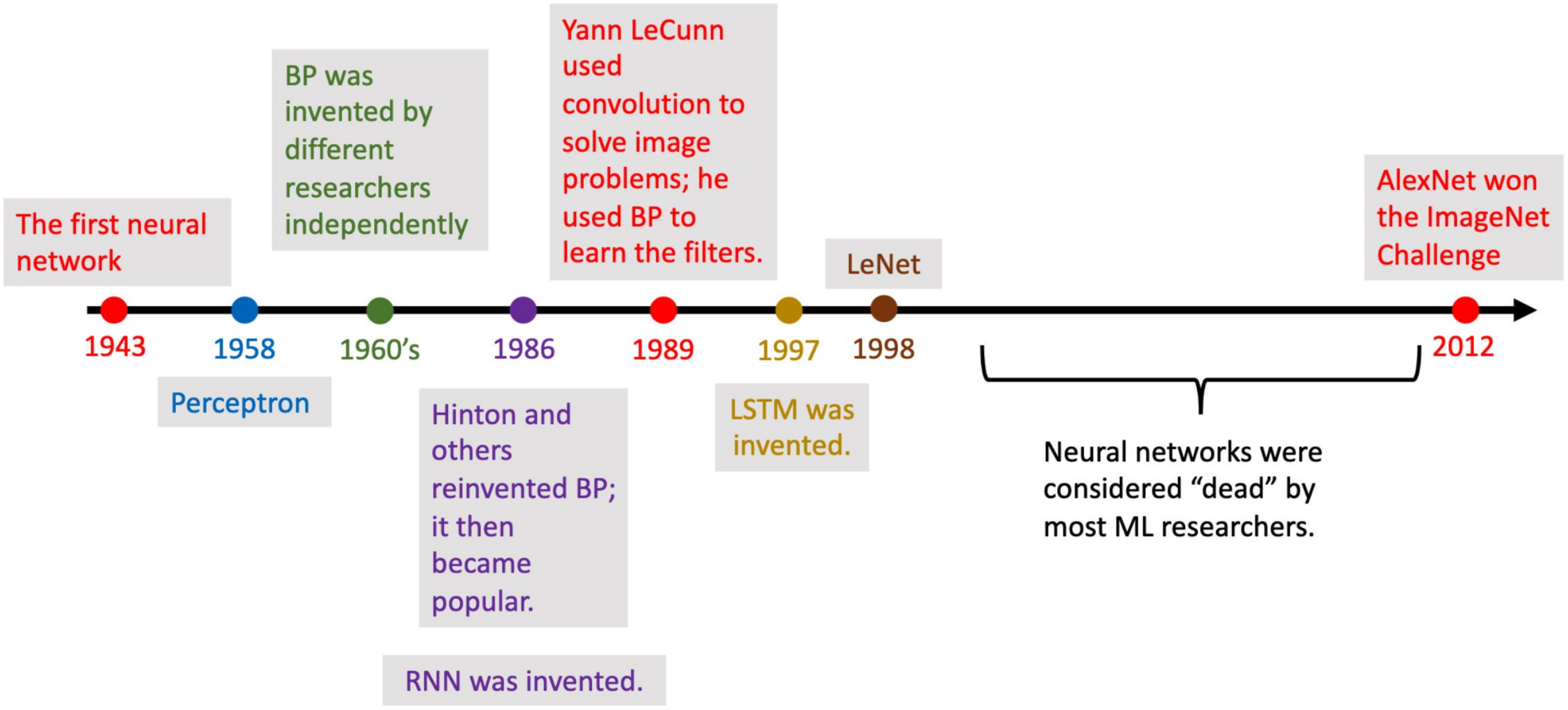
Convolutional Layer and Convolutional Neural Networks

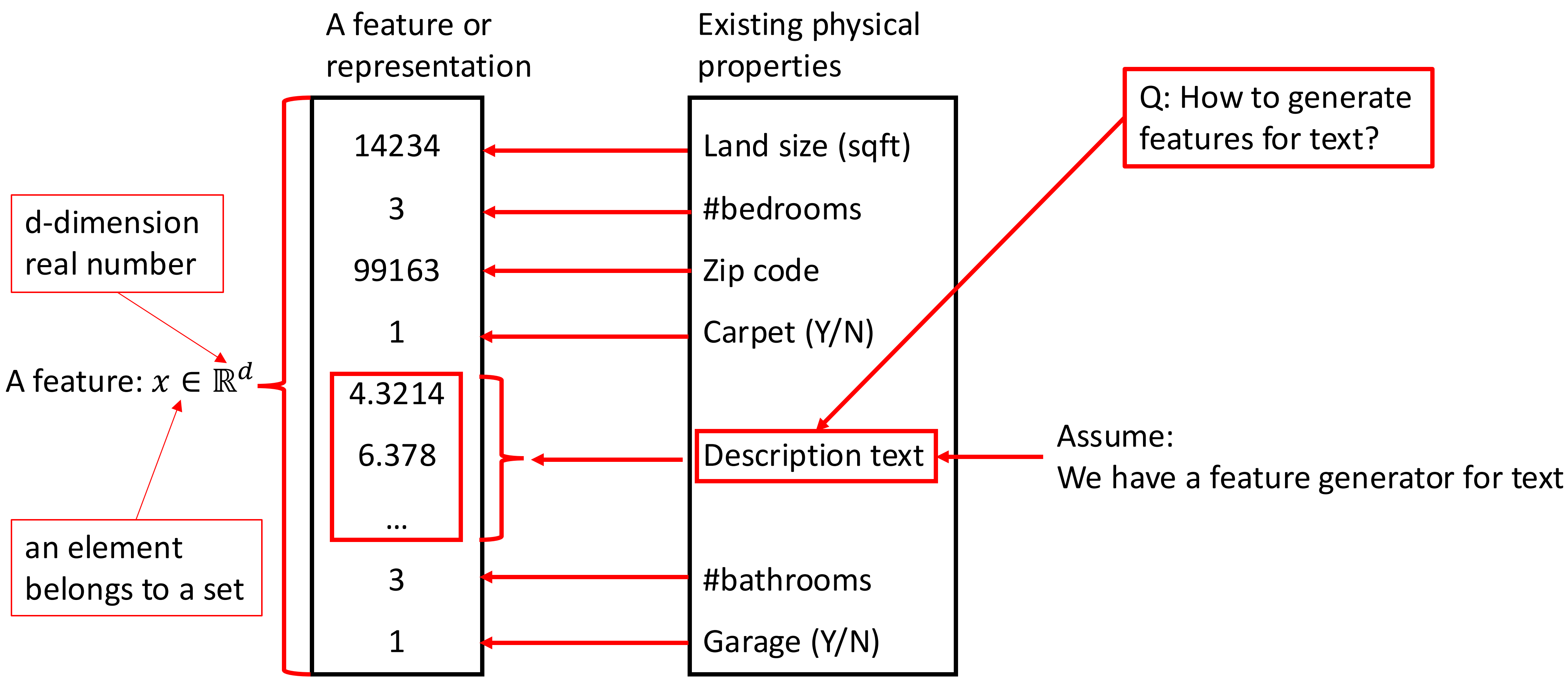
Neural Networks Design And Application

History



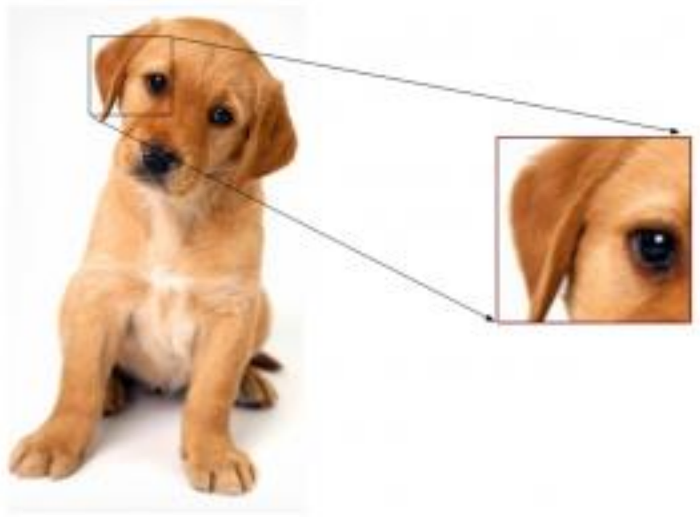
The first neural network
1943
1958
Perceptron
BP was invented by different researchers independently
1960's
1986
Hinton and others reinvented BP; it then became popular.
RNN was invented.
Yann LeCunn used convolution to solve image problems; he used BP to learn the filters.
1989
1997
LSTM was invented.
LeNet
1998
Neural networks were considered "dead" by most ML researchers.
AlexNet won the ImageNet Challenge
2012

Review: house price prediction



Review: histogram of oriented gradients

- Oriented gradients?
  - Gradients: changes in X and Y directions
  - Oriented:



121	10	78	96	125
48	152	68	125	111
145	78	85	89	65
154	214	56	200	66
214	87	45	102	45

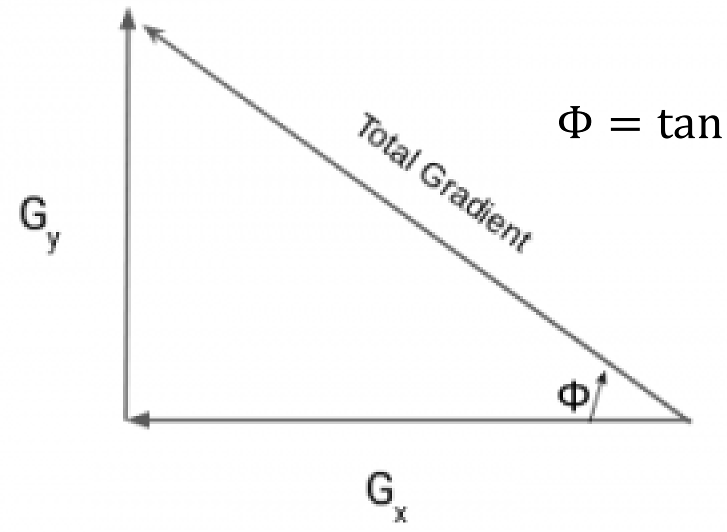
X direction G_x
Subtract the value on the left from the pixel value on the right:
89-78 = 11

Y direction G_y
Subtract the pixel value below from the pixel value above the selected pixel:
68-56=8

Credit for https://www.analyticsvidhya.com/blog/2019/09/feature-engineering-images-introduction-hog-feature-descriptor/

Review: histogram of oriented gradients

- Oriented gradients?
  - Gradients: changes in X and Y directions
  - Oriented:



$\Phi = \tan(G_y/G_x)$

$\Phi = \tan^{-1}(G_y/G_x)$

121	10	78	96	125
48	152	68	125	111
145	78	85	89	65
154	214	56	200	66
214	87	45	102	45

X direction G_x
Subtract the value on the left from the pixel value on the right:
89-78 = 11

Y direction G_y
Subtract the pixel value below from the pixel value above the selected pixel:
68-56=8

Credit for https://www.analyticsvidhya.com/blog/2019/09/feature-engineering-images-introduction-hog-feature-descriptor/

Review: histogram of oriented gradients

121	10	78	96	125
48	152	68	125	111
145	78	85	89	65
154	214	56	200	66
214	87	45	102	45

Frequency						1										
Angle	1	2	3	4 ...	35	36	37	38	39....	175	176	177	178	179	180

Credit for https://www.analyticsvidhya.com/blog/2019/09/feature-engineering-images-introduction-hog-feature-descriptor/

Review: ImageNet challenge 2012

AlexNet

Task 1

Team name	**Filename**	**Error (5 guesses)**	**Description**
SuperVision	test-preds-141-146.2009-131-137-145-146.2011-145f.	0.15315	Using extra training data from ImageNet Fall 2011 release
SuperVision	test-preds-131-137-145-135-145f.txt	0.16422	Using only supplied training data
ISI	pred_FVs_wLACs_weighted.txt	0.26172	Weighted sum of scores from each classifier with SIFT+FV, LBP+FV, GIST+FV, and CSIFT+FV, respectively.
ISI	pred_FVs_weighted.txt	0.26602	Weighted sum of scores from classifiers using each FV.
ISI	pred_FVs_summed.txt	0.26646	Naive sum of scores from classifiers using each FV.

Review: LeNet-5 in 1999

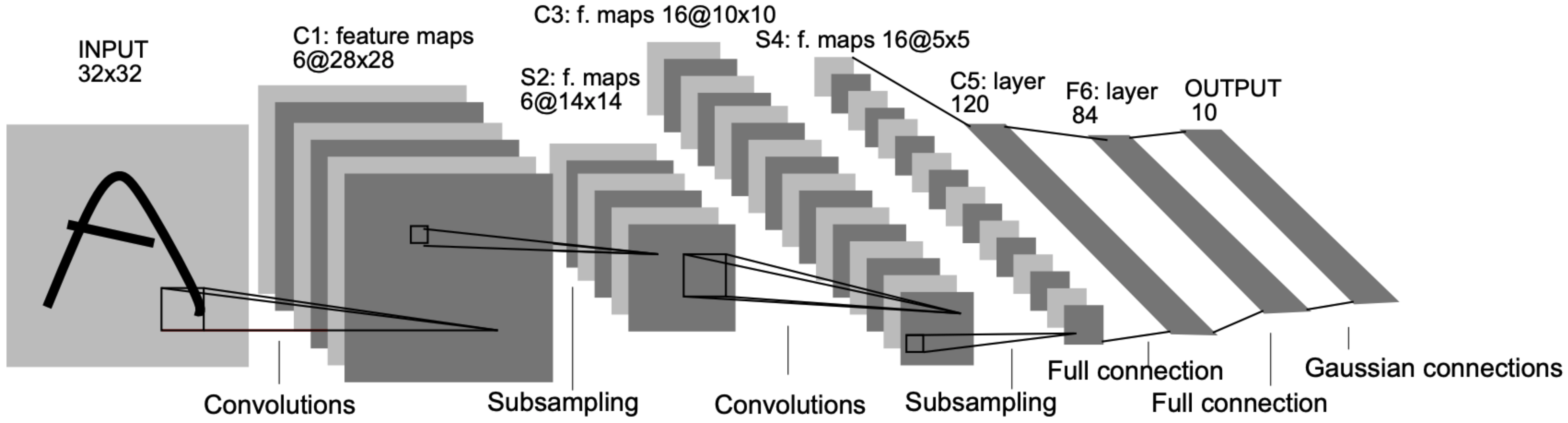
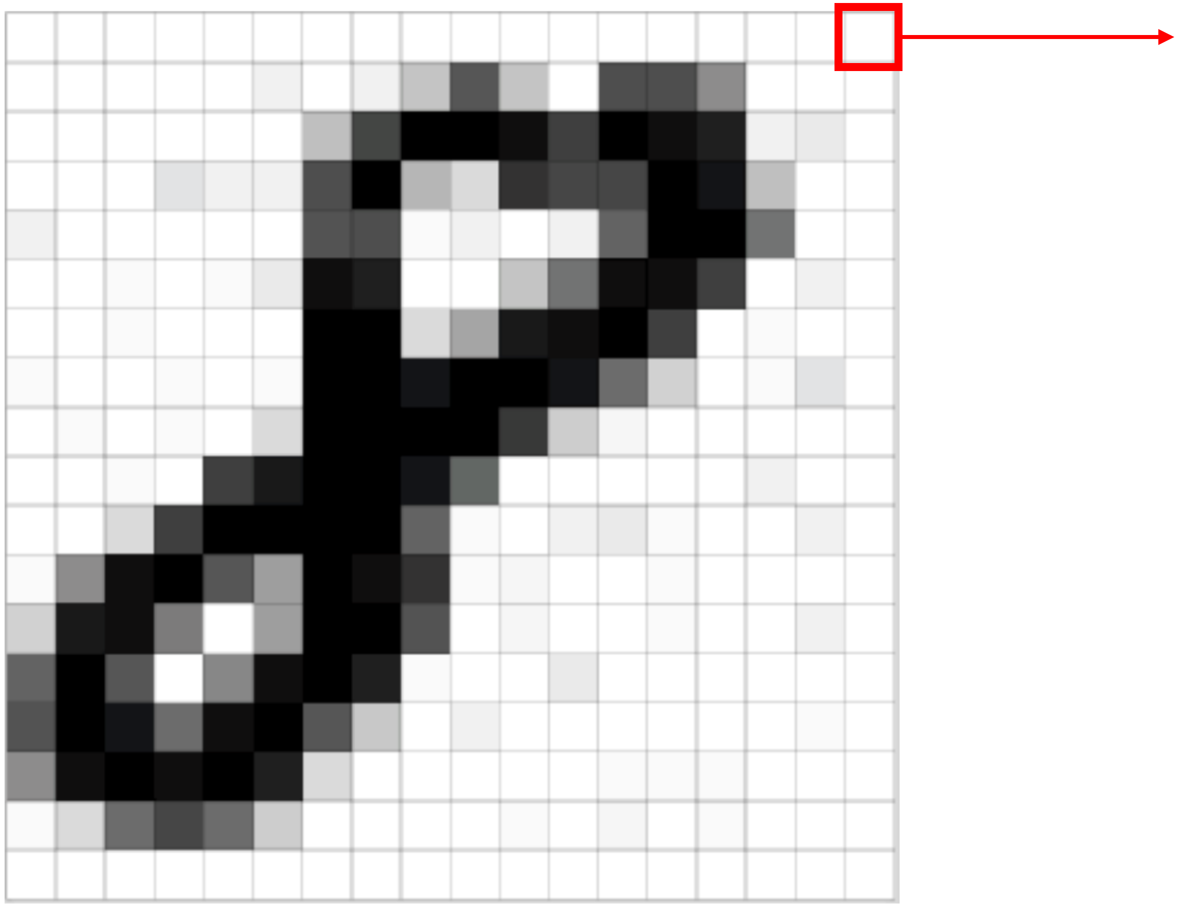


Fig. 1. Architecture of LeNet-5, a Convolutional Neural Network, here for digits recognition. Each plane is a feature map, i.e. a set of units whose weights are constrained to be identical.

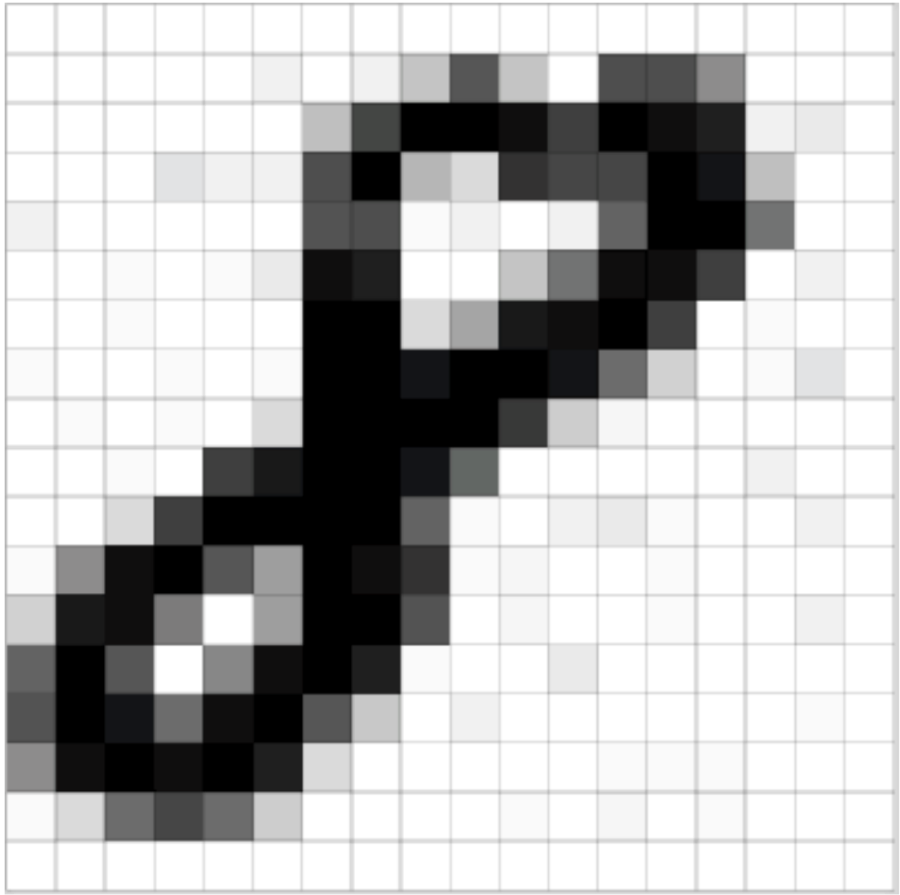
LeCun, Yann, Patrick Haffner, Léon Bottou, and Yoshua Bengio. "Object recognition with gradient-based learning." In *Shape, contour and grouping in computer vision*, pp. 319-345. Springer, Berlin, Heidelberg, 1999.

What is convolutional neural network?



A grayscale image

What is convolutional neural network?



A grayscale image

An image → a matrix

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

Convolution for images (matrices)

$$
\begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 \end{bmatrix} * \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}
$$

Involving two matrices

Convolution for images (matrices)

Given image

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

*

Filter

1	0	1
0	1	0
1	0	1

Involving two matrices

Convolution for images (matrices)

Larger

Given image

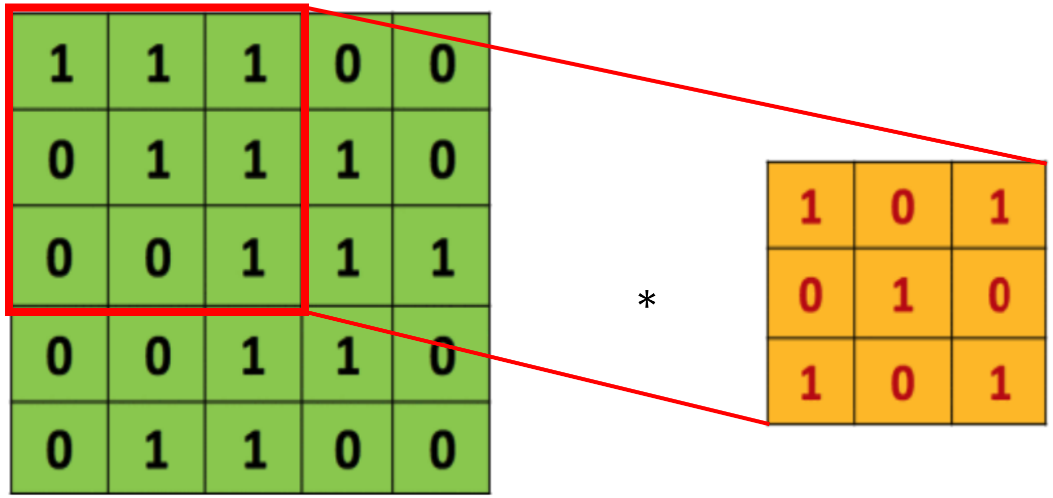
$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 \end{bmatrix} \quad * \quad \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}$$

Smaller

Filter

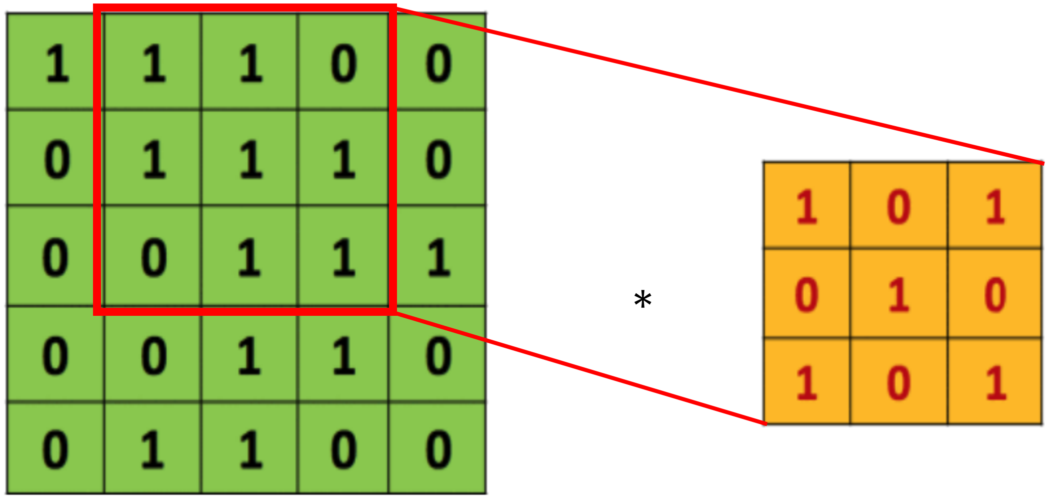
Involving two matrices

Convolution for images (matrices)



Finding pairs

Convolution for images (matrices)



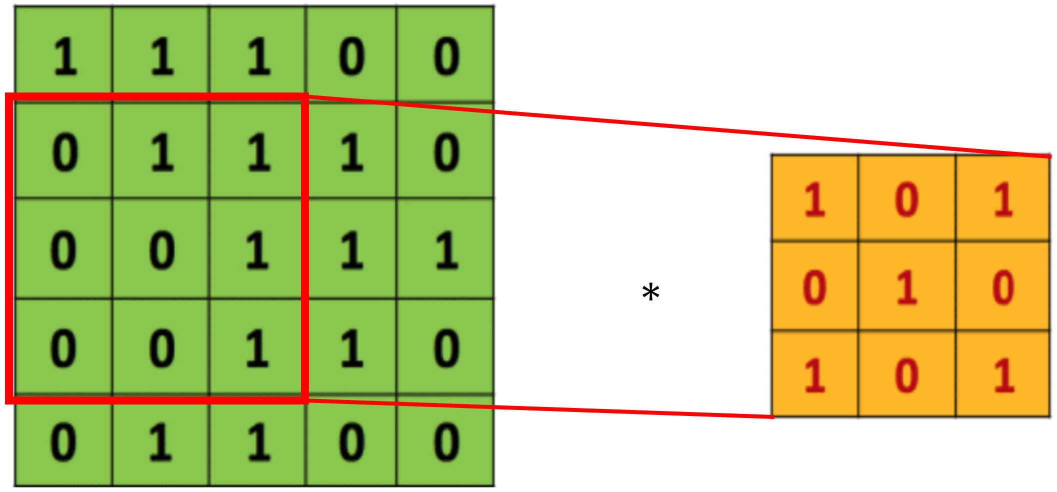
Finding pairs

Convolution for images (matrices)

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 \end{bmatrix} * \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}$$

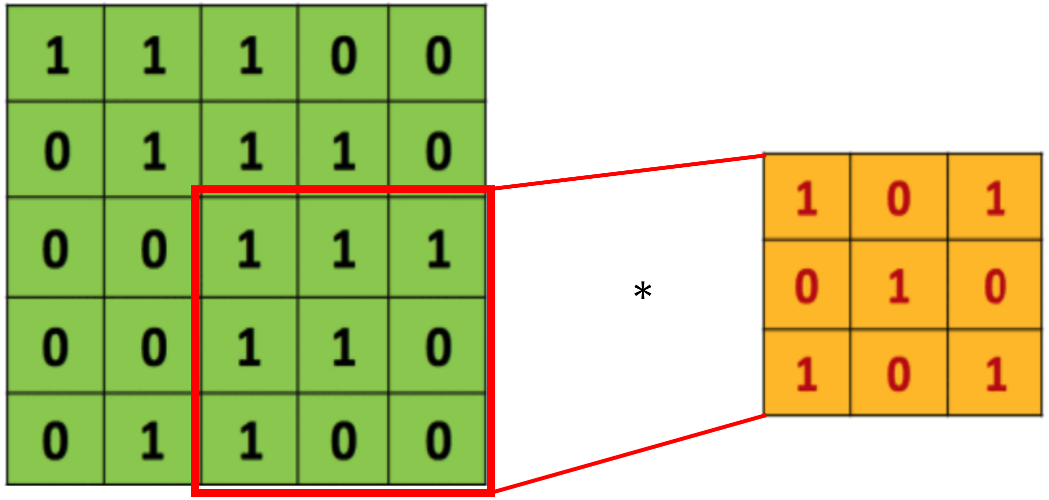
Finding pairs

Convolution for images (matrices)



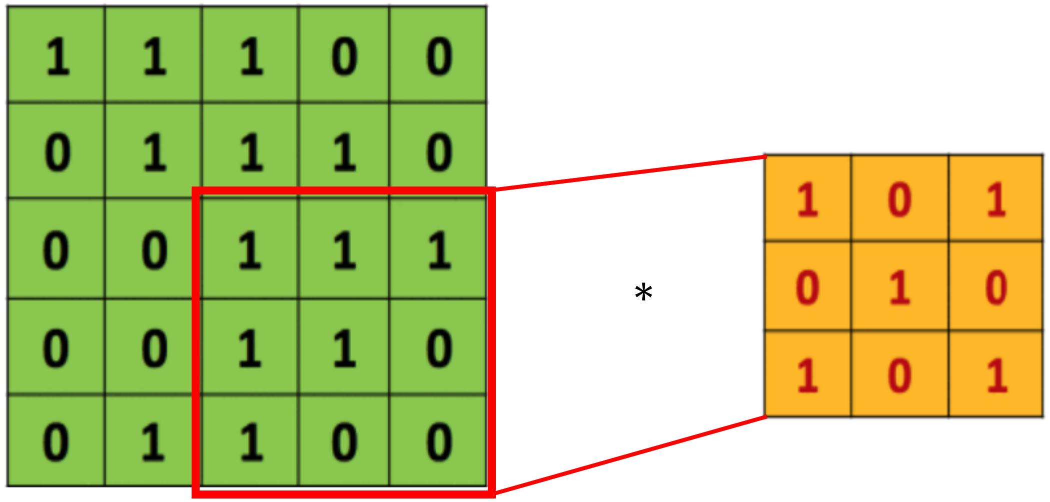
Finding pairs

Convolution for images (matrices)



Finding pairs

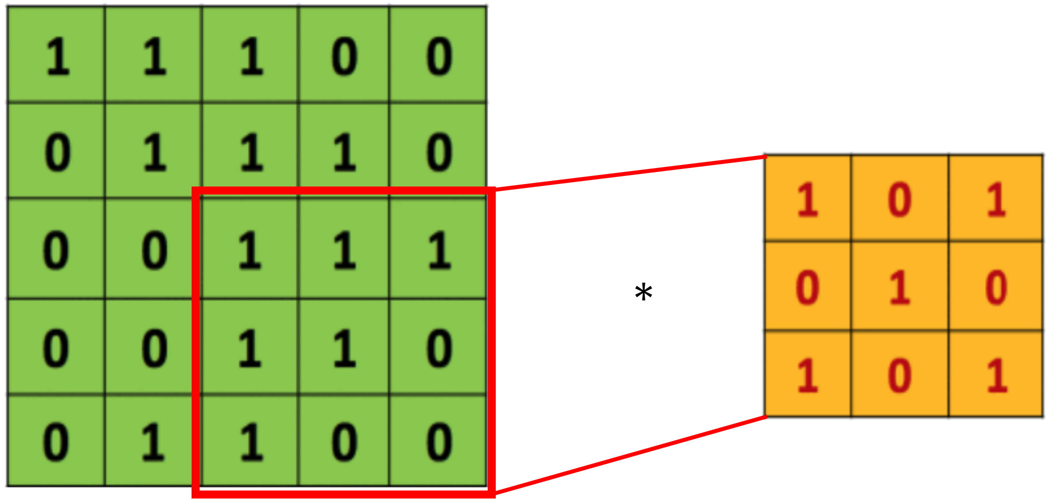
Convolution for images (matrices)



Finding pairs

Q: how many pairs we have?

Convolution for images (matrices)

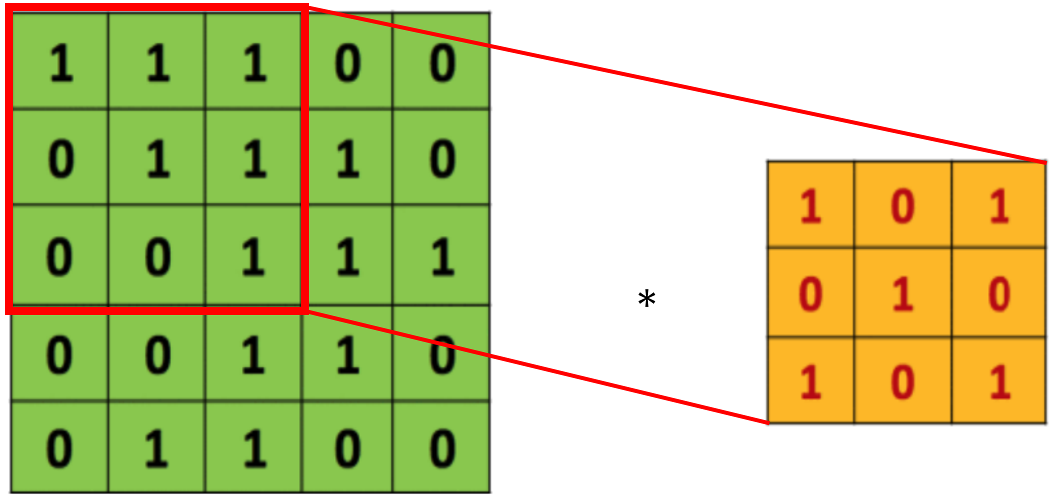


Finding pairs

Q: how many pairs we have?

