Convolutional Neural Networks

Neural Networks Design And Application

LeNet-5 in 1999

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

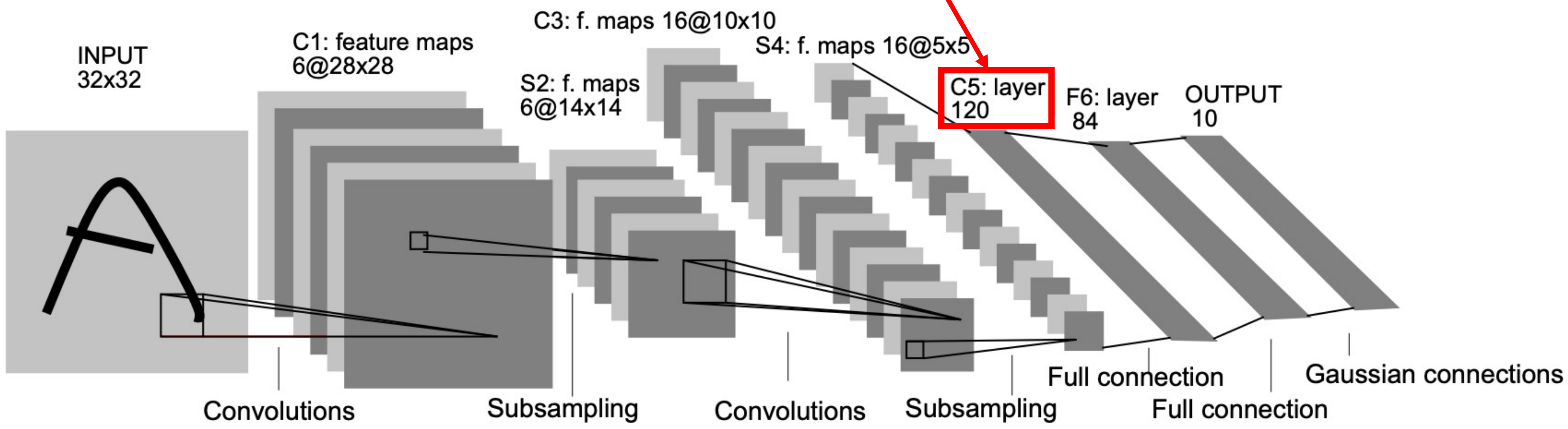
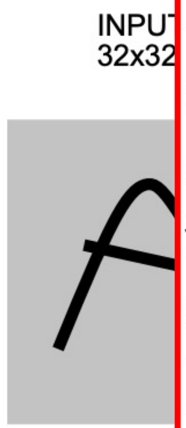


Fig. 1. Architecture of LeNet-5, a Convolutional Neural Network, here for digits recognition. Each plane is a feature map, i.e. a set of units whose weights are constrained to be identical.

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?



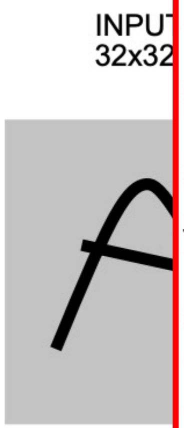
Layer C5 is a convolutional layer with 120 feature maps. Each unit is connected to a 5x5 neighborhood on all 16 of S4's feature maps. Here, because the size of S4 is also 5x5, the size of C5's feature maps is 1x1: this amounts to a full connection between S4 and C5. C5 is labeled as a convolutional layer, 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.

onnections

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.

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 (matrices)

$$\begin{array}{ccccc} \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 \end{bmatrix} & * & \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 \end{bmatrix} & \rightarrow & \begin{bmatrix} 4 & 3 & 4 \\ 2 & 4 & 3 \\ 2 & 3 & 4 \end{bmatrix} \\ n=5 & & m=3 & & n-m+1=3 \end{array}$$

One matrix — One matrix — One matrix

One input matrix * one filter → one feature matrix

Convolution for images (matrices)

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

n=5

One 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

m=**5**

One matrix

→

4	3	4
2	4	3
2	3	4

n-m+1=3

One matrix

One input matrix * one filter → one feature matrix

Convolution for images (matrices)

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

n=5 | m=**5** | n-m+1=**1**

One matrix | One matrix | One matrix

One input matrix * one filter → one feature matrix

Convolution for images (matrices)

Step 1: finding pairs

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

n=5

One 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

m=**5**

One matrix

→ ???

n-m+1=**1**

One matrix

One input matrix * one filter → one feature matrix

Convolution for images (matrices)

Step 1: finding pairs
1 pair

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

n=5

One 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

m=**5**

One matrix

→

???

n-m+1=**1**

One matrix

One input matrix * one filter → one feature matrix

Convolution for images (matrices)

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

n=5

One matrix

m=**5**

One matrix

n-m+1=**1**

One matrix

Step 1: finding pairs

1 pair

Step 2: elementwise summation

One input matrix * one filter → one feature matrix

Convolution for images (matrices)

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

n=5 | m=**5** | n-m+1=**1**

One matrix | One matrix | One matrix

Step 1: finding pairs
1 pair
Step 2: elementwise summation

One input matrix * one filter → one feature matrix

Convolution for images (matrices)

Step 1: finding pairs
1 pair
Step 2: elementwise summation

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 \end{bmatrix} * \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 \end{bmatrix} \rightarrow \mathbf{13}$$

n=5 | m=**5** | n-m+1=**1**

One matrix | One matrix | One matrix

One input matrix * one filter → one feature matrix

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

$$
\begin{array}{c}
\begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 \end{bmatrix} \\ n=5
\end{array}
\;*\;
\begin{array}{c}
\begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} \\ m=3
\end{array}
\;\rightarrow\;
\begin{array}{c}
\begin{bmatrix} 4 & 3 & 4 \\ 2 & 4 & 3 \\ 2 & 3 & 4 \end{bmatrix} \\ n-m+1=3
\end{array}
$$

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

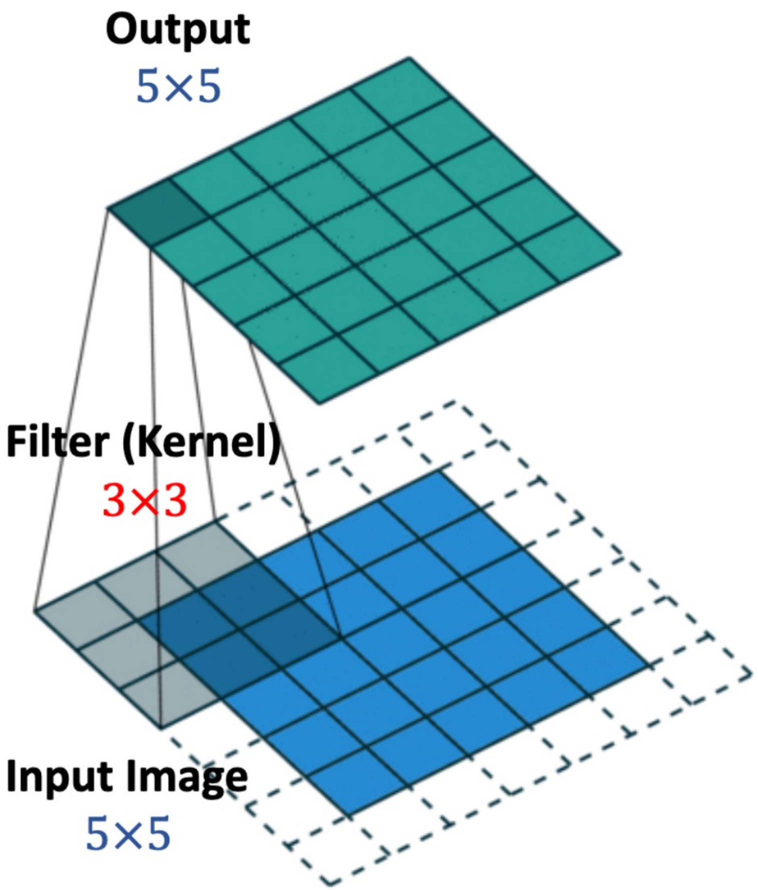
$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 \end{bmatrix} * \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} \rightarrow \begin{bmatrix} 4 & 3 & 4 \\ 2 & 4 & 3 \\ 2 & 3 & 4 \end{bmatrix}$$

