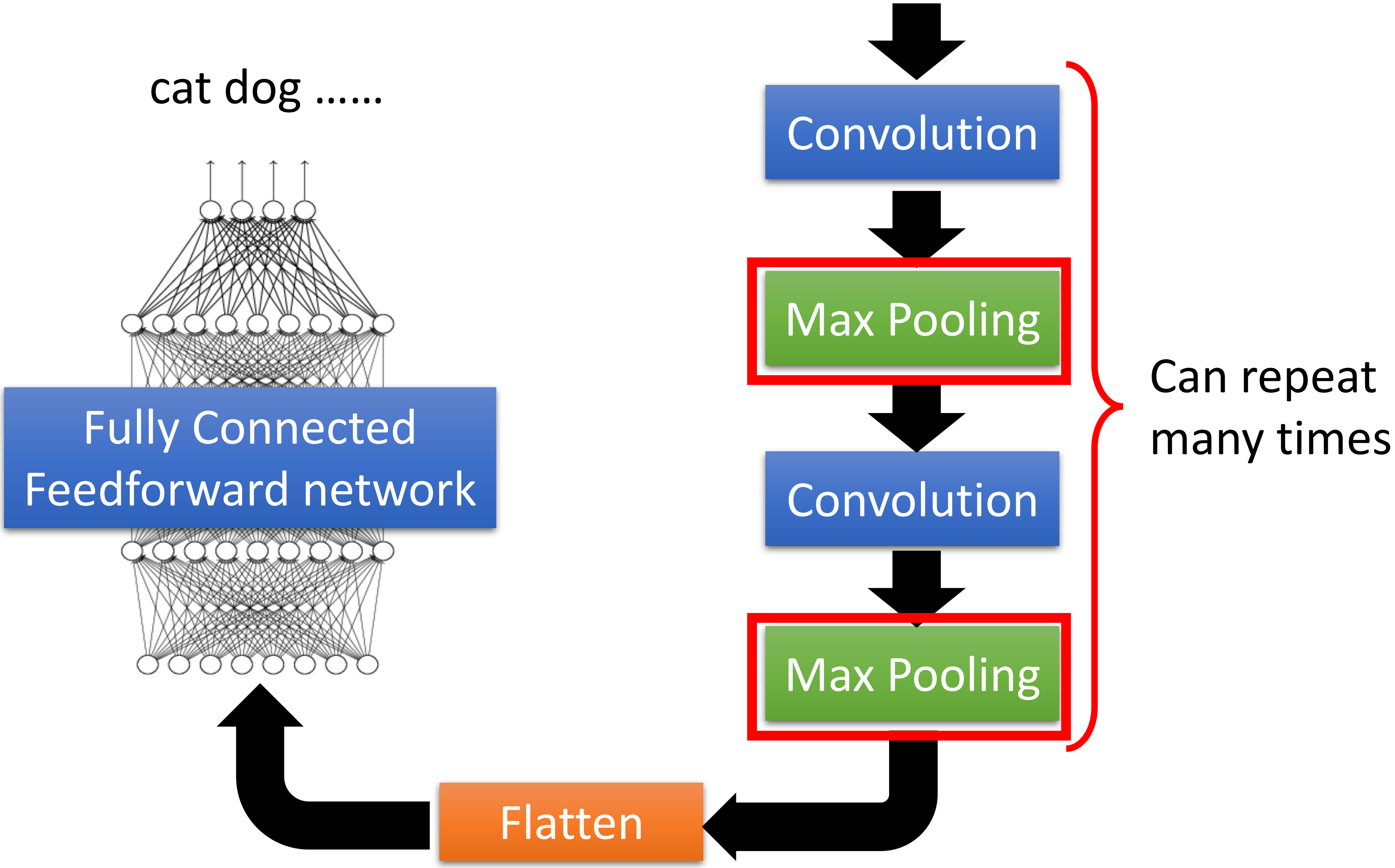
Less parameters!

Even less parameters!



Shared weights

The whole CNN



Convolution

Max Pooling

Convolution

Max Pooling

Can repeat many times

Flatten

Fully Connected Feedforward network

cat dog ......

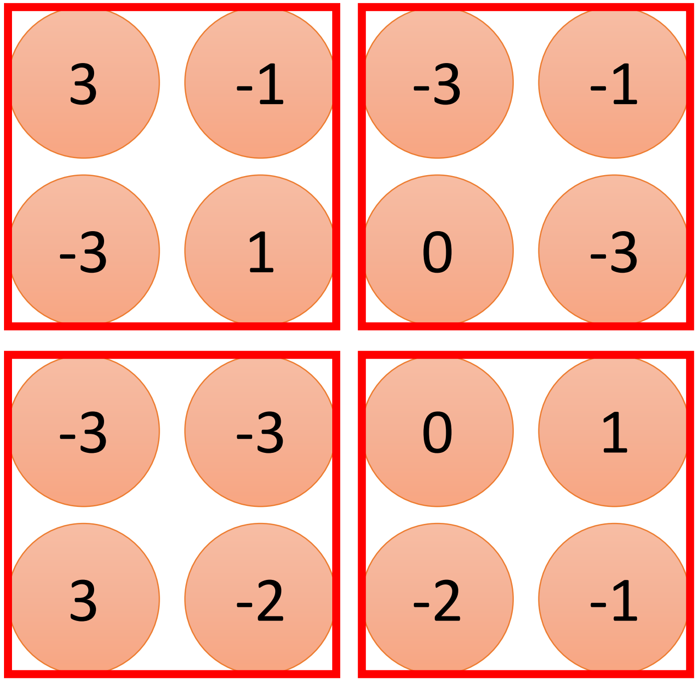
CNN – Max Pooling

1	-1	-1
-1	1	-1
-1	-1	1

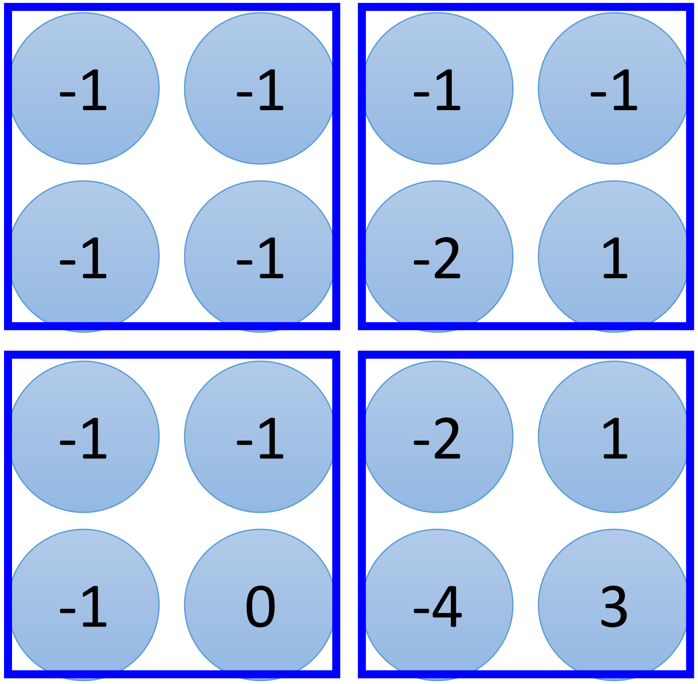
Filter 1

-1	1	-1
-1	1	-1
-1	1	-1

Filter 2



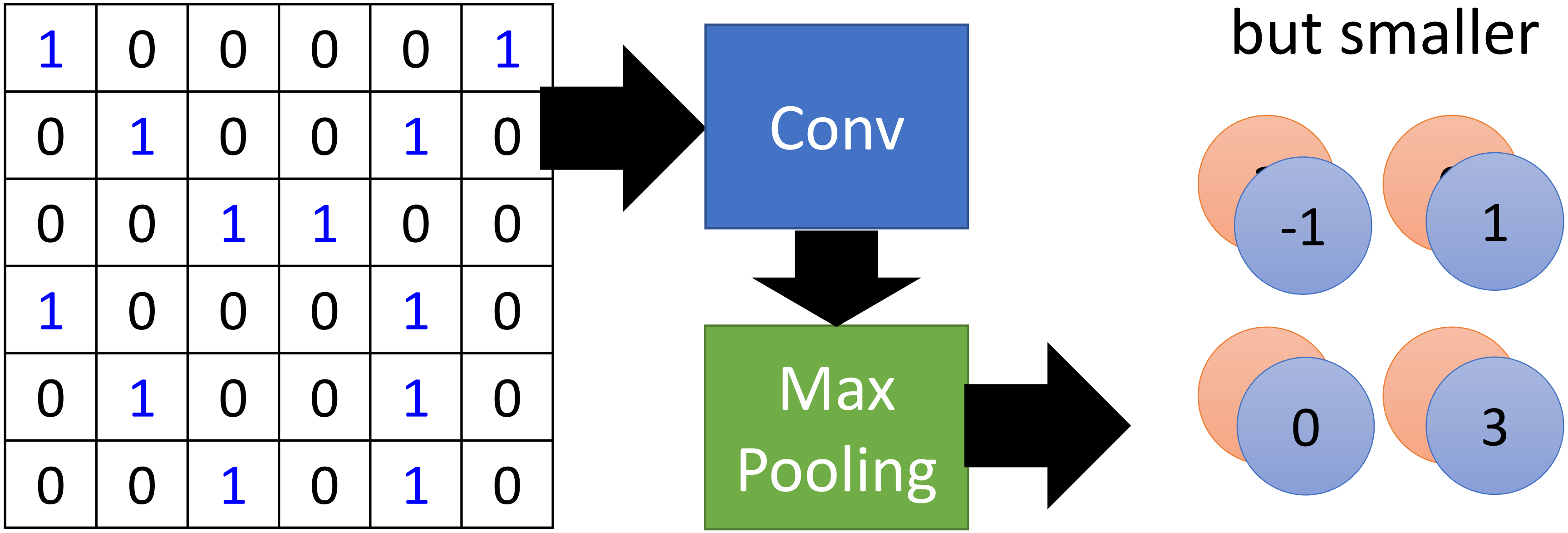
3	-1	-3	-1
-3	1	0	-3
-3	-3	0	1
3	-2	-2	-1