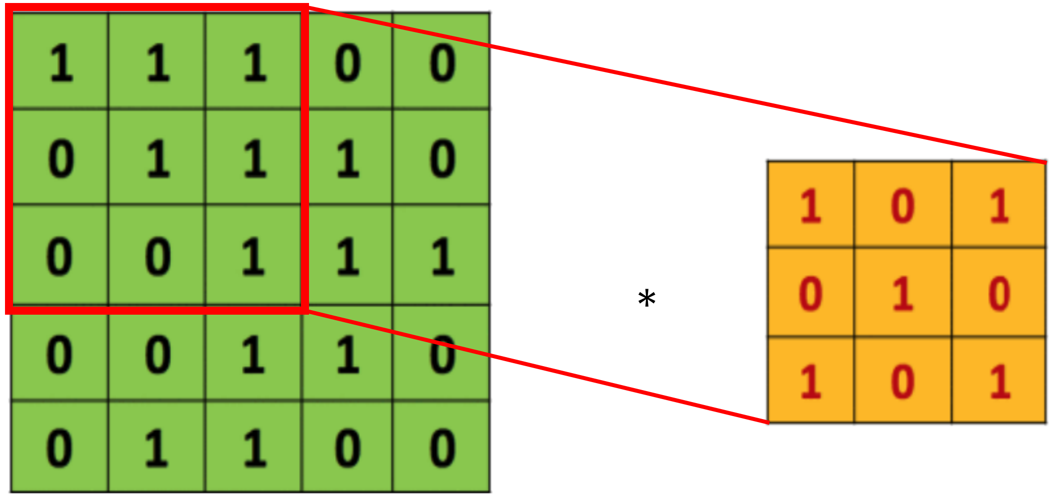
(5-3+1) * (5-3+1)=9

Convolution for images (matrices)



Inner product of each pair

Convolution for images (matrices)



Inner product of each pair

Elementwise multiplication + summation

Convolution for images (matrices)

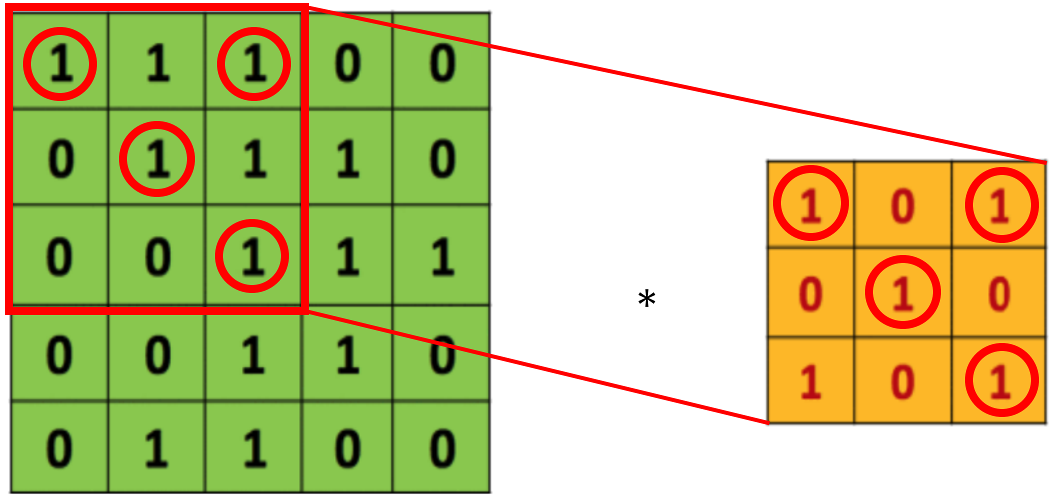
$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 \end{bmatrix} * \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}$$

Inner product of each pair

Elementwise multiplication + summation

Q: what is your result for the first pair?

Convolution for images (matrices)



$\boxed{\text{Inner product}}$ of each pair

$\boxed{\text{Elementwise multiplication + summation}}$

Convolution for images (matrices)

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 \end{bmatrix} * \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} = 4$$

$\boxed{\text{Inner product}}$ of each pair

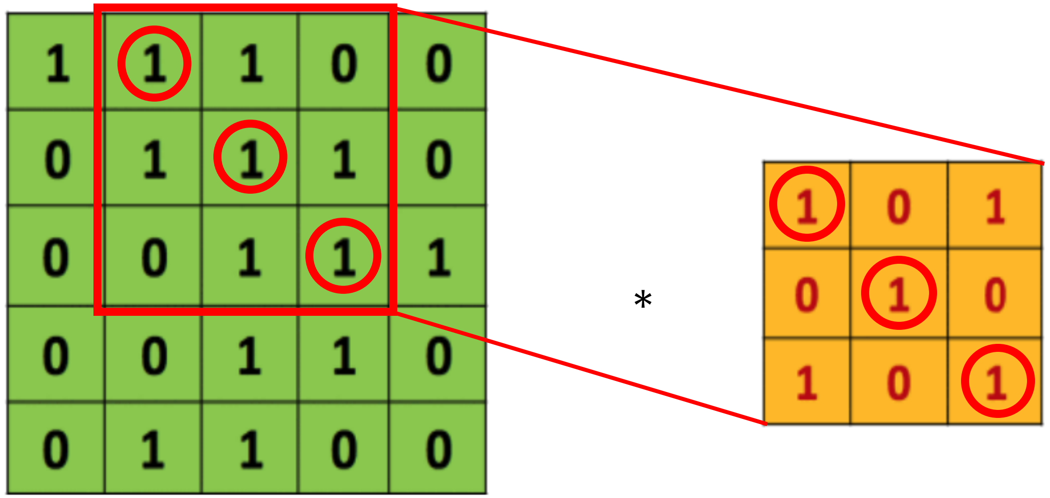
$\boxed{\text{Elementwise multiplication + summation}}$

Convolution for images (matrices)

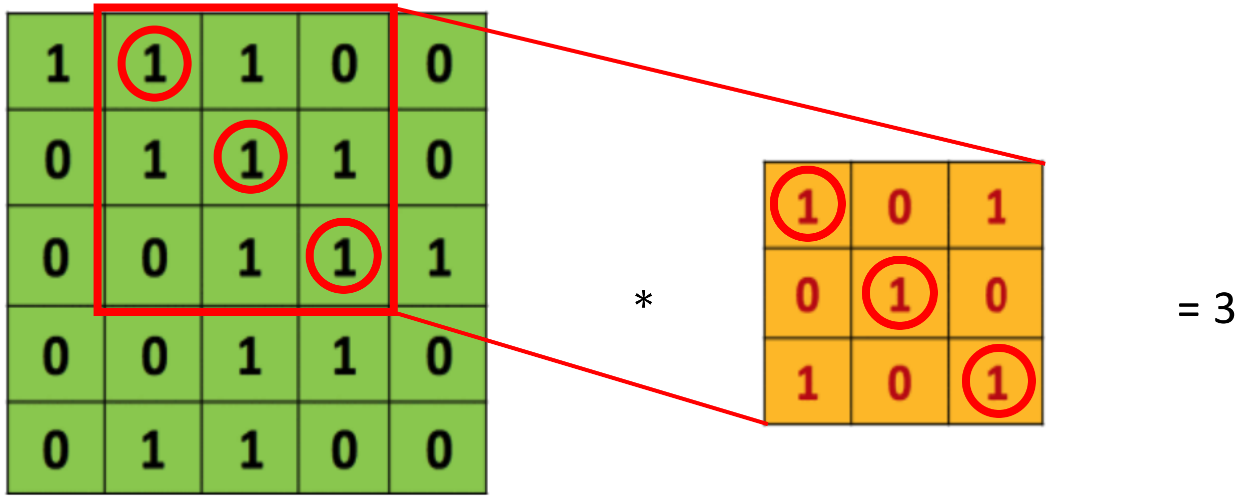
$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 \end{bmatrix} * \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}$$

Q: the second pair?

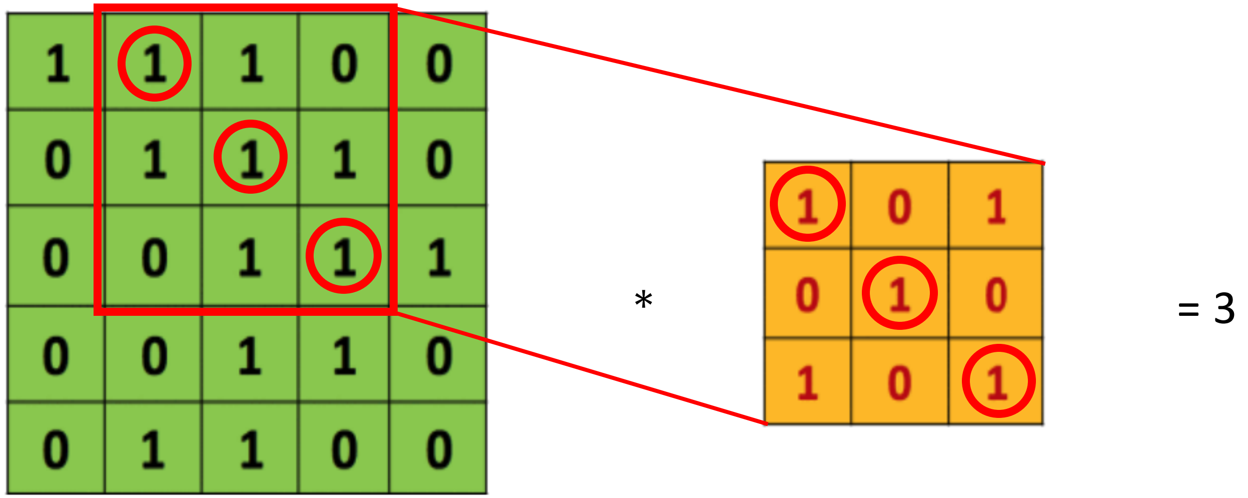
Convolution for images (matrices)



Convolution for images (matrices)



Convolution for images (matrices)



We can repeat for each pair

Convolution for images (matrices)

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 \end{bmatrix} * \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} \rightarrow \begin{bmatrix} 4 & 3 & 4 \\ 2 & 4 & 3 \\ 2 & 3 & 4 \end{bmatrix}$$

Convolution for images (matrices)

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 \end{bmatrix} * \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} \rightarrow \begin{bmatrix} 4 & 3 & 4 \\ 2 & 4 & 3 \\ 2 & 3 & 4 \end{bmatrix}$$

Place each element
according to their positions

Convolution for images (matrices)

Row: 1
Column: 1

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 \end{bmatrix} * \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} \rightarrow \begin{bmatrix} \boxed{4} & 3 & 4 \\ 2 & 4 & 3 \\ 2 & 3 & 4 \end{bmatrix}$$

Place each element according to their positions

Convolution for images (matrices)

Row: 1
Column: 2

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 \end{bmatrix} * \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} \rightarrow \begin{bmatrix} 4 & \boxed{3} & 4 \\ 2 & 4 & 3 \\ 2 & 3 & 4 \end{bmatrix}$$

Place each element according to their positions

Convolution for images (matrices)

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 \end{bmatrix} * \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} \rightarrow \begin{bmatrix} 4 & 3 & 4 \\ 2 & 4 & 3 \\ 2 & 3 & \boxed{4} \end{bmatrix}$$

Row: 3
Column: 3

Place each element according to their positions

Convolution for images (matrices)

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 \end{bmatrix} * \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} \rightarrow \begin{bmatrix} 4 & 3 & 4 \\ 2 & 4 & 3 \\ 2 & 3 & 4 \end{bmatrix}$$

n=5 m=3 Q: dimension?

Convolution for images (matrices)

$$
\begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 \end{bmatrix}
*
\begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}
\rightarrow
\begin{bmatrix} 4 & 3 & 4 \\ 2 & 4 & 3 \\ 2 & 3 & 4 \end{bmatrix}
$$

n=5 m=3 n-m+1=3

Convolution for images (matrices)

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 \end{bmatrix} * \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} \rightarrow \begin{bmatrix} 4 & 3 & 4 \\ 2 & 4 & 3 \\ 2 & 3 & 4 \end{bmatrix}$$

n=5	m=3	n-m+1=3
One matrix	One matrix	One matrix

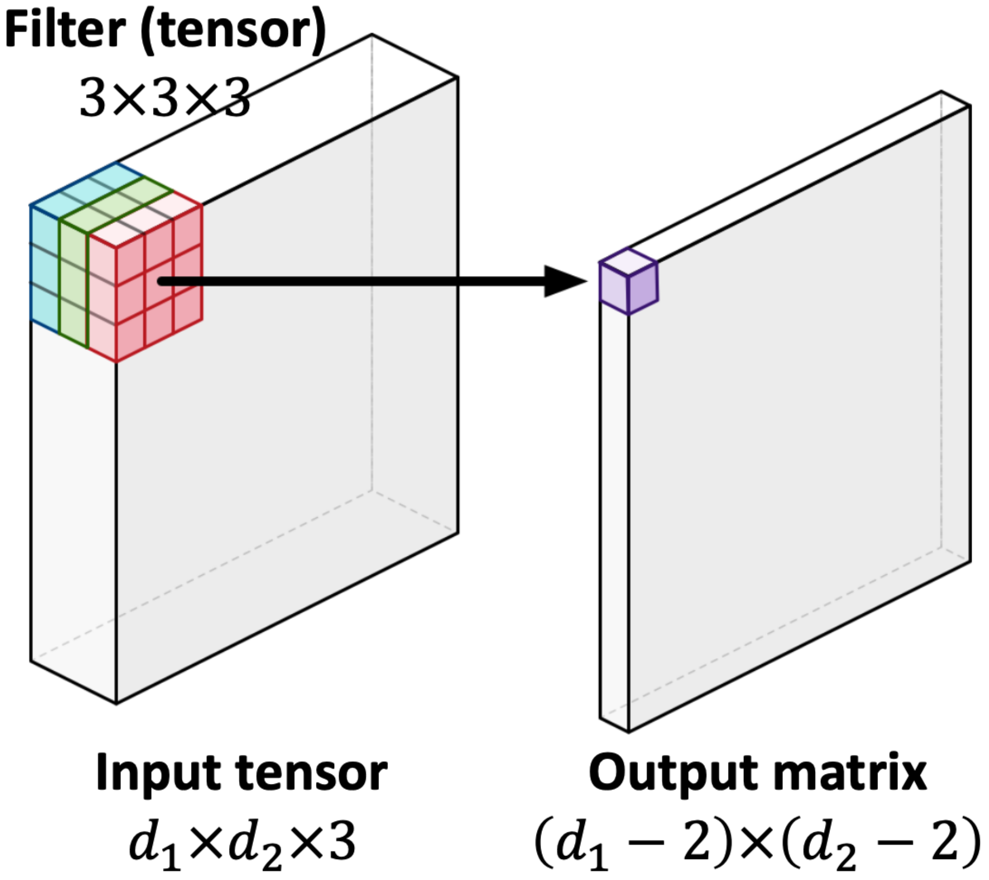
Convolution for images (matrices)

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 \end{bmatrix} * \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} \rightarrow \begin{bmatrix} 4 & 3 & 4 \\ 2 & 4 & 3 \\ 2 & 3 & 4 \end{bmatrix}$$

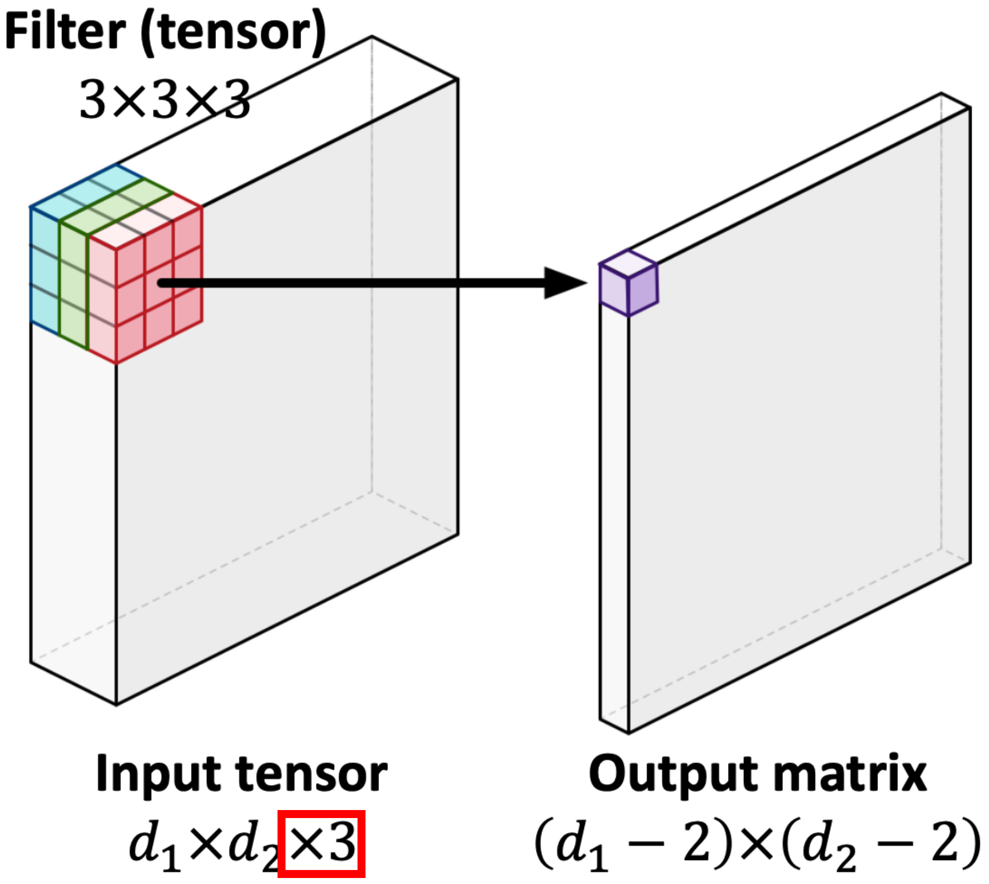
n=5	m=3	n-m+1=3
One matrix	One matrix	One matrix

One input matrix * one filter → one feature matrix

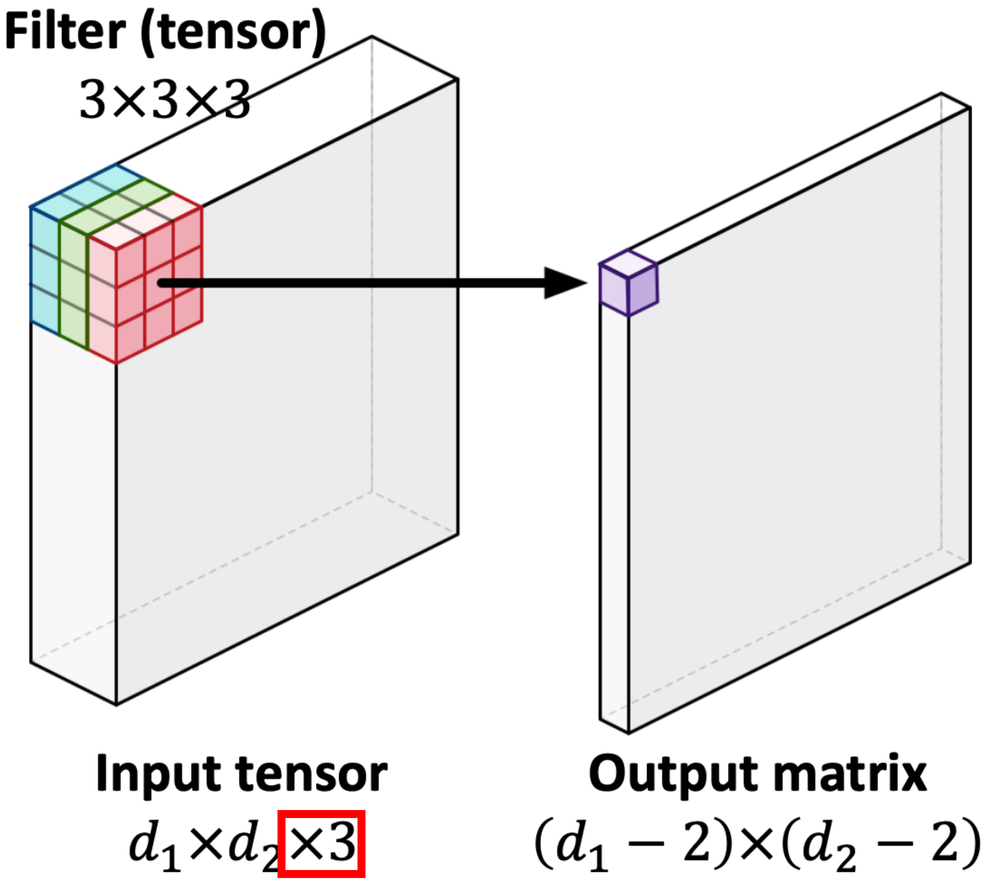
Convolution for images (tensors)



Convolution for images (tensors)

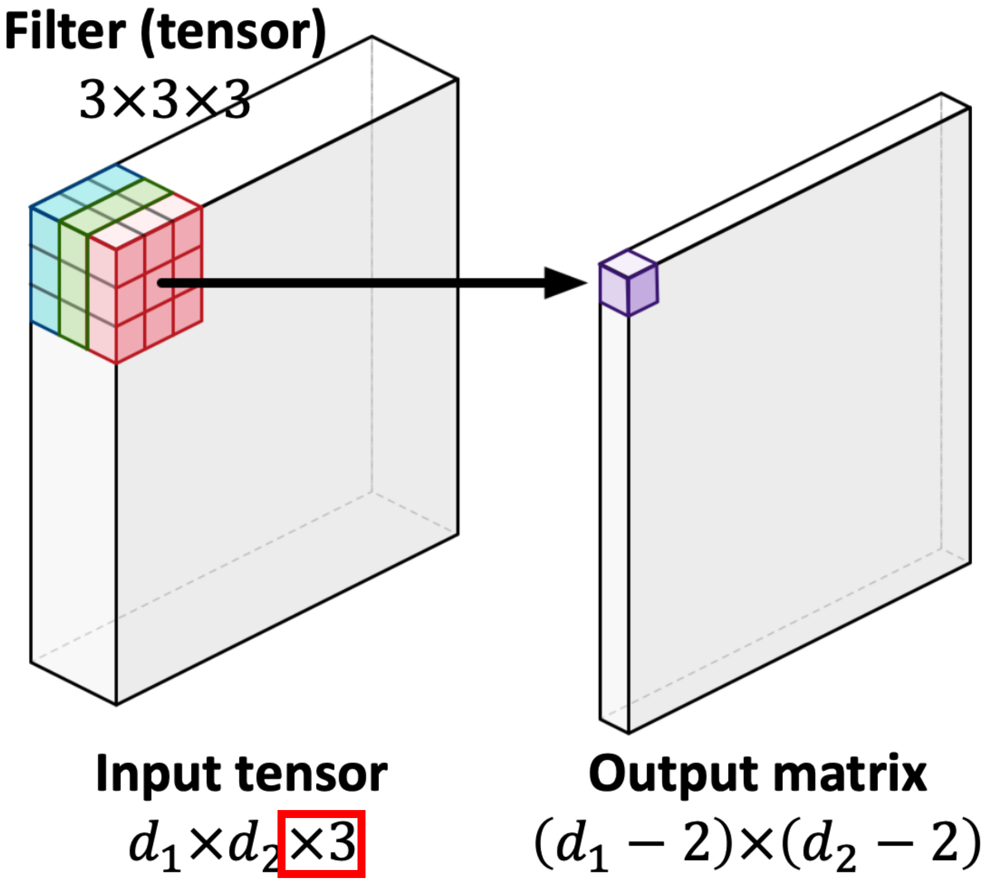


Convolution for images (tensors)



Q: why we care about tensors?

Convolution for images (tensors)



Q: why we care about tensors?

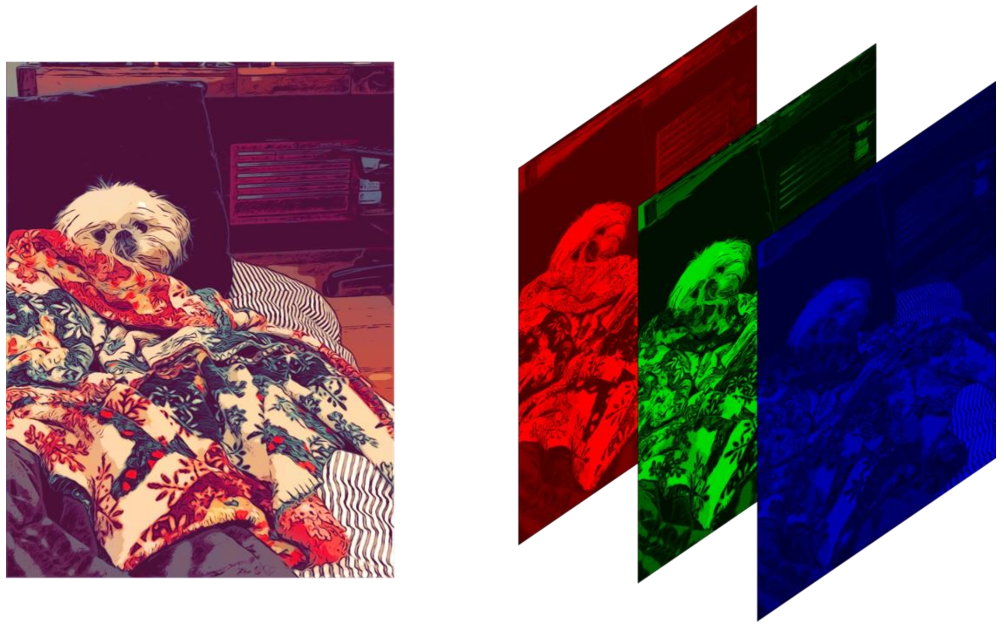
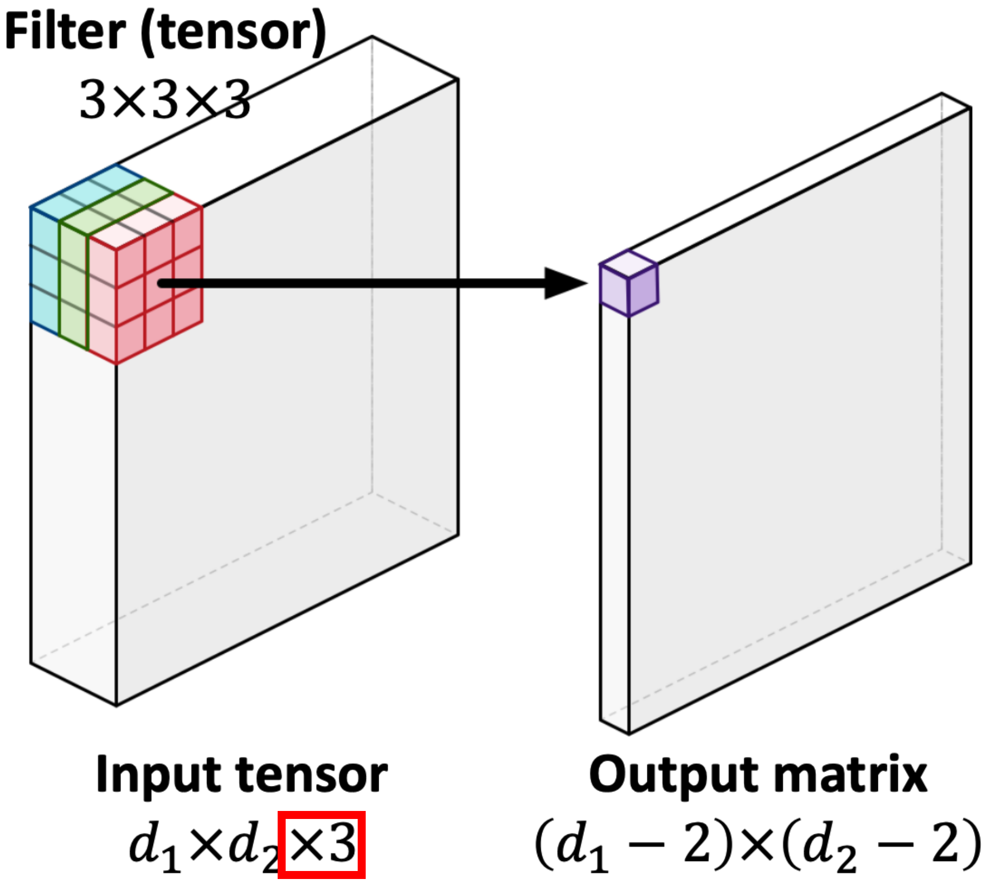
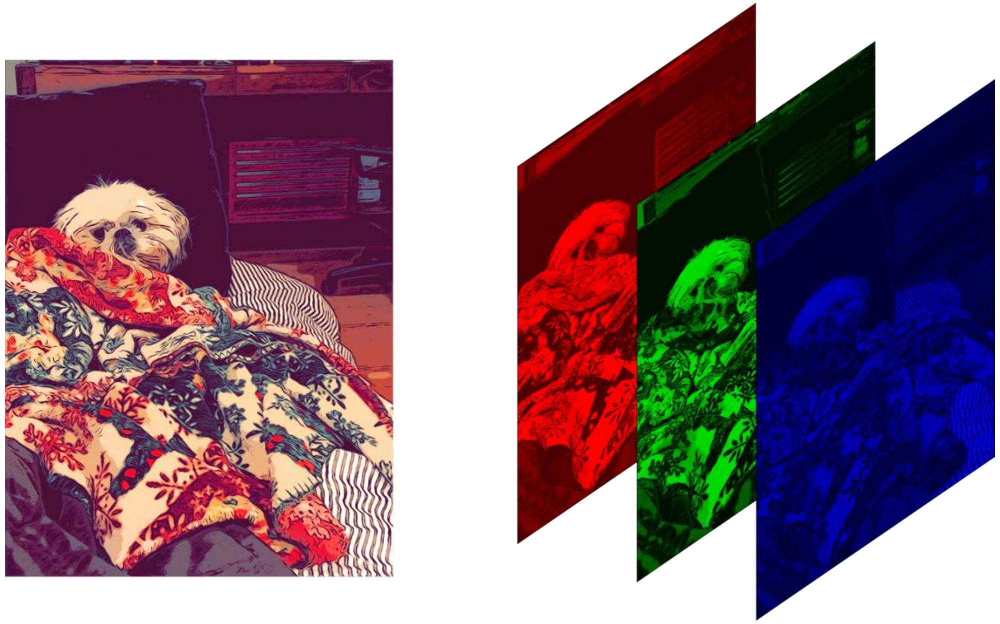


Image from https://e2eml.school/convert_rgb_to_grayscale.html

Convolution for images (tensors)



Q: why we care about tensors?