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

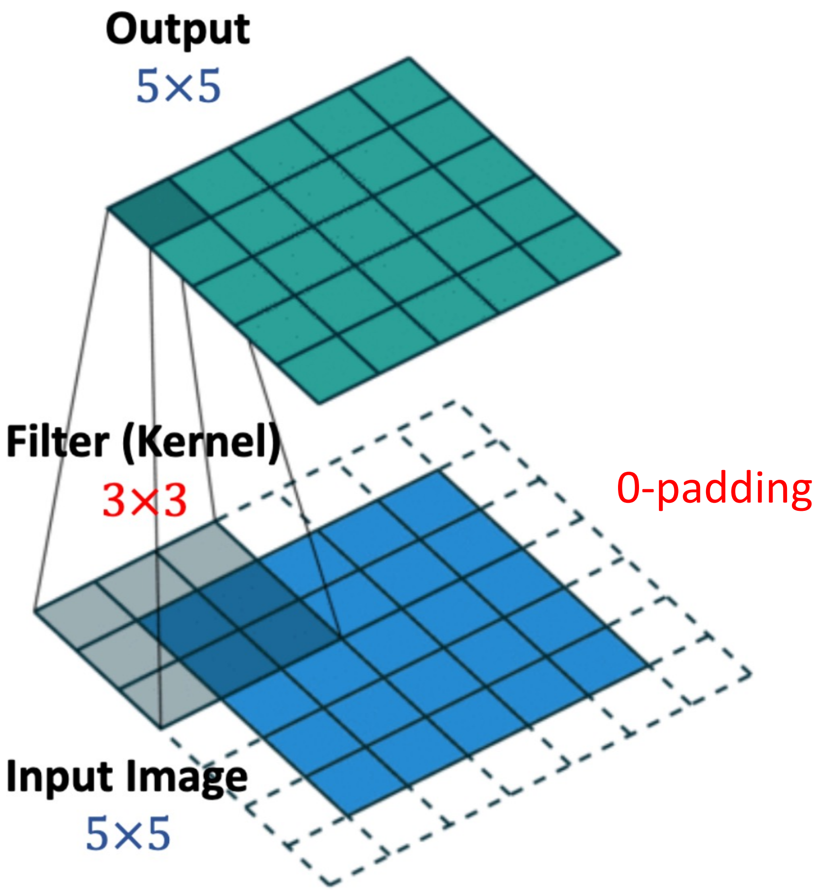
Operations with convolution layers

- Padding: convolution operation reduces the size of feature maps



Operations with convolution layers

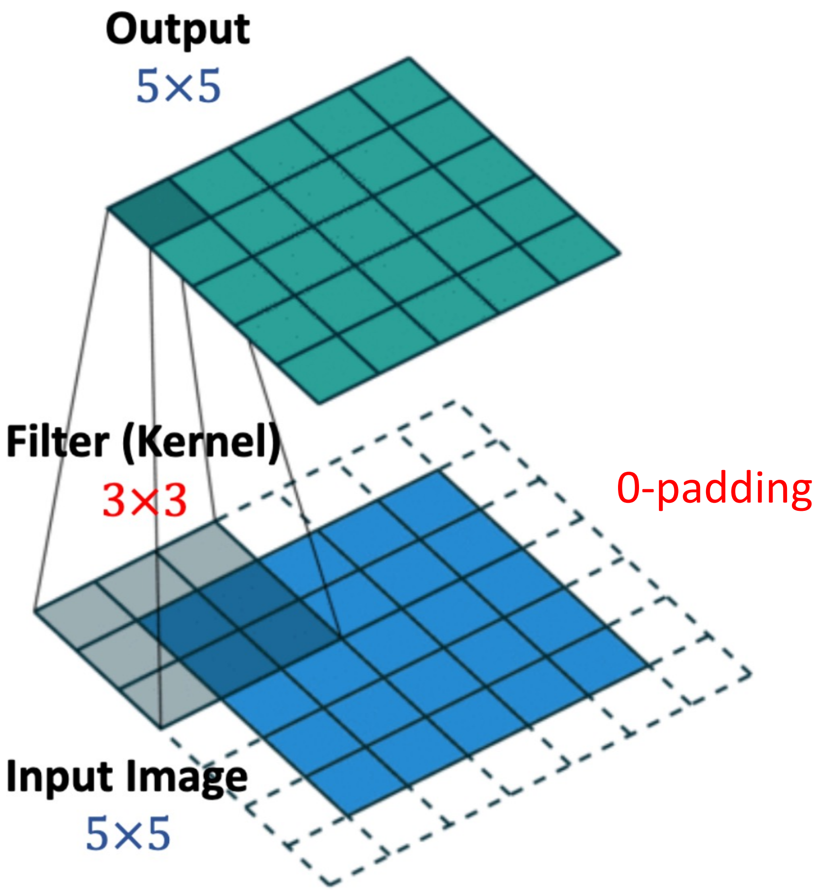
- Padding: convolution operation reduces the size of feature maps