-1	-1	-1	-1
-1	-1	-2	1
-1	-1	-2	1
-1	0	-4	3

CNN – Max Pooling

New image
but smaller



1	0	0	0	0	1
0	1	0	0	1	0
0	0	1	1	0	0
1	0	0	0	1	0
0	1	0	0	1	0
0	0	1	0	1	0

Conv

Max
Pooling

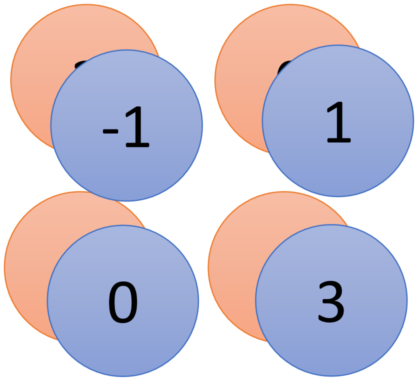
-1	1
0	3

6 x 6 image

2 x 2 image

Each filter
is a channel

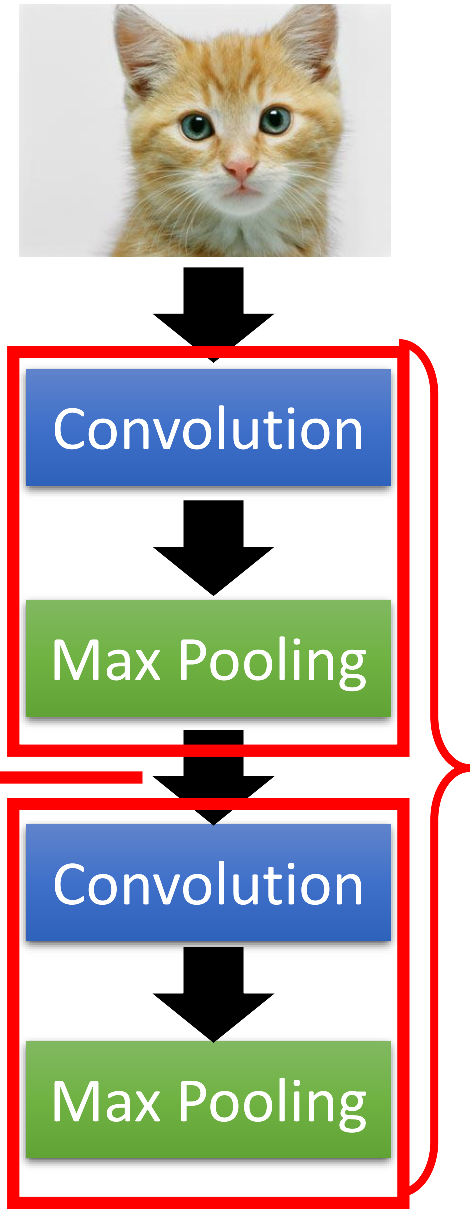
The whole CNN



A new image

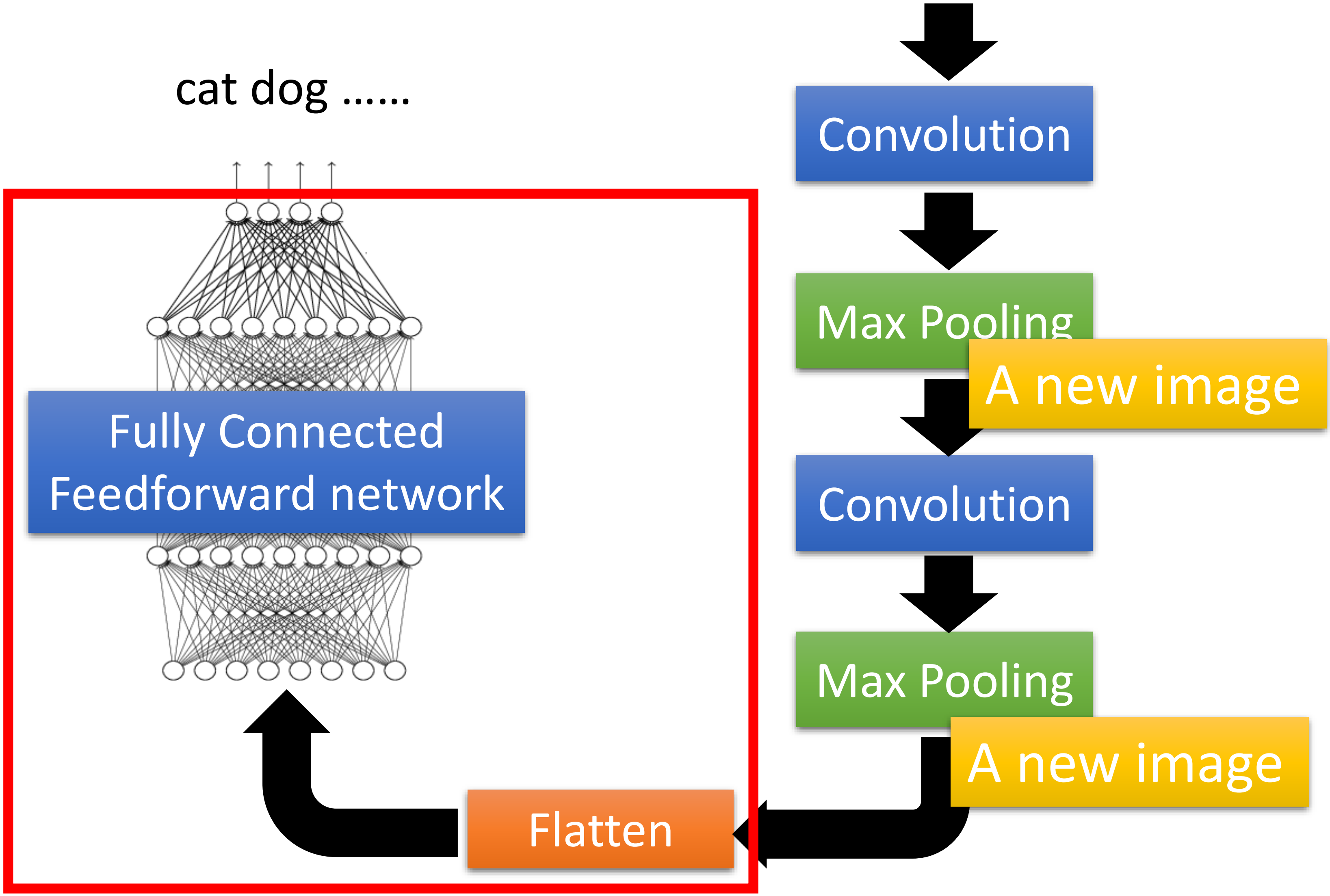
Smaller than the original image

The number of the channel is the number of filters



Can repeat many times

The whole CNN



cat dog ......

Convolution

Max Pooling

A new image

Convolution

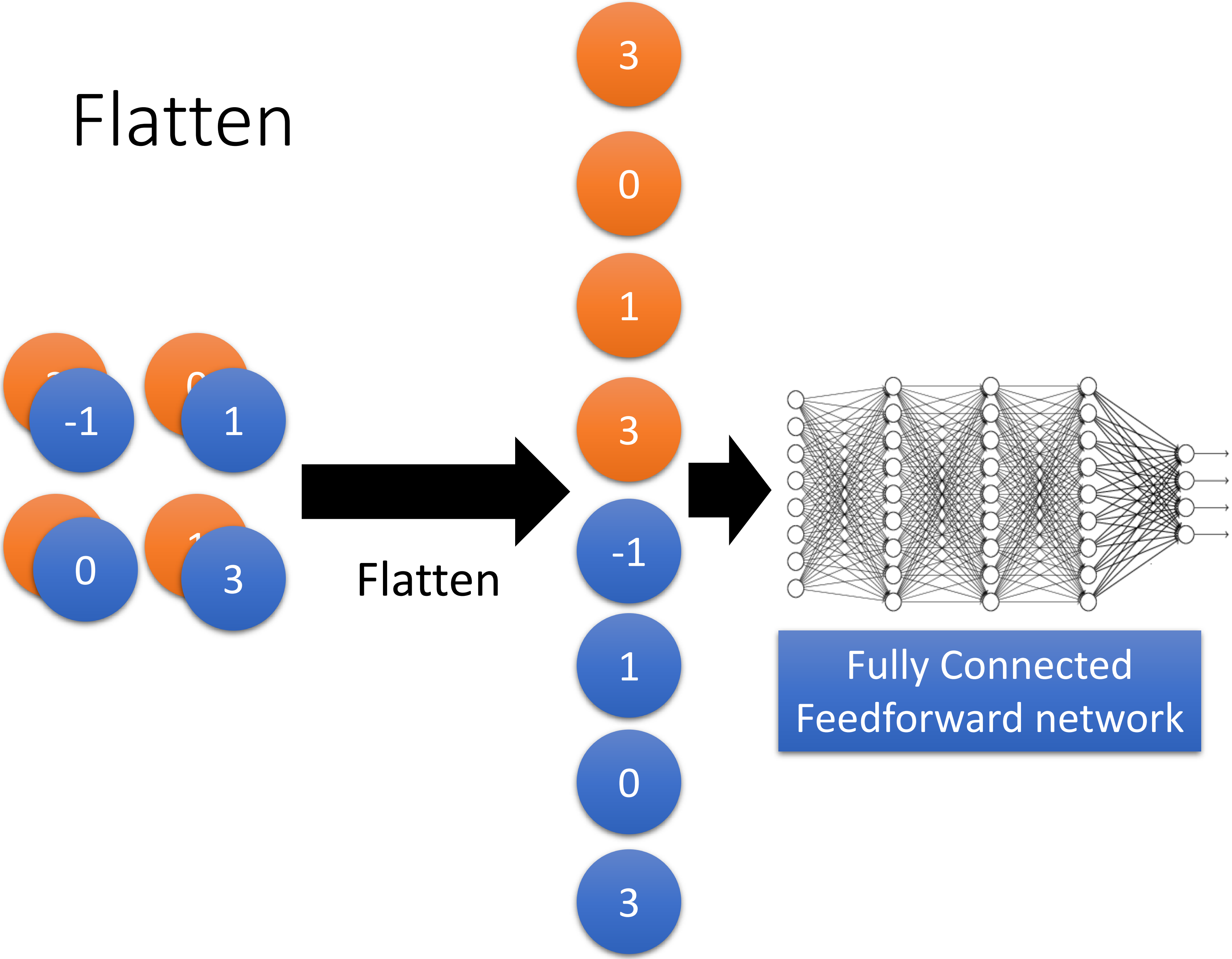
Max Pooling

A new image

Flatten

Fully Connected Feedforward network

Flatten

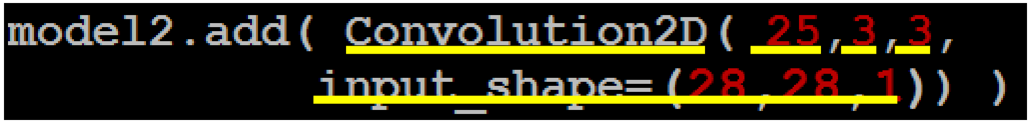


Flatten

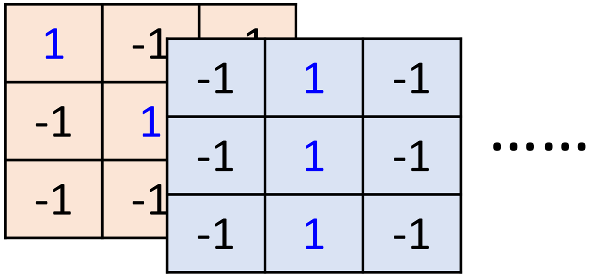
Fully Connected Feedforward network

CNN in Keras

Only modified the ***network structure*** and ***input format (vector -> 3-D tensor)***



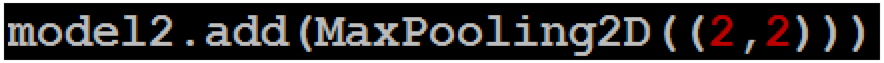
```
model2.add( Convolution2D( 25,3,3,
                 input_shape=(28,28,1)) )
```



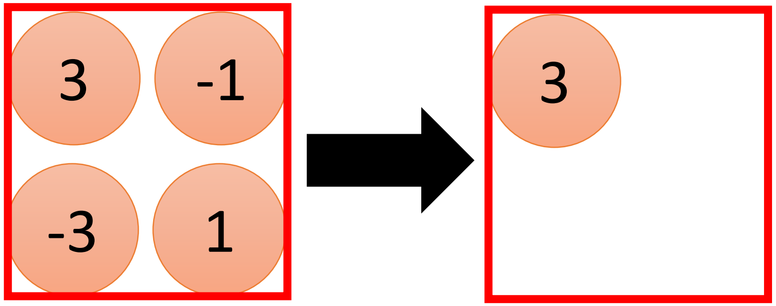
There are 25 3x3 filters.

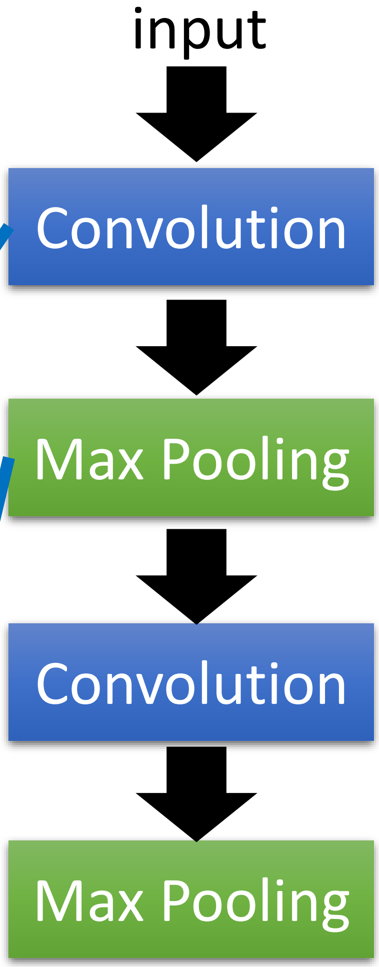
Input_shape = (28 , 28 , 1)