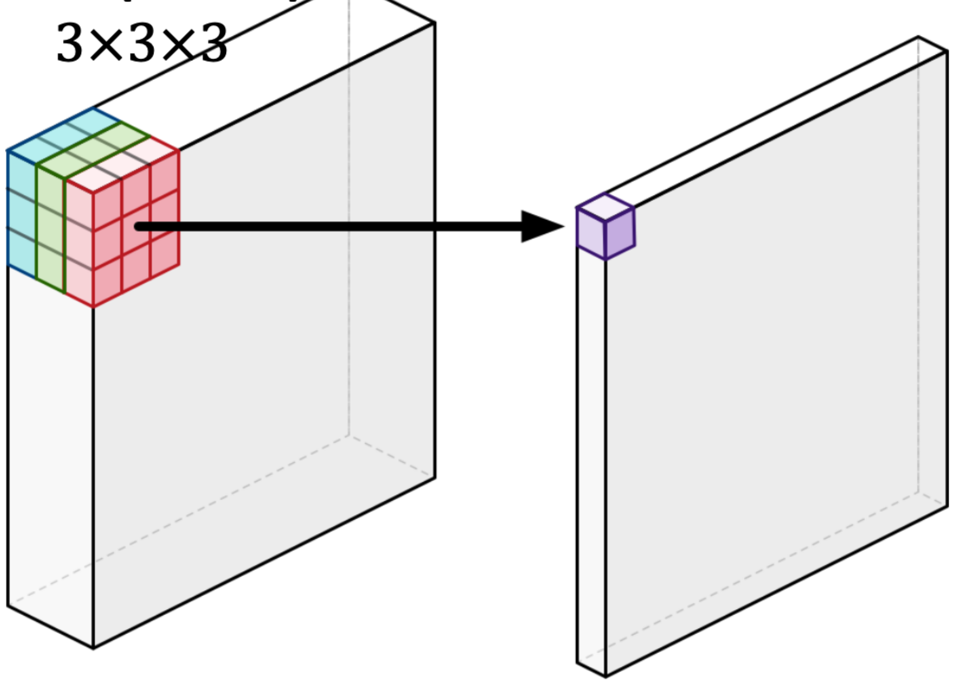


Reason 1:
RGB channels are more common

Image from https://e2eml.school/convert_rgb_to_grayscale.html

Convolution for images (tensors)

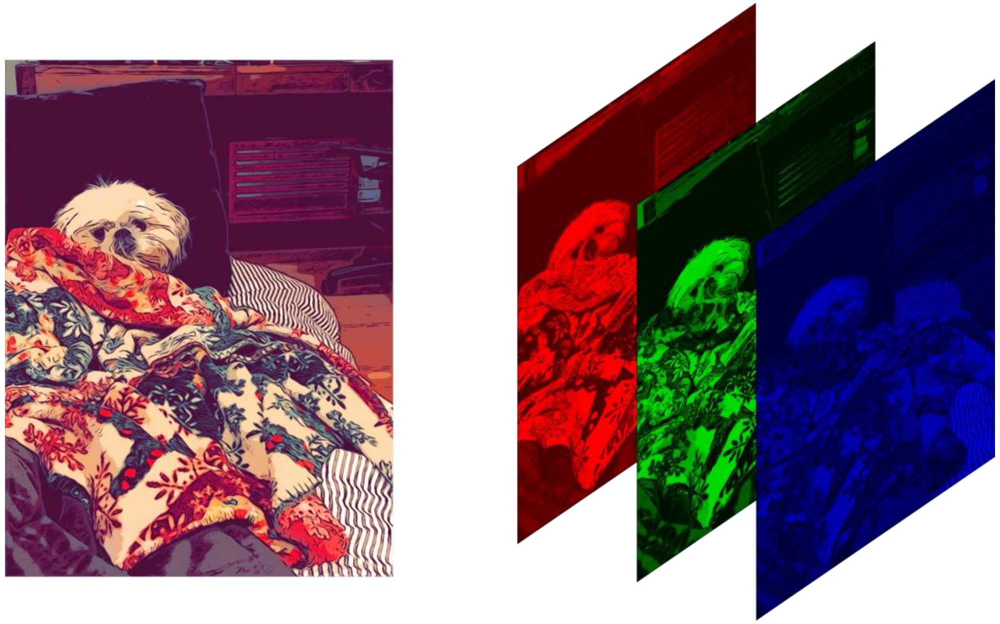
Filter (tensor)
$3\times3\times3$



Input tensor
$d_1\times d_2\boxed{\times 3}$

Output matrix
$(d_1-2)\times(d_2-2)$

Q: why we care about tensors?



Reason 1:
RGB channels are more common

Each channel → a matrix

Image from https://e2eml.school/convert_rgb_to_grayscale.html

LeNet-5 in 1999

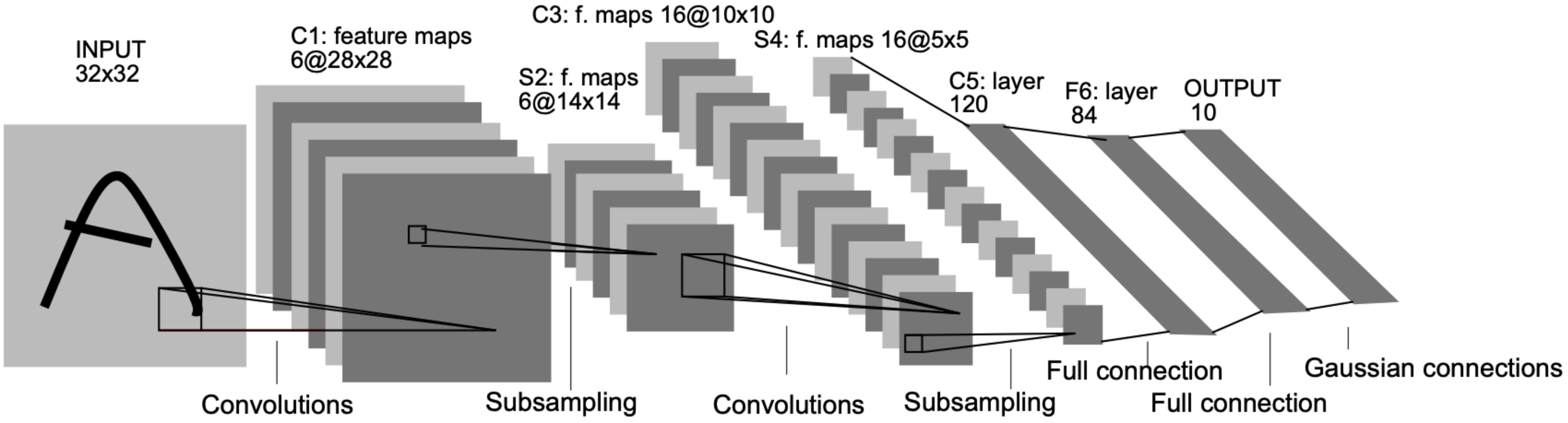


Fig. 1. Architecture of LeNet-5, a Convolutional Neural Network, here for digits recognition. Each plane is a feature map, i.e. a set of units whose weights are constrained to be identical.

LeCun, Yann, Patrick Haffner, Léon Bottou, and Yoshua Bengio. "Object recognition with gradient-based learning." In *Shape, contour and grouping in computer vision*, pp. 319-345. Springer, Berlin, Heidelberg, 1999.

LeNet-5 in 1999

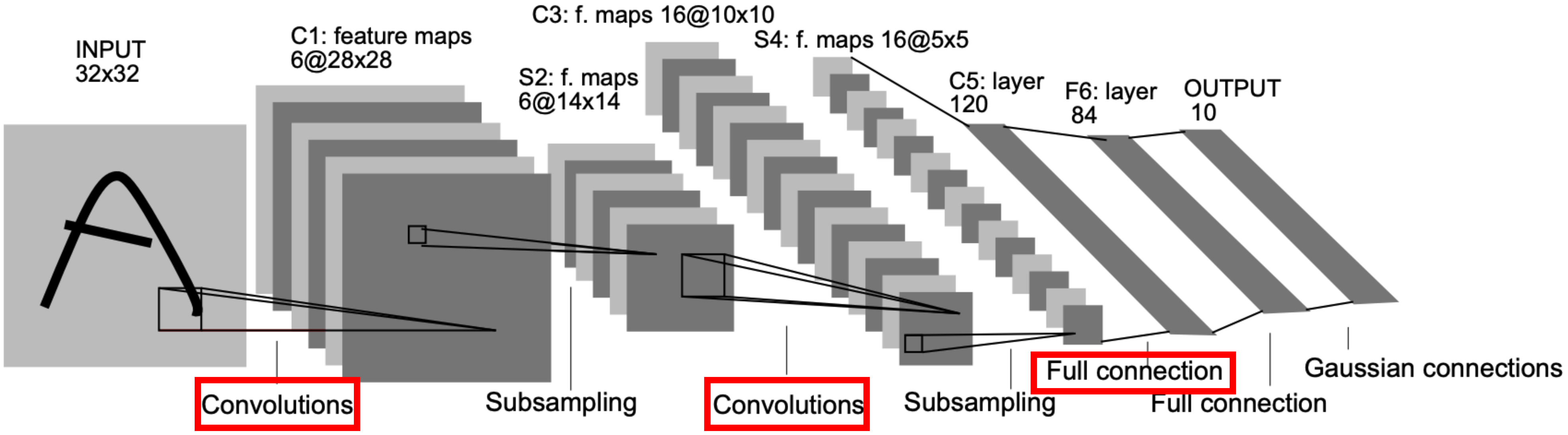


Fig. 1. Architecture of LeNet-5, a Convolutional Neural Network, here for digits recognition. Each plane is a feature map, i.e. a set of units whose weights are constrained to be identical.

LeCun, Yann, Patrick Haffner, Léon Bottou, and Yoshua Bengio. "Object recognition with gradient-based learning." In *Shape, contour and grouping in computer vision*, pp. 319-345. Springer, Berlin, Heidelberg, 1999.

LeNet-5 in 1999

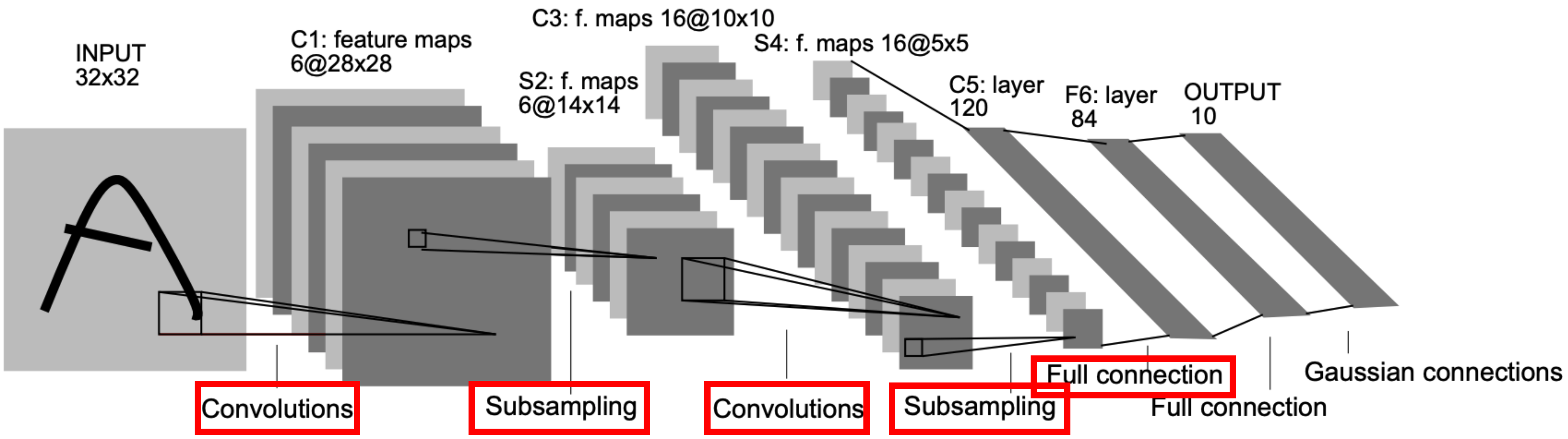
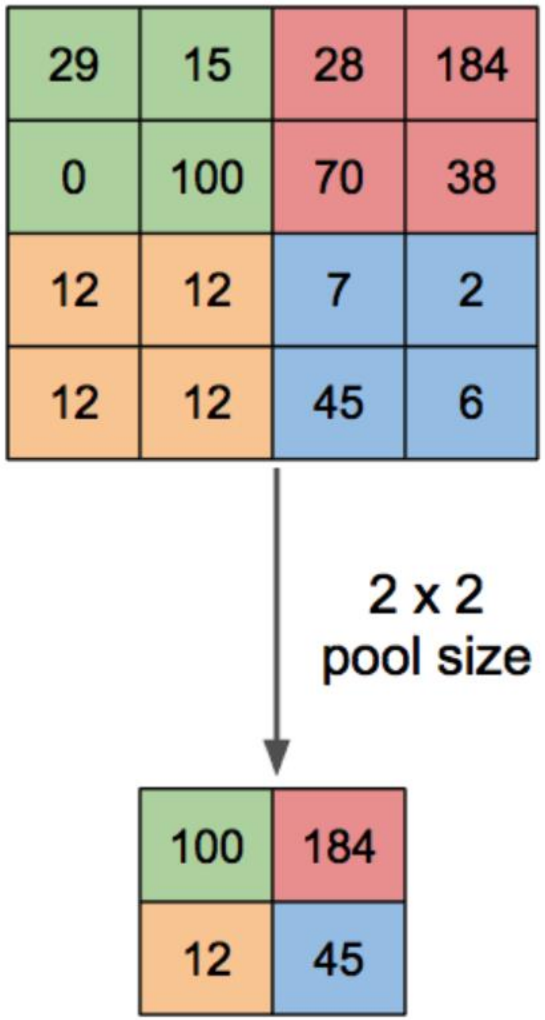


Fig. 1. Architecture of LeNet-5, a Convolutional Neural Network, here for digits recognition. Each plane is a feature map, i.e. a set of units whose weights are constrained to be identical.

LeCun, Yann, Patrick Haffner, Léon Bottou, and Yoshua Bengio. "Object recognition with gradient-based learning." In *Shape, contour and grouping in computer vision*, pp. 319-345. Springer, Berlin, Heidelberg, 1999.

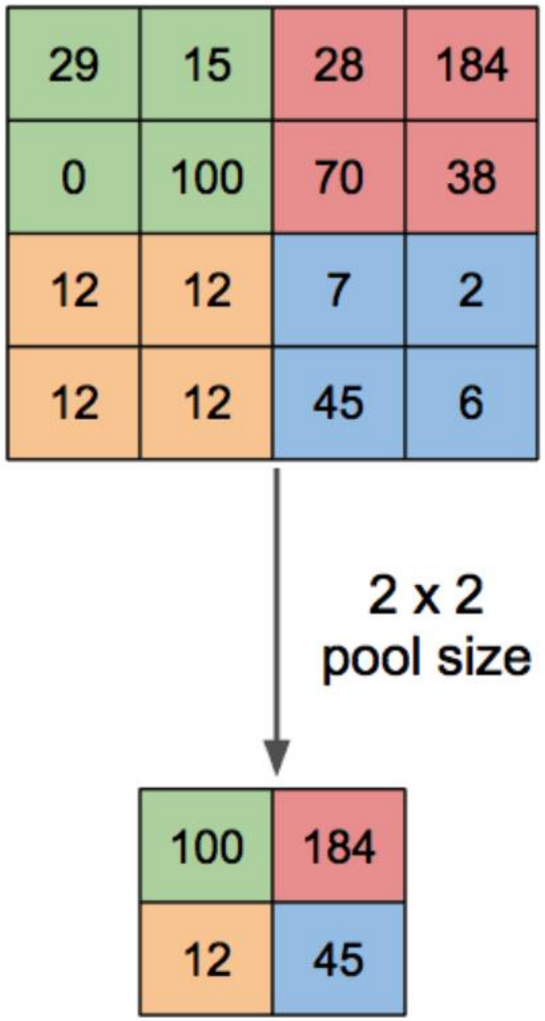
Subsampling operations

- Max pooling



Subsampling operations

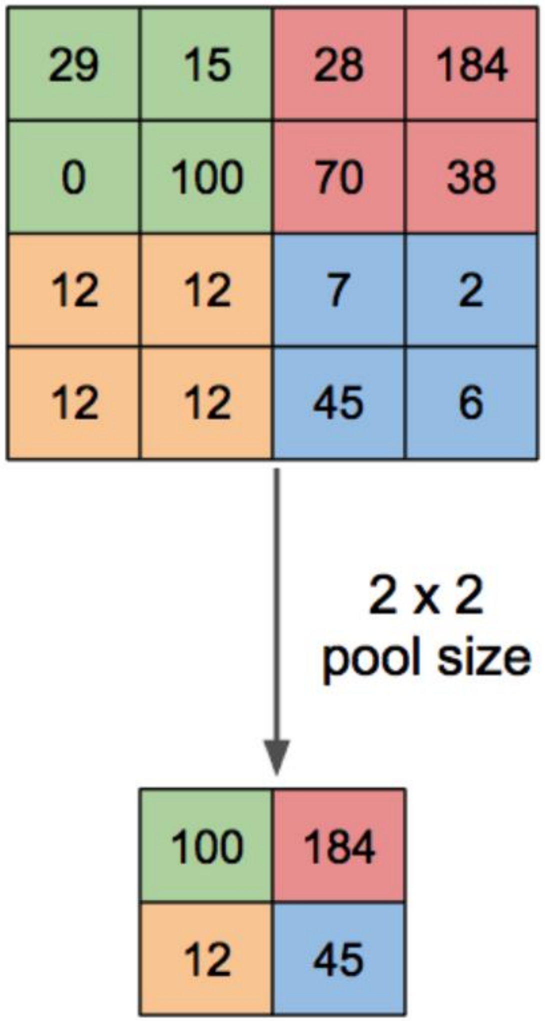
- Max pooling

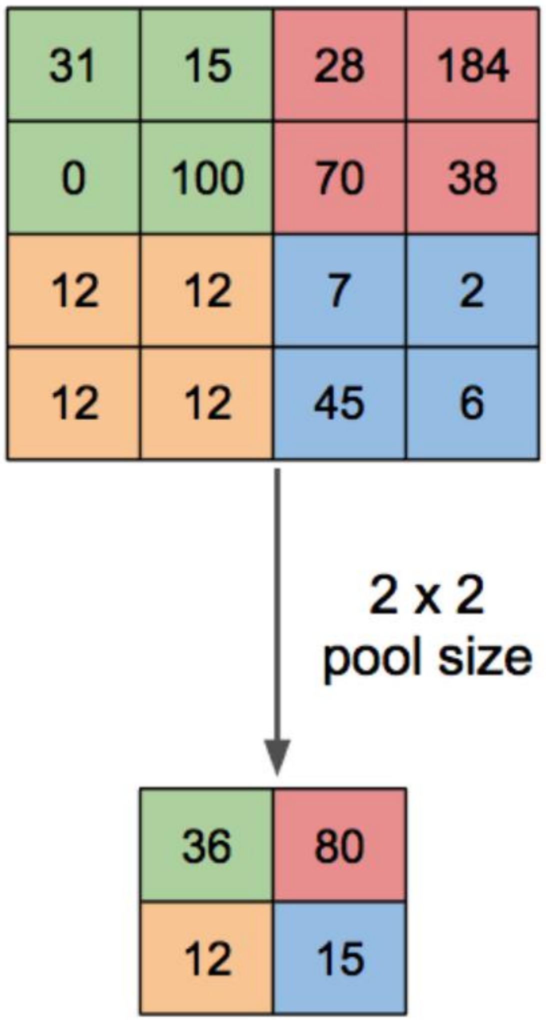


Q: what does max Pooling really do?

Subsampling operations

- Max pooling
- Average pooling

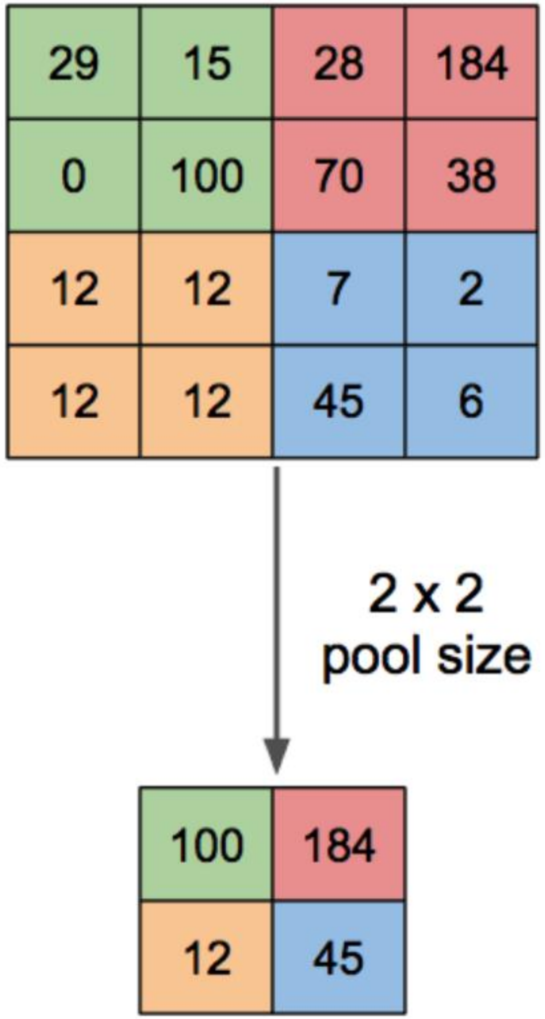


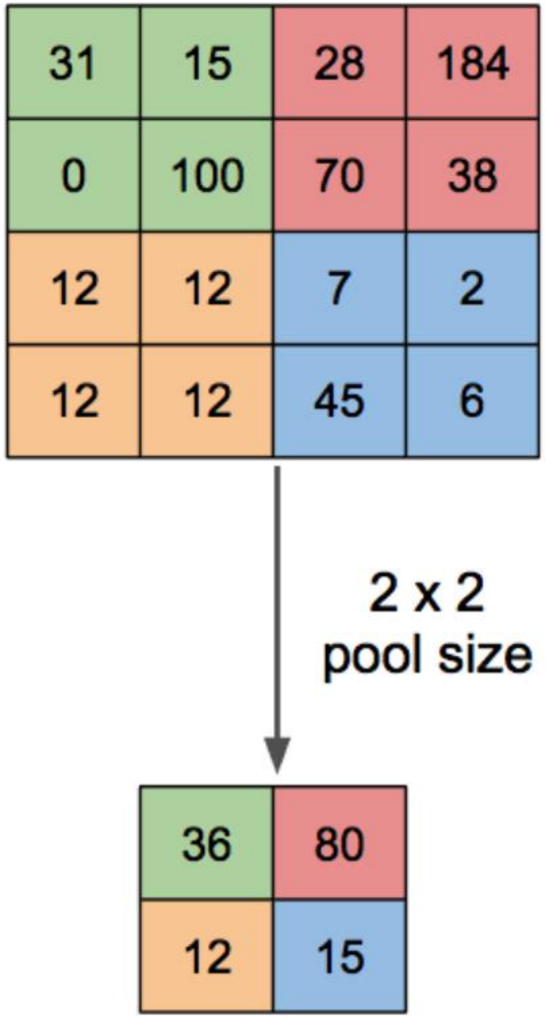


Subsampling operations

Q: what does average Pooling really do?

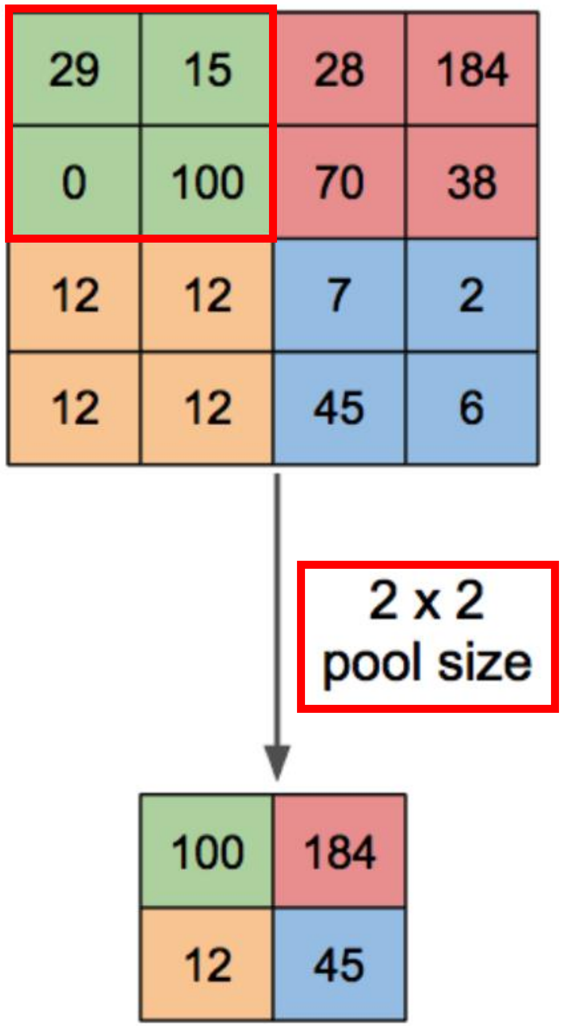
- Max pooling
- Average pooling

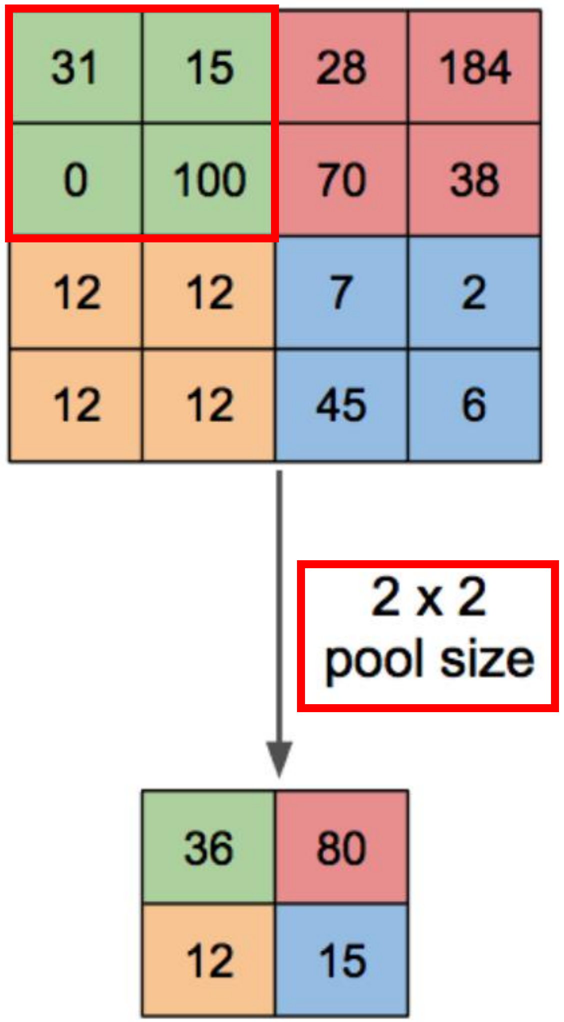




Subsampling operations

- Max pooling
- Average pooling

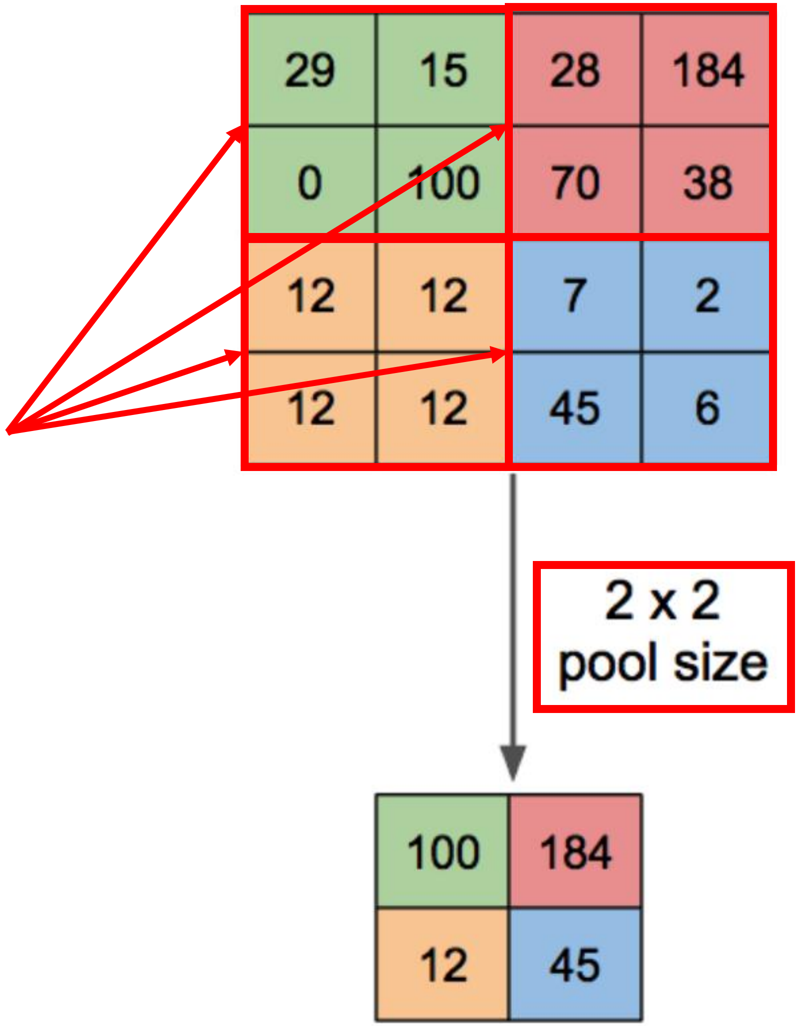


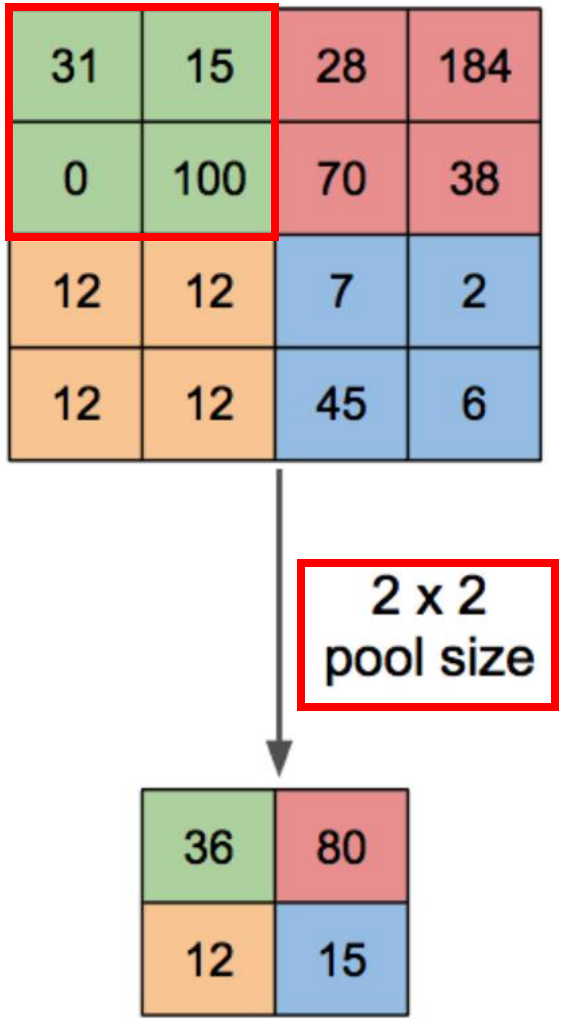


Subsampling operations

- Max pooling
- Average pooling

No overlapping

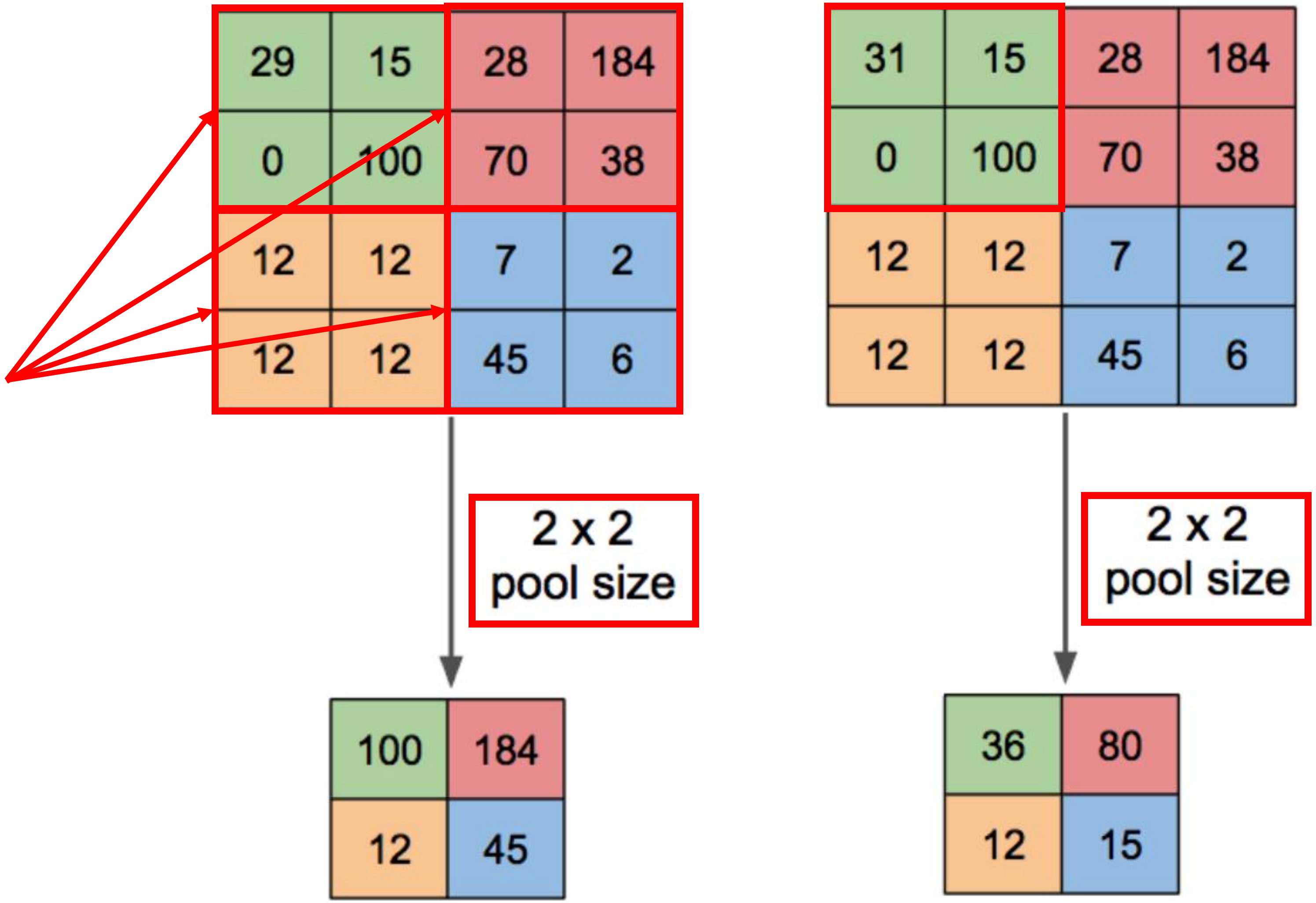




Subsampling operations

- Max pooling
- Average pooling

No overlapping
(stride=2*2)