Operations with convolution layers

- Padding: convolution operation reduces the size of feature maps

Input size
n→7



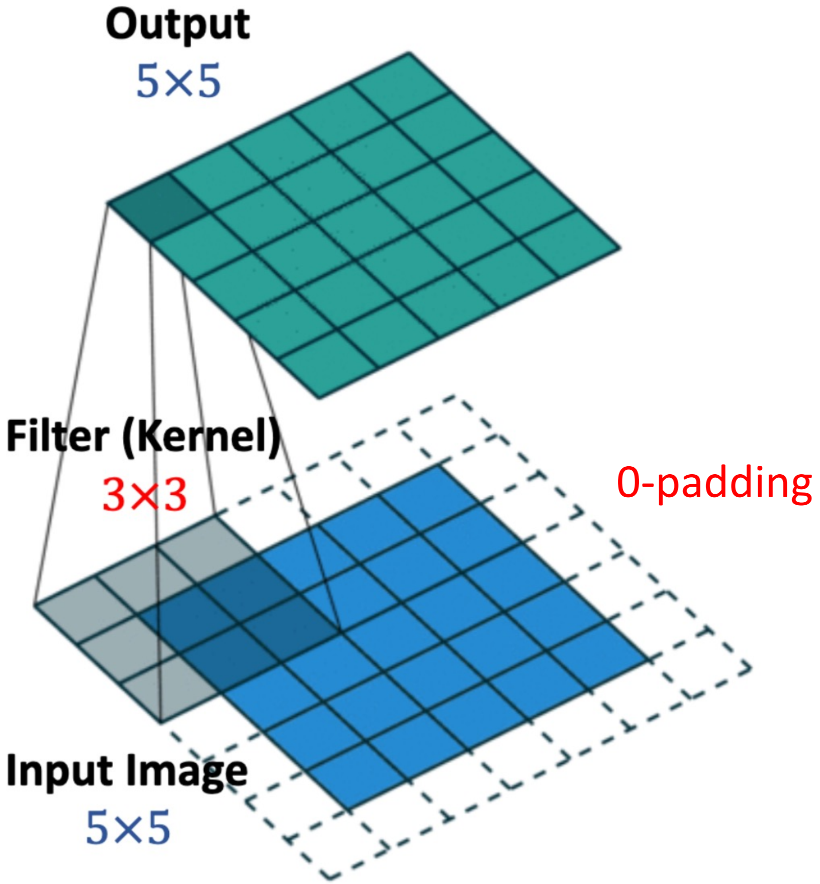
Operations with convolution layers

- Padding: convolution operation reduces the size of feature maps

Input size

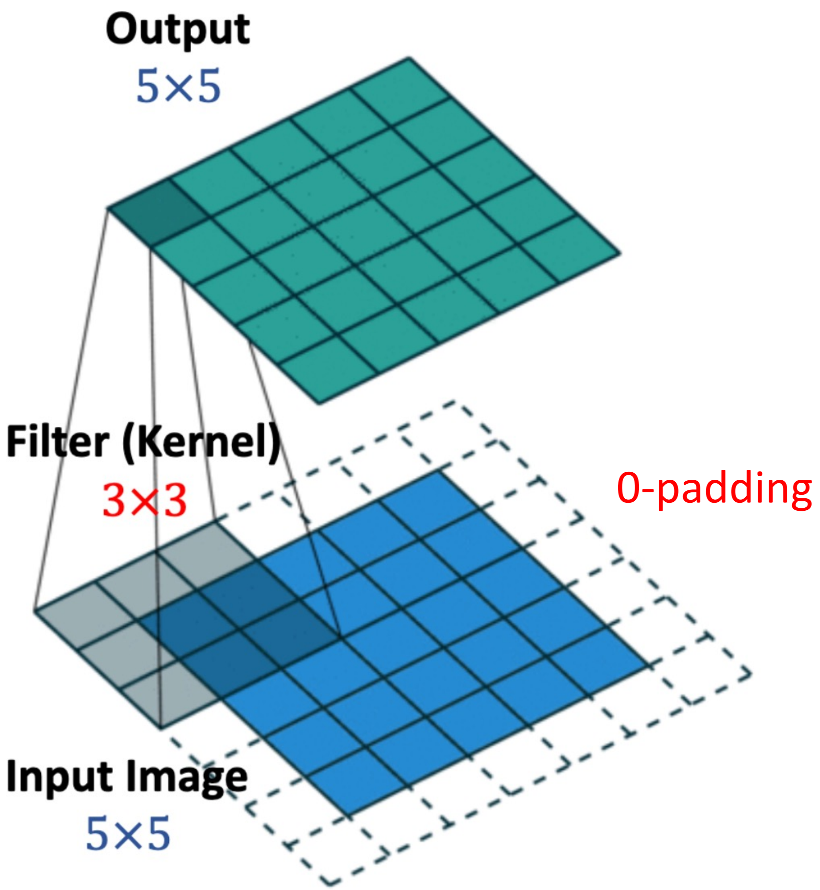
n→7 → n-m+1=7-3+1=5

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

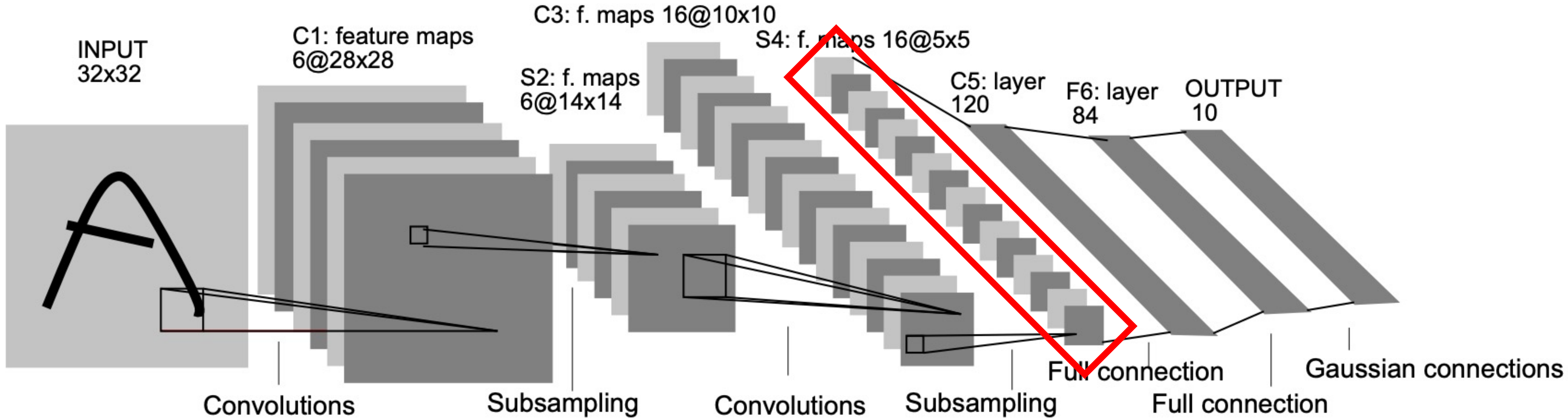


Fig. 1. Architecture of LeNet-5, a Convolutional Neural Network, here for digits recognition. Each plane is a feature map, i.e. a set of units whose weights are constrained to be identical.

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

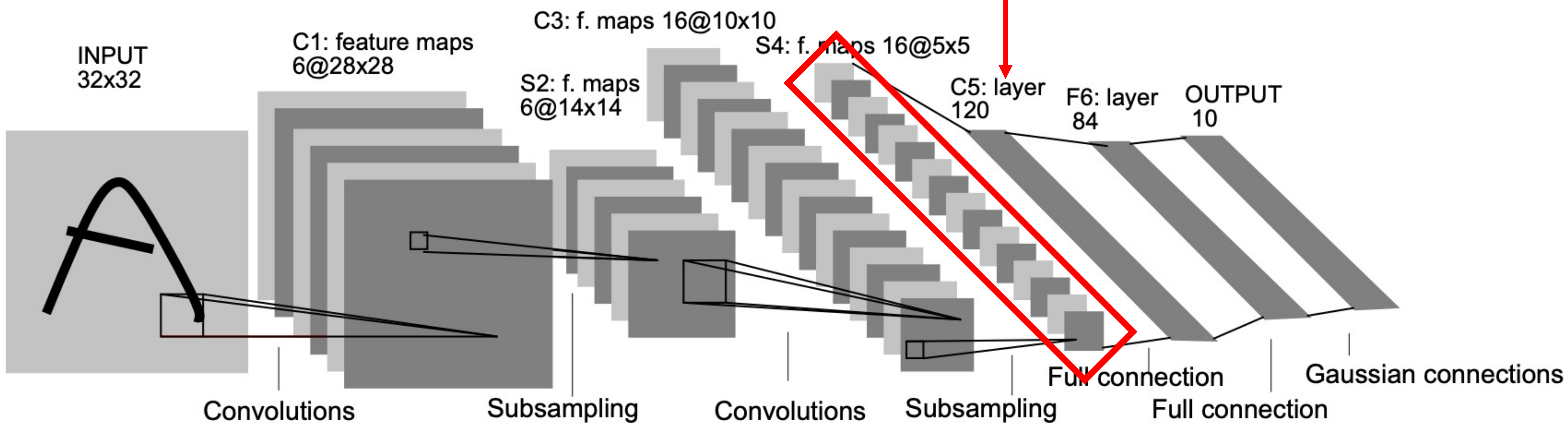


Fig. 1. Architecture of LeNet-5, a Convolutional Neural Network, here for digits recognition. Each plane is a feature map, i.e. a set of units whose weights are constrained to be identical.