28 x 28 pixels   1: black/white, 3: RGB



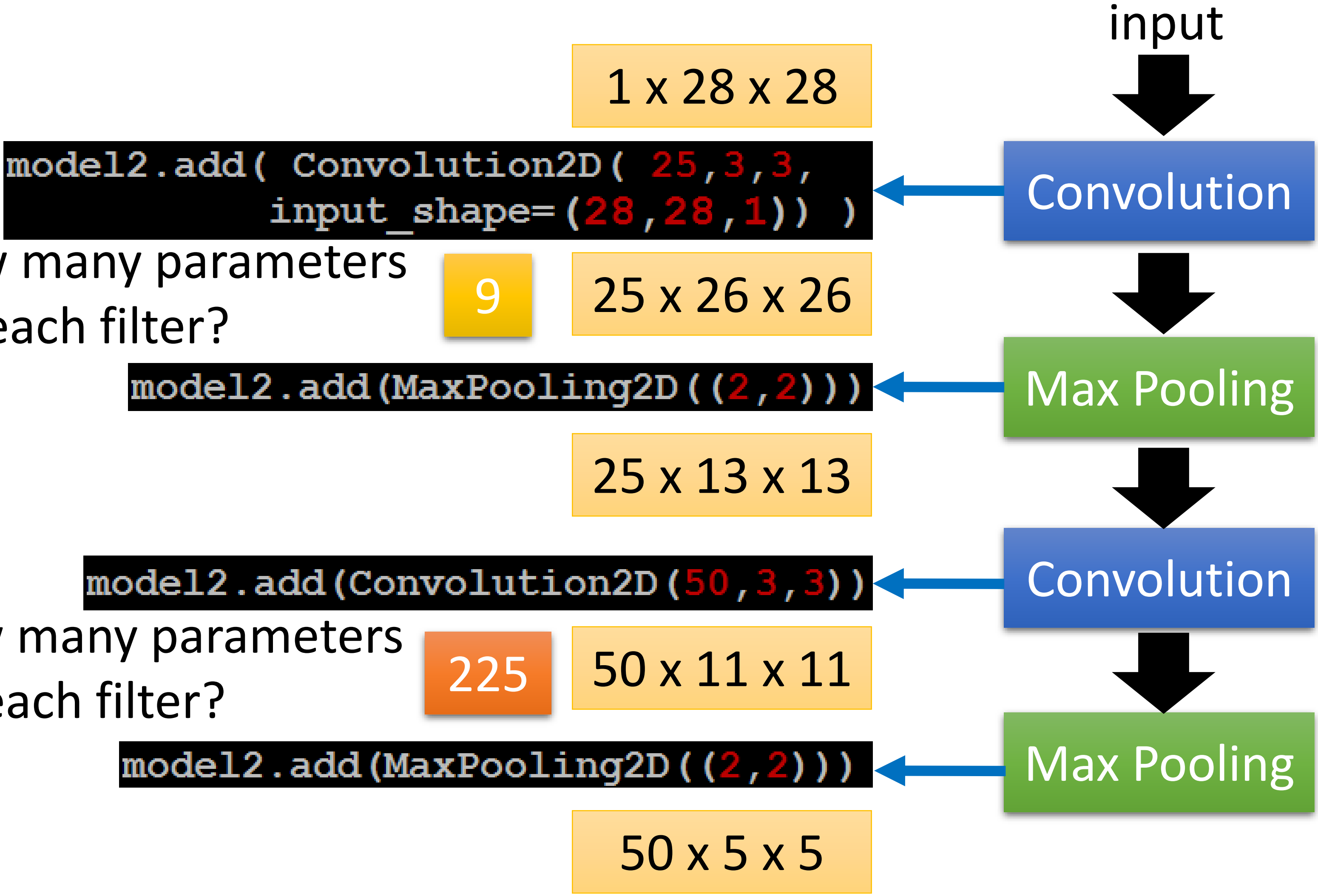
```
model2.add(MaxPooling2D((2,2)))
```





CNN in Keras

Only modified the ***network structure*** and ***input format (vector -> 3-D tensor)***



input

1 x 28 x 28

Convolution

```
model2.add( Convolution2D( 25,3,3,
            input_shape=(28,28,1)) )
```

How many parameters for each filter? 9

25 x 26 x 26

Max Pooling

```
model2.add(MaxPooling2D((2,2)))
```

25 x 13 x 13

Convolution

```
model2.add(Convolution2D(50,3,3))
```

How many parameters for each filter? 225

50 x 11 x 11

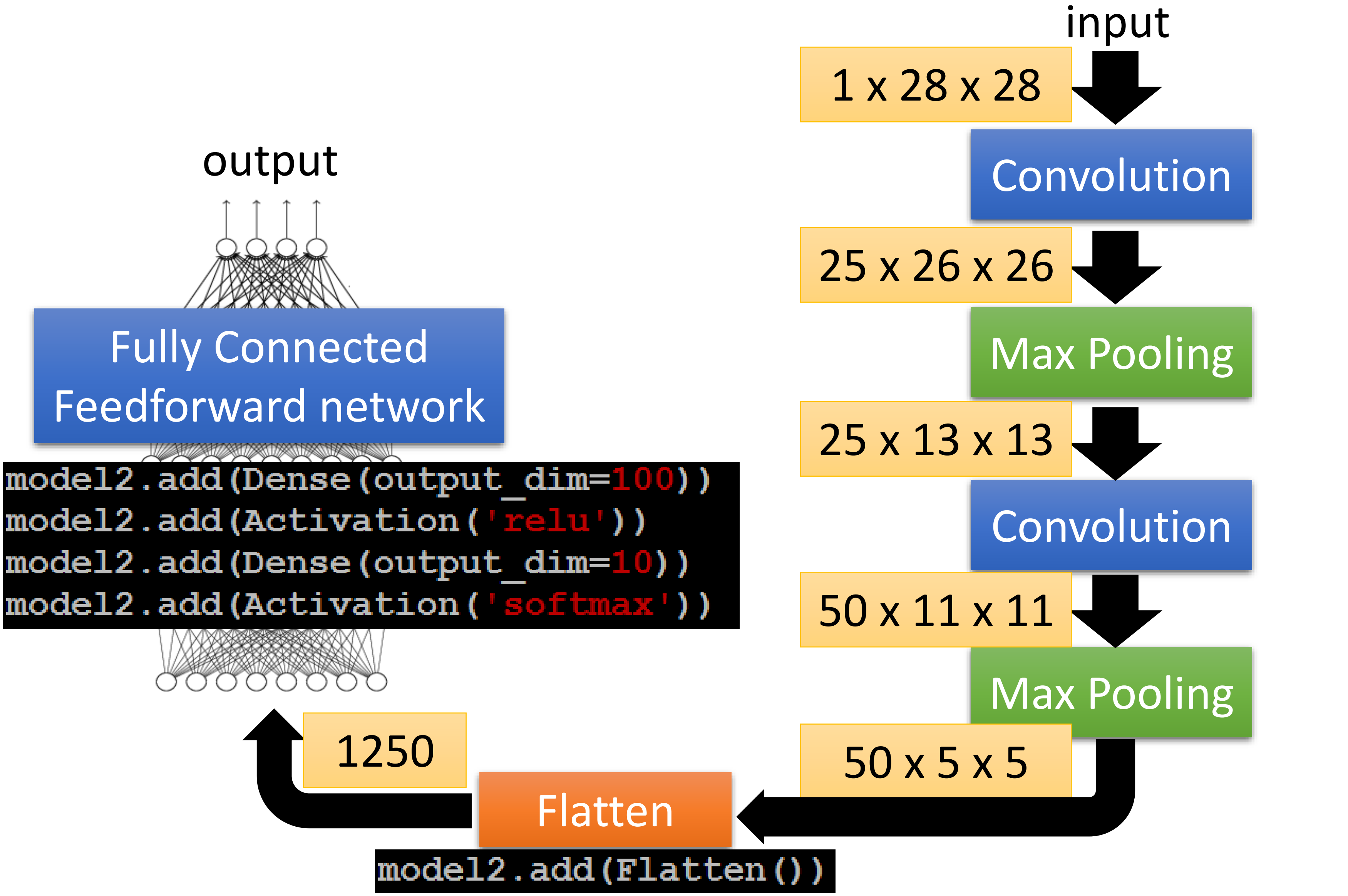
Max Pooling

```
model2.add(MaxPooling2D((2,2)))
```

50 x 5 x 5

CNN in Keras

Only modified the ***network structure*** and ***input format (vector -> 3-D tensor)***

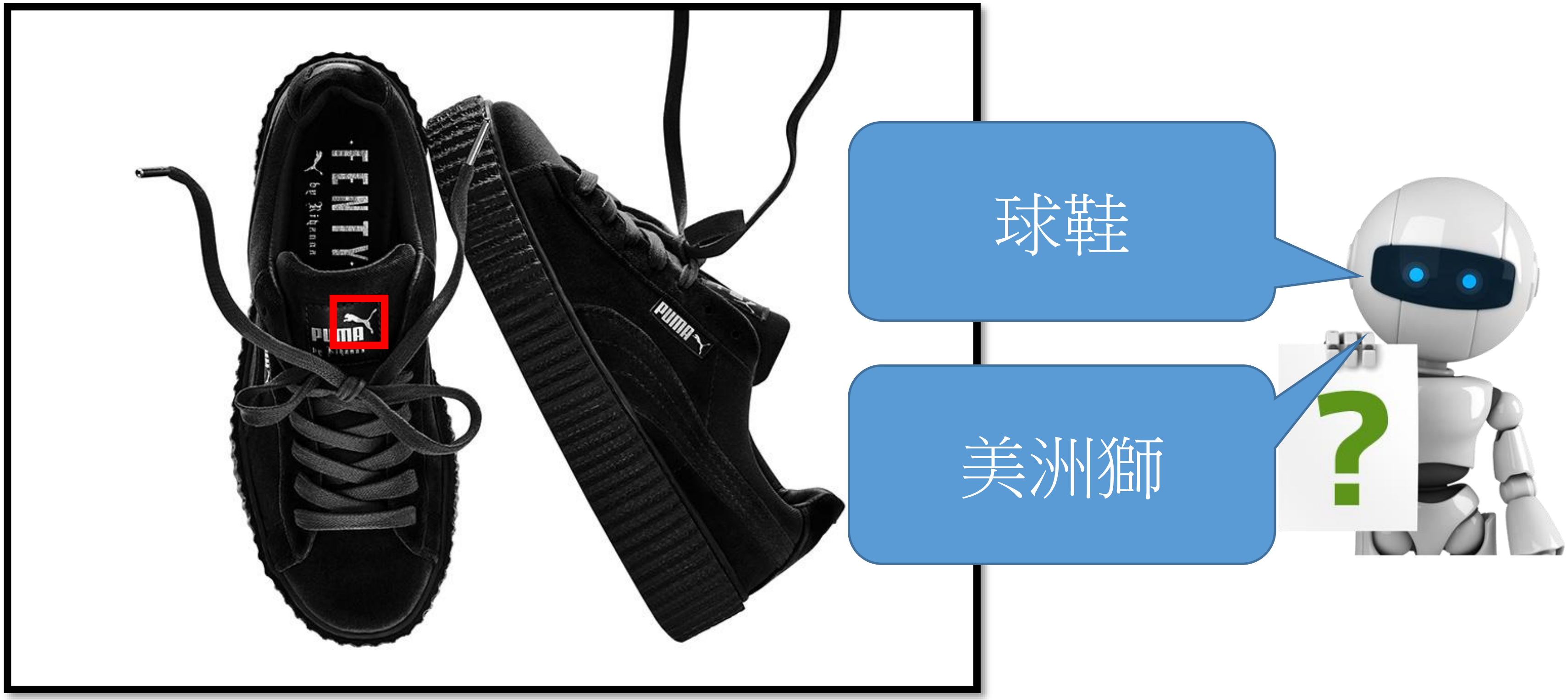


```
model2.add(Dense(output_dim=100))
model2.add(Activation('relu'))
model2.add(Dense(output_dim=10))
model2.add(Activation('softmax'))
```

```
model2.add(Flatten())
```

Live Demo

What does machine learn?



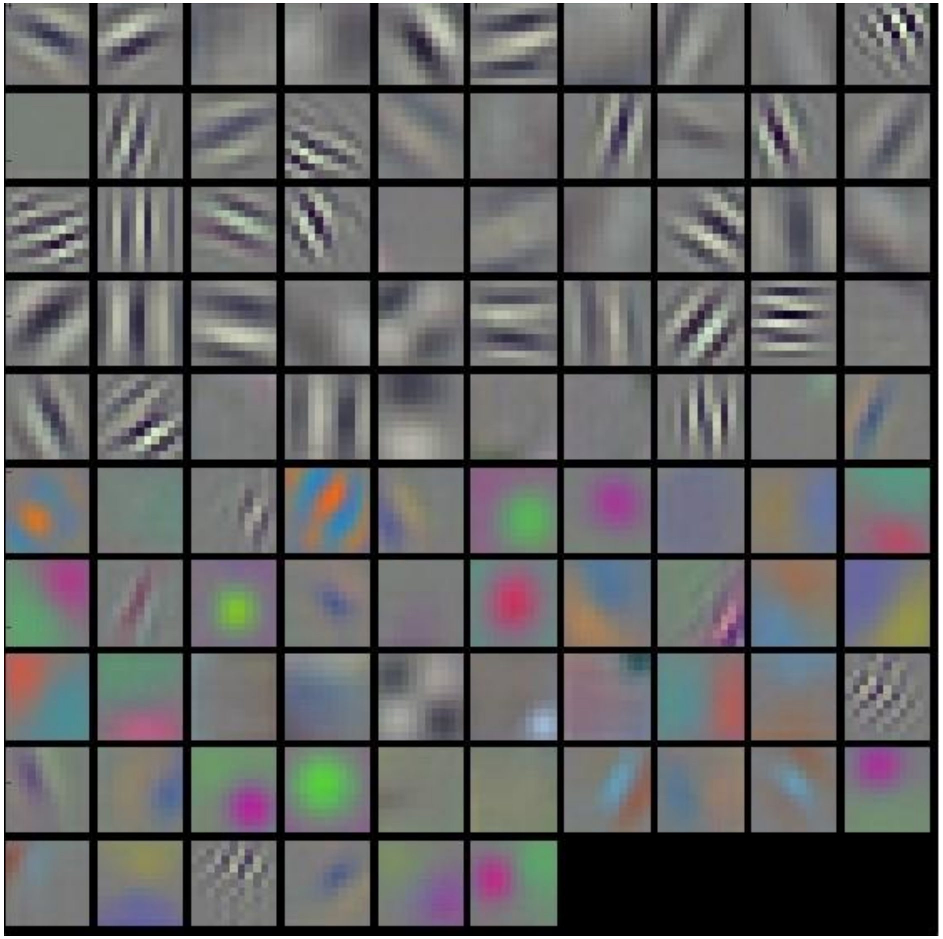
球鞋

美洲獅

http://newsneakernews.wpengine.netdna-cdn.com/wp-content/uploads/2016/11/rihanna-puma-creeper-velvet-release-date-02.jpg