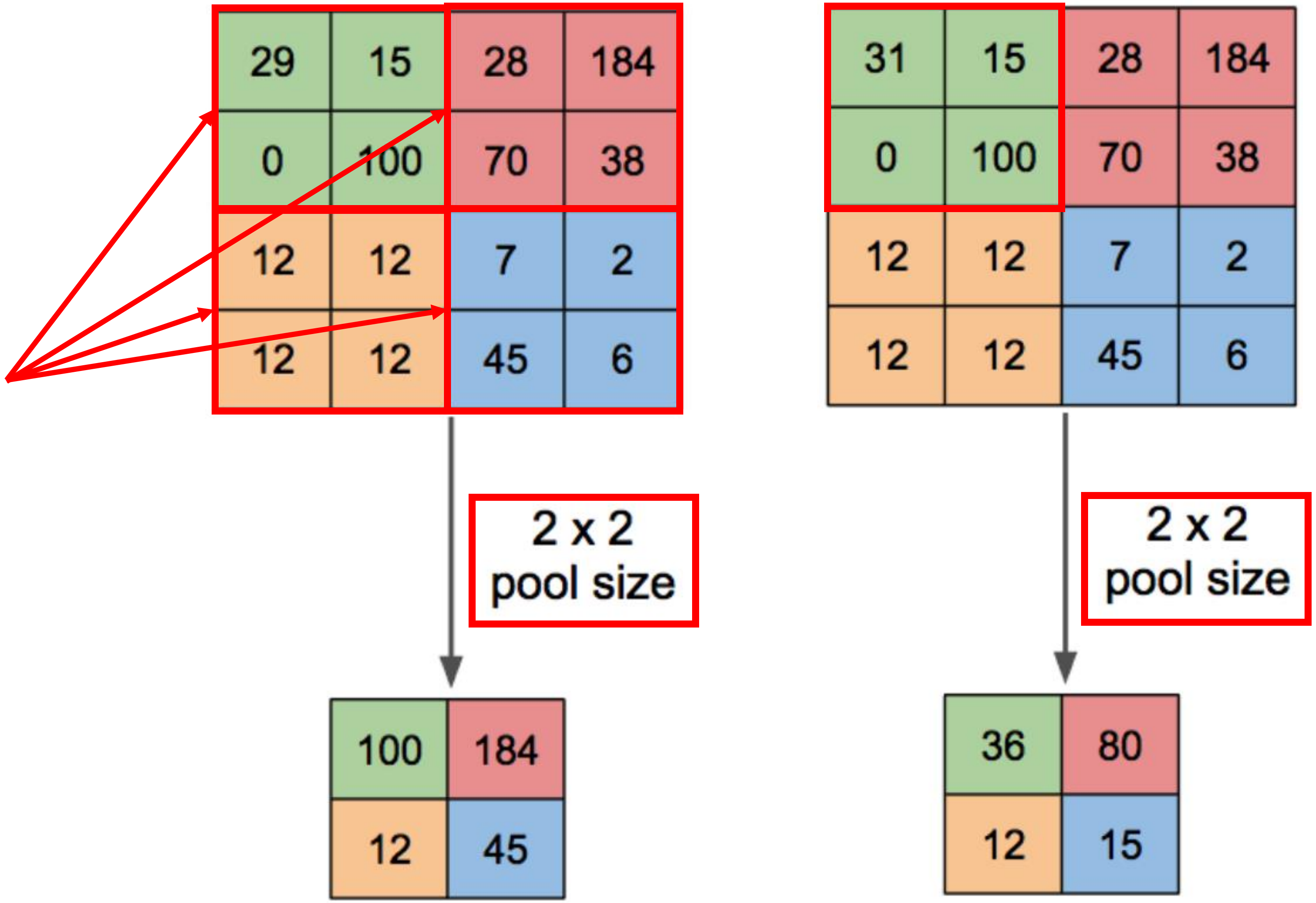


Subsampling operations

- Max pooling
- Average pooling

No overlapping
(stride=2*2)

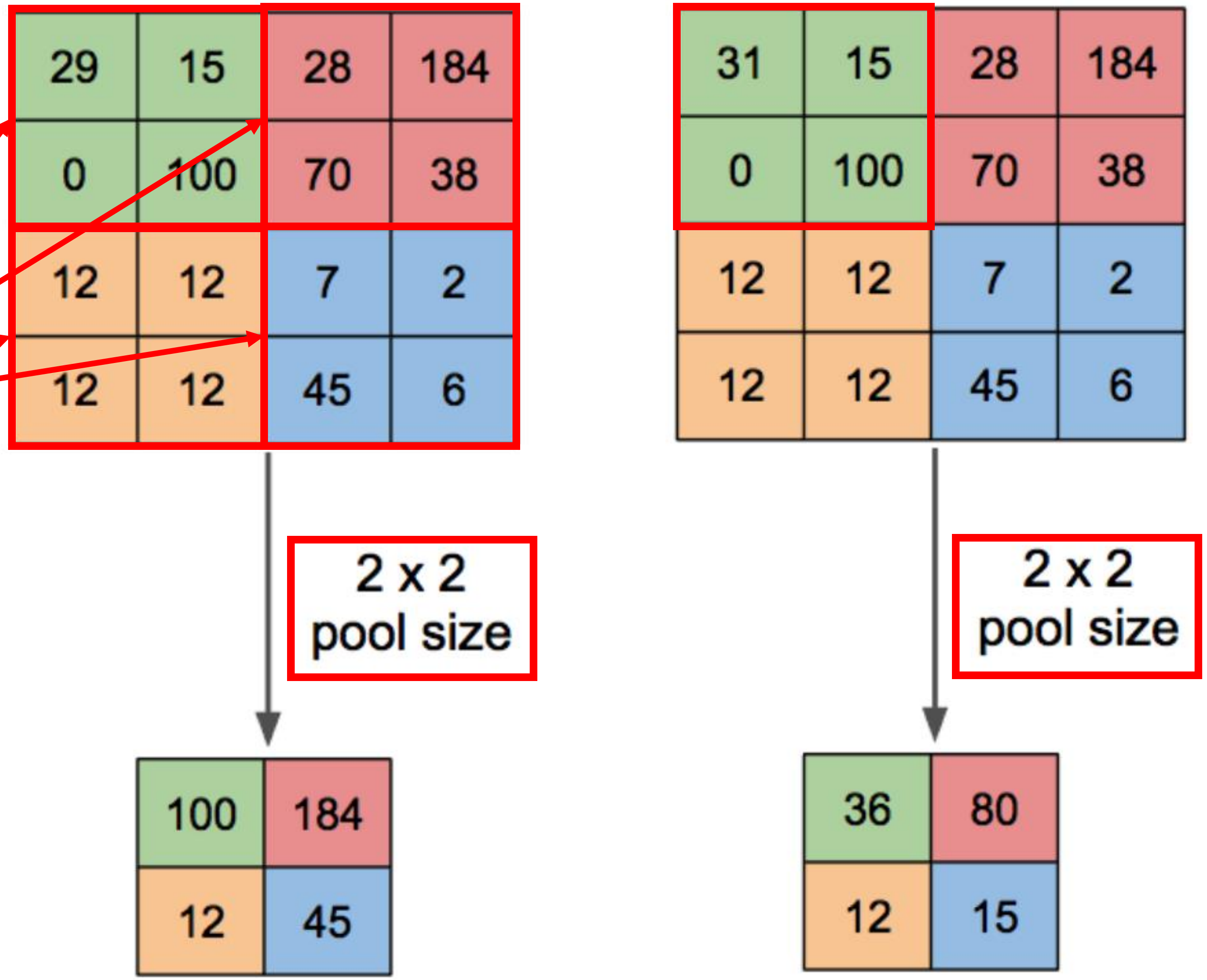
Row stride = 2
Column stride = 2



Subsampling operations

- Max pooling
- Average pooling

No overlapping
(stride=2*2)



Row stride = 2
Column stride = 2

Q: Why pooling?
Connection to subsampling?

Subsampling operations

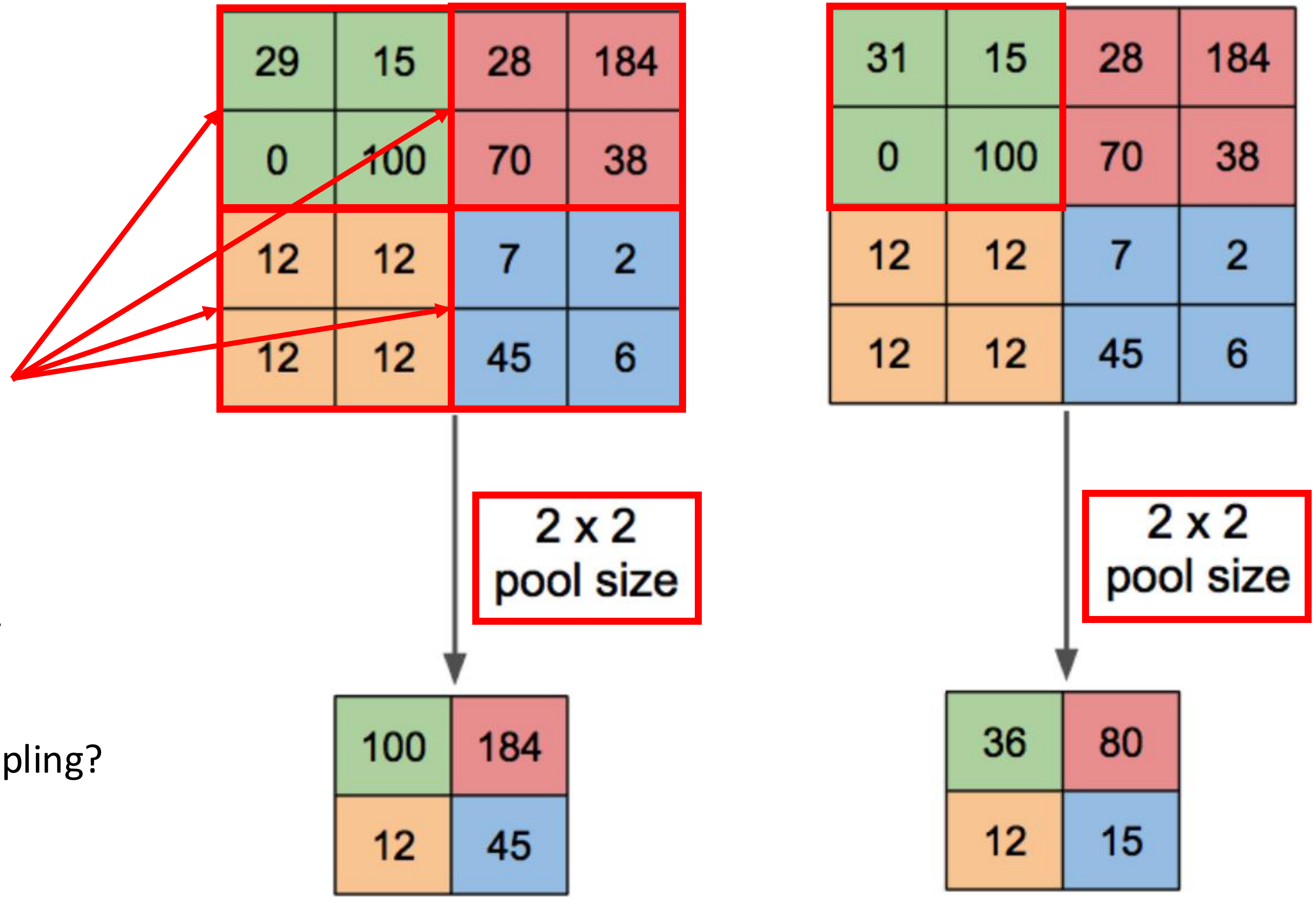
- Max pooling
- Average pooling

No overlapping
(stride=2*2)

Row stride = 2
Column stride = 2

Q: Why pooling?
Connection to subsampling?

4*4 → 2*2



Subsampling operations

- Max pooling
- Average pooling

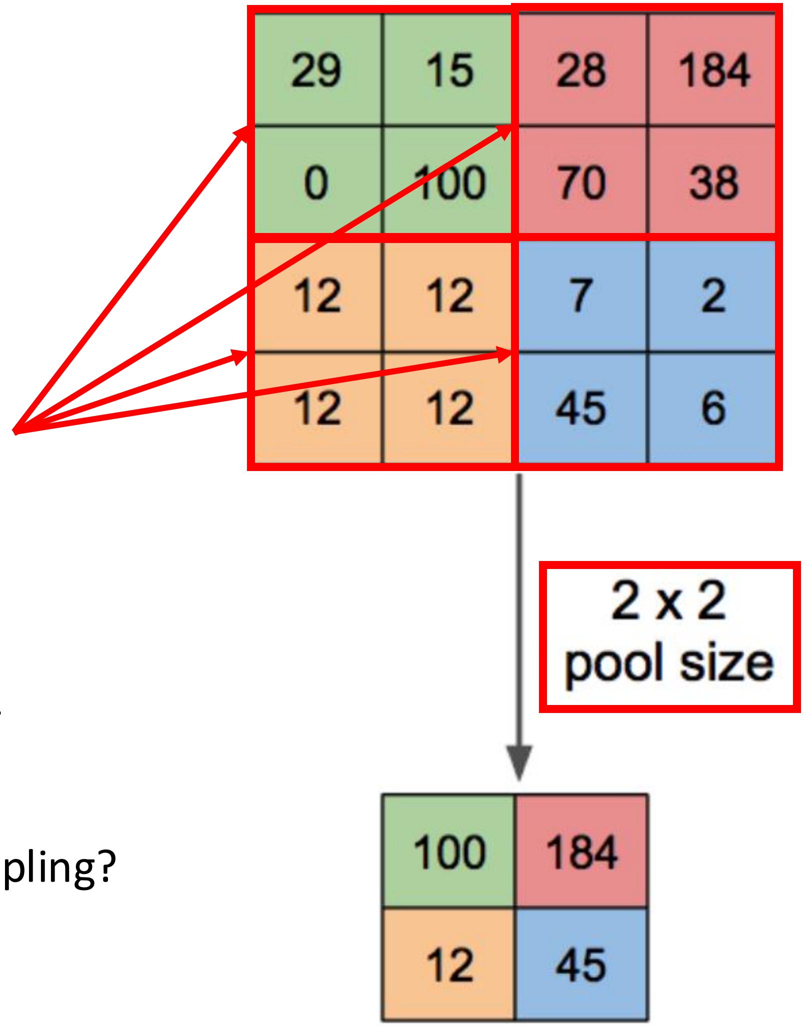
No overlapping
(stride=2*2)

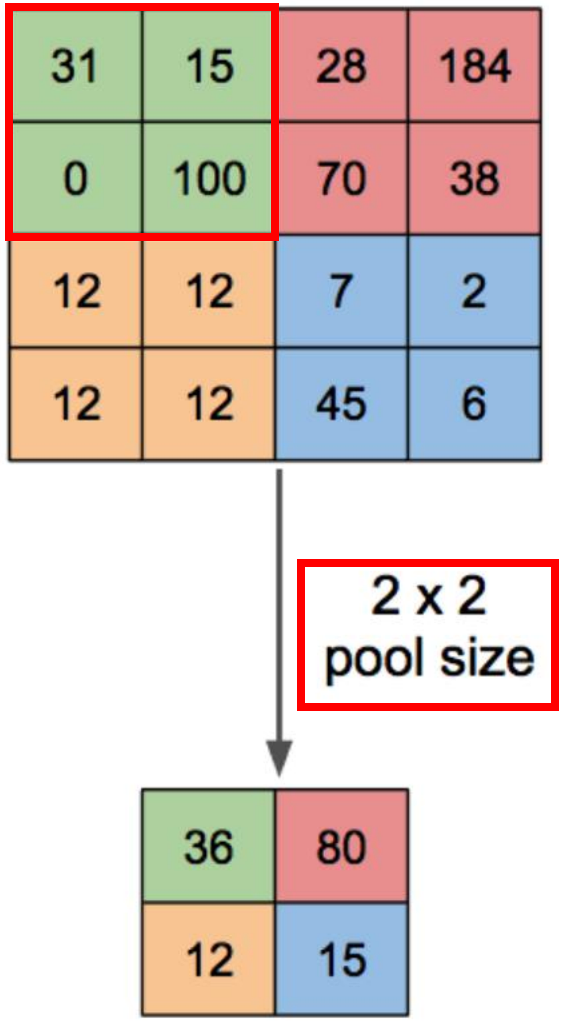
Row stride = 2
Column stride = 2

Q: Why pooling?
Connection to subsampling?

4*4 → 2*2

Dimension reduced





Subsampling operations

- Max pooling
- Average pooling

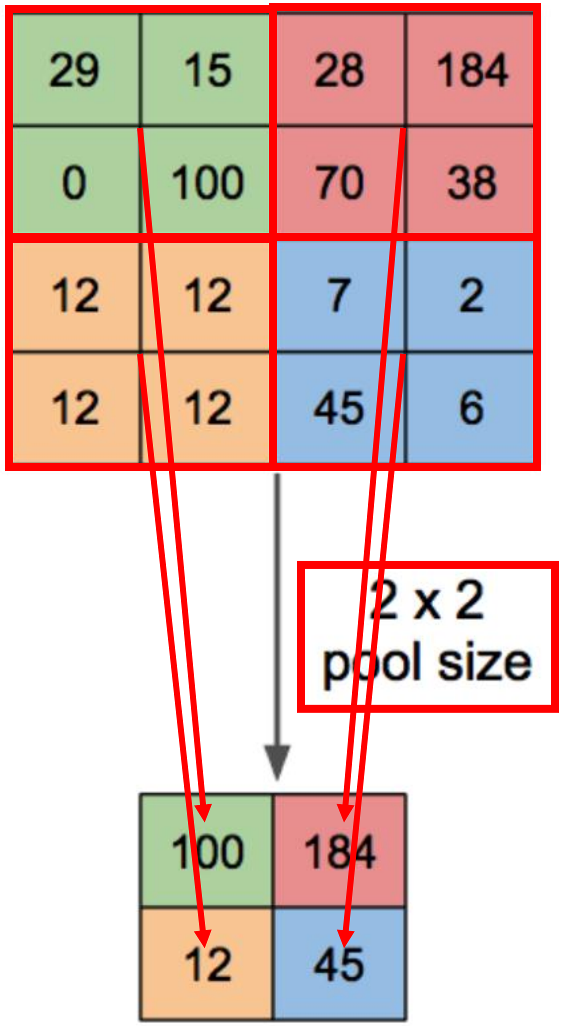
No overlapping
(stride=2*2)

Row stride = 2
Column stride = 2

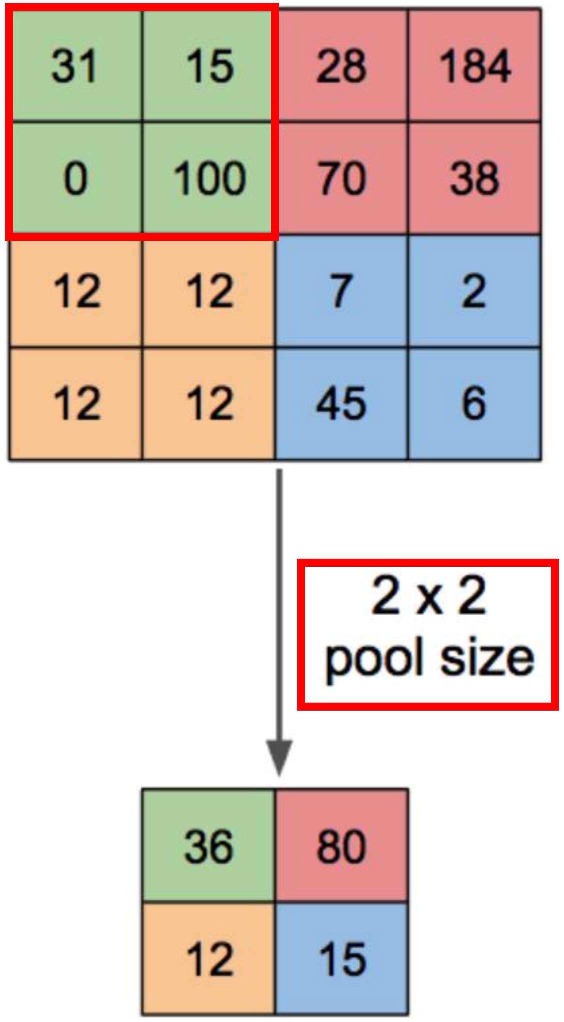
Q: Why pooling?
Connection to subsampling?

4*4 → 2*2

Dimension reduced



Use one to represent all



LeNet-5 in 1999

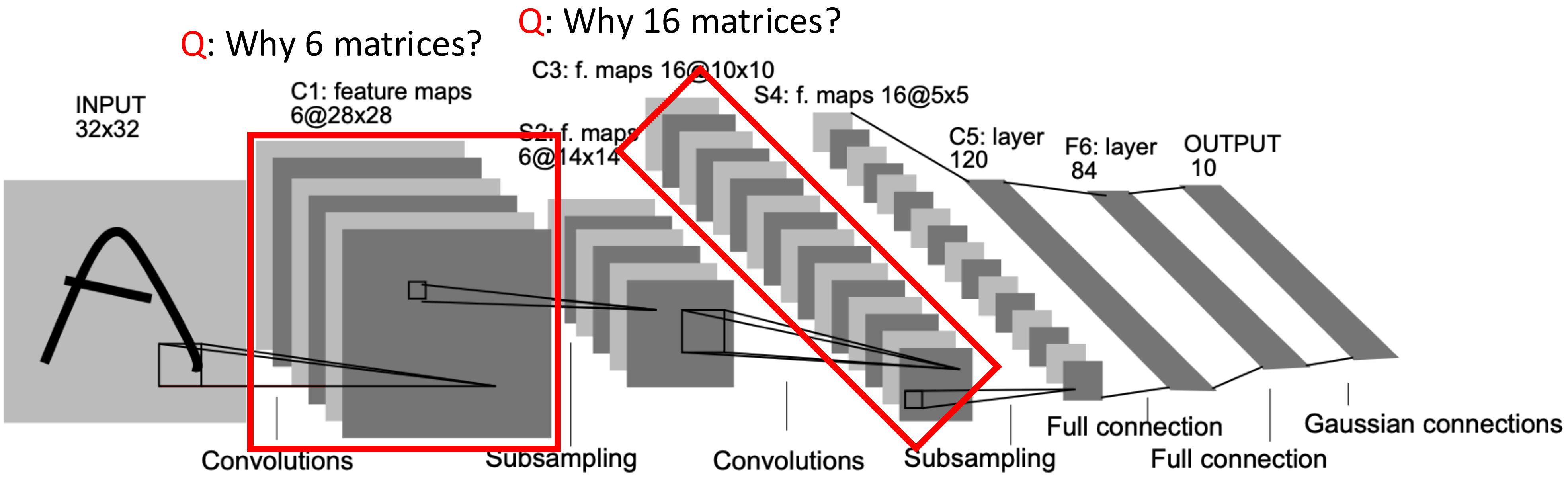


Fig. 1. Architecture of LeNet-5, a Convolutional Neural Network, here for digits recognition. Each plane is a feature map, i.e. a set of units whose weights are constrained to be identical.

LeCun, Yann, Patrick Haffner, Léon Bottou, and Yoshua Bengio. "Object recognition with gradient-based learning." In *Shape, contour and grouping in computer vision*, pp. 319-345. Springer, Berlin, Heidelberg, 1999.

LeNet-5 in 1999

Q: Why 6 matrices?

Q: Why 16 matrices?

A (reason 2): we can use multiple filters at each layer

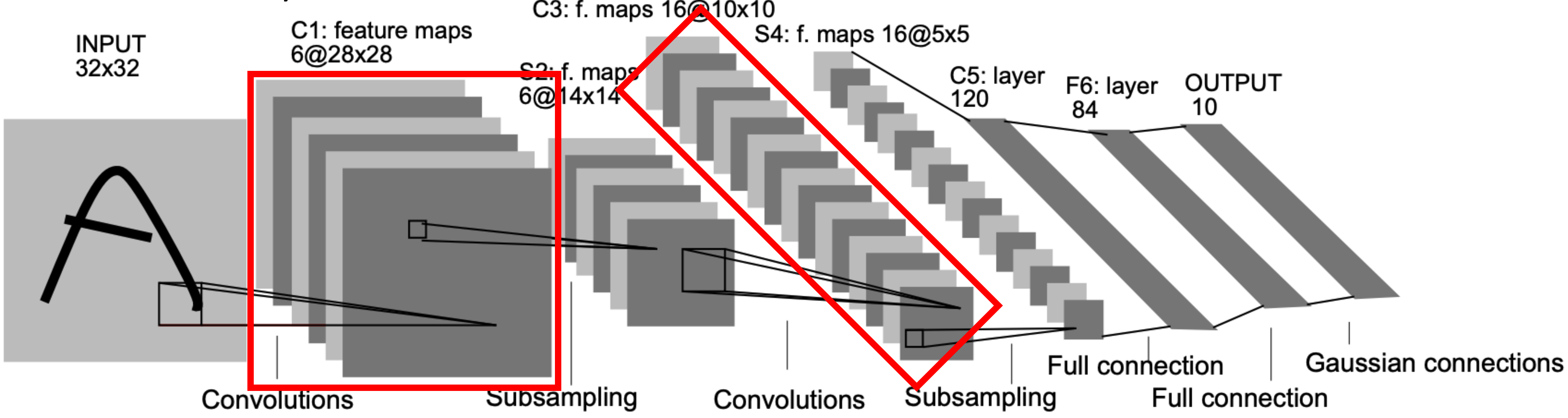


Fig. 1. Architecture of LeNet-5, a Convolutional Neural Network, here for digits recognition. Each plane is a feature map, i.e. a set of units whose weights are constrained to be identical.

LeCun, Yann, Patrick Haffner, Léon Bottou, and Yoshua Bengio. "Object recognition with gradient-based learning." In *Shape, contour and grouping in computer vision*, pp. 319-345. Springer, Berlin, Heidelberg, 1999.

LeNet-5 in 1999

Q: Why 6 matrices?

Q: Why 16 matrices?

A (reason 2): we can use multiple filters at each layer

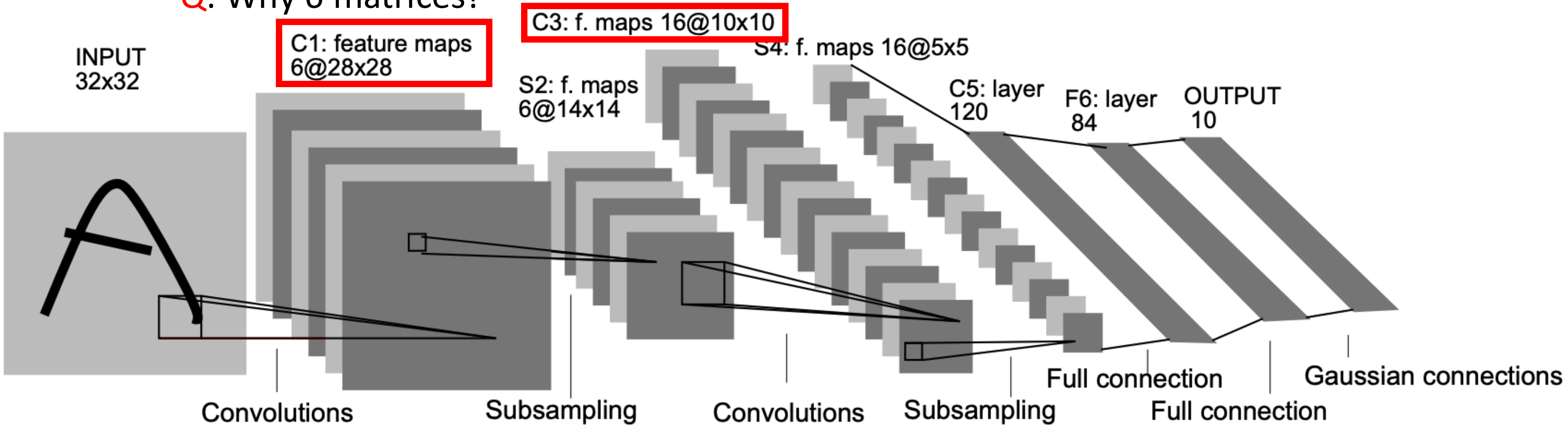


Fig. 1. Architecture of LeNet-5, a Convolutional Neural Network, here for digits recognition. Each plane is a feature map, i.e. a set of units whose weights are constrained to be identical.

LeCun, Yann, Patrick Haffner, Léon Bottou, and Yoshua Bengio. "Object recognition with gradient-based learning." In *Shape, contour and grouping in computer vision*, pp. 319-345. Springer, Berlin, Heidelberg, 1999.

LeNet-5 in 1999

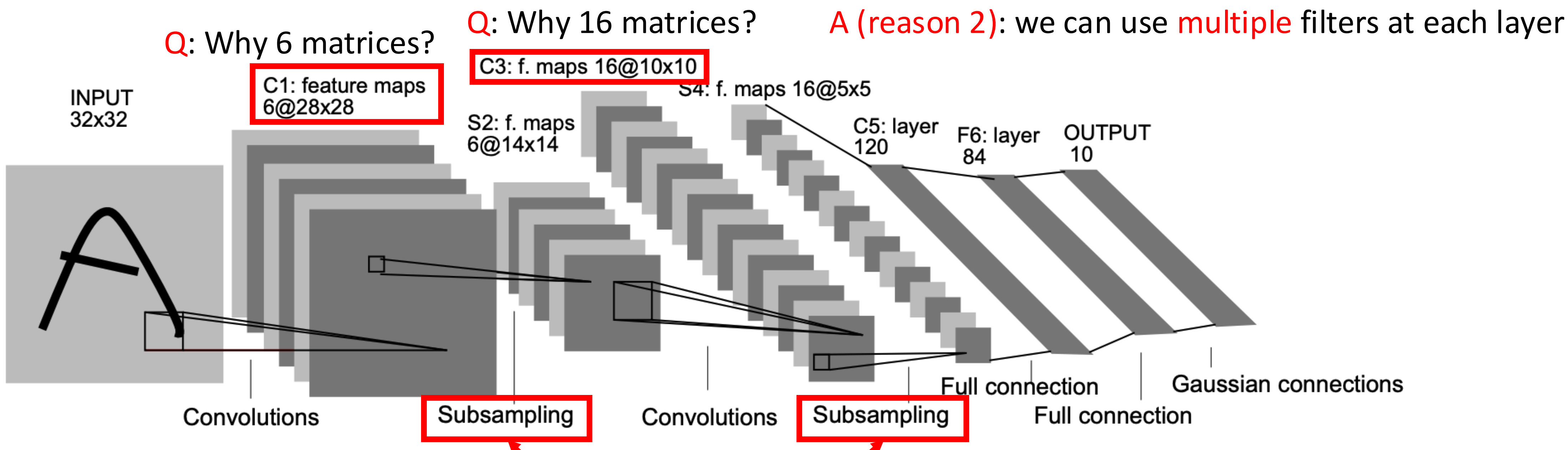


Fig. 1. Architecture of LeNet-5, a Convolutional Neural Network, here for digits recognition. Each plane is a feature map, i.e. a set of units whose weights are constrained to be identical.

Subsampling layer: max/average pooling

LeCun, Yann, Patrick Haffner, Léon Bottou, and Yoshua Bengio. "Object recognition with gradient-based learning." In *Shape, contour and grouping in computer vision*, pp. 319-345. Springer, Berlin, Heidelberg, 1999.

LeNet-5 in 1999

One more question:
How C5 comes from?