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

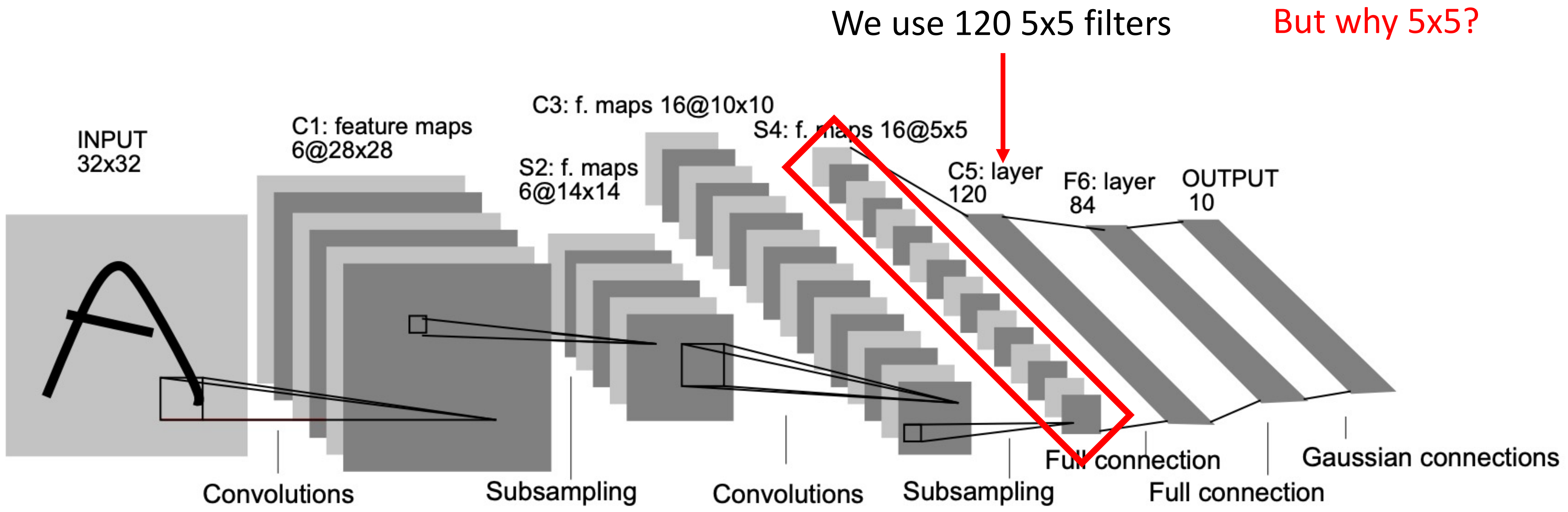


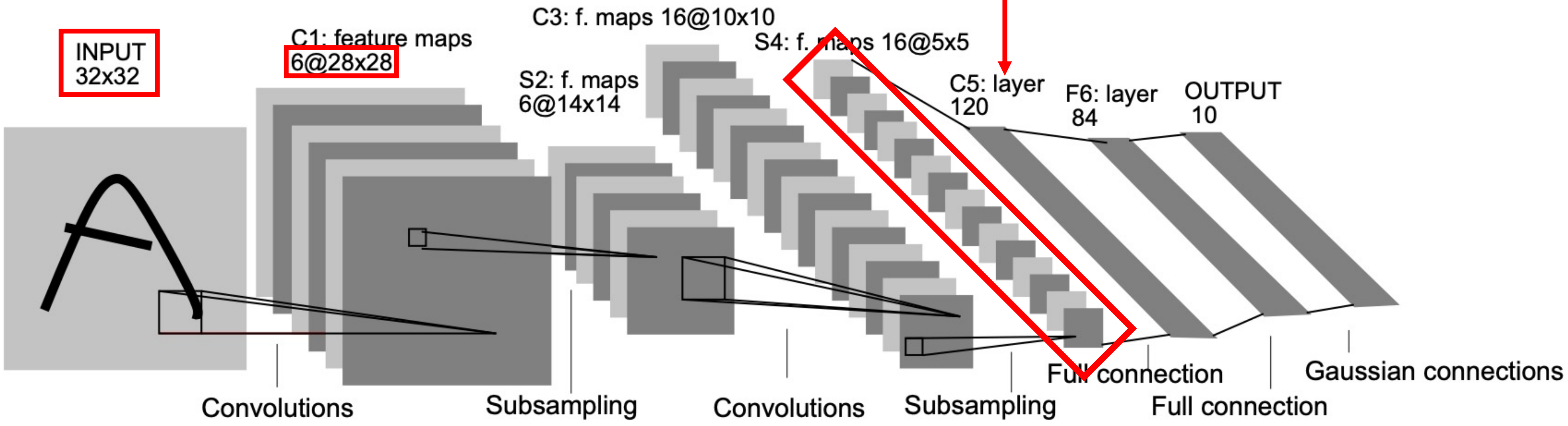
Fig. 1. Architecture of LeNet-5, a Convolutional Neural Network, here for digits recognition. Each plane is a feature map, i.e. a set of units whose weights are constrained to be identical.

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

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?

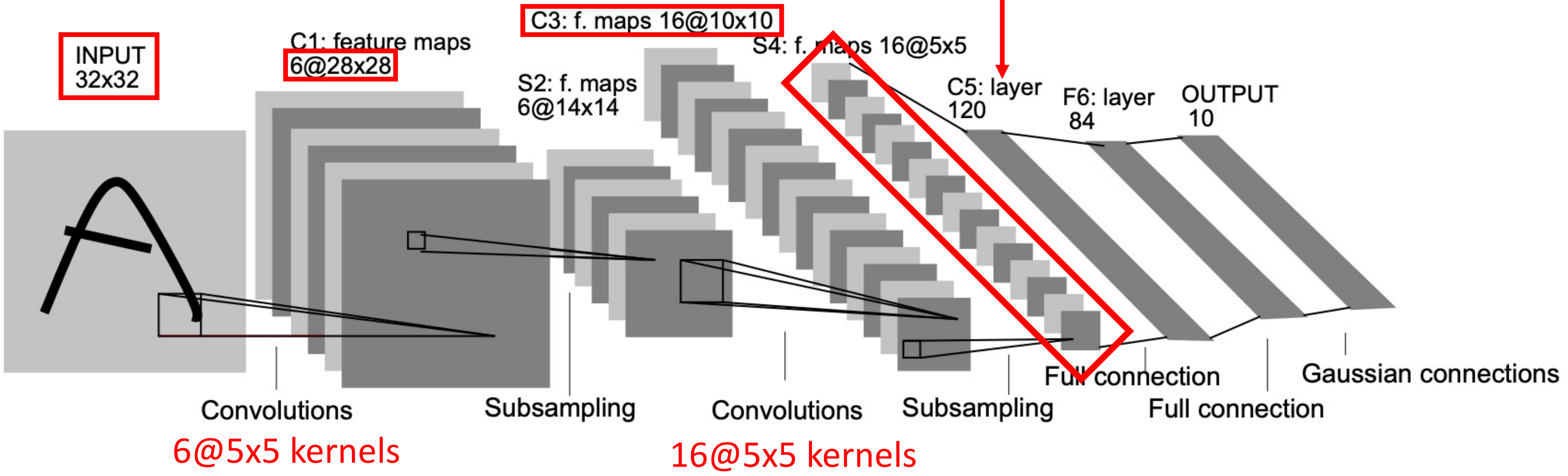


Fig. 1. Architecture of LeNet-5, a Convolutional Neural Network, here for digits recognition. Each plane is a feature map, i.e. a set of units whose weights are constrained to be identical.