First Convolution Layer

- Typical-looking filters on the trained first layer

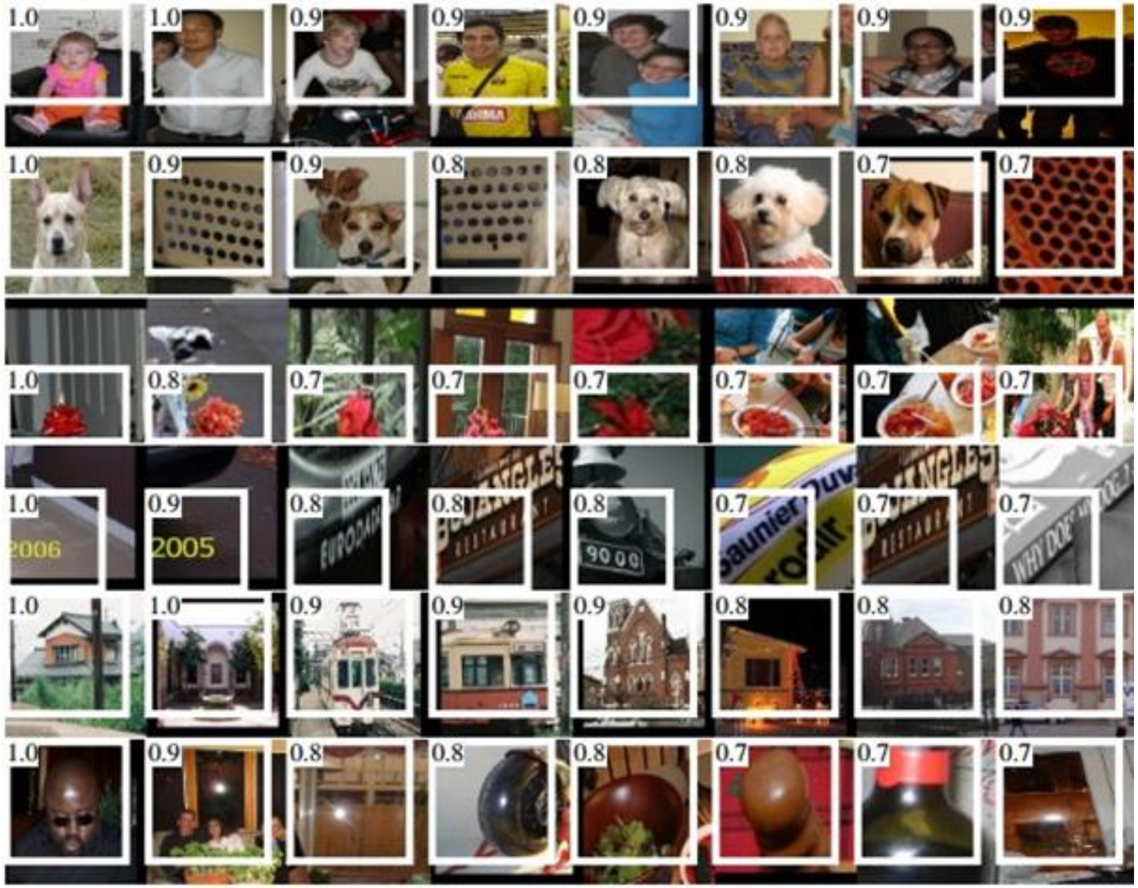


11 x 11
(AlexNet)

http://cs231n.github.io/understanding-cnn/

How about higher layers?

- Which images make a specific neuron activate



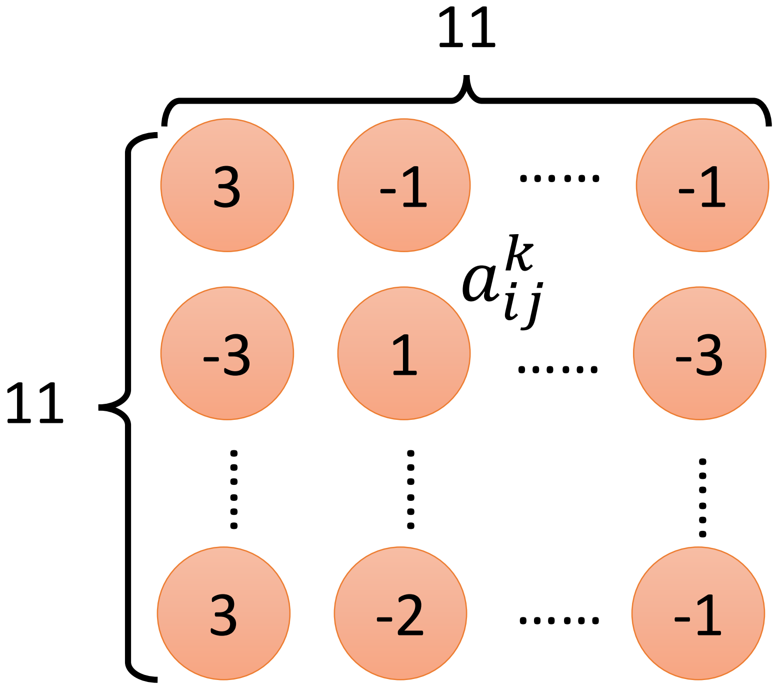
Ross Girshick, Jeff Donahue, Trevor Darrell, Jitendra Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation”, CVPR, 2014

What does CNN learn?

The output of the k-th filter is a 11 x 11 matrix.

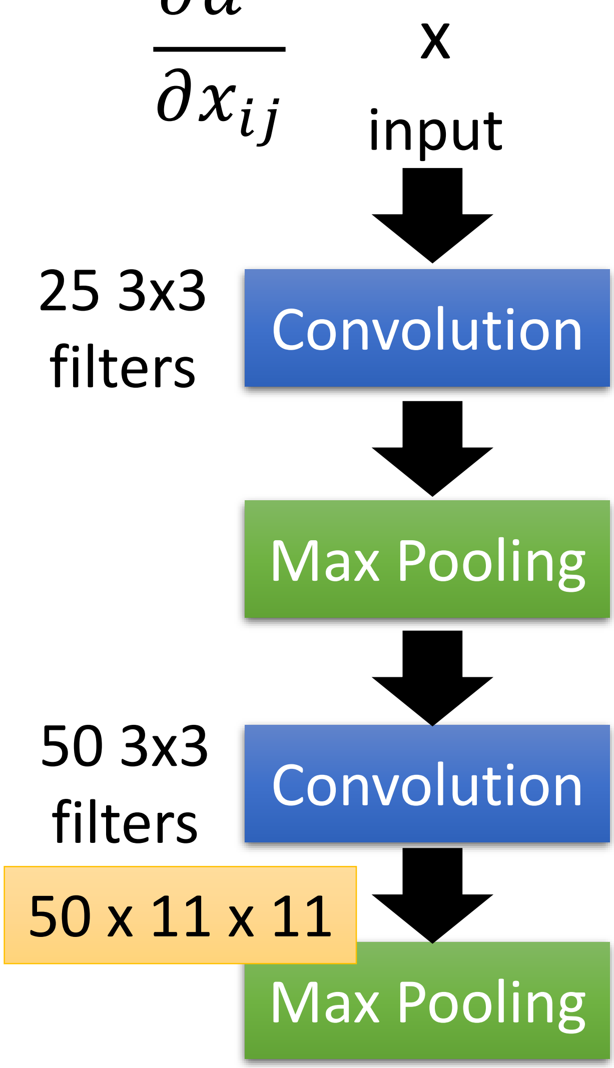
Degree of the activation of the k-th filter: $a^k = \sum_{i=1}^{11} \sum_{j=1}^{11} a_{ij}^k$

$x^* = arg \max_x a^k$ (gradient ascent)



$$\begin{array}{c|cccc} & & 11 & & \\ \hline & 3 & -1 & \cdots\cdots & -1 \\ & -3 & 1 & \cdots\cdots & -3 \\ 11 & \vdots & \vdots & & \vdots \\ & 3 & -2 & \cdots\cdots & -1 \end{array}$$

$$\frac{\partial a^k}{\partial x_{ij}}$$



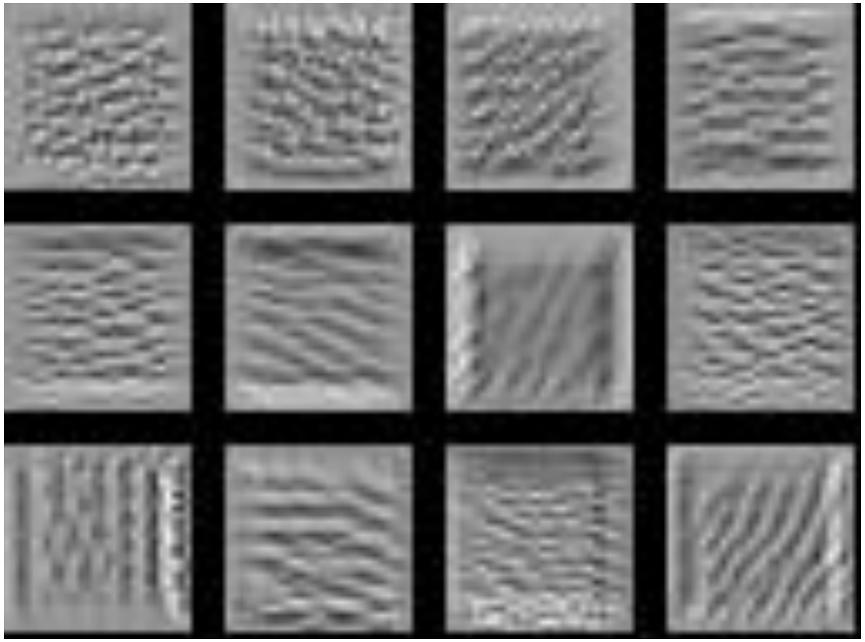
What does CNN learn?

The output of the k-th filter is a 11 x 11 matrix.

Degree of the activation of the k-th filter:

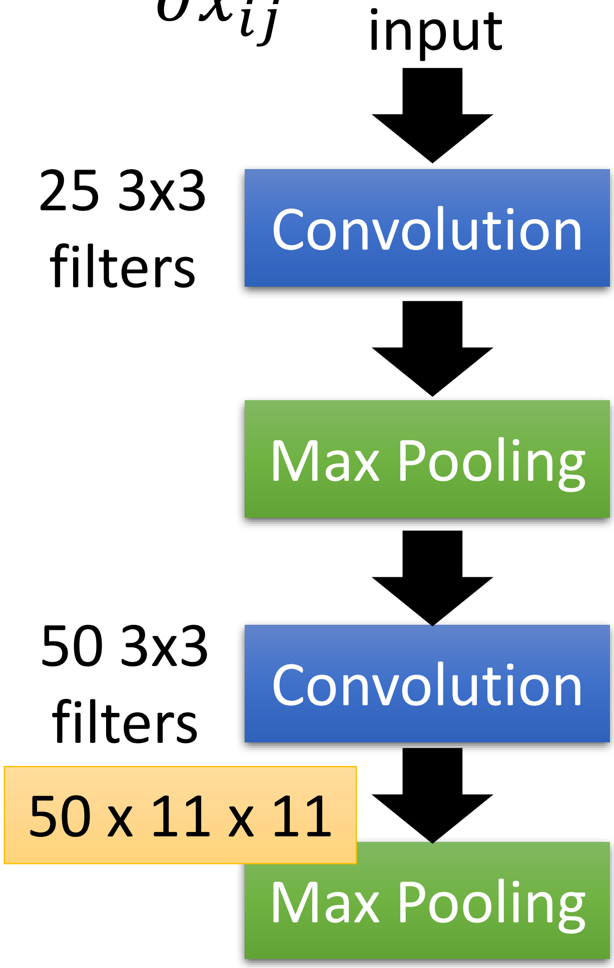
$$a^k = \sum_{i=1}^{11} \sum_{j=1}^{11} a_{ij}^k$$

$$x^* = arg \max_x a^k$$ (gradient ascent)