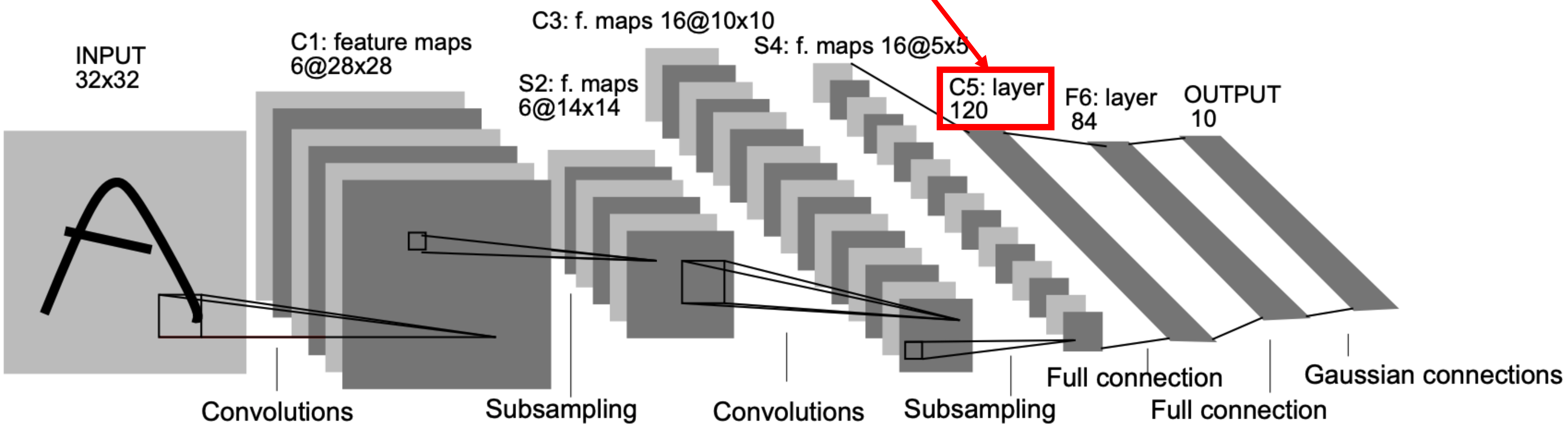


Fig. 1. Architecture of LeNet-5, a Convolutional Neural Network, here for digits recognition. Each plane is a feature map, i.e. a set of units whose weights are constrained to be identical.

LeCun, Yann, Patrick Haffner, Léon Bottou, and Yoshua Bengio. "Object recognition with gradient-based learning." In *Shape, contour and grouping in computer vision*, pp. 319-345. Springer, Berlin, Heidelberg, 1999.

LeNet-5 in 1999

One more question:
How C5 comes from? Matrices → a vector?

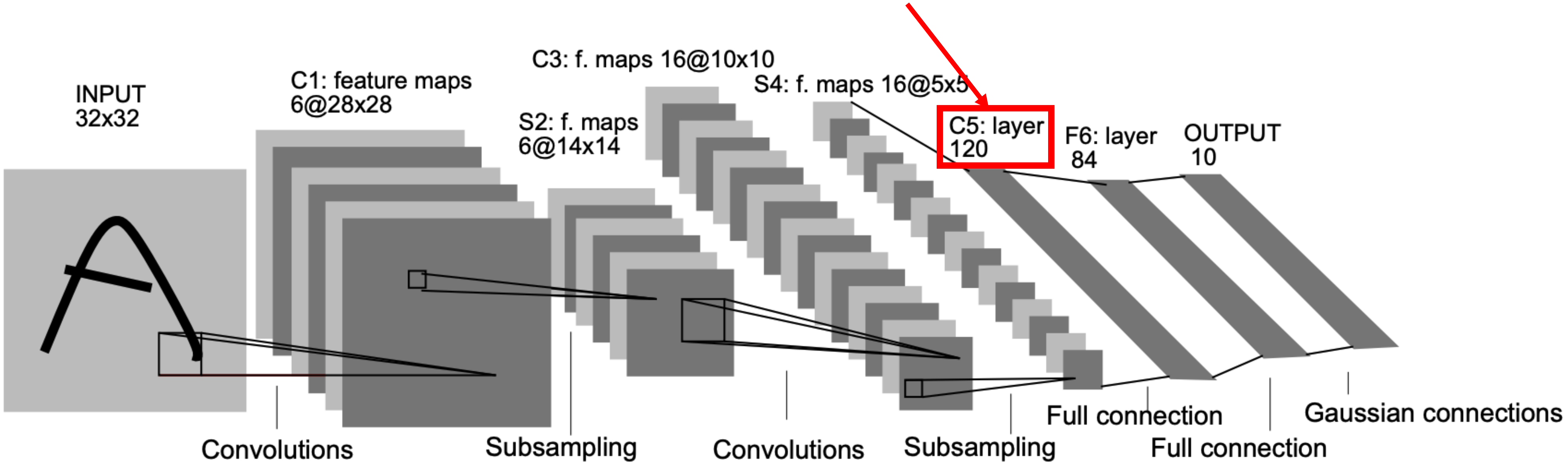


Fig. 1. Architecture of LeNet-5, a Convolutional Neural Network, here for digits recognition. Each plane is a feature map, i.e. a set of units whose weights are constrained to be identical.

LeCun, Yann, Patrick Haffner, Léon Bottou, and Yoshua Bengio. "Object recognition with gradient-based learning." In *Shape, contour and grouping in computer vision*, pp. 319-345. Springer, Berlin, Heidelberg, 1999.

LeNet-5 in 1999

One more question:
Where C5 comes from? 16 matrices → a 120d vector?

Layer C5 is a convolutional layer with 120 feature maps. Each unit is connected to a 5x5 neighborhood on all 16 of S4's feature maps. Here, because the size of S4 is also 5x5, the size of C5's feature maps is 1x1: this amounts to a full connection between S4 and C5. C5 is labeled as a convolutional layer, instead of a fully-connected layer, because if LeNet-5 input were made bigger with everything else kept constant, the feature map dimension would be larger than 1x1. This process of dynamically increasing the size of a convolutional network is described in the section Section VII. Layer C5 has 48,120 trainable connections.

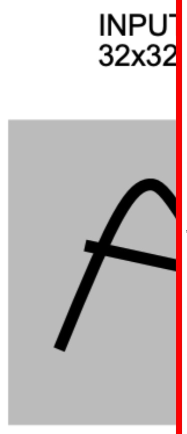


Fig. 1. … s recog- nition. … strained to be identical.

LeCun, Yann, Patrick Haffner, Léon Bottou, and Yoshua Bengio. "Object recognition with gradient-based learning." In *Shape, contour and grouping in computer vision*, pp. 319-345. Springer, Berlin, Heidelberg, 1999.

LeNet-5 in 1999

One more question:

Where C5 comes from? 16 matrices → a 120d vector?

Layer C5 is a convolutional layer with $\boxed{120}$ feature maps. Each unit is connected to a $\boxed{5x5}$ neighborhood on all 16 of S4's feature maps. Here, because the size of S4 is also 5x5, the size of C5's feature maps is 1x1: this amounts to a full connection between S4 and C5. C5 is labeled as a convolutional layer, instead of a fully-connected layer, because if LeNet-5 input were made bigger with everything else kept constant, the feature map dimension would be larger than 1x1. This process of dynamically increasing the size of a convolutional network is described in the section Section VII. Layer C5 has 48,120 trainable connections.

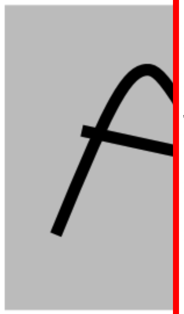


Fig. 1. ... s recog- nition. ... strained to be identical.

LeCun, Yann, Patrick Haffner, Léon Bottou, and Yoshua Bengio. "Object recognition with gradient-based learning." In *Shape, contour and grouping in computer vision*, pp. 319-345. Springer, Berlin, Heidelberg, 1999.

Operations with convolution layers

- Padding
- Pooling layers for arbitrary input resolution

Operations with convolution layers

- Padding: convolution operation reduces the size of feature maps

Operations with convolution layers

- Padding: convolution operation reduces the size of feature maps

$$
\begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 \end{bmatrix}
*
\begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}
\rightarrow
\begin{bmatrix} 4 & 3 & 4 \\ 2 & 4 & 3 \\ 2 & 3 & 4 \end{bmatrix}
$$

n=5 m=3 n-m+1=3

Operations with convolution layers

- Padding: convolution operation reduces the size of feature maps

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 \end{bmatrix} * \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} \rightarrow \begin{bmatrix} 4 & 3 & 4 \\ 2 & 4 & 3 \\ 2 & 3 & 4 \end{bmatrix}$$

n=5 m=3 n-m+1=3

If m>1 → ??

Operations with convolution layers

- Padding: convolution operation reduces the size of feature maps

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 \end{bmatrix} * \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} \rightarrow \begin{bmatrix} 4 & 3 & 4 \\ 2 & 4 & 3 \\ 2 & 3 & 4 \end{bmatrix}$$

n=5 m=3 n-m+1=3

If m>1 → convolution will reduce the dimension

Operations with convolution layers

- Padding: convolution operation reduces the size of feature maps

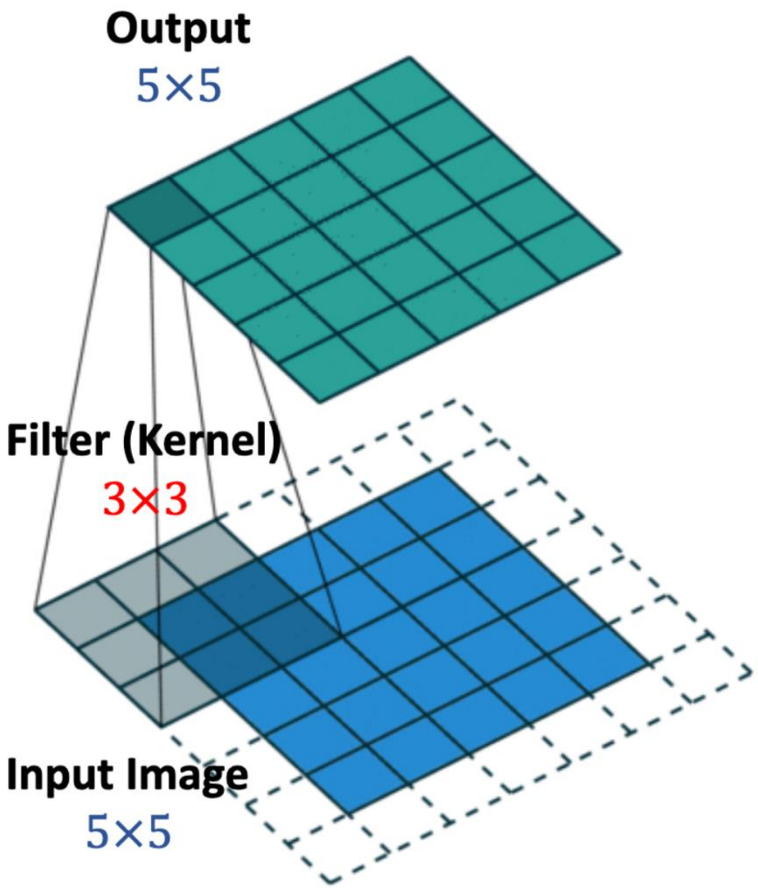
$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 \end{bmatrix} * \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} \rightarrow \begin{bmatrix} 4 & 3 & 4 \\ 2 & 4 & 3 \\ 2 & 3 & 4 \end{bmatrix}$$

n=5 m=3 n-m+1=3

If m>1 → convolution will reduce the dimension
The input resolution introduces a limits of #convolution layers

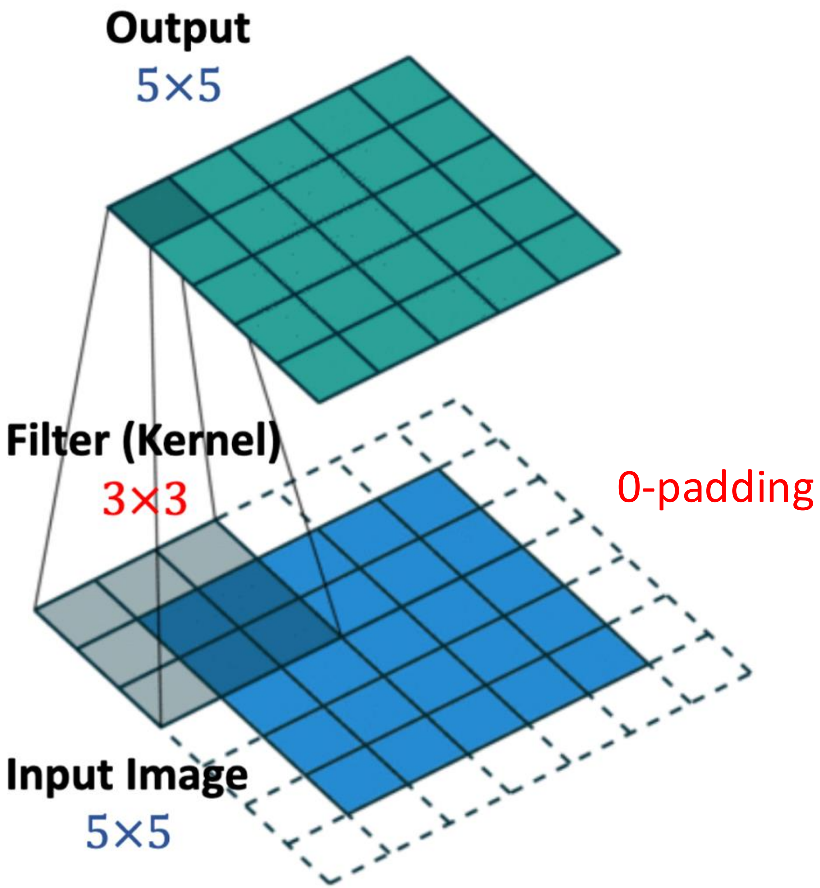
Operations with convolution layers

- Padding: convolution operation reduces the size of feature maps



Operations with convolution layers

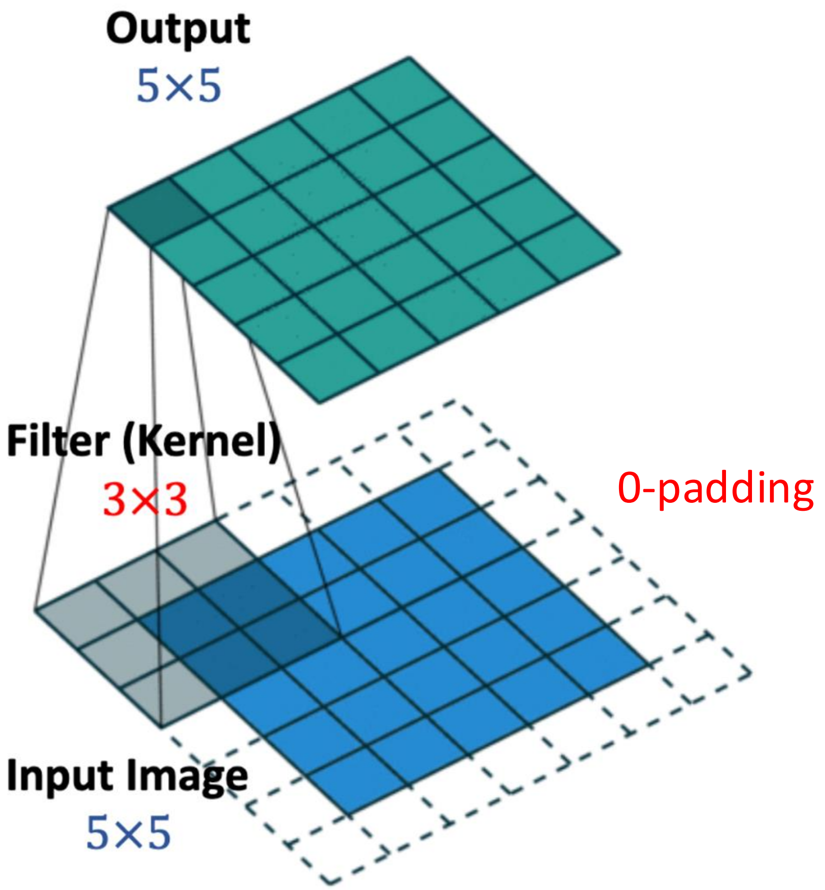
- Padding: convolution operation reduces the size of feature maps



Operations with convolution layers

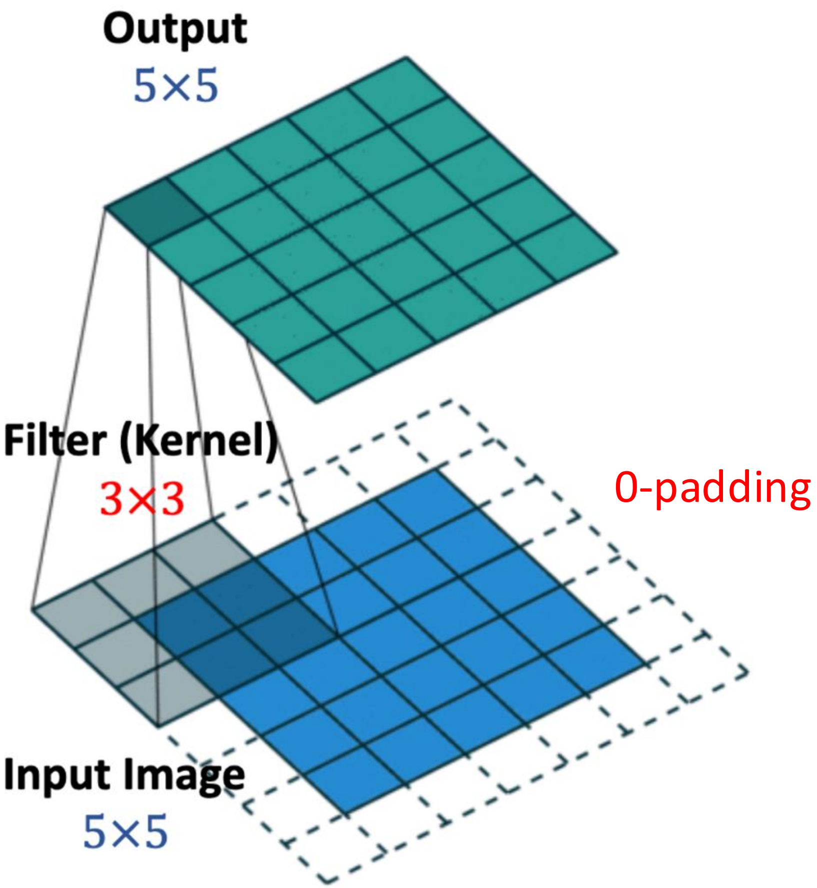
- Padding: convolution operation reduces the size of feature maps

Input size
n→7



Operations with convolution layers

- Padding: convolution operation reduces the size of feature maps



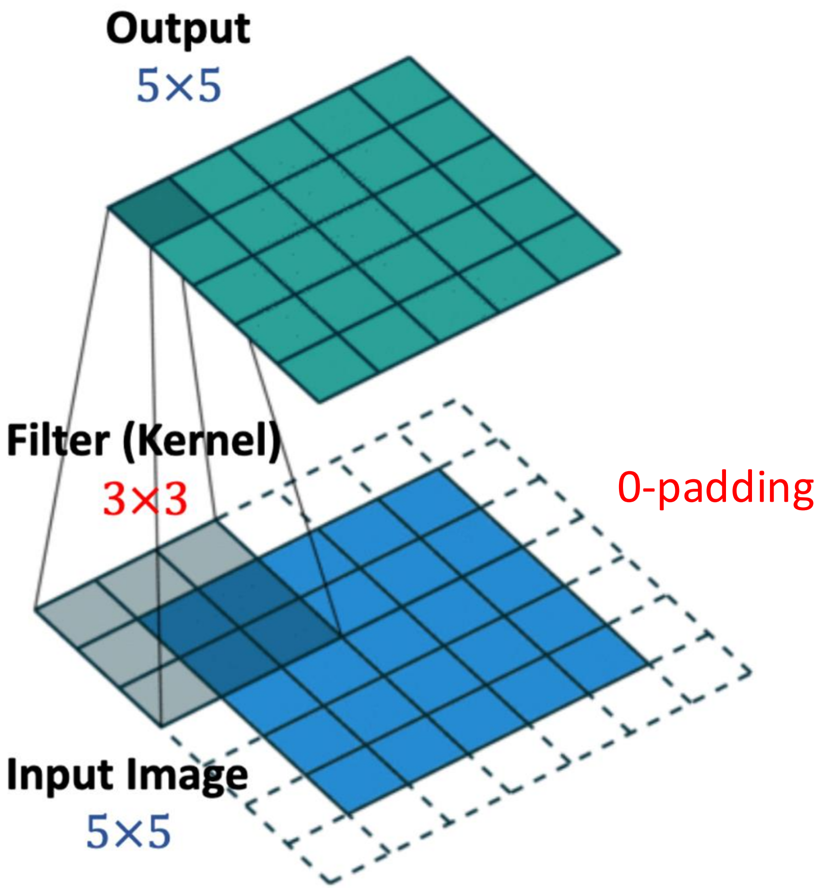
Input size

n→7 → n-m+1=7-3+1=5

Output size

Operations with convolution layers

- Padding: convolution operation reduces the size of feature maps



n→7 → n-m+1=7-3+1=5

Conclusion:
dimension of feature maps remains the same

Operations with convolution layers

- Padding: convolution operation reduces the size of feature maps
- Pooling layers for an arbitrary input resolution

Input resolution issue

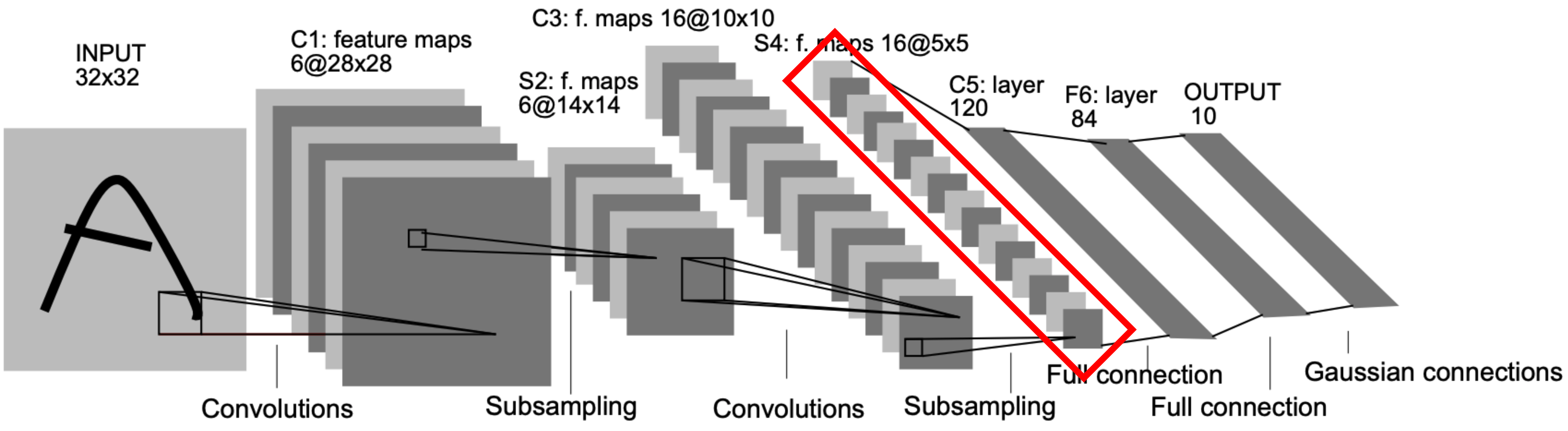


Fig. 1. Architecture of LeNet-5, a Convolutional Neural Network, here for digits recognition. Each plane is a feature map, i.e. a set of units whose weights are constrained to be identical.

LeCun, Yann, Patrick Haffner, Léon Bottou, and Yoshua Bengio. "Object recognition with gradient-based learning." In *Shape, contour and grouping in computer vision*, pp. 319-345. Springer, Berlin, Heidelberg, 1999.

Input resolution issue

We use 120 5x5 filters

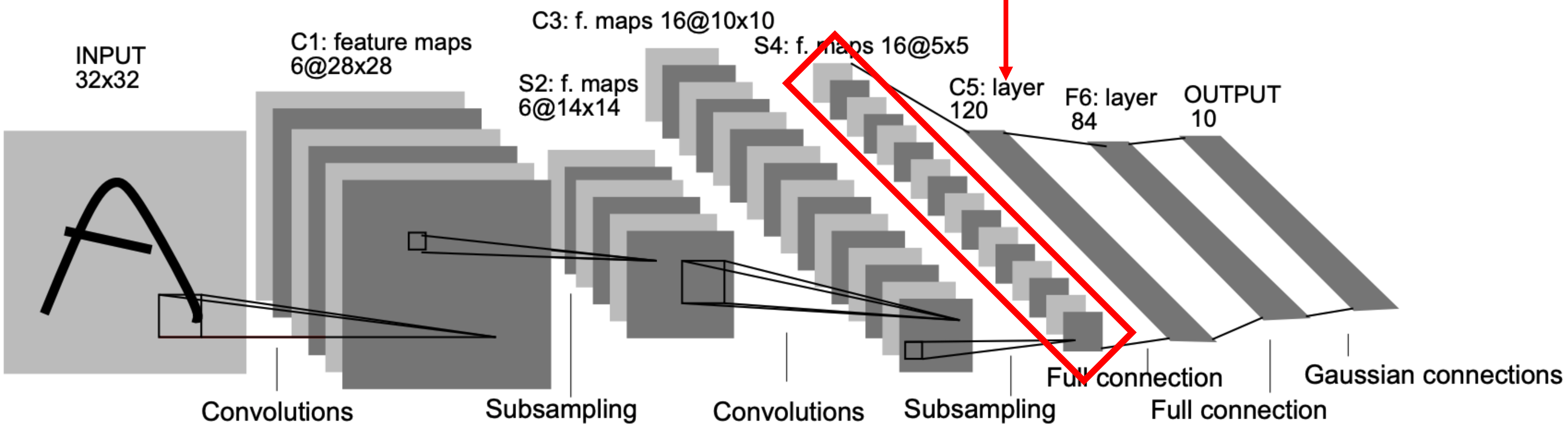


Fig. 1. Architecture of LeNet-5, a Convolutional Neural Network, here for digits recognition. Each plane is a feature map, i.e. a set of units whose weights are constrained to be identical.

LeCun, Yann, Patrick Haffner, Léon Bottou, and Yoshua Bengio. "Object recognition with gradient-based learning." In *Shape, contour and grouping in computer vision*, pp. 319-345. Springer, Berlin, Heidelberg, 1999.

Input resolution issue

We use 120 5x5 filters

But why 5x5?

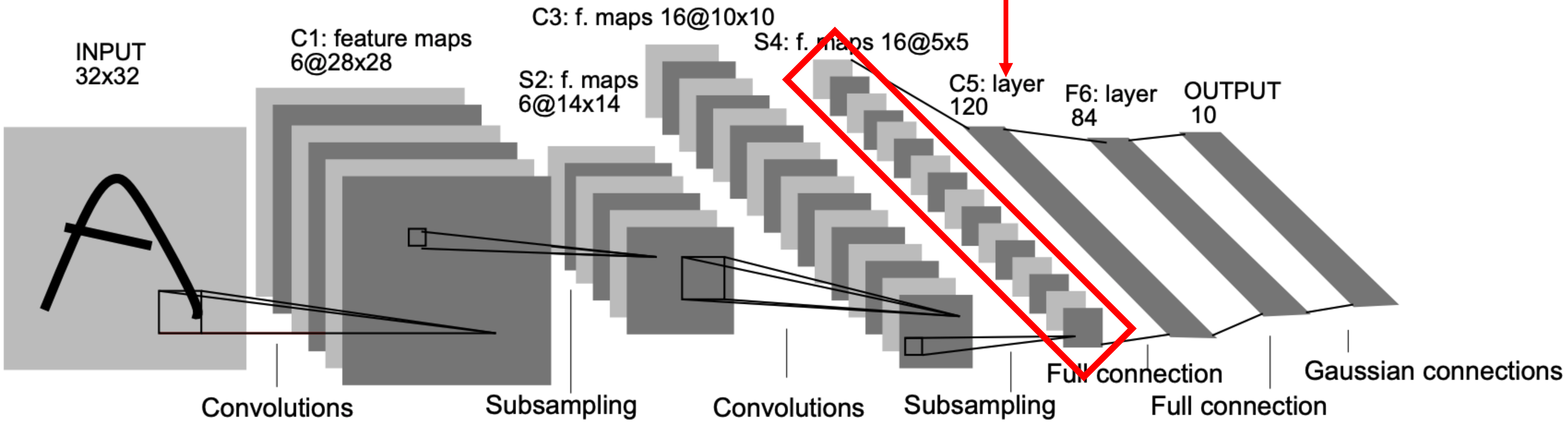


Fig. 1. Architecture of LeNet-5, a Convolutional Neural Network, here for digits recognition. Each plane is a feature map, i.e. a set of units whose weights are constrained to be identical.

LeCun, Yann, Patrick Haffner, Léon Bottou, and Yoshua Bengio. "Object recognition with gradient-based learning." In *Shape, contour and grouping in computer vision*, pp. 319-345. Springer, Berlin, Heidelberg, 1999.

Input resolution issue

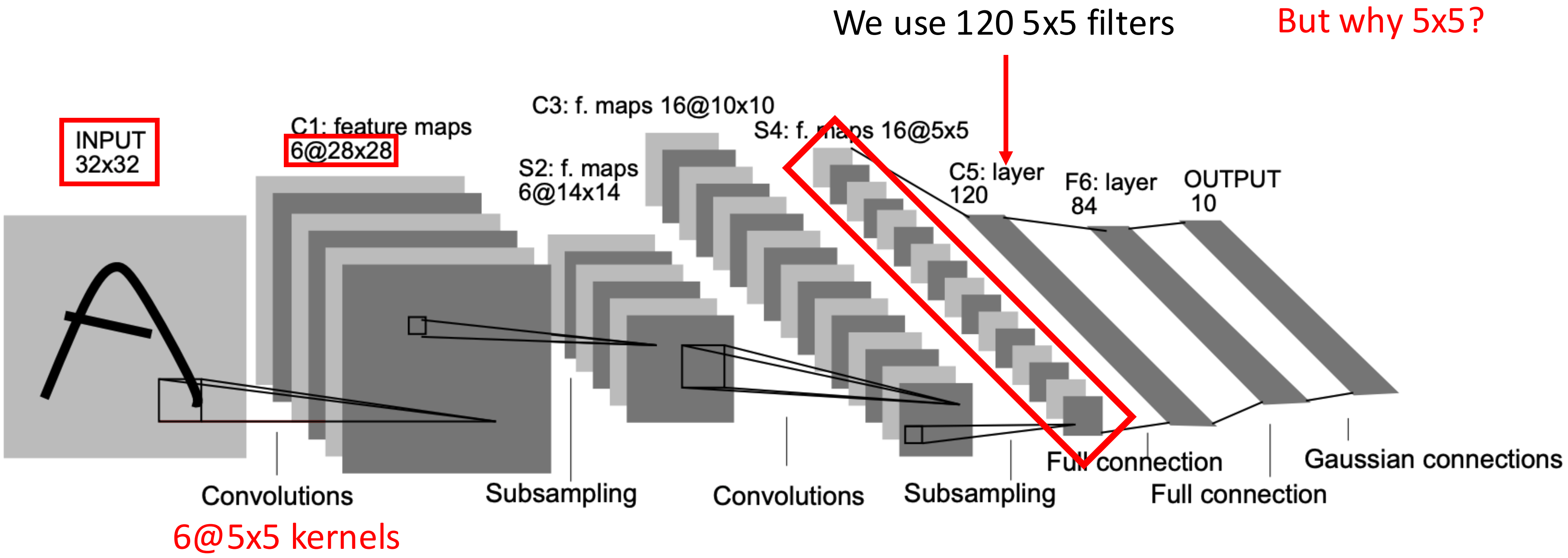


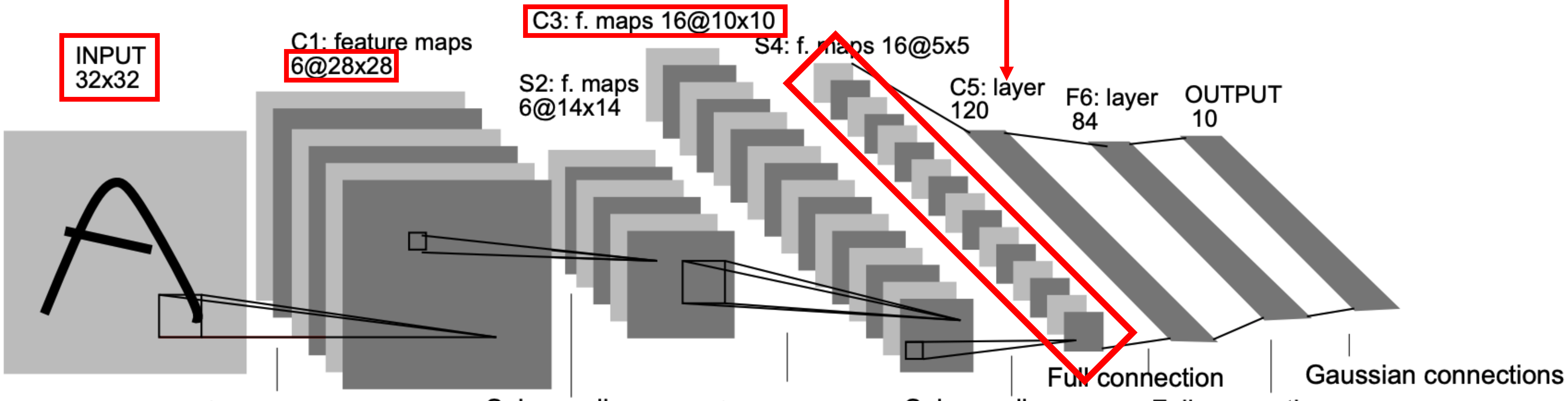
Fig. 1. Architecture of LeNet-5, a Convolutional Neural Network, here for digits recognition. Each plane is a feature map, i.e. a set of units whose weights are constrained to be identical.