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

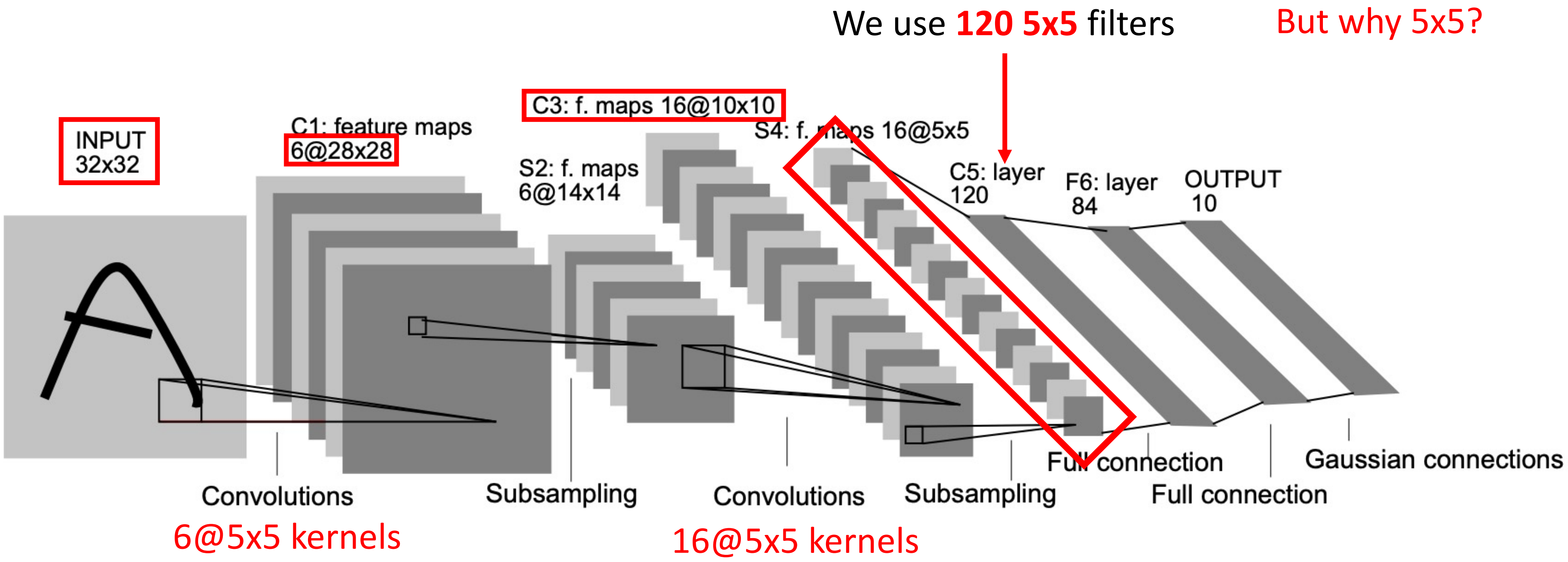


Fig. 1. Architecture of LeNet-5, a Convolutional Neural Network, here for digits recognition. Each plane is a feature map, i.e. a set of units whose weights are constrained to be identical.

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

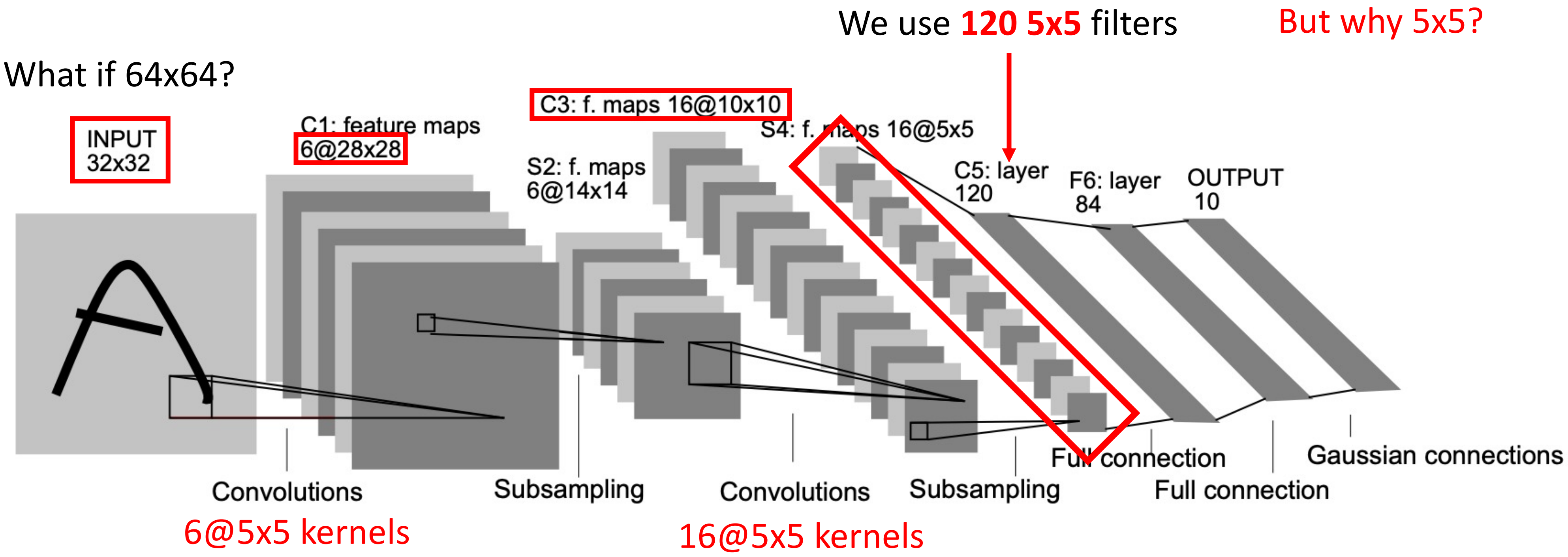
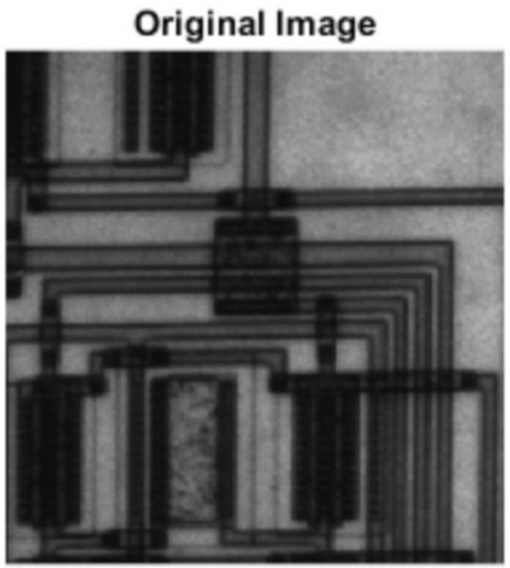


Fig. 1. Architecture of LeNet-5, a Convolutional Neural Network, here for digits recognition. Each plane is a feature map, i.e. a set of units whose weights are constrained to be identical.

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



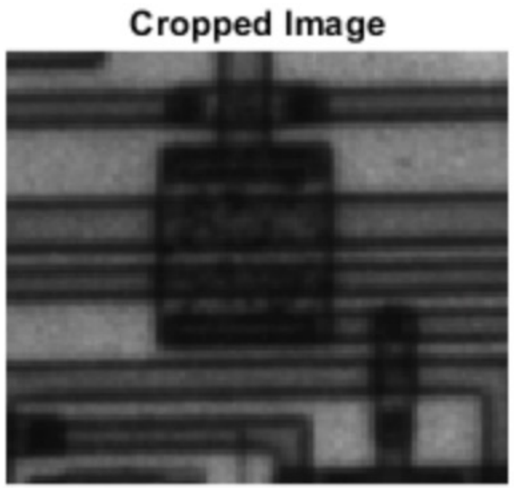
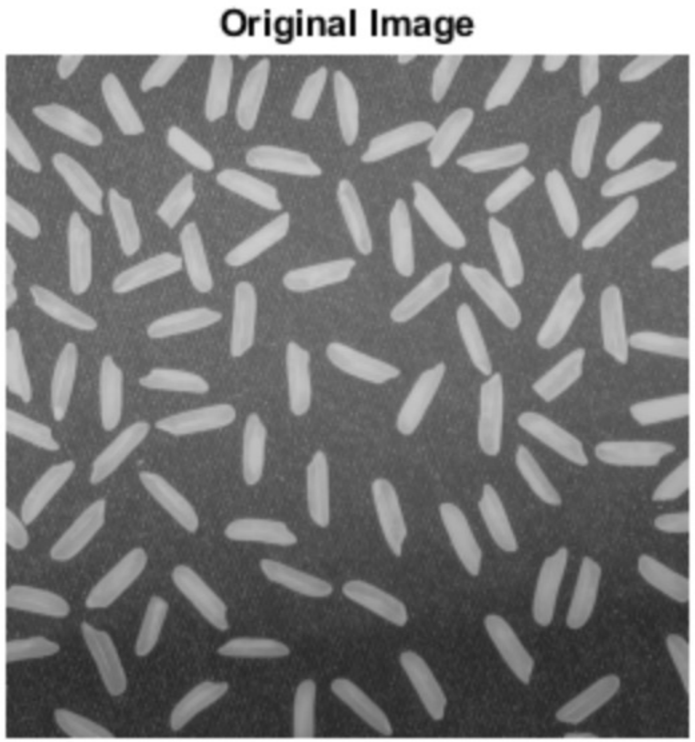