For each filter

$$\frac{\partial a^k}{\partial x_{ij}}$$



input

25 3x3 filters

Convolution

Max Pooling

50 3x3 filters

Convolution

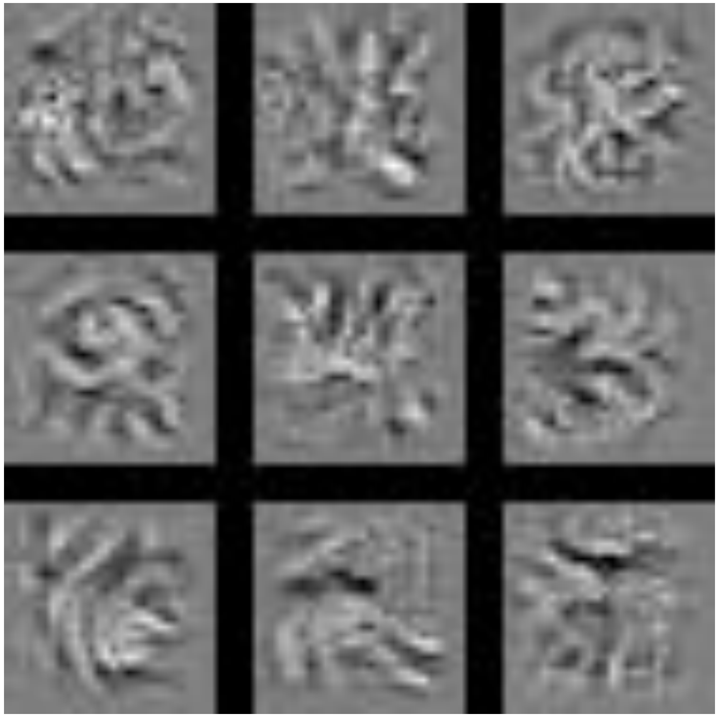
50 x 11 x 11

Max Pooling

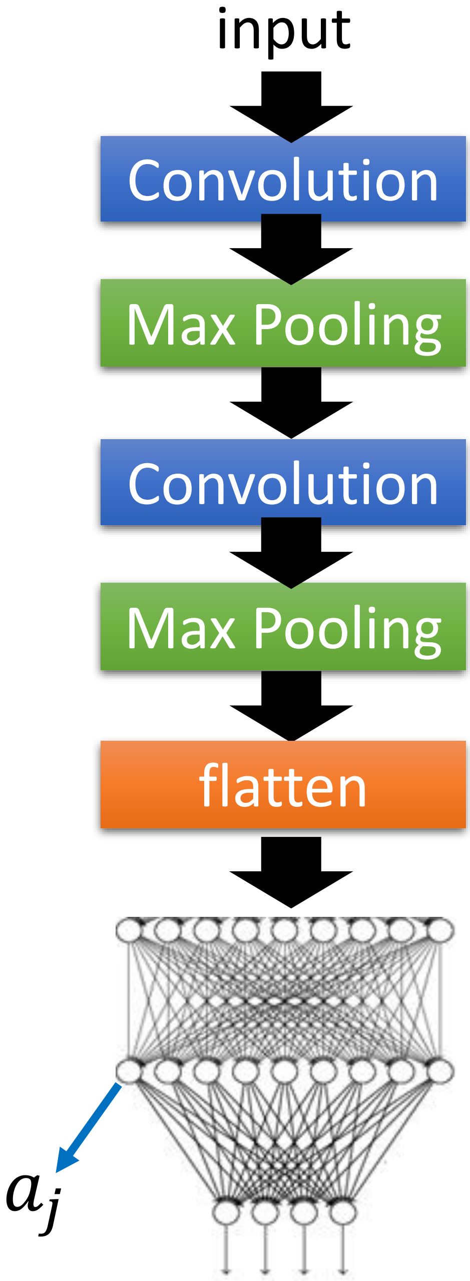
What does CNN learn?

Find an image maximizing the output of neuron:

$$x^* = arg\max_{x} a^j$$



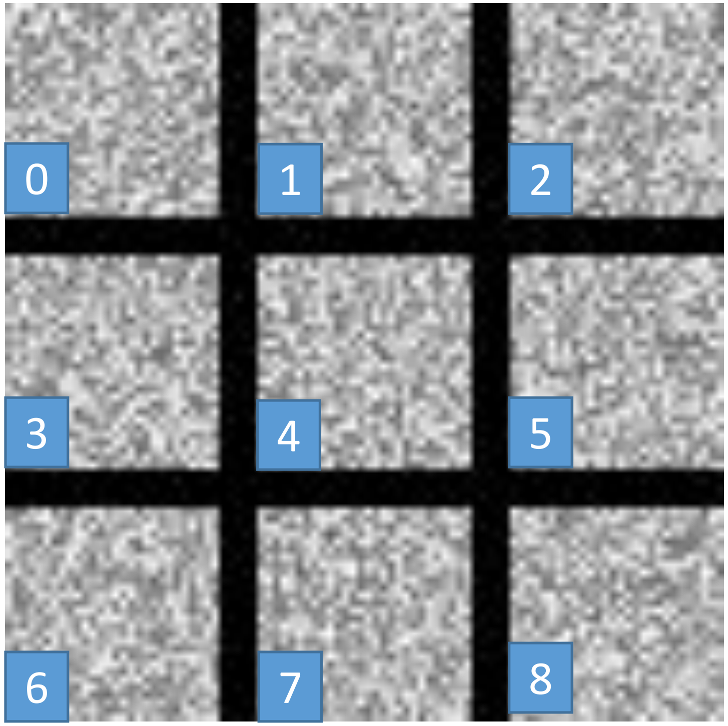
Each figure corresponds to a neuron



What does CNN learn?

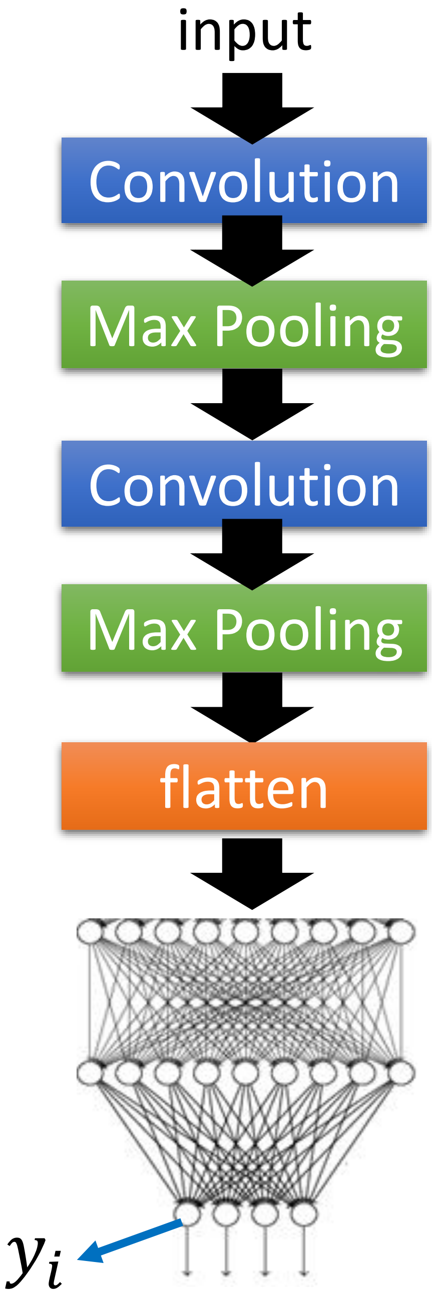
$$x^* = arg\max_x y^i$$

Can we see digits?



Deep Neural Networks are Easily Fooled

https://www.youtube.com/watch?v=M2IebCN9Ht4



input

Convolution

Max Pooling

Convolution

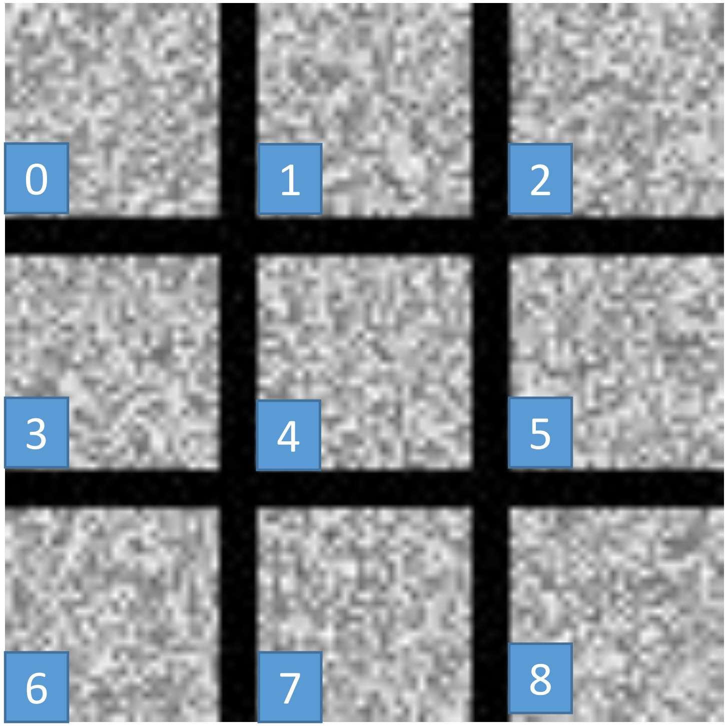
Max Pooling

flatten

y_i

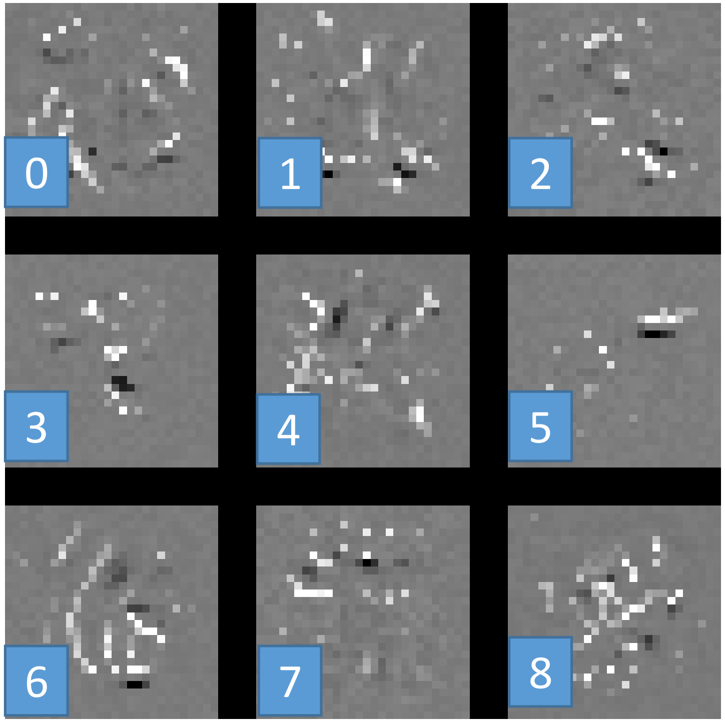
What does CNN learn?