LeCun, Yann, Patrick Haffner, Léon Bottou, and Yoshua Bengio. "Object recognition with gradient-based learning." In *Shape, contour and grouping in computer vision*, pp. 319-345. Springer, Berlin, Heidelberg, 1999.

Input resolution issue

We use 120 5x5 filters

But why 5x5?



6@5x5 kernels

16@5x5 kernels

Fig. 1. Architecture of LeNet-5, a Convolutional Neural Network, here for digits recognition. Each plane is a feature map, i.e. a set of units whose weights are constrained to be identical.

LeCun, Yann, Patrick Haffner, Léon Bottou, and Yoshua Bengio. "Object recognition with gradient-based learning." In *Shape, contour and grouping in computer vision*, pp. 319-345. Springer, Berlin, Heidelberg, 1999.

Input resolution issue

We use **120 5x5** filters

But why 5x5?

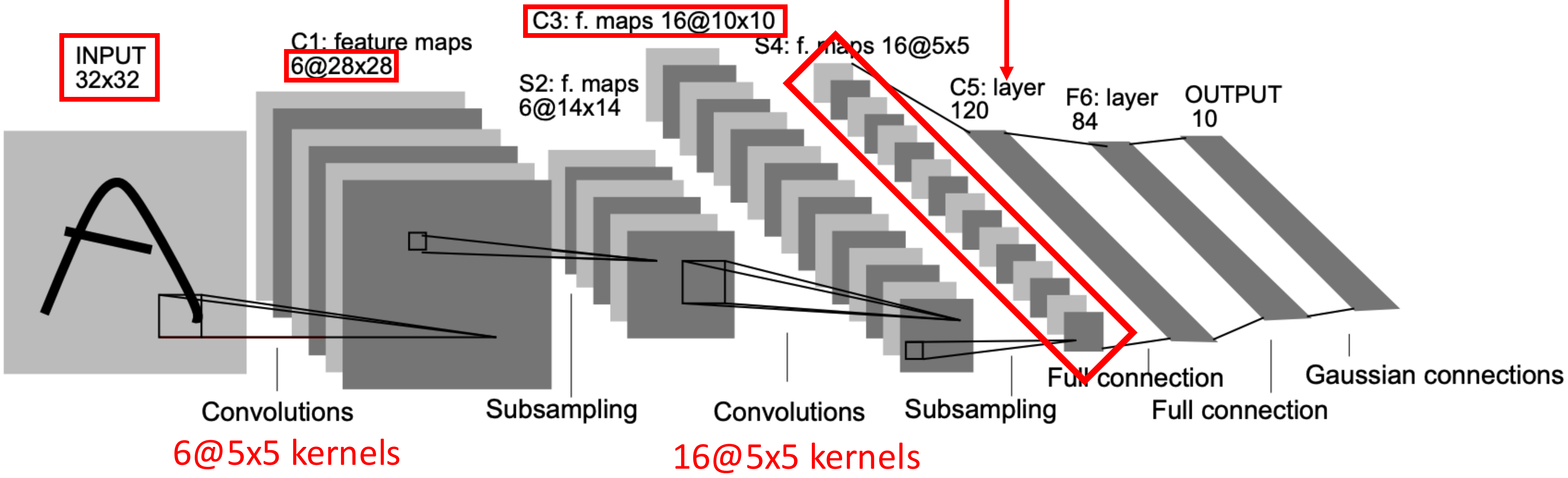


Fig. 1. Architecture of LeNet-5, a Convolutional Neural Network, here for digits recognition. Each plane is a feature map, i.e. a set of units whose weights are constrained to be identical.

LeCun, Yann, Patrick Haffner, Léon Bottou, and Yoshua Bengio. "Object recognition with gradient-based learning." In *Shape, contour and grouping in computer vision*, pp. 319-345. Springer, Berlin, Heidelberg, 1999.

Input resolution issue

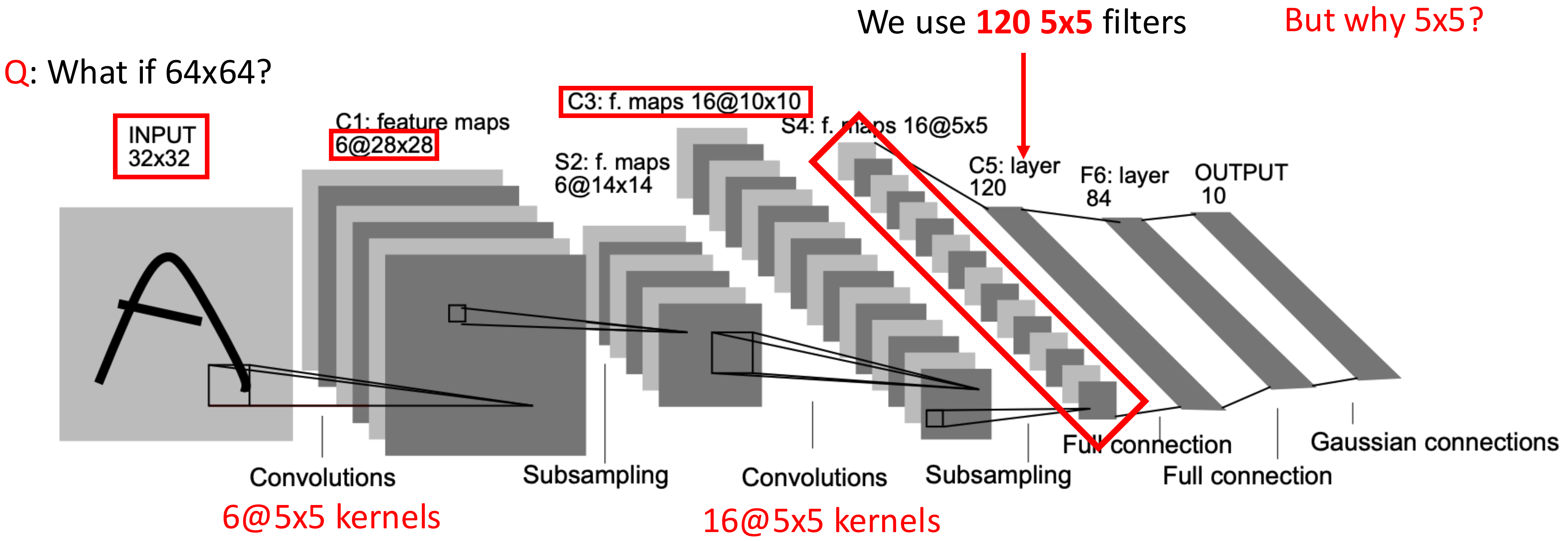
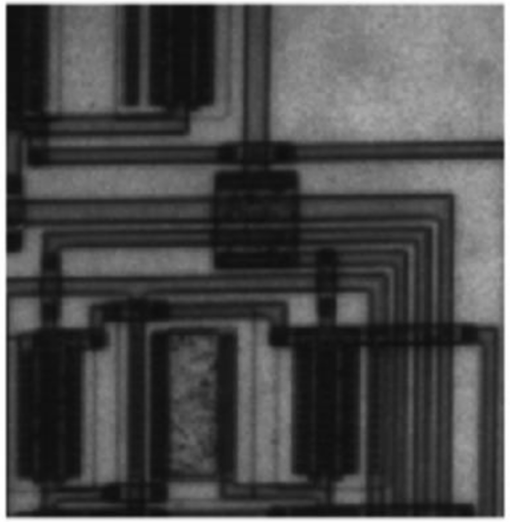


Fig. 1. Architecture of LeNet-5, a Convolutional Neural Network, here for digits recognition. Each plane is a feature map, i.e. a set of units whose weights are constrained to be identical.

LeCun, Yann, Patrick Haffner, Léon Bottou, and Yoshua Bengio. "Object recognition with gradient-based learning." In *Shape, contour and grouping in computer vision*, pp. 319-345. Springer, Berlin, Heidelberg, 1999.

Input resolution issue

Original Image



Cropped Image

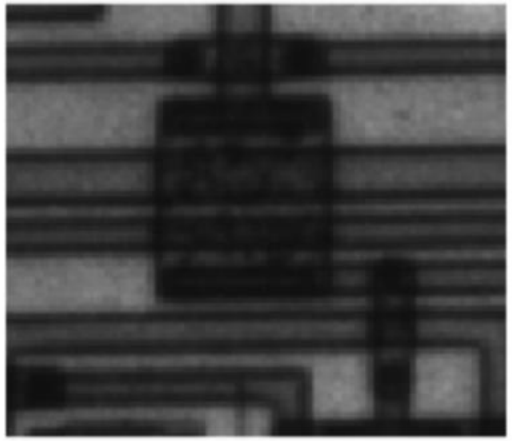
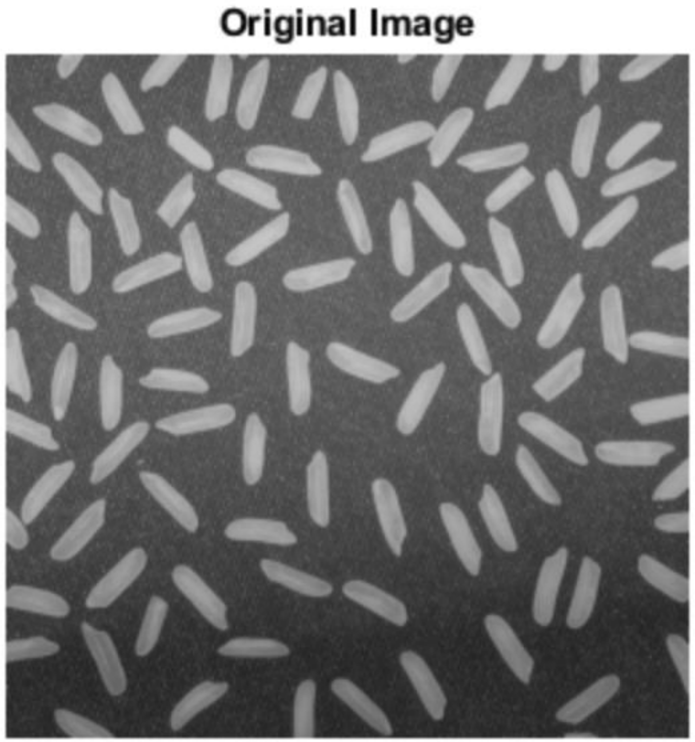


Image from https://www.mathworks.com/help/images/ref/imcrop.html

Input resolution issue



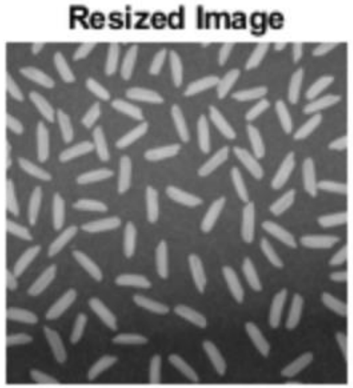
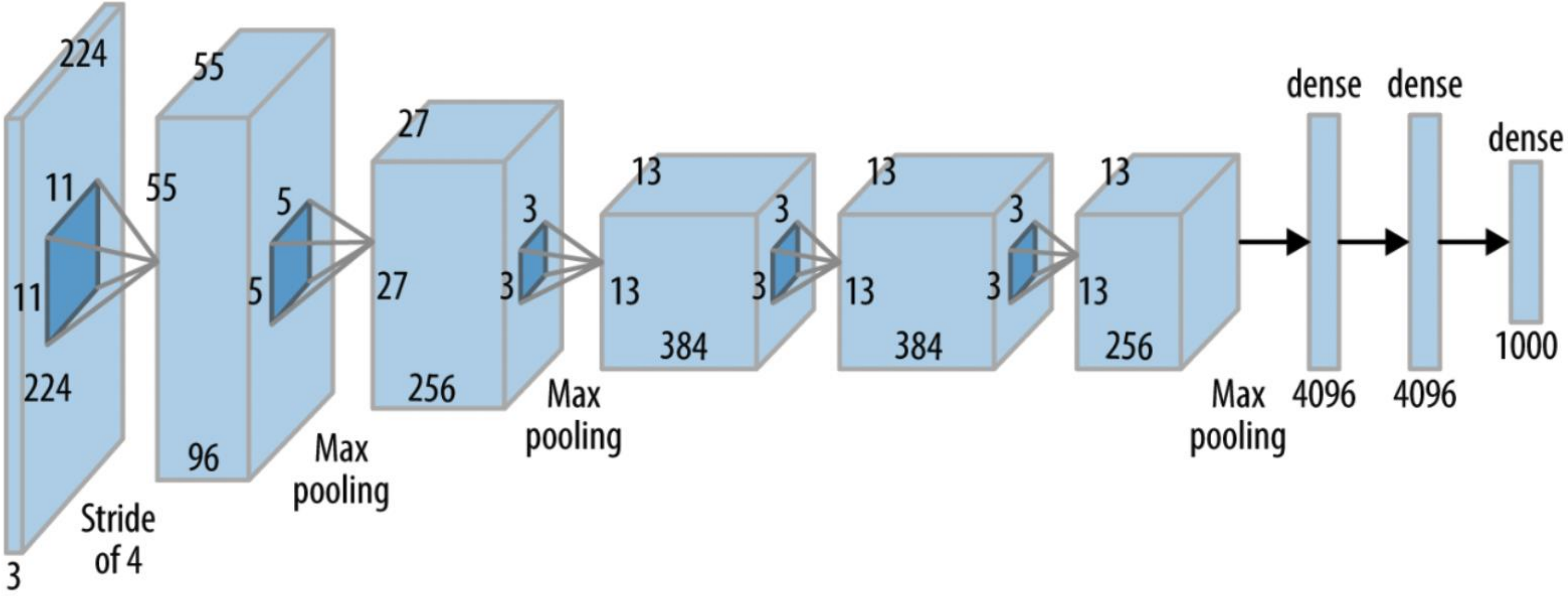


Image from https://www.mathworks.com/help/images/ref/imresize.html

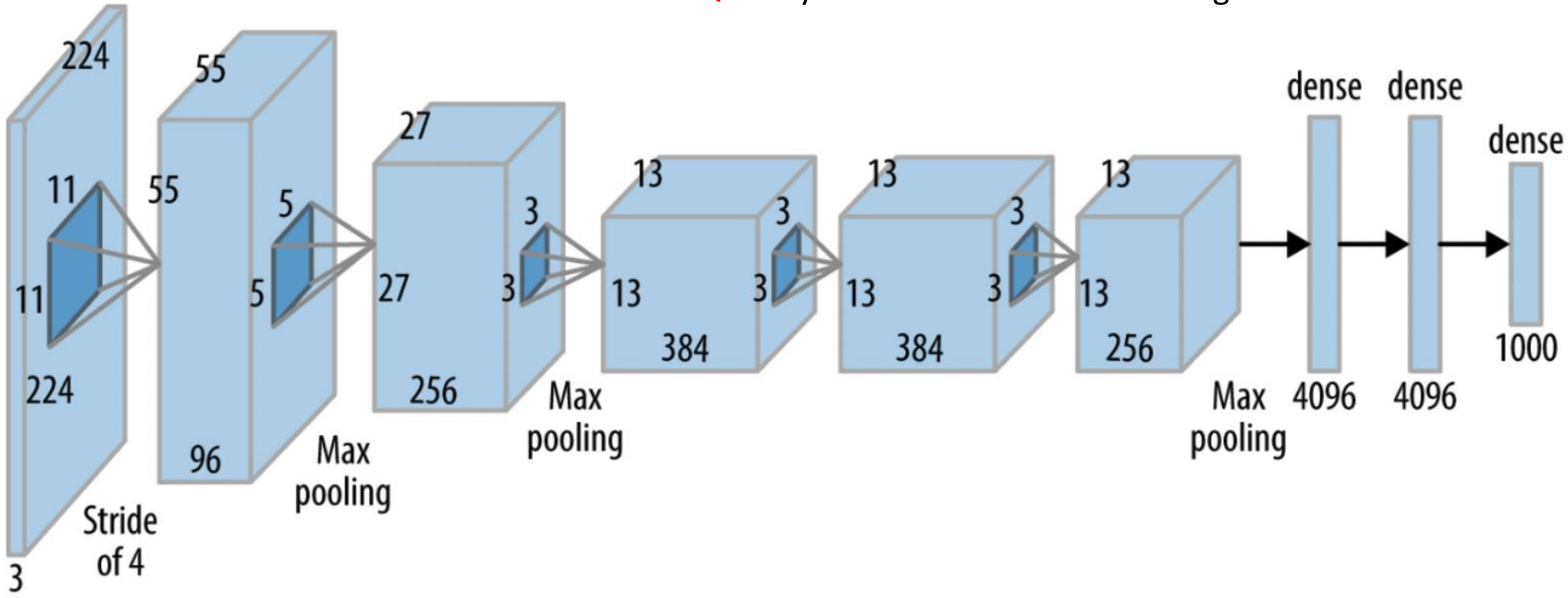
Input resolution issue



[Alexnet]

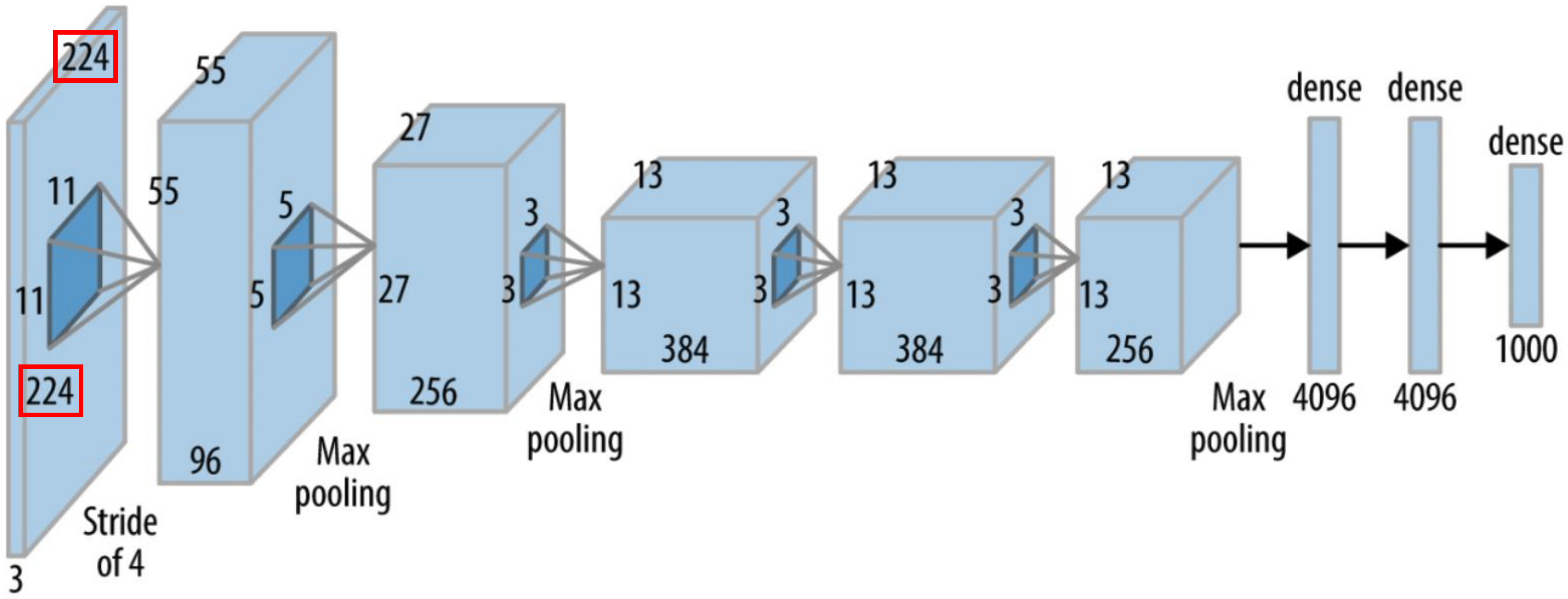
Input resolution issue

Q: can you understand the following architecture?



[Alexnet]

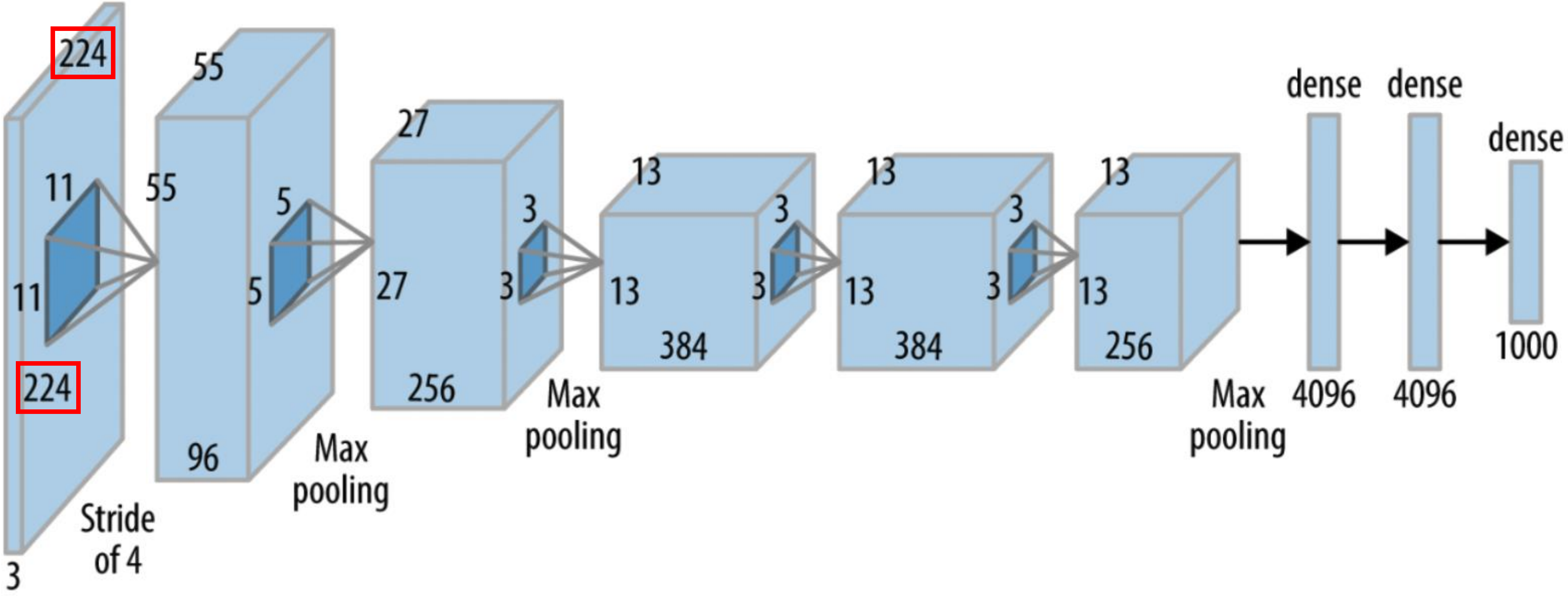
Input resolution issue



Any input image must be 224x224

[Alexnet]

Input resolution issue



Any input image must be 224x224

Q: how to handle an arbitrary resolution?

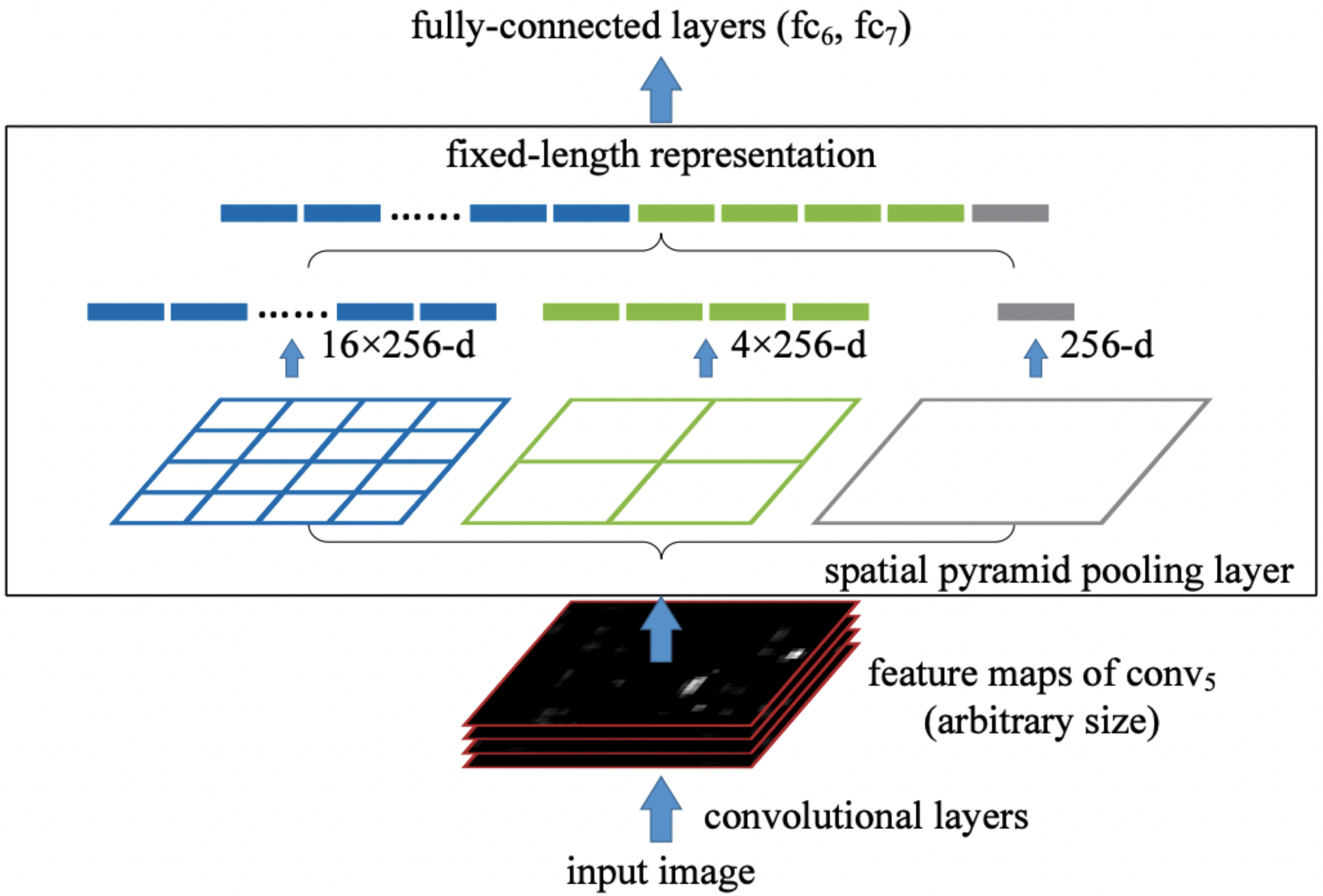
[Alexnet]