Image from https://www.mathworks.com/help/images/ref/imcrop.html

Input resolution issue



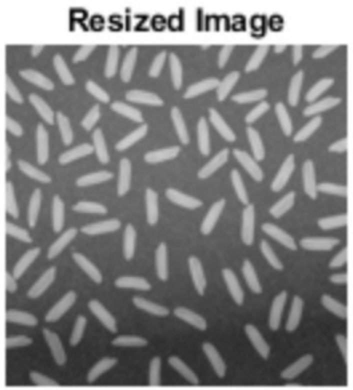
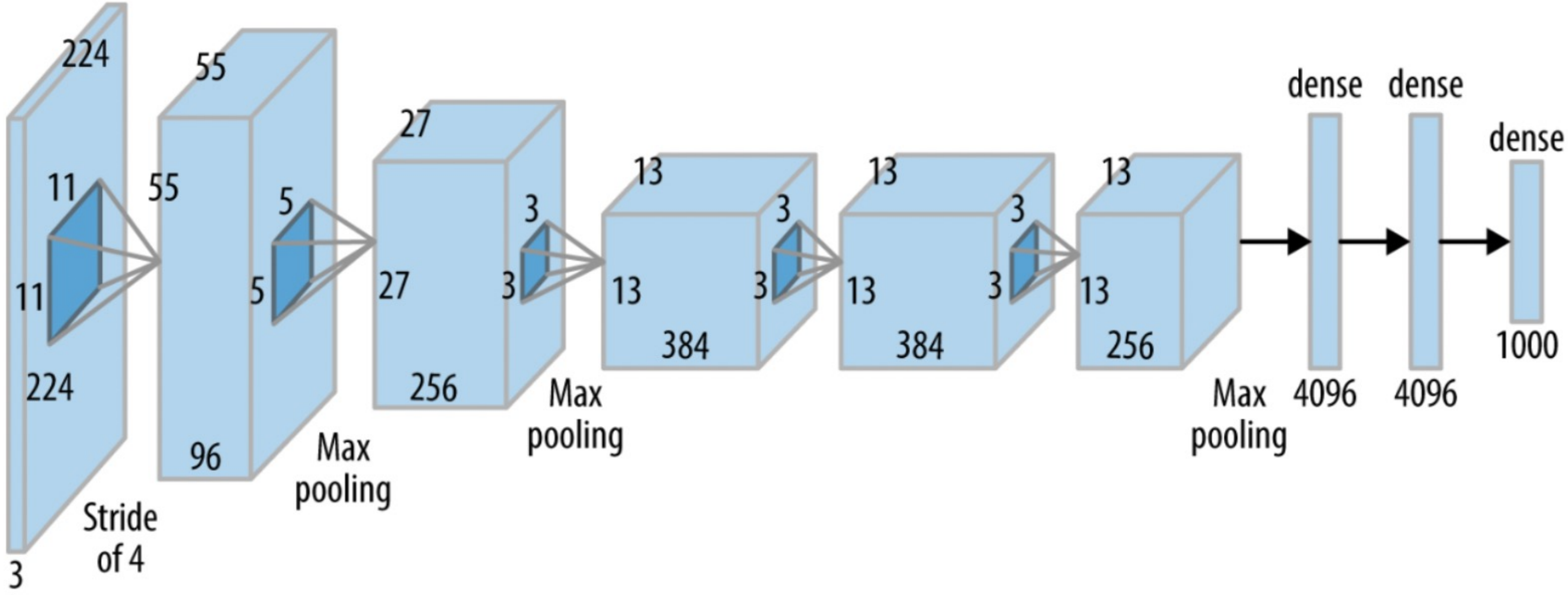


Image from https://www.mathworks.com/help/images/ref/imresize.html

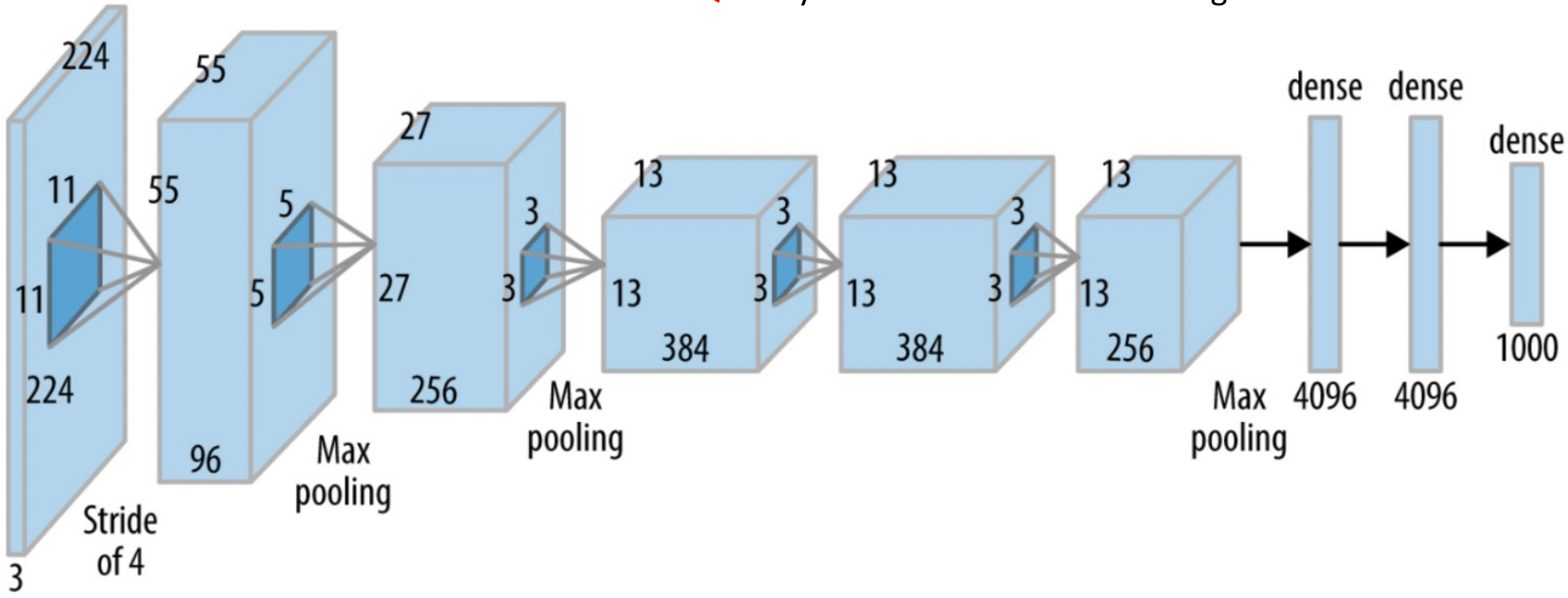
Input resolution issue



[Alexnet]