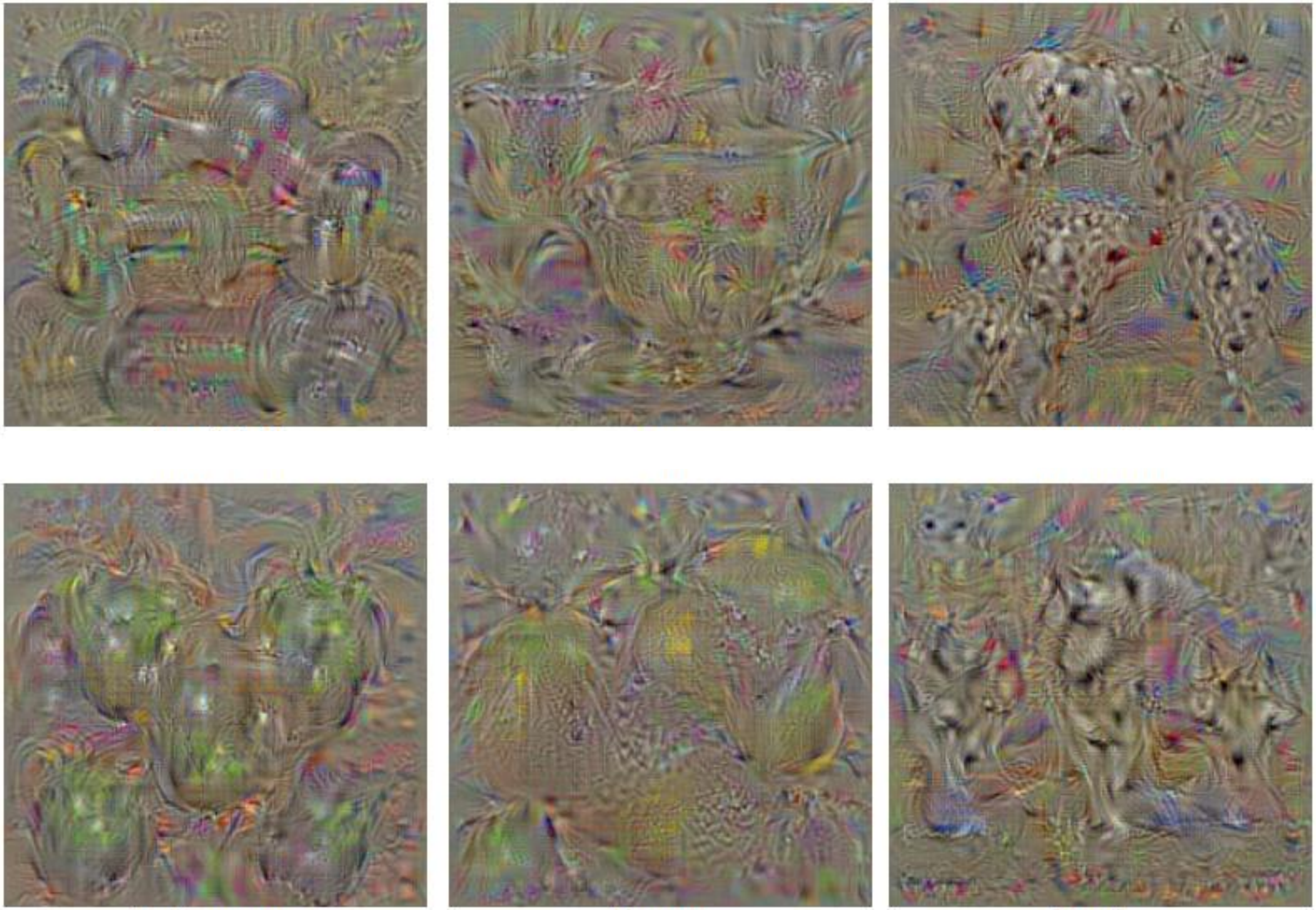
$$x^* = arg \max_x y^i$$



Over all pixel values

$$x^* = arg \max_x \left(y^i - \boxed{\sum_{i,j} \left| x_{ij} \right|} \right)$$

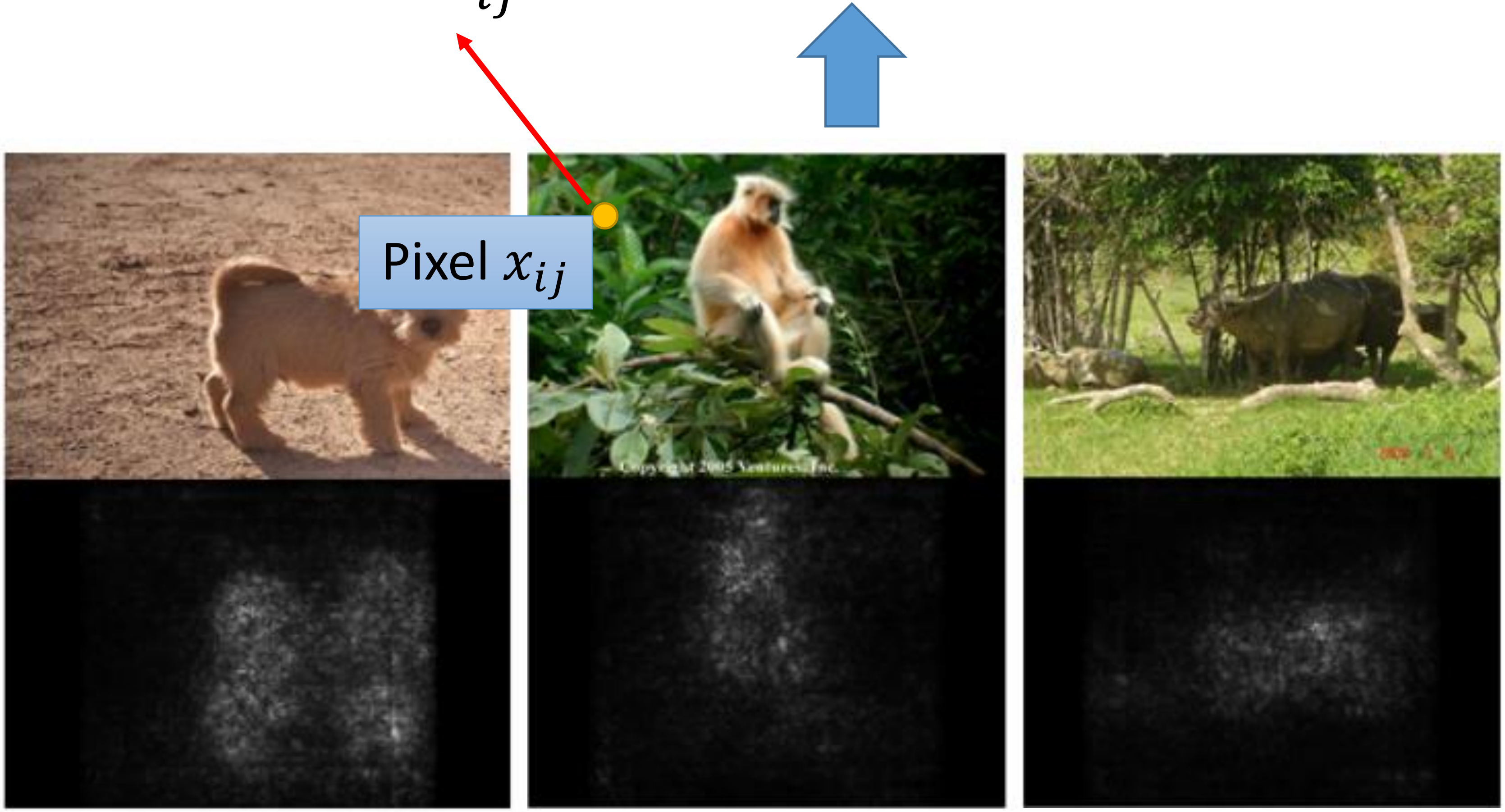




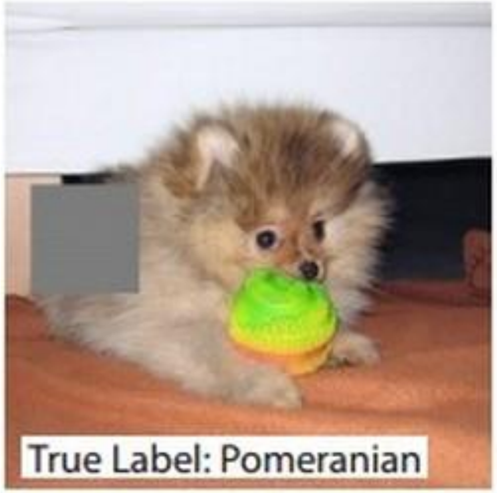
Karen Simonyan, Andrea Vedaldi, Andrew Zisserman, “Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps”, ICLR, 2014

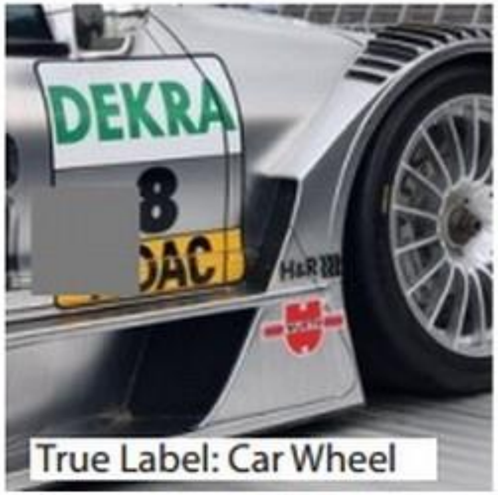
$$\left|\frac{\partial y_k}{\partial x_{ij}}\right|$$

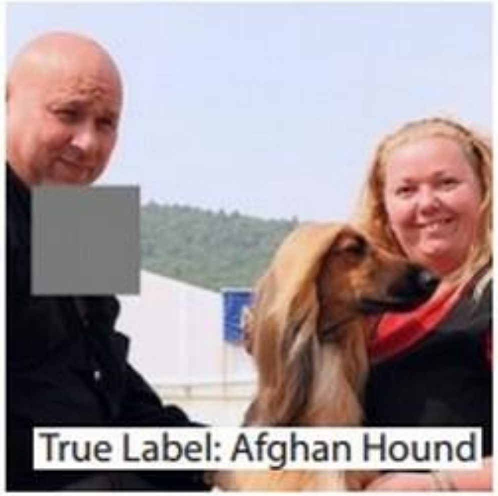
y_k: the predicted class of the model



Karen Simonyan, Andrea Vedaldi, Andrew Zisserman, "Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps", ICLR, 2014

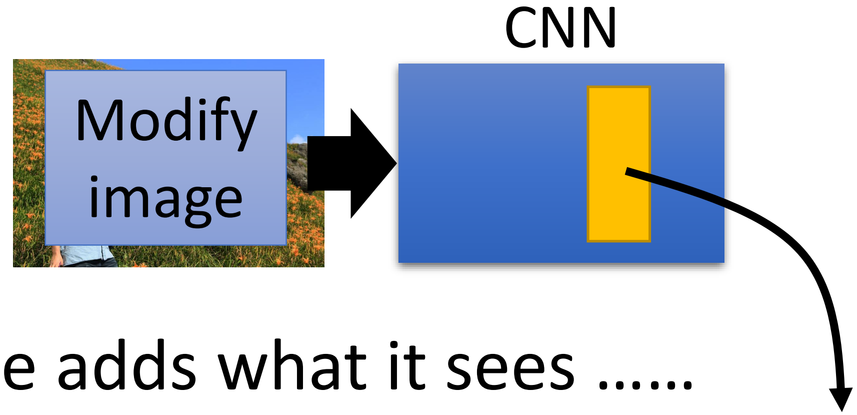






Reference: Zeiler, M. D., & Fergus, R. (2014). Visualizing and understanding convolutional networks. In *Computer Vision–ECCV 2014* (pp. 818-833)

Deep Dream



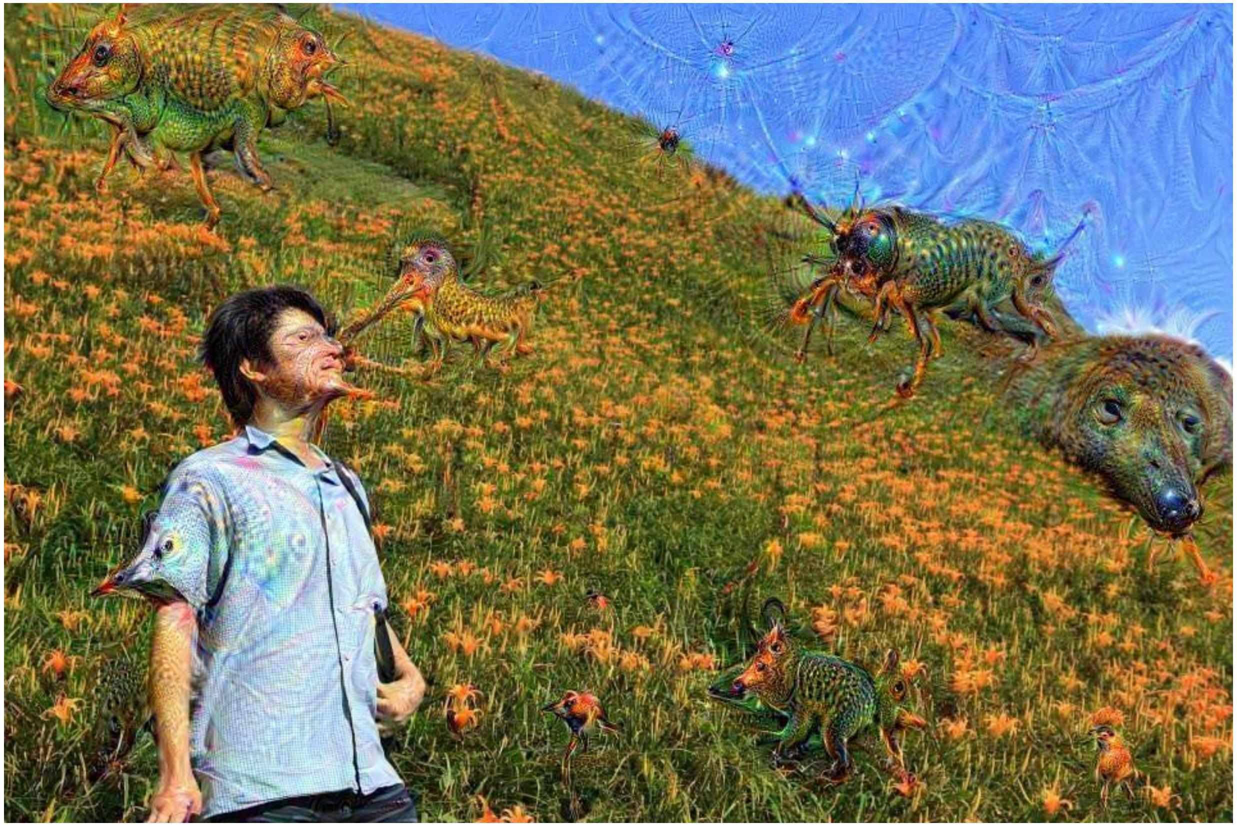
- Given a photo, machine adds what it sees ……