Input resolution issue

- Spatial pyramid pooling [pyramid]
- Global average pooling [NIN]
- ……

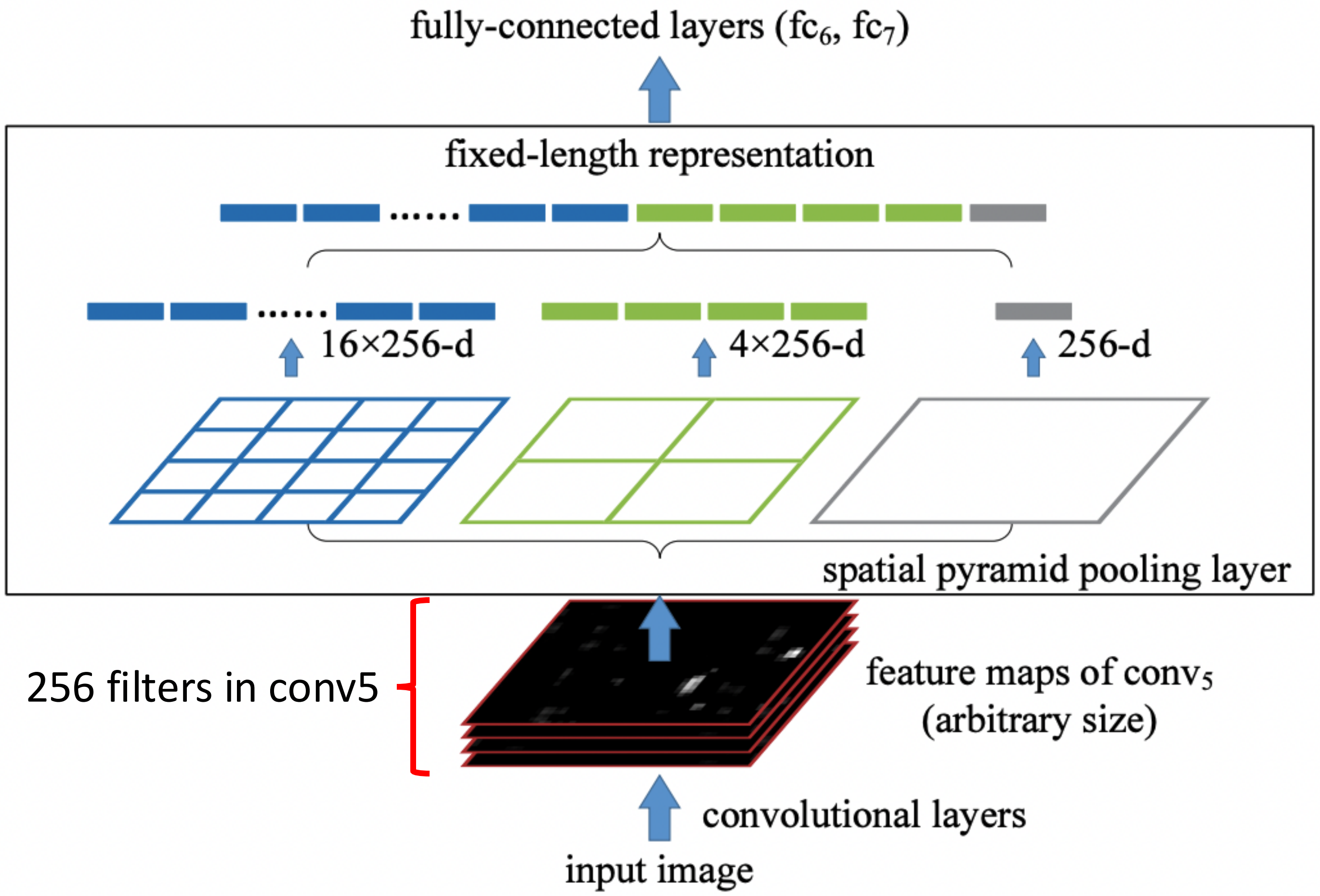
Input resolution issue

- Spatial pyramid pooling



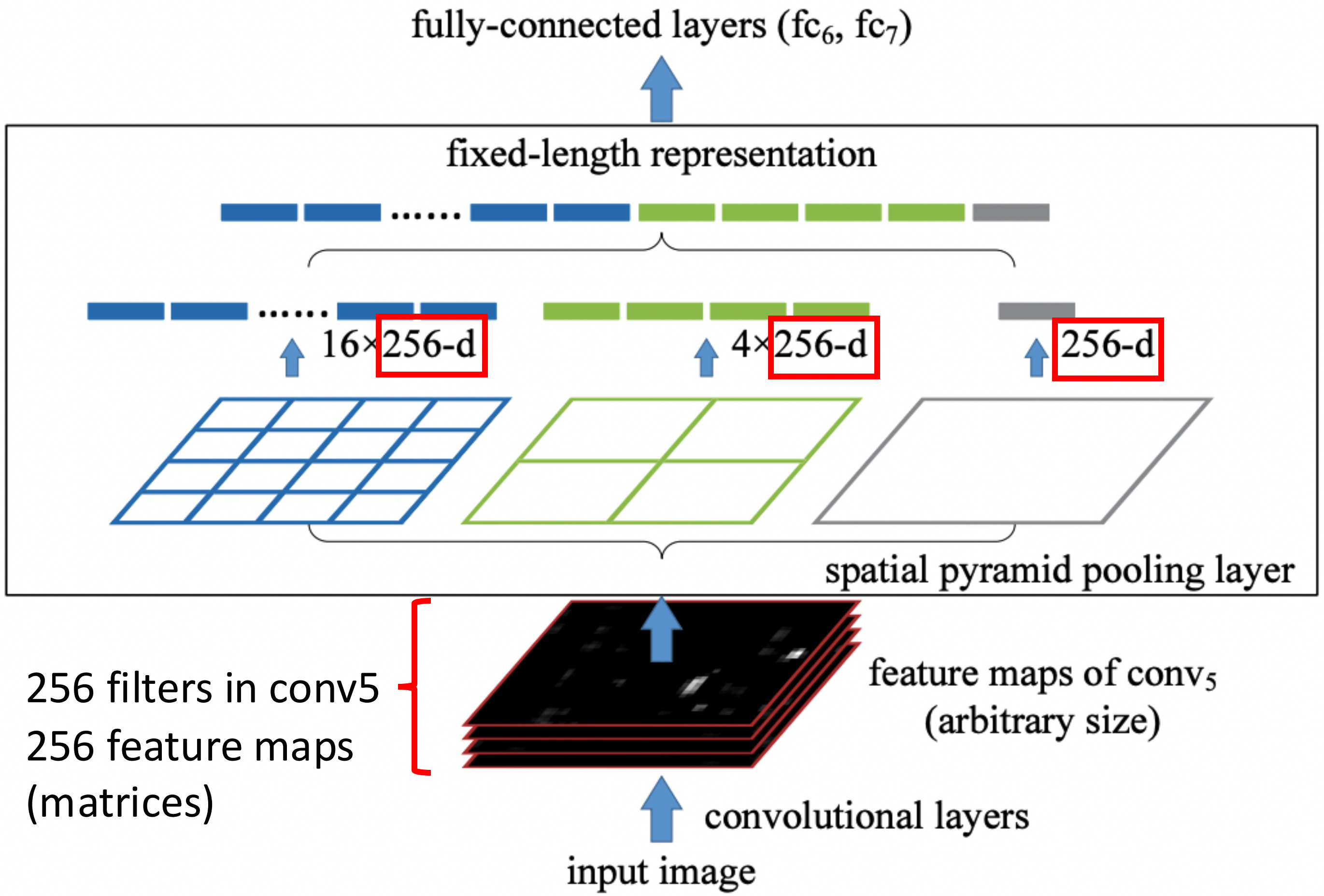
Input resolution issue

- Spatial pyramid pooling

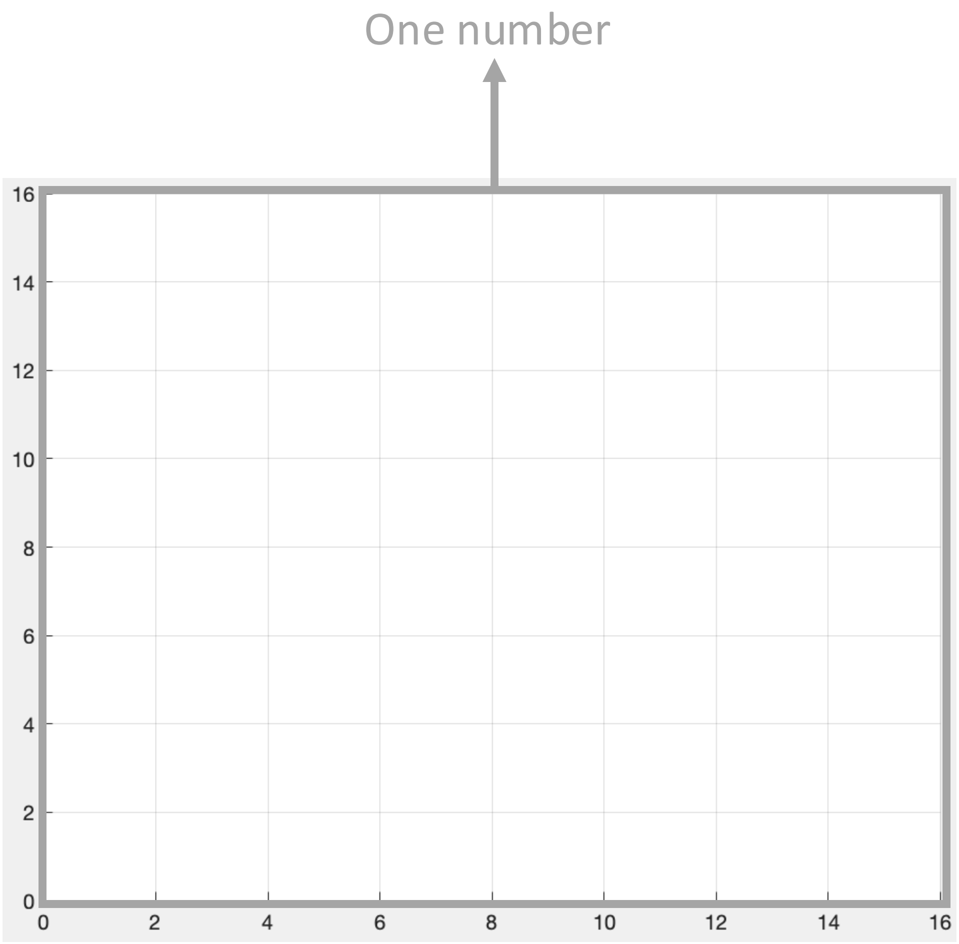


Input resolution issue

- Spatial pyramid pooling



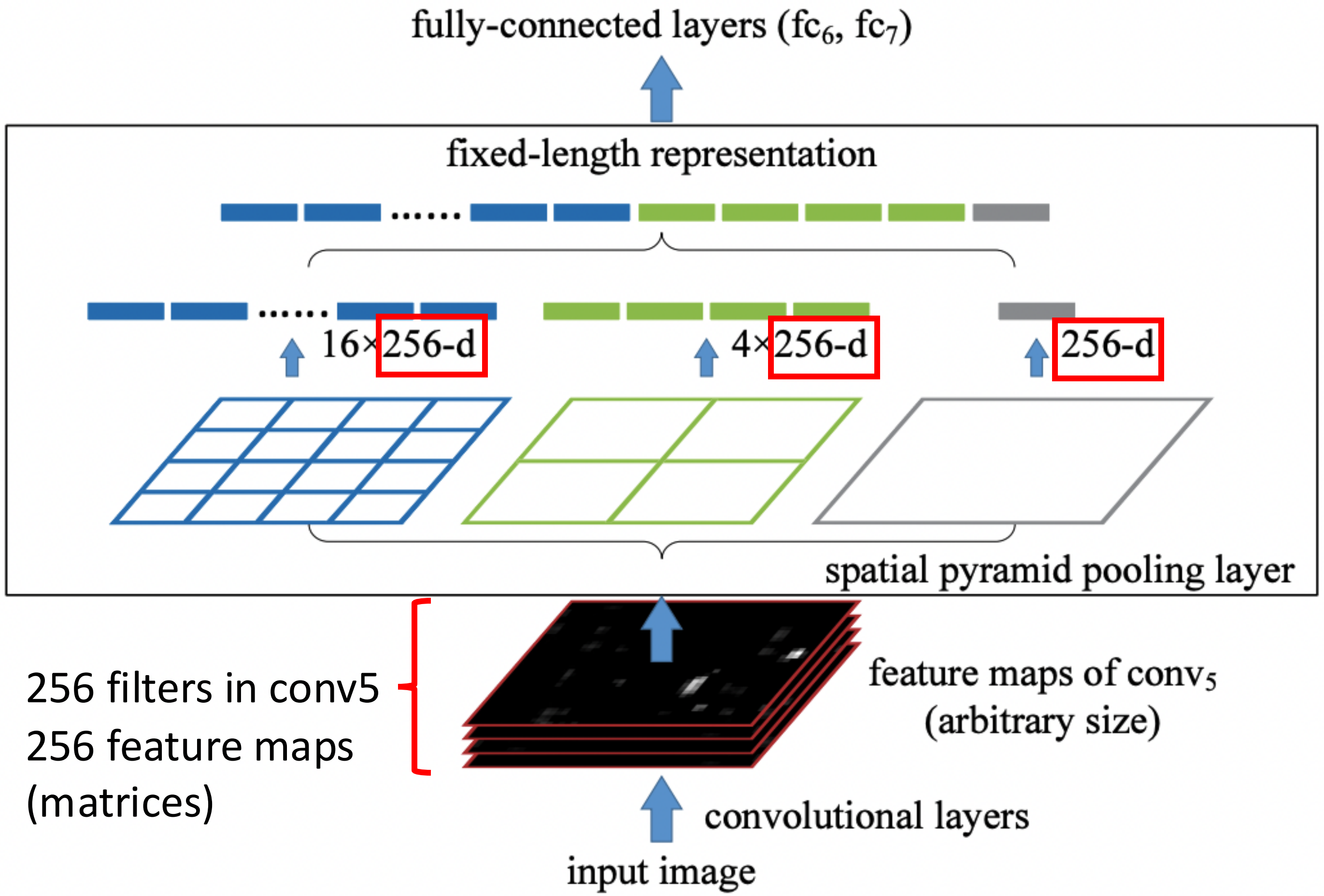
256 filters in conv5
256 feature maps
(matrices)



Some pooling (max/average)

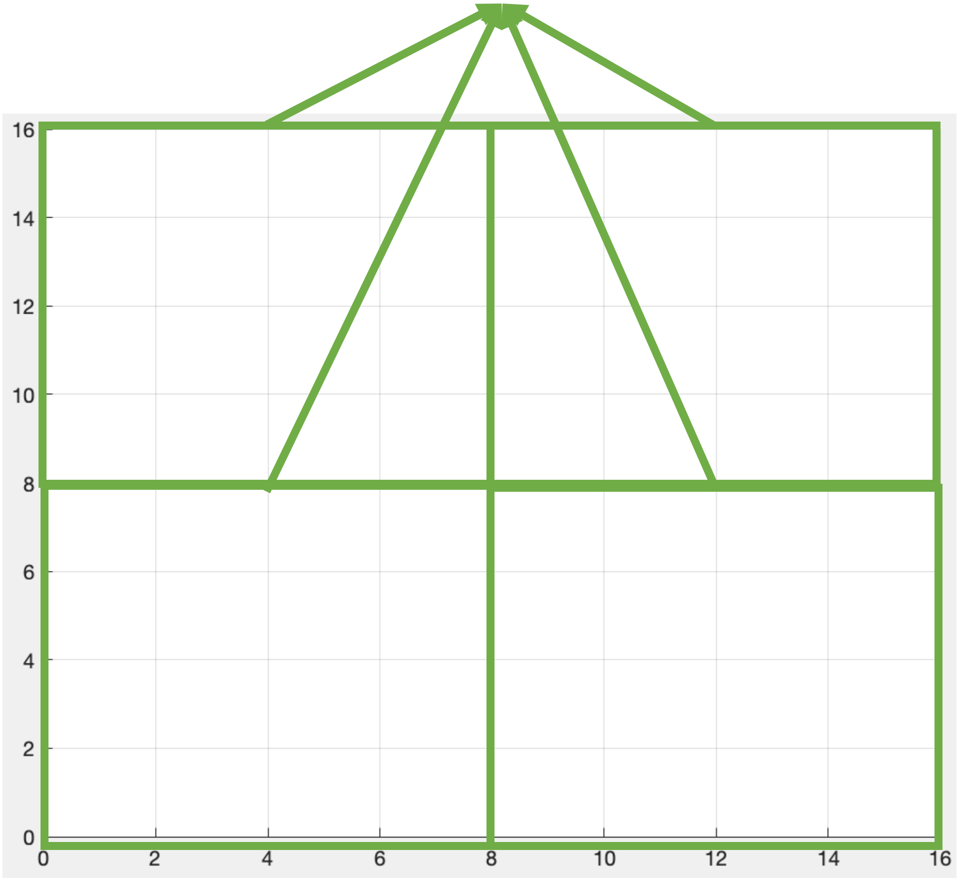
Input resolution issue

- Spatial pyramid pooling



256 filters in conv5
256 feature maps
(matrices)

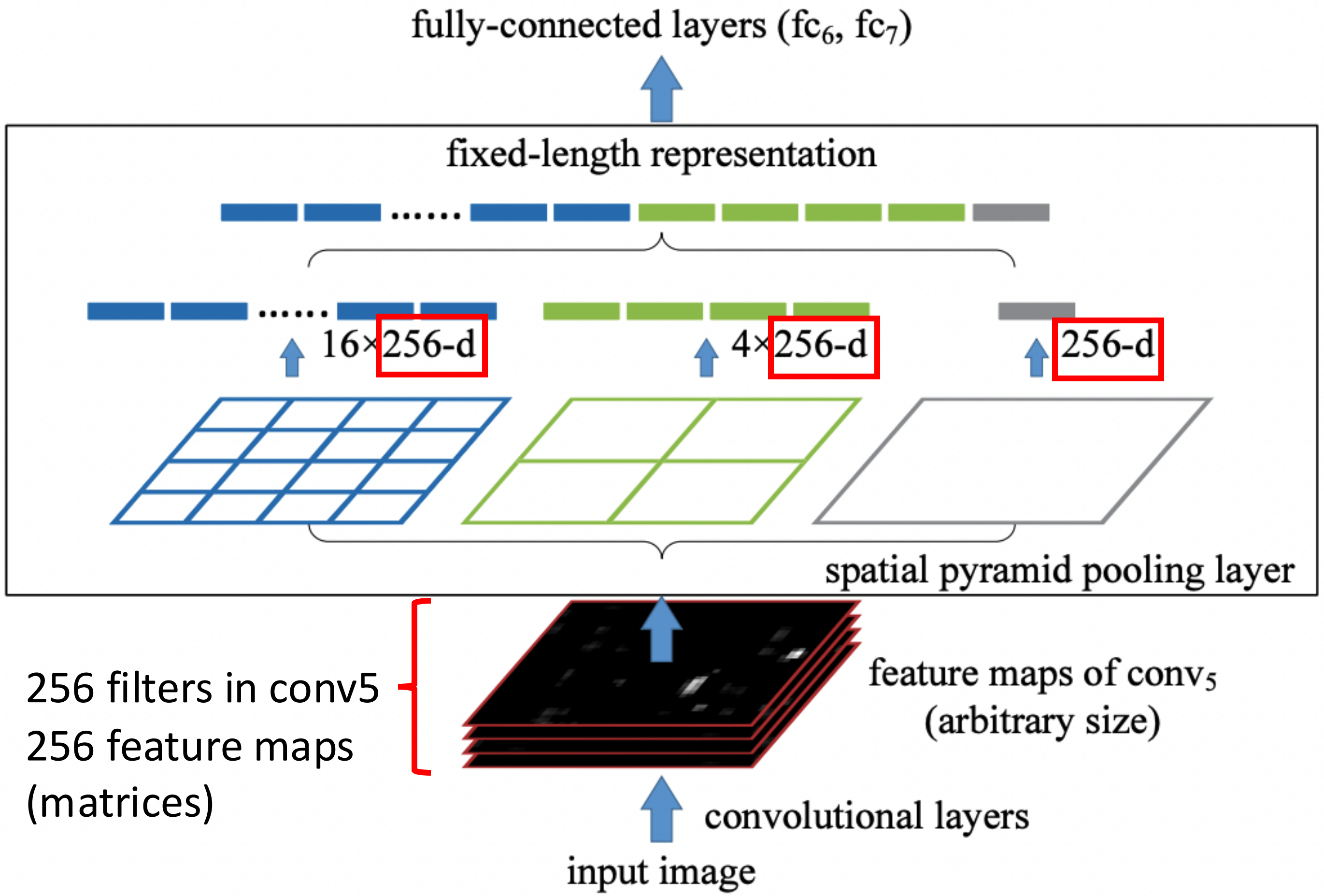
Four numbers



Some pooling (max/average)

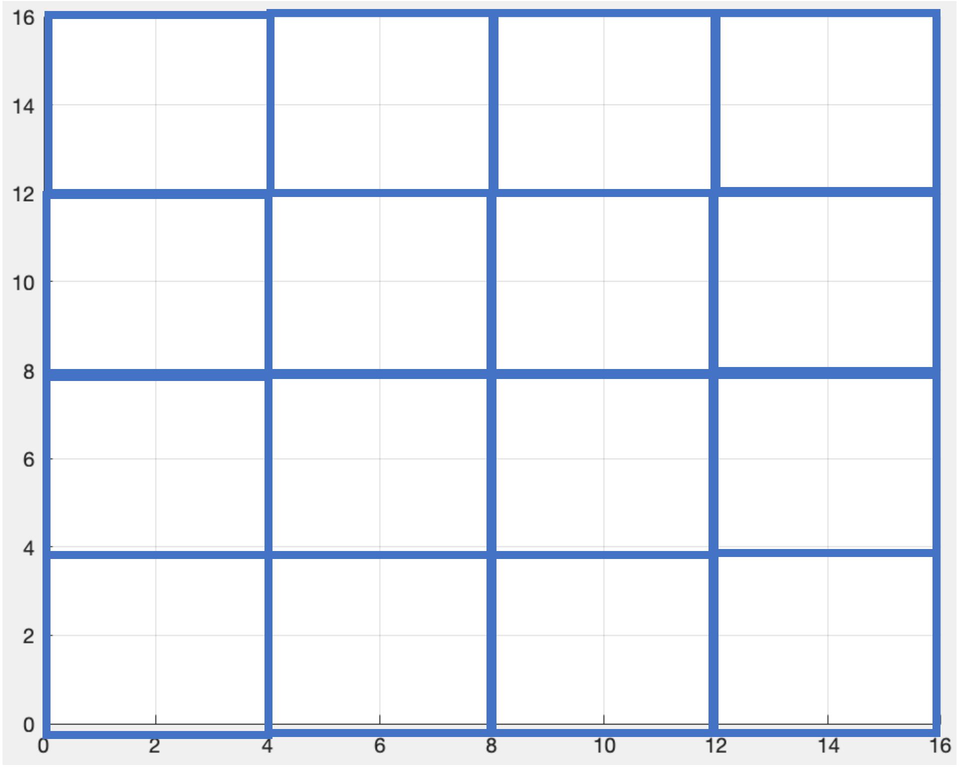
Input resolution issue

- Spatial pyramid pooling



256 filters in conv5
256 feature maps
(matrices)

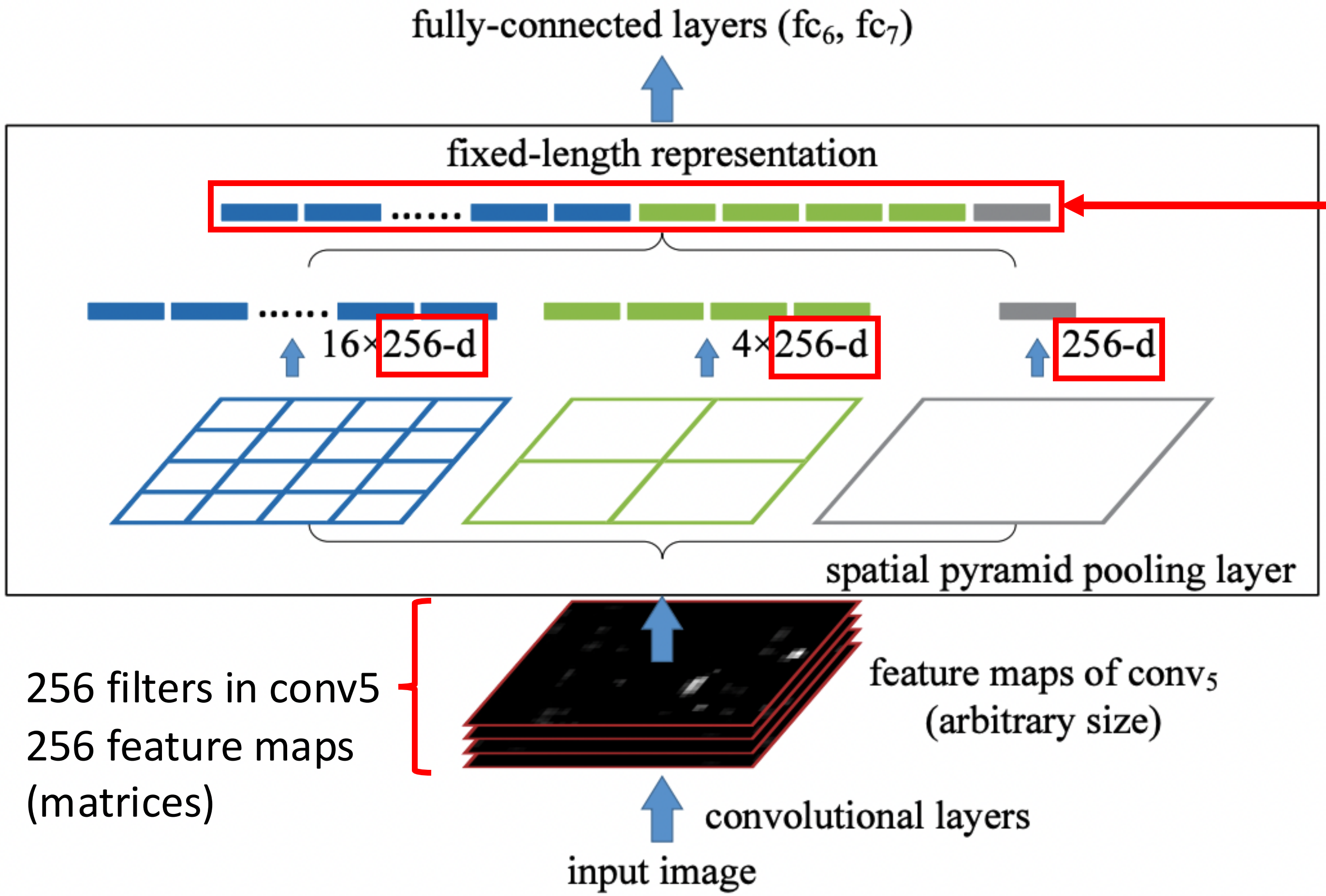
16 numbers



Some pooling (max/average)

Input resolution issue

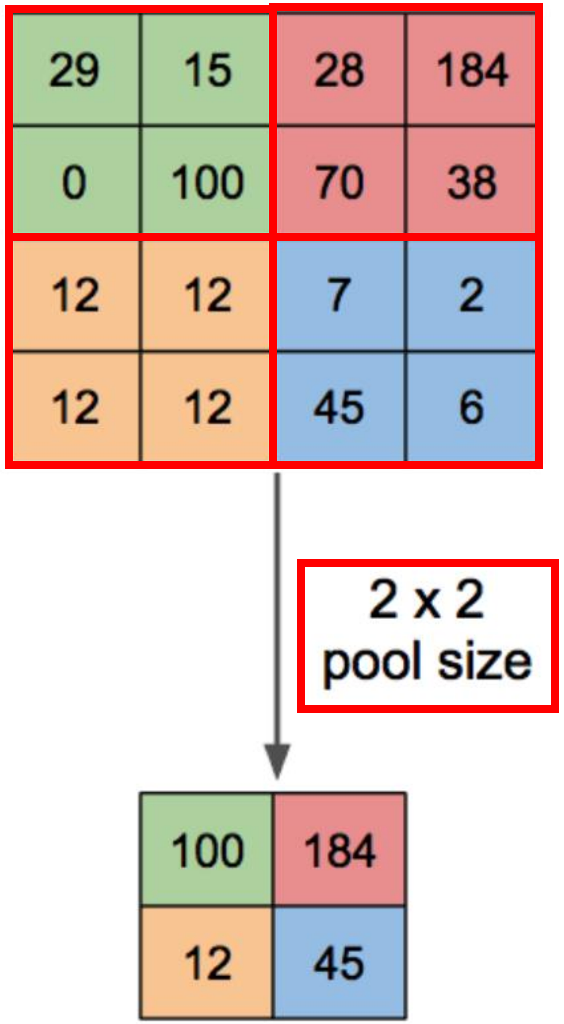
- Spatial pyramid pooling



Concatenation:
(1+4+16) x 256 numbers

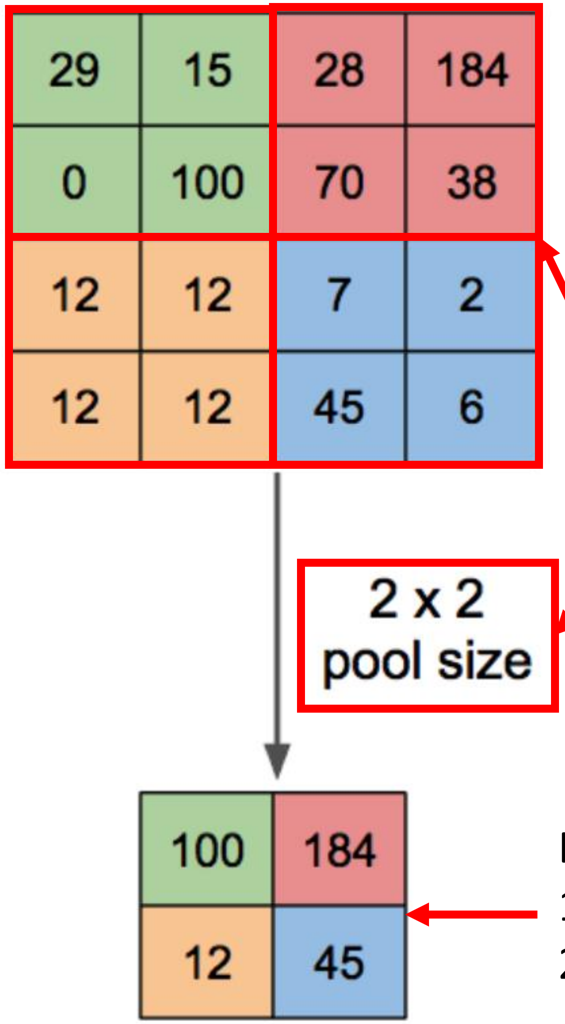
256 filters in conv5
256 feature maps
(matrices)