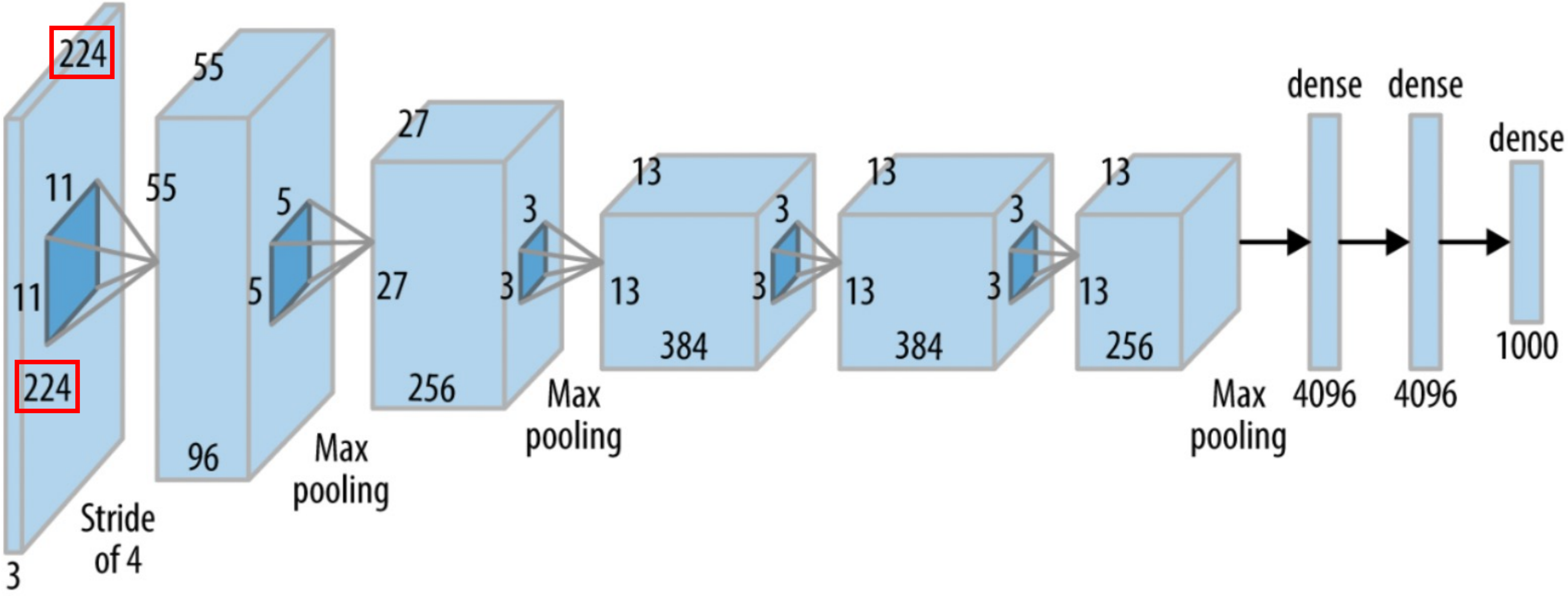
Input resolution issue

Q: can you understand the following architecture?



[Alexnet]

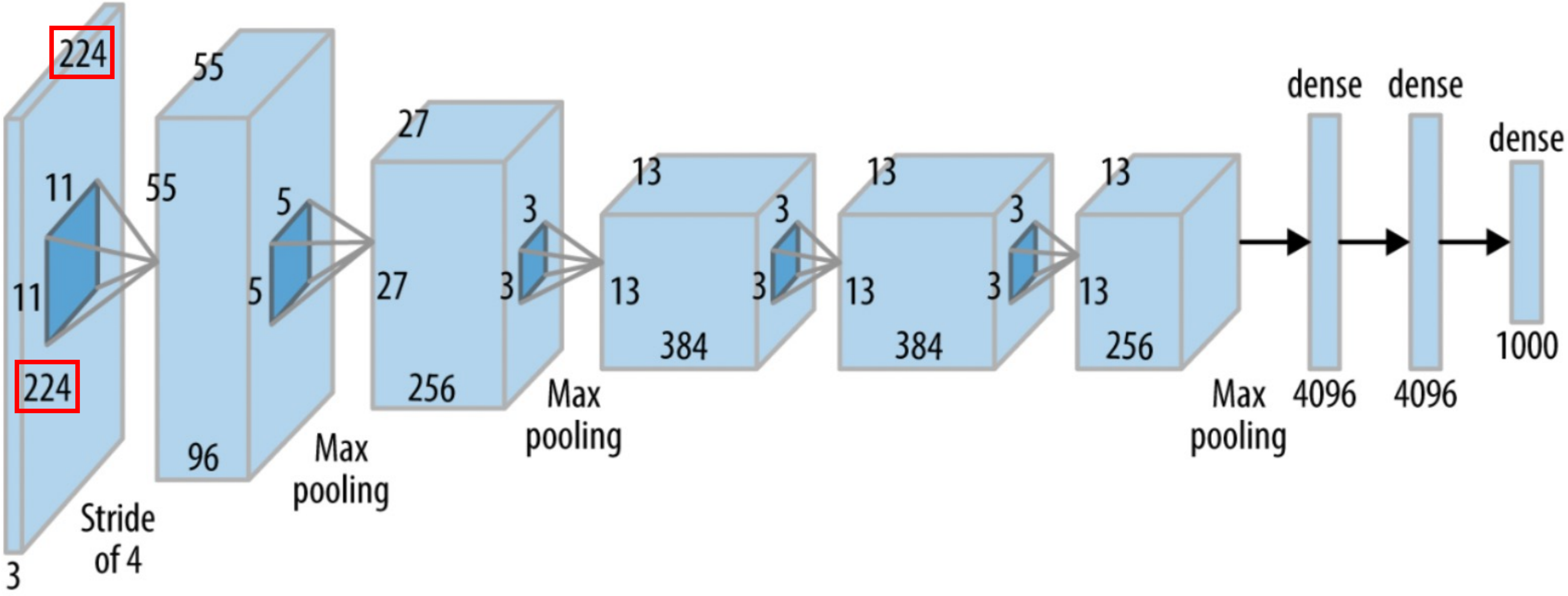
Input resolution issue



Any input image must be 224x224

[Alexnet]

Input resolution issue



Any input image must be 224x224

Q: how to handle an arbitrary resolution?

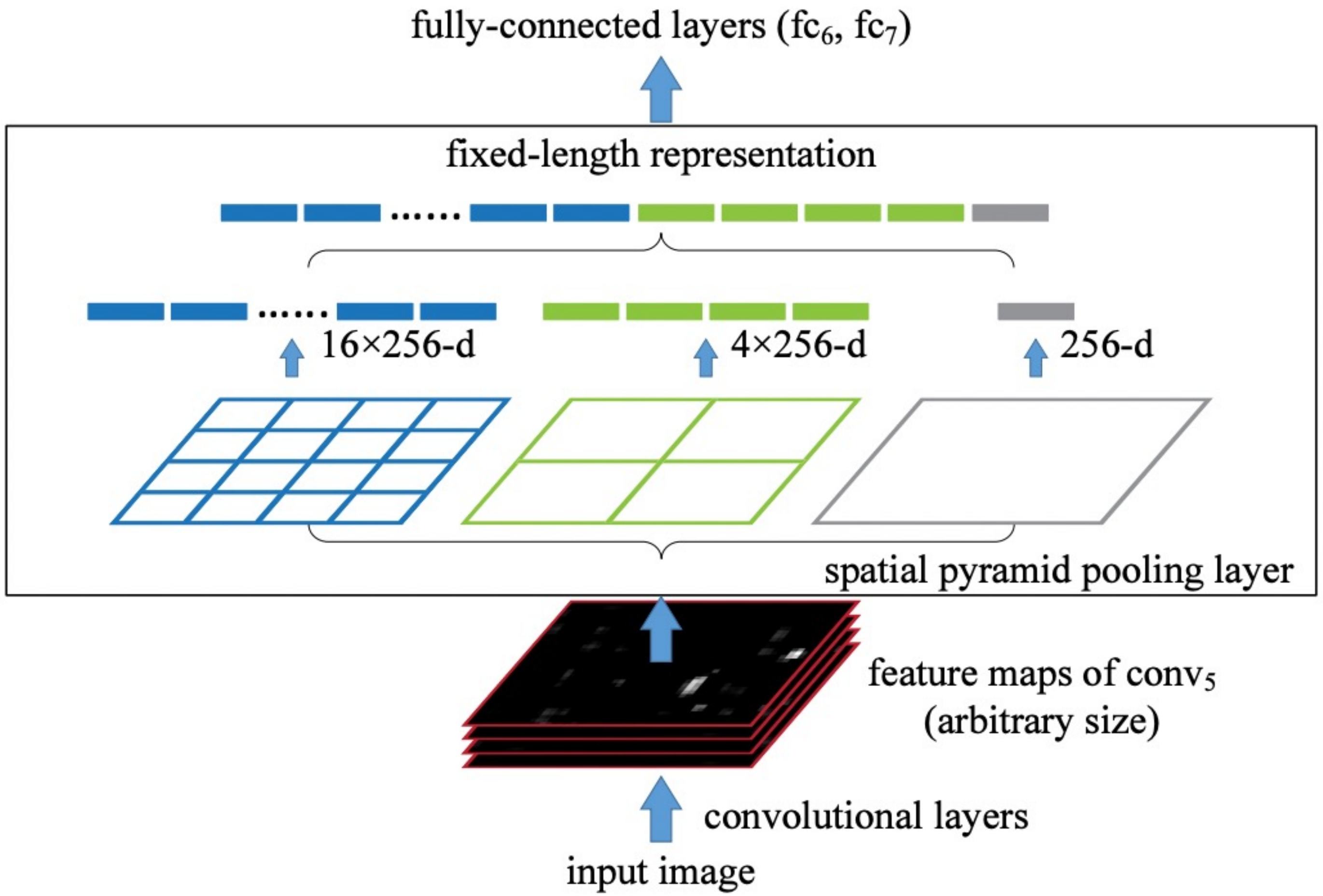
[Alexnet]