$$\begin{bmatrix} 3.9 \\ -1.5 \\ 2.3 \\ \vdots \end{bmatrix}$$

http://deepdreamgenerator.com/

Deep Dream

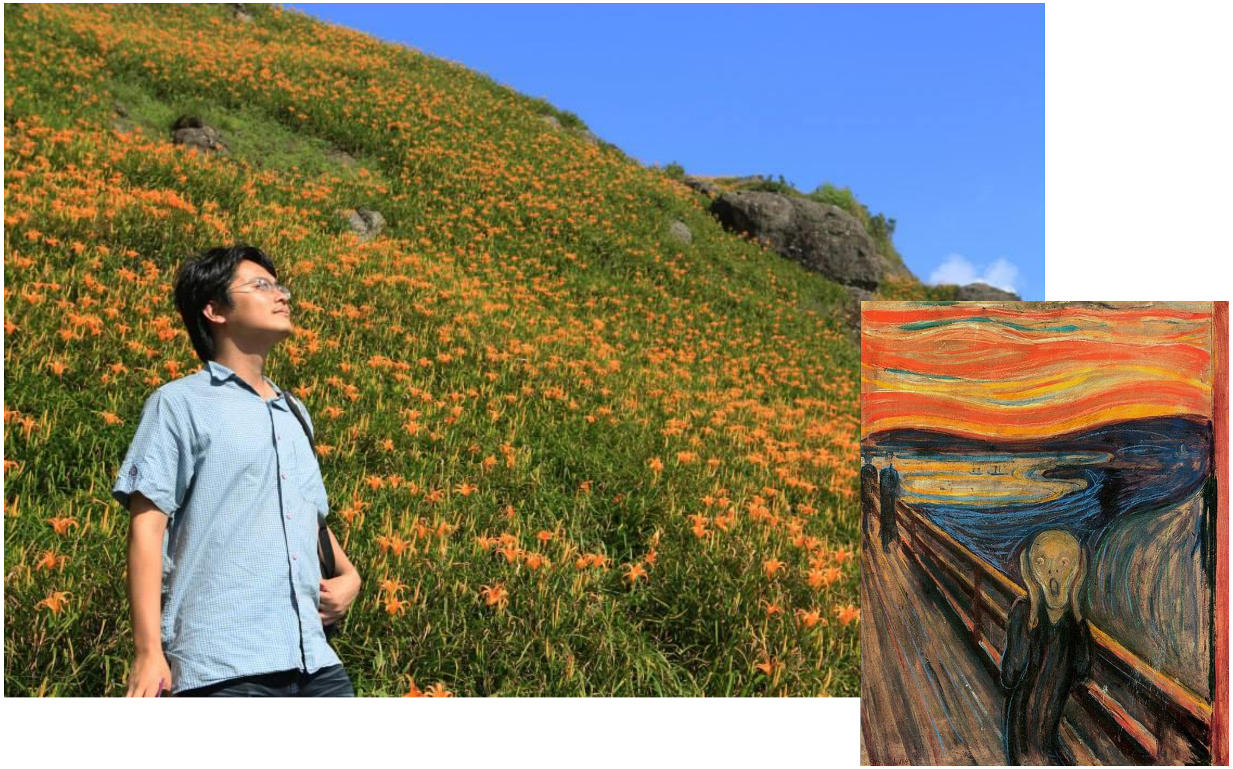
- Given a photo, machine adds what it sees ……



http://deepdreamgenerator.com/

Deep Style

- Given a photo, make its style like famous paintings



https://dreamscopeapp.com/

Deep Style

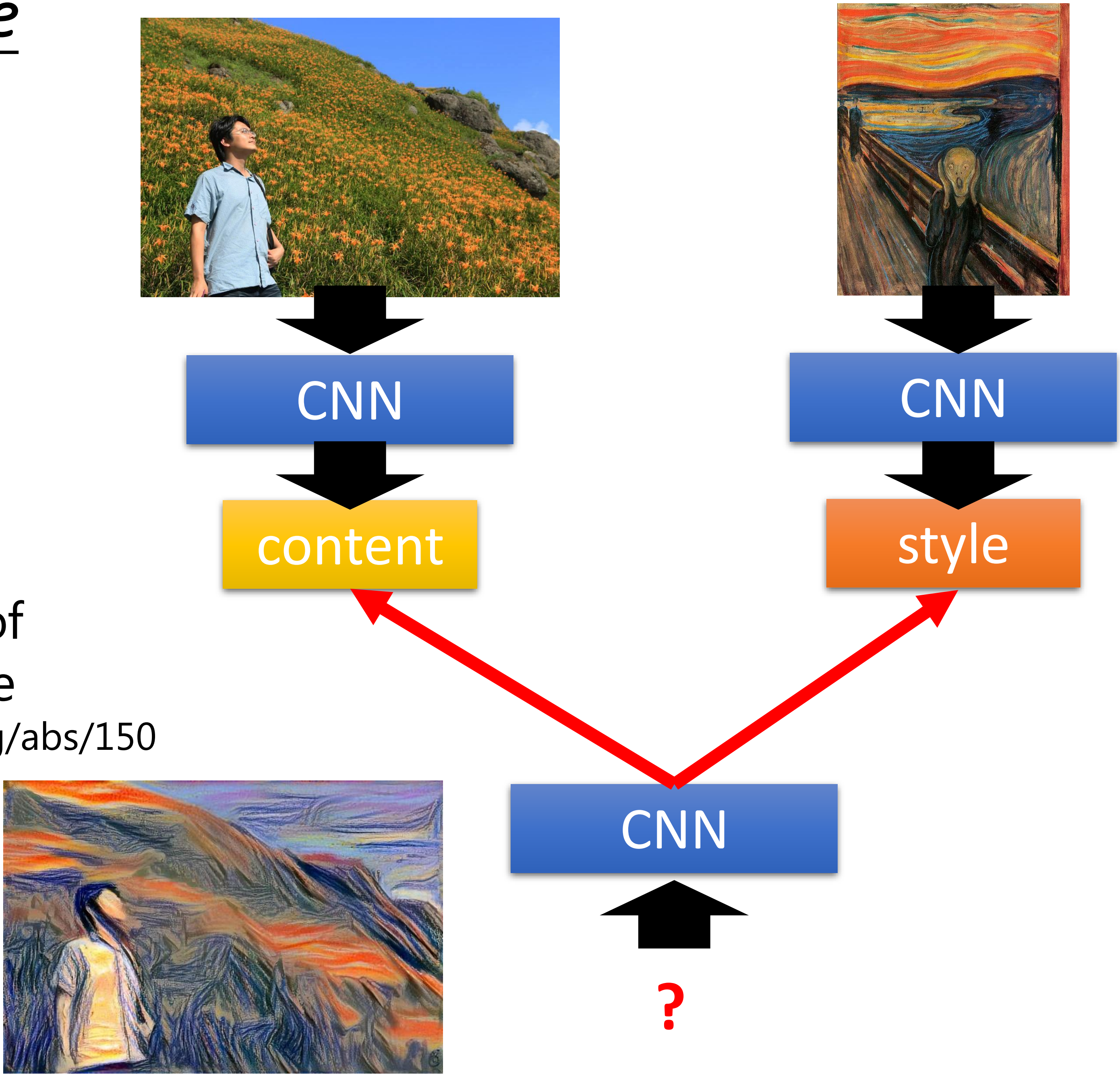
- Given a photo, make its style like famous paintings

https://dreamscopeapp.com/

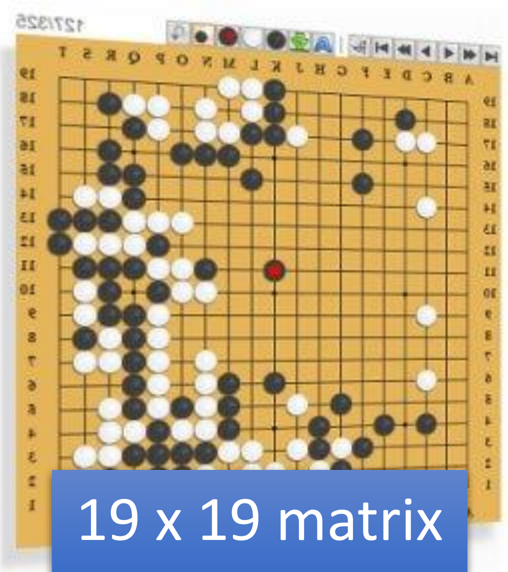
Deep Style

A Neural Algorithm of Artistic Style

https://arxiv.org/abs/1508.06576



More Application: Playing Go



19 x 19 matrix (image)

Network

Next move (19 x 19 positions)

19 x 19 vector

Black: 1

white: -1

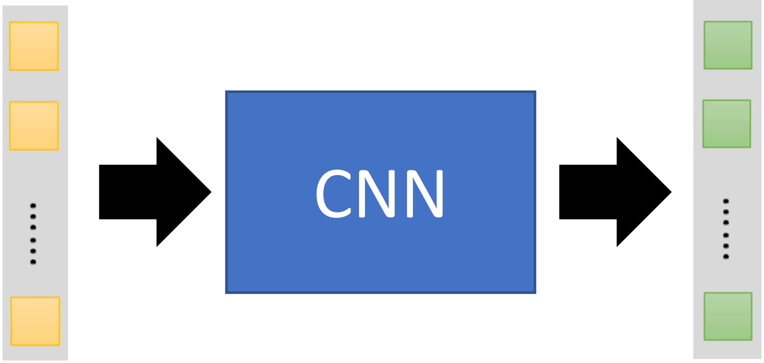
none: 0

Fully-connected feedforward network can be used

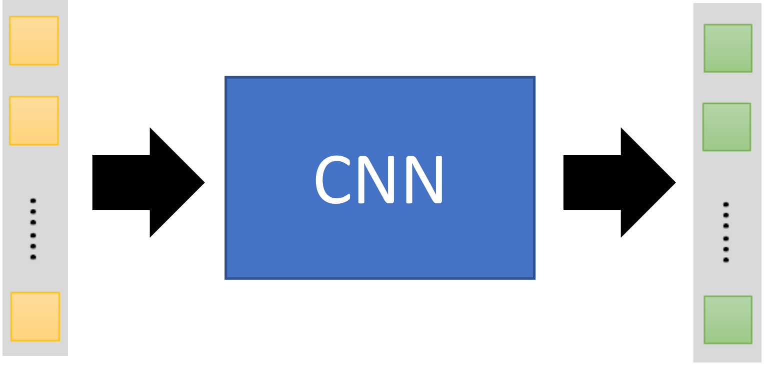
But CNN performs much better.

More Application: Playing Go

Training: record of previous plays 黑: 5之五 → 白: 天元 → 黑: 五之5 …



Target:
“天元” = 1
else = 0

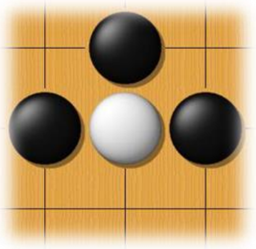


Target:
“五之 5” = 1
else = 0

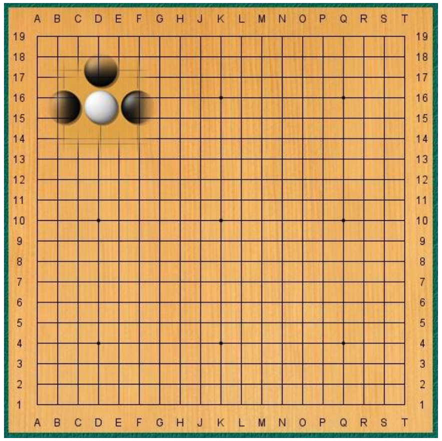
Why CNN for playing Go?