Input resolution issue



Max pooling

Input resolution issue



Max pooling

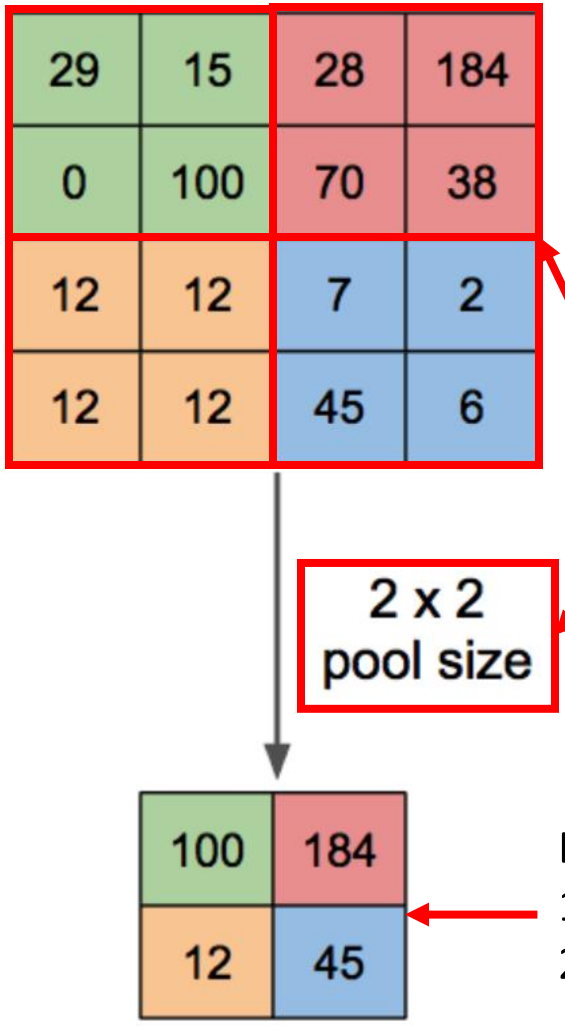
Input dimension

Predefined

Pool size

Depends on:
1. Previous feat map size
2. Pooling size

Input resolution issue



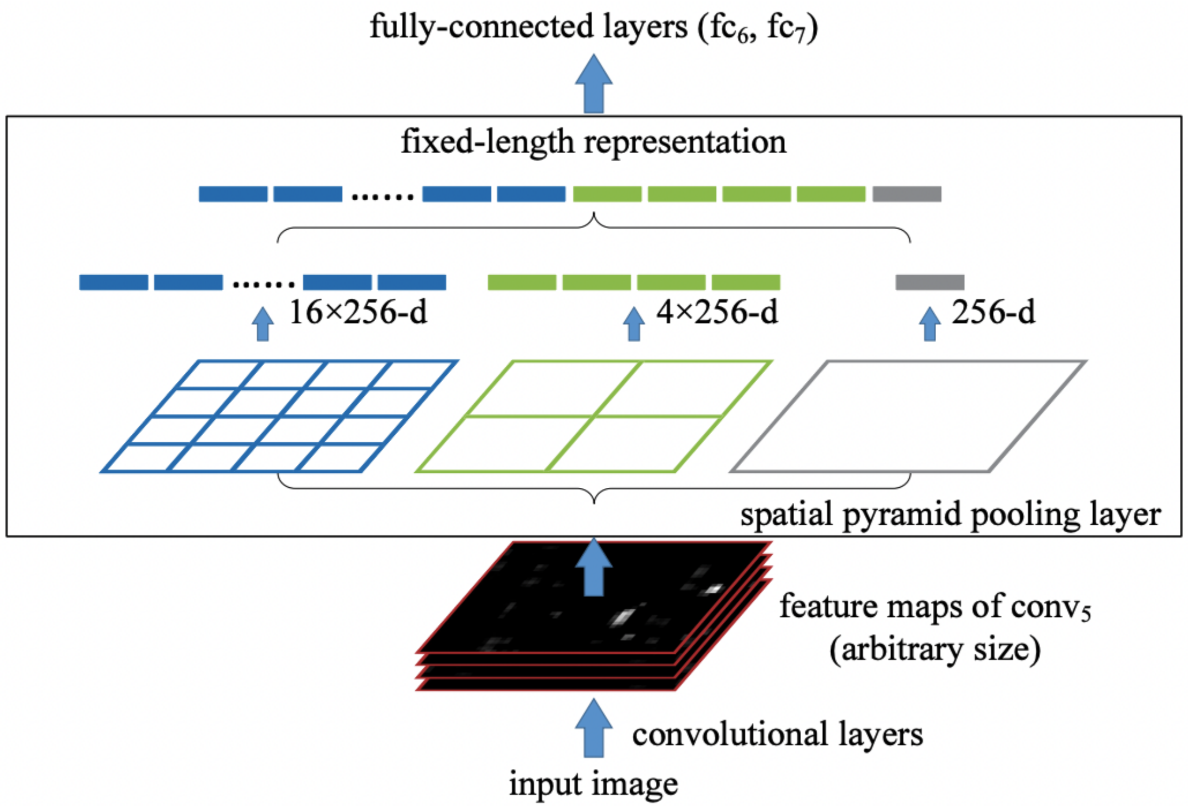
Input dimension

Predefined

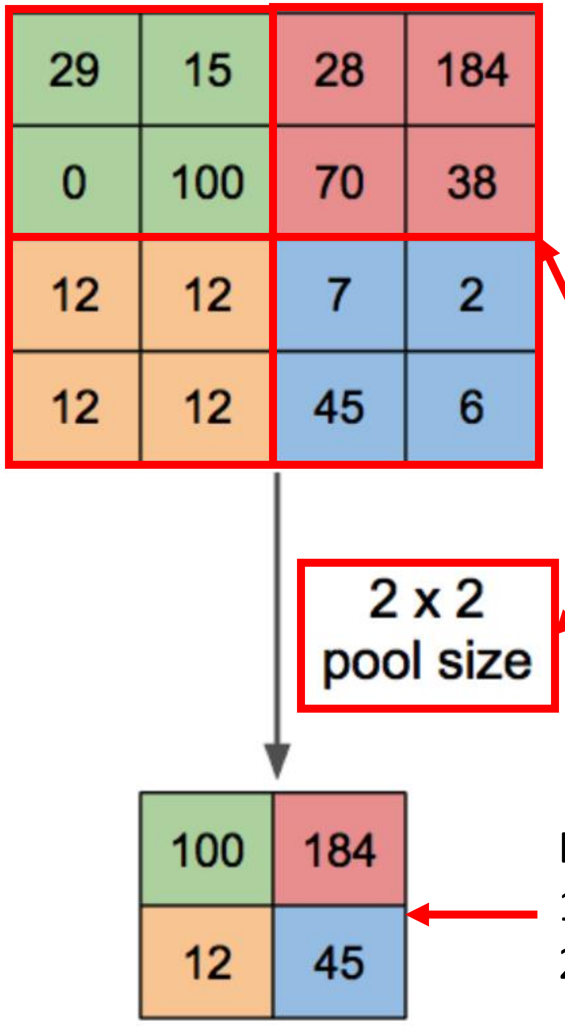
Pool size

Depends on:
1. Previous feat map size
2. Pooling size

Max pooling



Input resolution issue



Input dimension

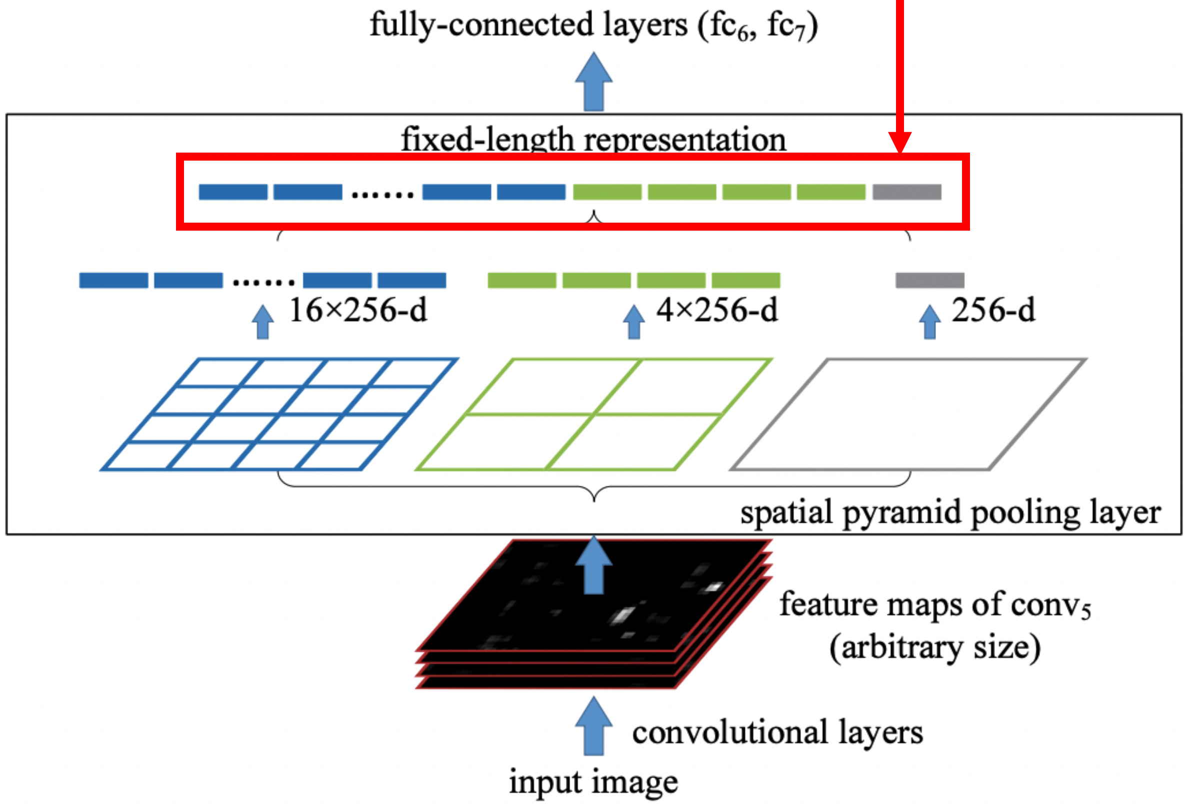
Predefined

Pool size

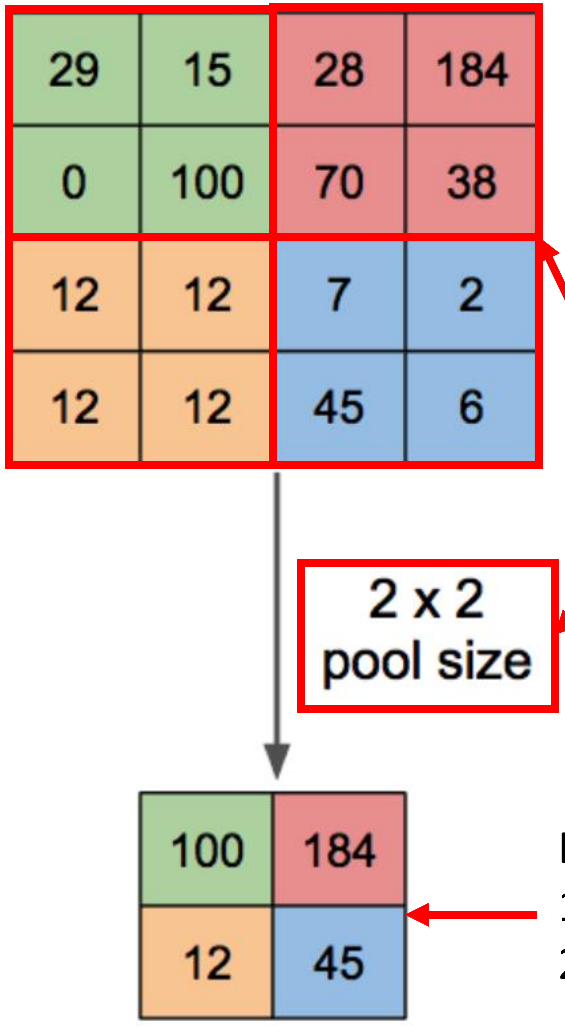
Depends on:
1. Previous feat map size
2. Pooling size

Max pooling

Concatenation:
(1+4+16) x 256 numbers



Input resolution issue



Input dimension

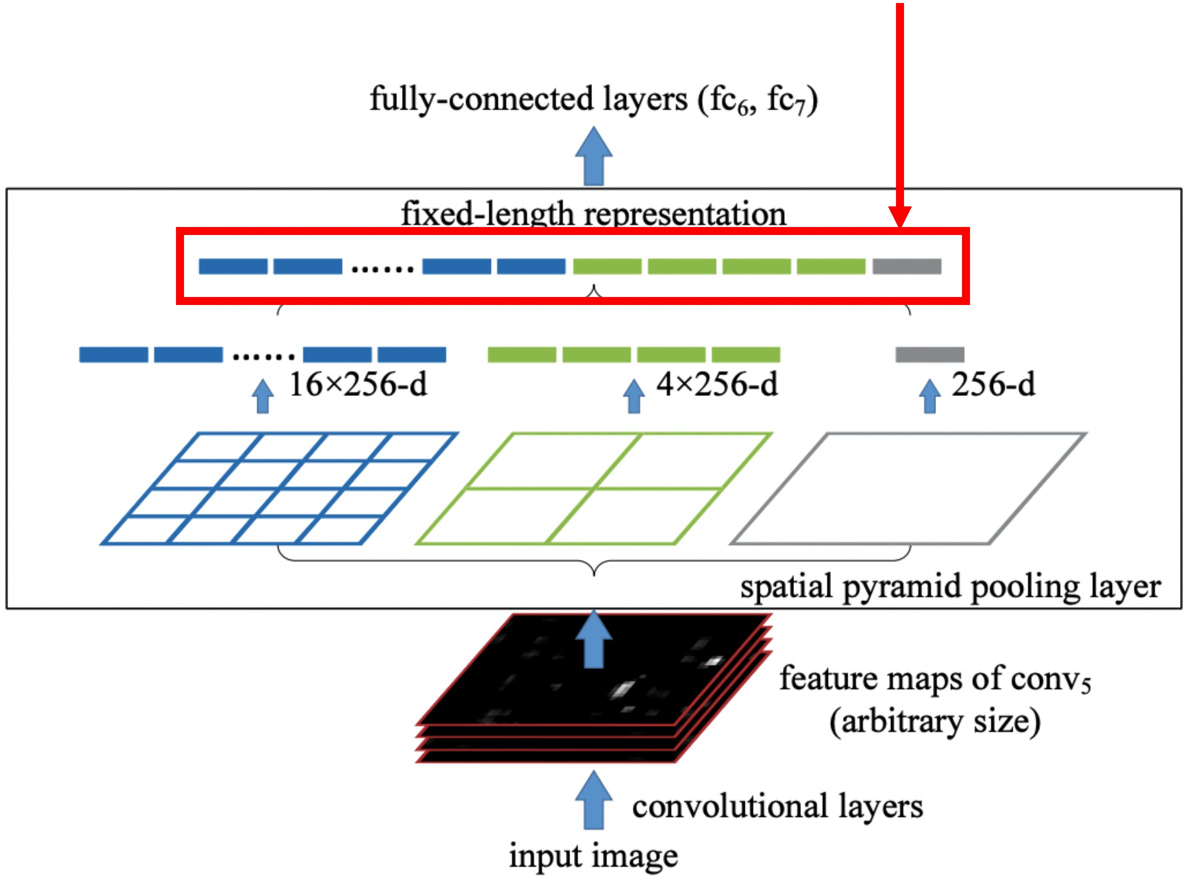
Predefined

Pool size

Depends on:
1. Previous feat map size
2. Pooling size

Max pooling

Concatenation:
(1+4+16) x 256 numbers



Arbitrary size

Input resolution issue

- Global average pooling

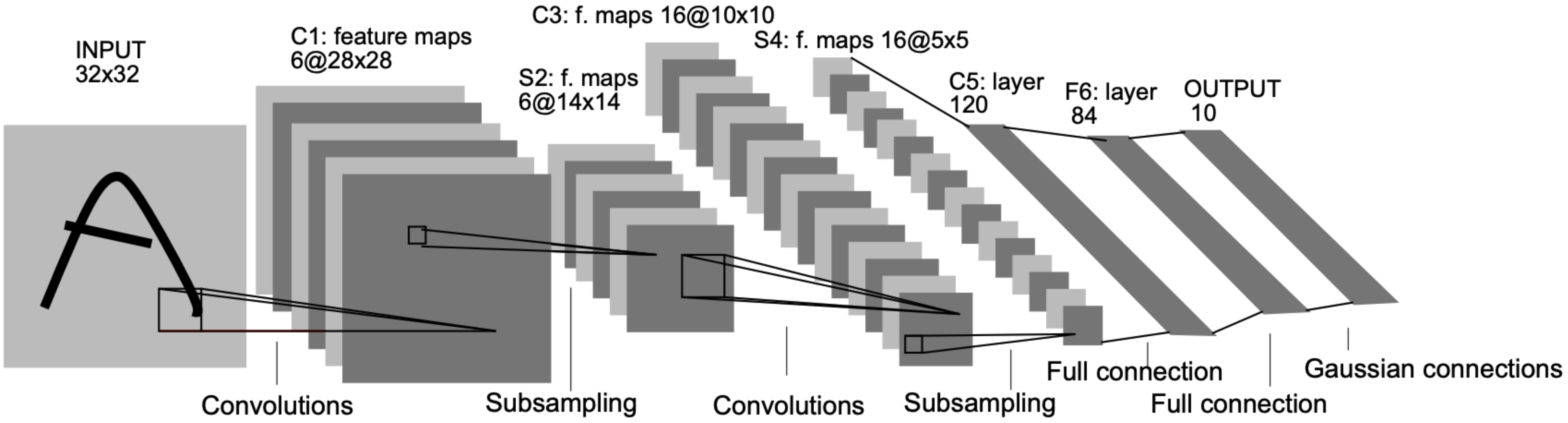


Fig. 1. Architecture of LeNet-5, a Convolutional Neural Network, here for digits recognition. Each plane is a feature map, i.e. a set of units whose weights are constrained to be identical.

Input resolution issue

- Global average pooling

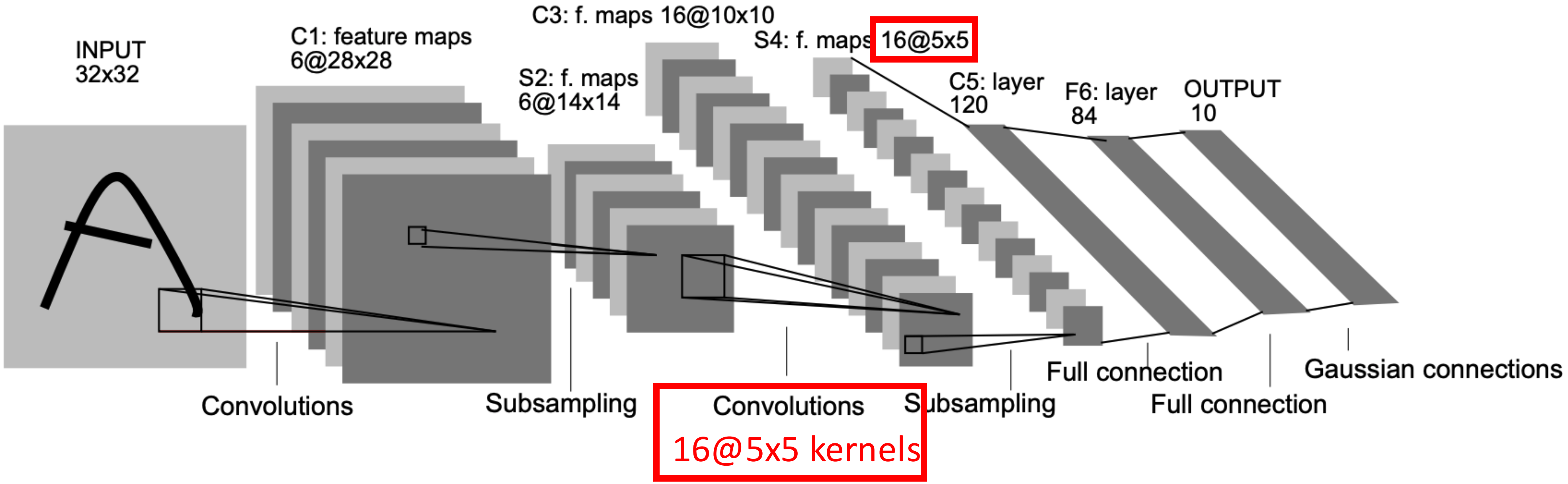


Fig. 1. Architecture of LeNet-5, a Convolutional Neural Network, here for digits recognition. Each plane is a feature map, i.e. a set of units whose weights are constrained to be identical.

Input resolution issue

- Global average pooling

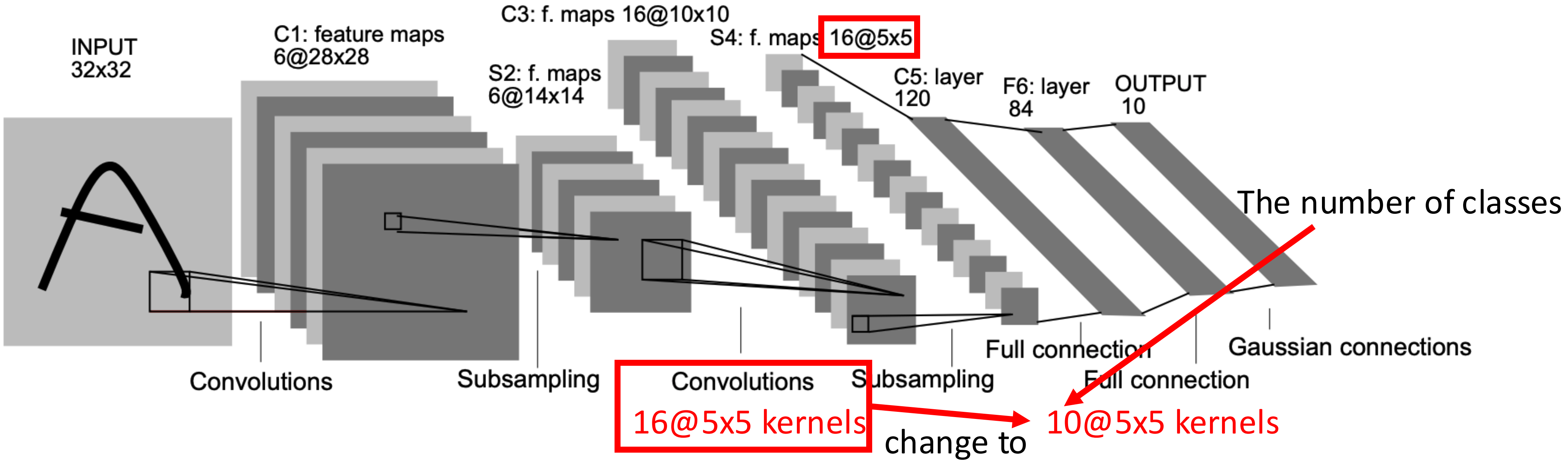


Fig. 1. Architecture of LeNet-5, a Convolutional Neural Network, here for digits recognition. Each plane is a feature map, i.e. a set of units whose weights are constrained to be identical.

Input resolution issue

- Global average pooling

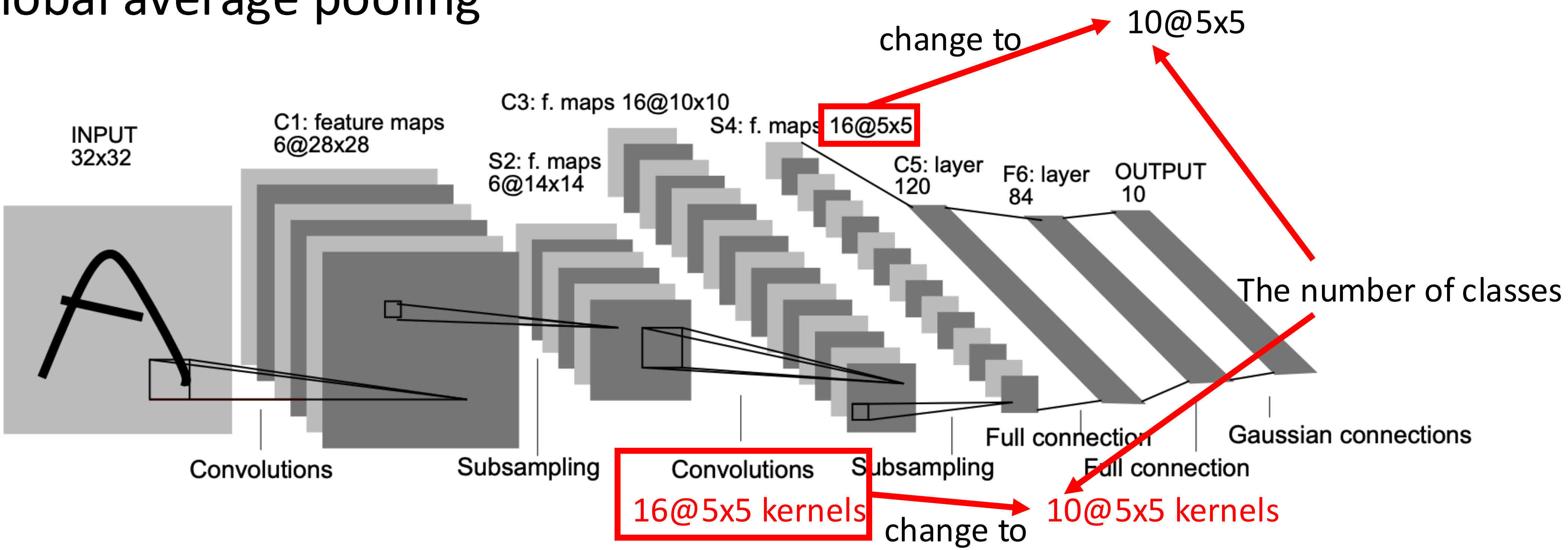


Fig. 1. Architecture of LeNet-5, a Convolutional Neural Network, here for digits recognition. Each plane is a feature map, i.e. a set of units whose weights are constrained to be identical.

Input resolution issue

- Global average pooling

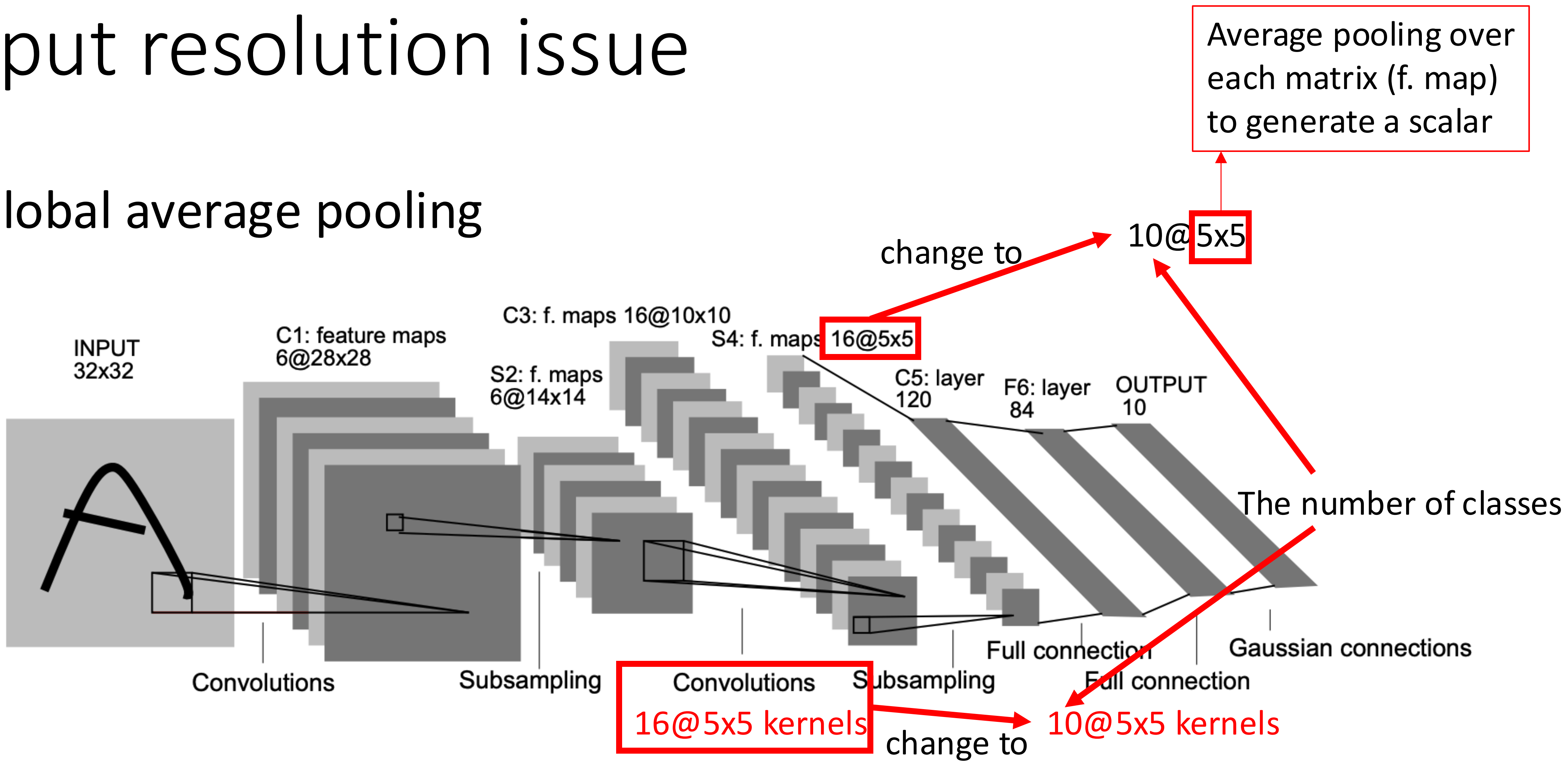


Fig. 1. Architecture of LeNet-5, a Convolutional Neural Network, here for digits recognition. Each plane is a feature map, i.e. a set of units whose weights are constrained to be identical.

Input resolution issue

- Global average pooling

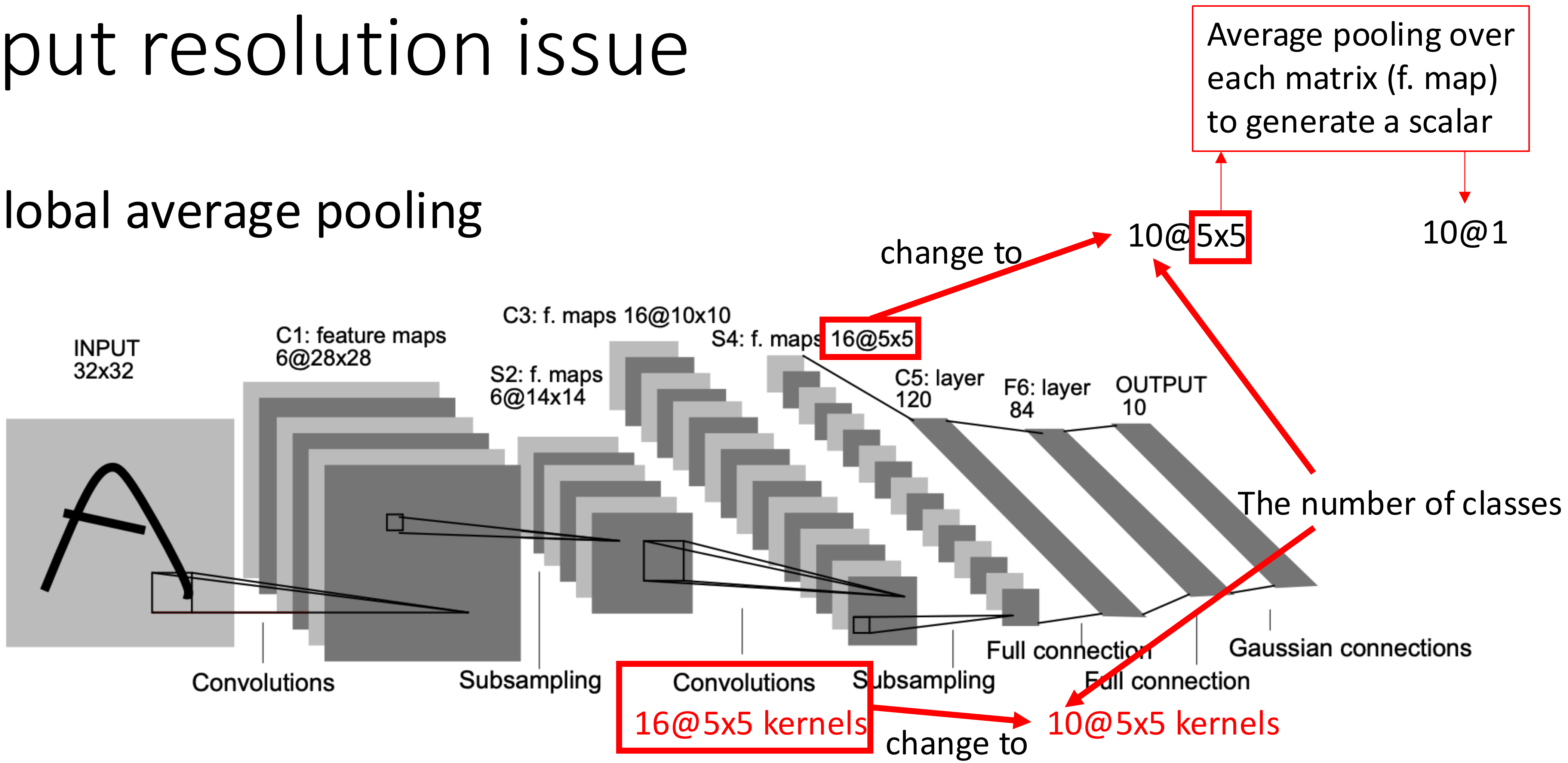


Fig. 1. Architecture of LeNet-5, a Convolutional Neural Network, here for digits recognition. Each plane is a feature map, i.e. a set of units whose weights are constrained to be identical.

Input resolution issue

- Global average pooling

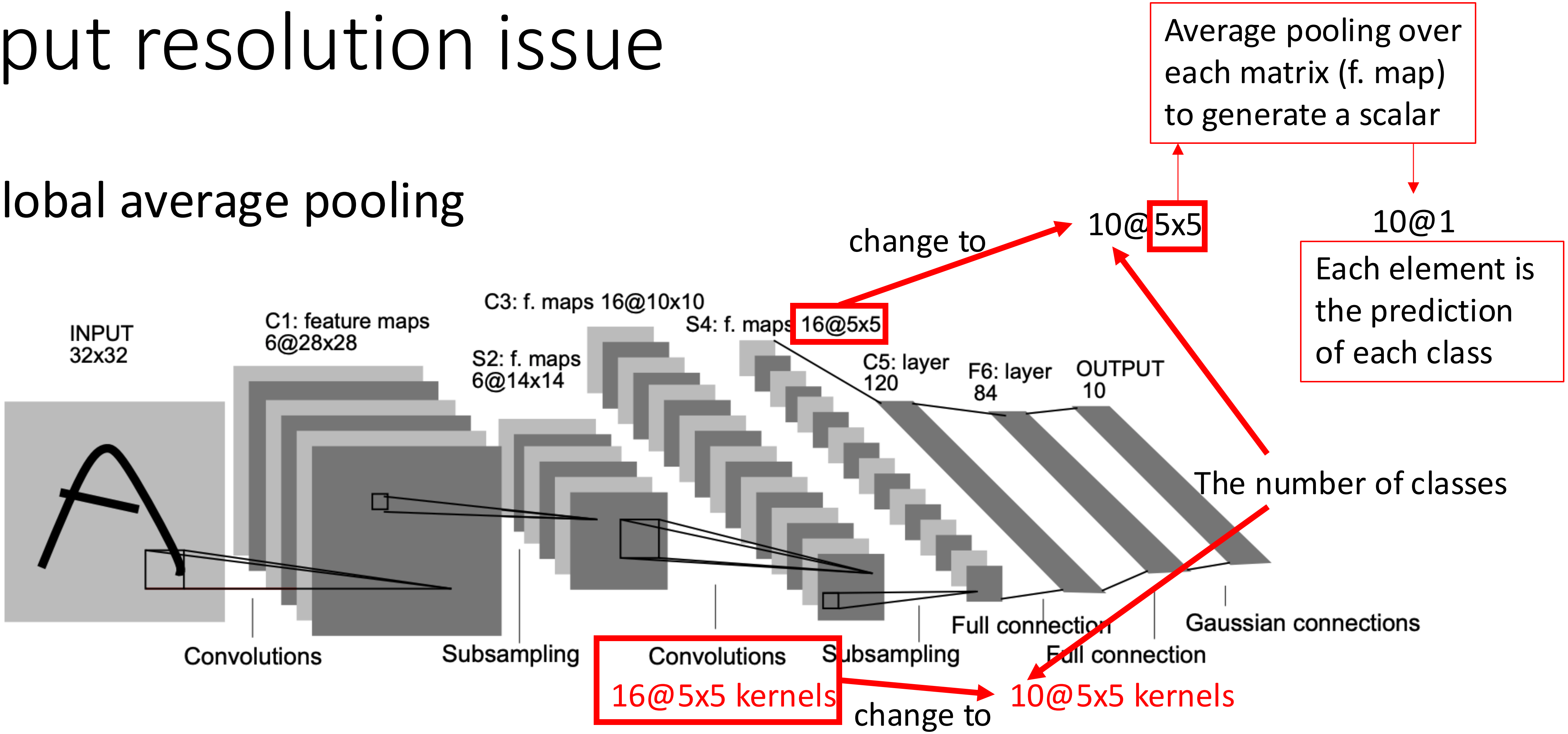


Fig. 1. Architecture of LeNet-5, a Convolutional Neural Network, here for digits recognition. Each plane is a feature map, i.e. a set of units whose weights are constrained to be identical.

References

- LeCun, Yann, Patrick Haffner, Léon Bottou, and Yoshua Bengio. "Object recognition with gradient-based learning." In *Shape, contour and grouping in computer vision*, pp. 319-345. Springer, Berlin, Heidelberg, 1999.
  - Online at http://yann.lecun.com/exdb/publis/pdf/lecun-99.pdf
  - Section 2.2
  - Understand architecture of LeNet-5
- LeCun, Yann, Léon Bottou, Yoshua Bengio, and Patrick Haffner. "Gradient-based learning applied to document recognition." *Proceedings of the IEEE* 86, no. 11 (1998): 2278-2324.
  - Online at http://vision.stanford.edu/cs598_spring07/papers/Lecun98.pdf
  - Section II.B

References

- [Alexnet] Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." *Advances in neural information processing systems* 25 (2012): 1097-1105. Conference proceeding version at https://papers.nips.cc/paper/2012/hash/c399862d3b9d6b76c8436e924a68c45b-Abstract.html or https://papers.nips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf (Section 3.5)
- [pyramid] He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Spatial pyramid pooling in deep convolutional networks for visual recognition." *IEEE transactions on pattern analysis and machine intelligence* 37, no. 9 (2015): 1904-1916. ArXiv version at https://arxiv.org/abs/1406.4729 (Section 2.2)
- [NIN] Lin, Min, Qiang Chen, and Shuicheng Yan. "Network in network." *arXiv preprint arXiv:1312.4400* (2013). ArXiv version at https://arxiv.org/abs/1312.4400 (Section 3.2)