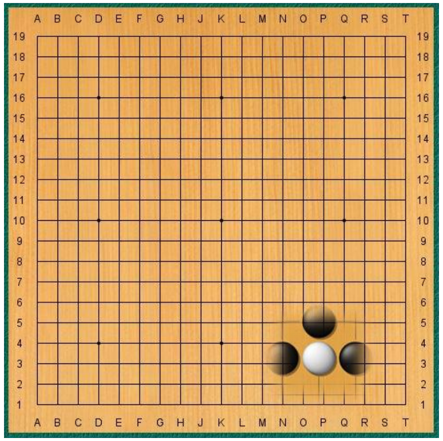
- Some patterns are much smaller than the whole image

Alpha Go uses 5 x 5 for first layer



- The same patterns appear in different regions.



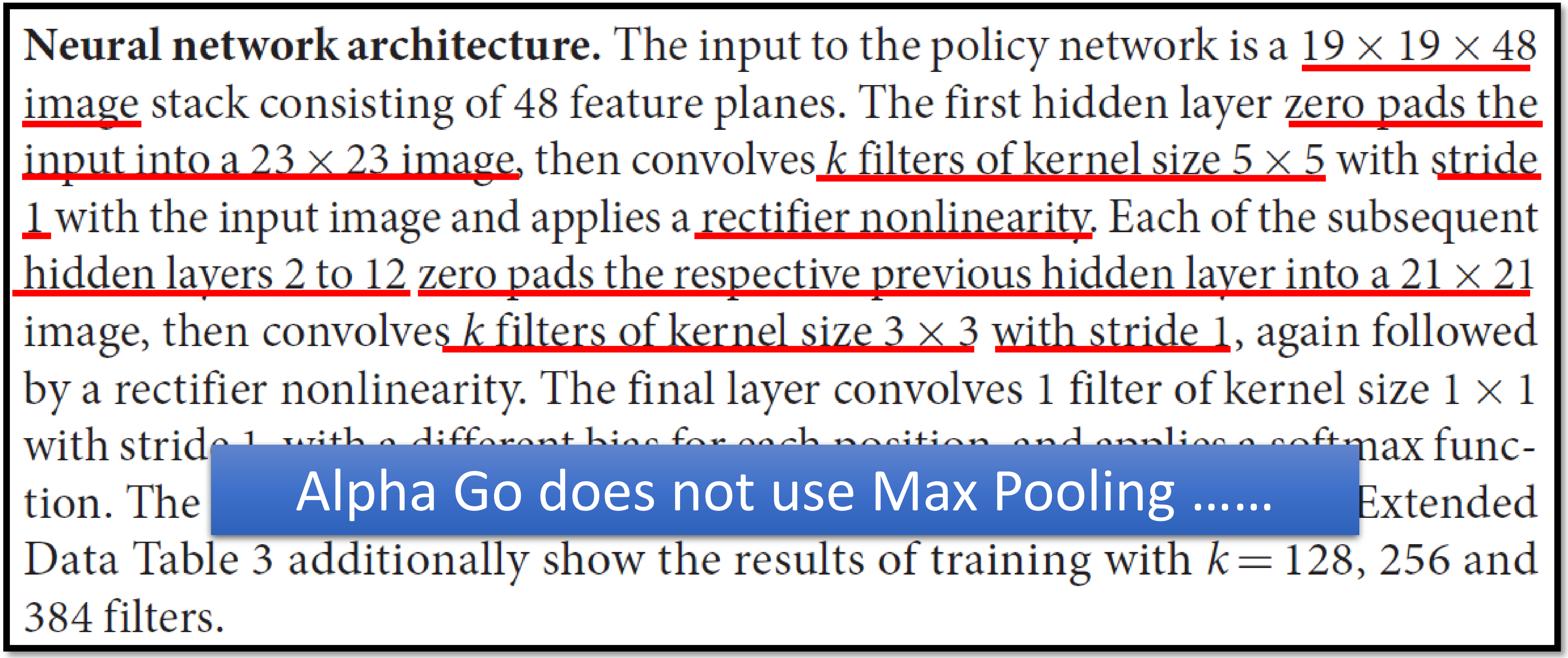


Why CNN for playing Go?

- Subsampling the pixels will not change the object

Max Pooling

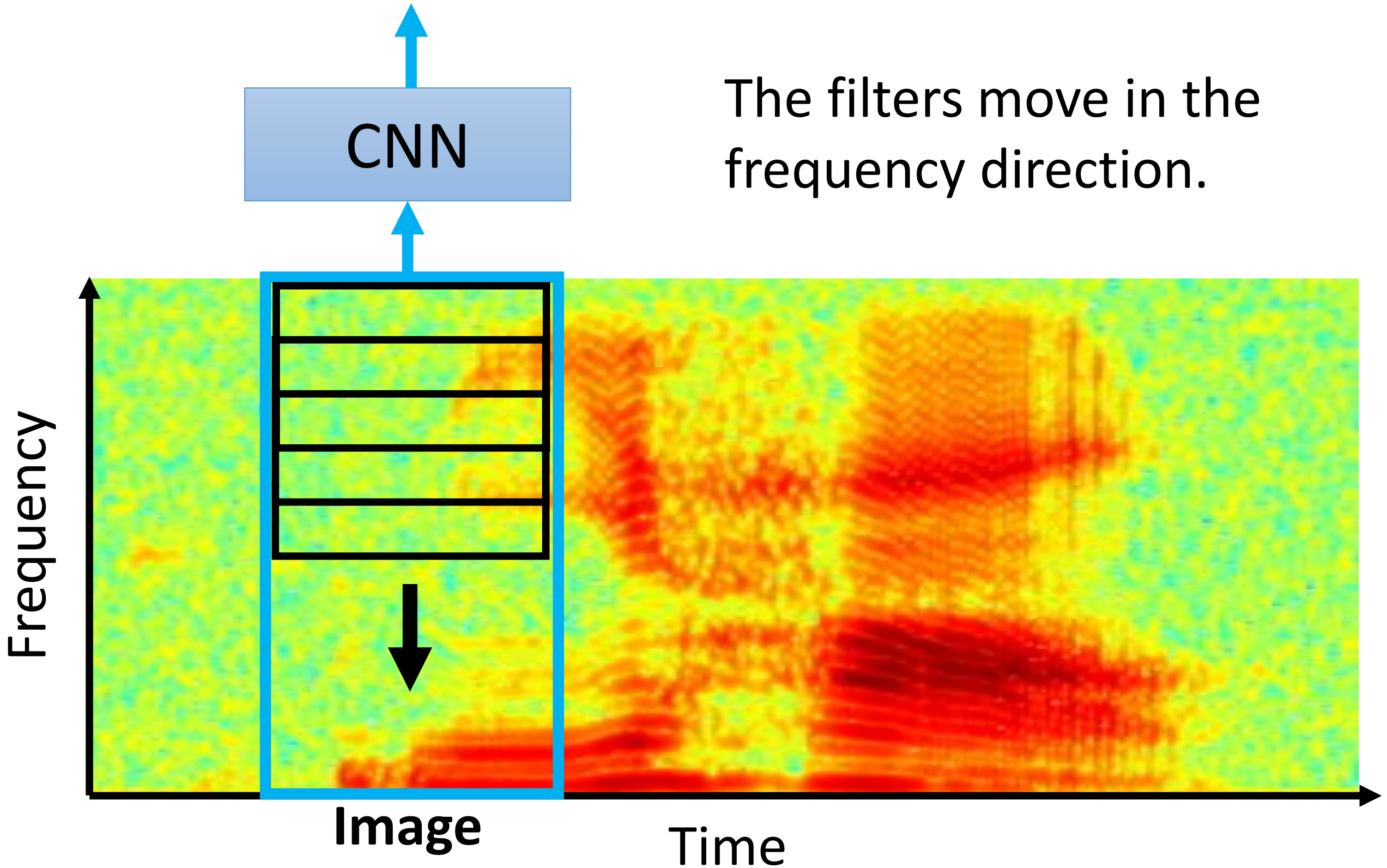
How to explain this???



Neural network architecture. The input to the policy network is a 19 × 19 × 48 image stack consisting of 48 feature planes. The first hidden layer zero pads the input into a 23 × 23 image, then convolves *k* filters of kernel size 5 × 5 with stride 1 with the input image and applies a rectifier nonlinearity. Each of the subsequent hidden layers 2 to 12 zero pads the respective previous hidden layer into a 21 × 21 image, then convolves *k* filters of kernel size 3 × 3 with stride 1, again followed by a rectifier nonlinearity. The final layer convolves 1 filter of kernel size 1 × 1 with stride 1, with a different bias for each position, and applies a softmax function. The [illegible] Extended Data Table 3 additionally show the results of training with *k* = 128, 256 and 384 filters.

Alpha Go does not use Max Pooling ……

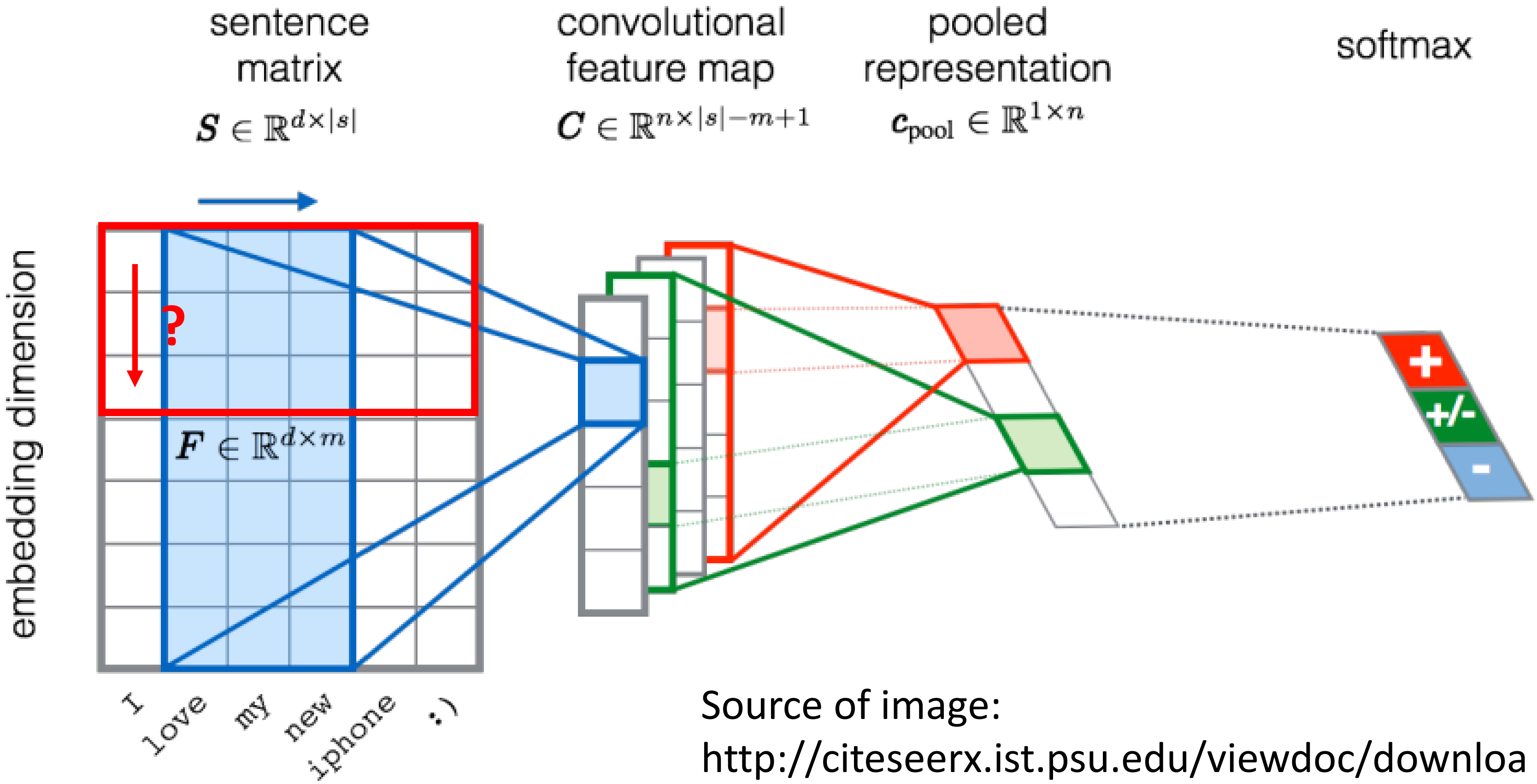
More Application: Speech



The filters move in the frequency direction.

Spectrogram

More Application: Text


sentence matrix
$S \in \mathbb{R}^{d \times |s|}$
convolutional feature map
$C \in \mathbb{R}^{n \times |s|-m+1}$
pooled representation
$c_{\text{pool}} \in \mathbb{R}^{1 \times n}$
softmax
embedding dimension
?
$F \in \mathbb{R}^{d \times m}$
I
love
my
new
iphone
:)
+
+/-
-

Source of image: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.703.6858&rep=rep1&type=pdf

Acknowledgment

- 感謝 Guobiao Mo 發現投影片上的打字錯誤