Input resolution issue

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

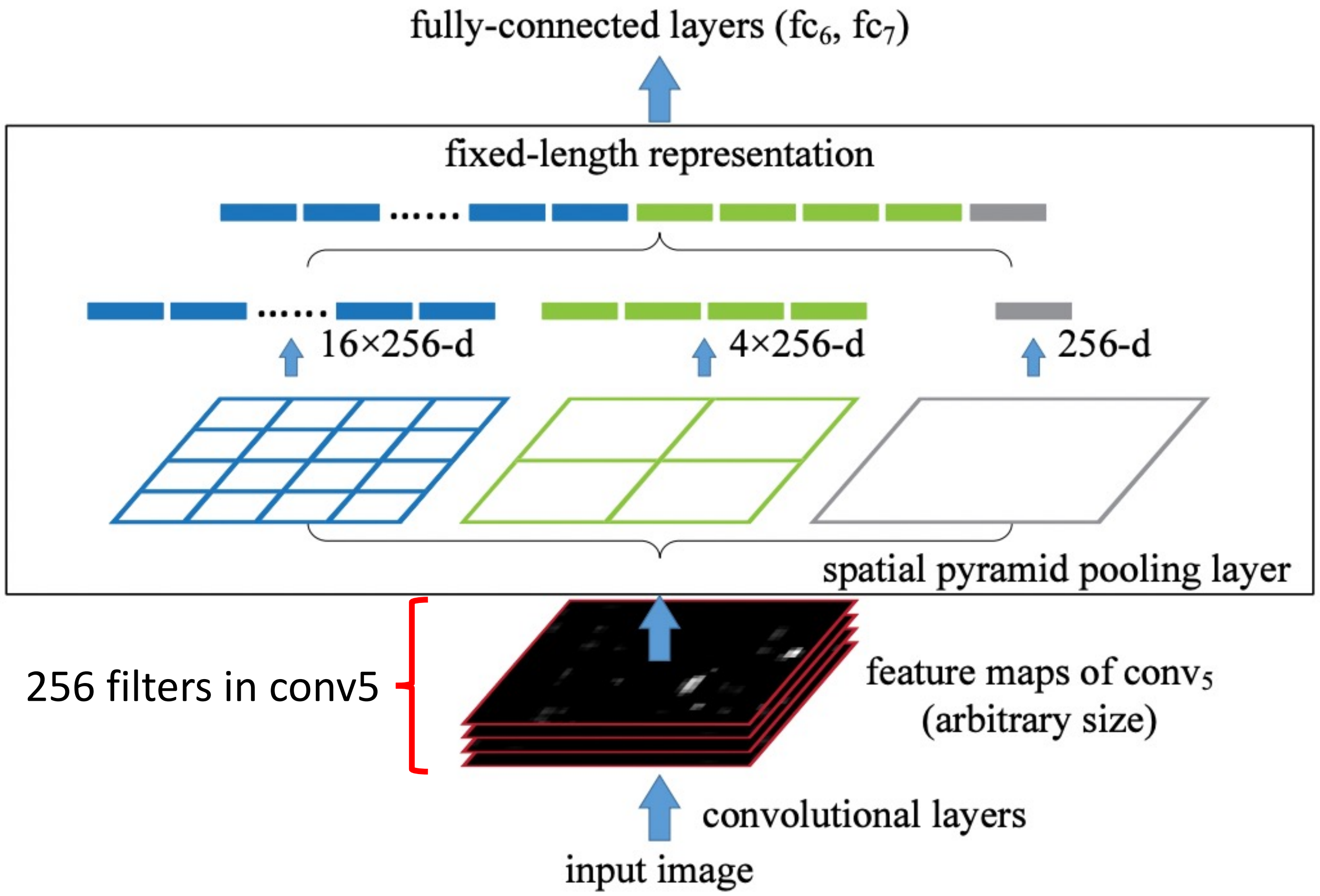
Input resolution issue

- Spatial pyramid pooling



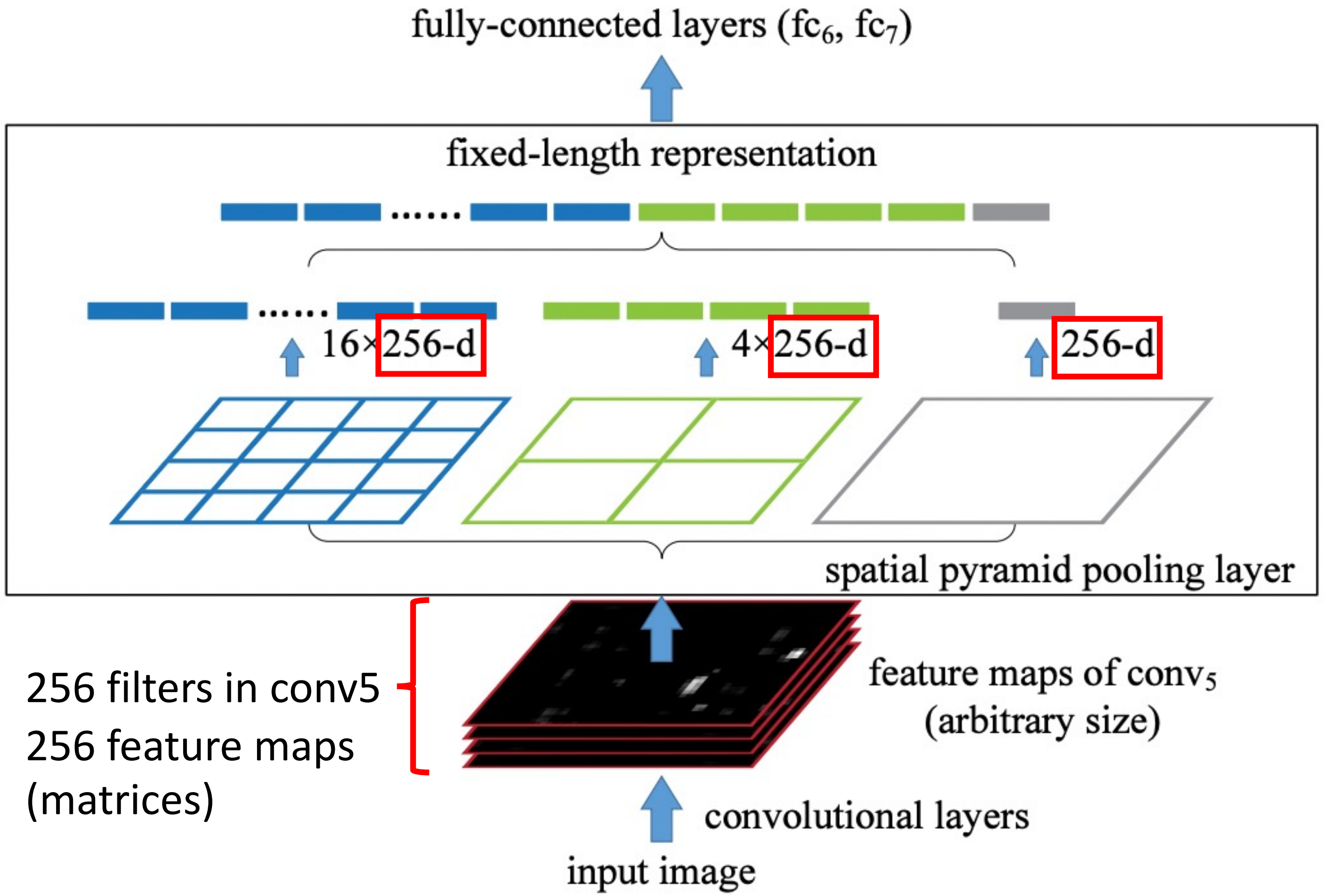
Input resolution issue

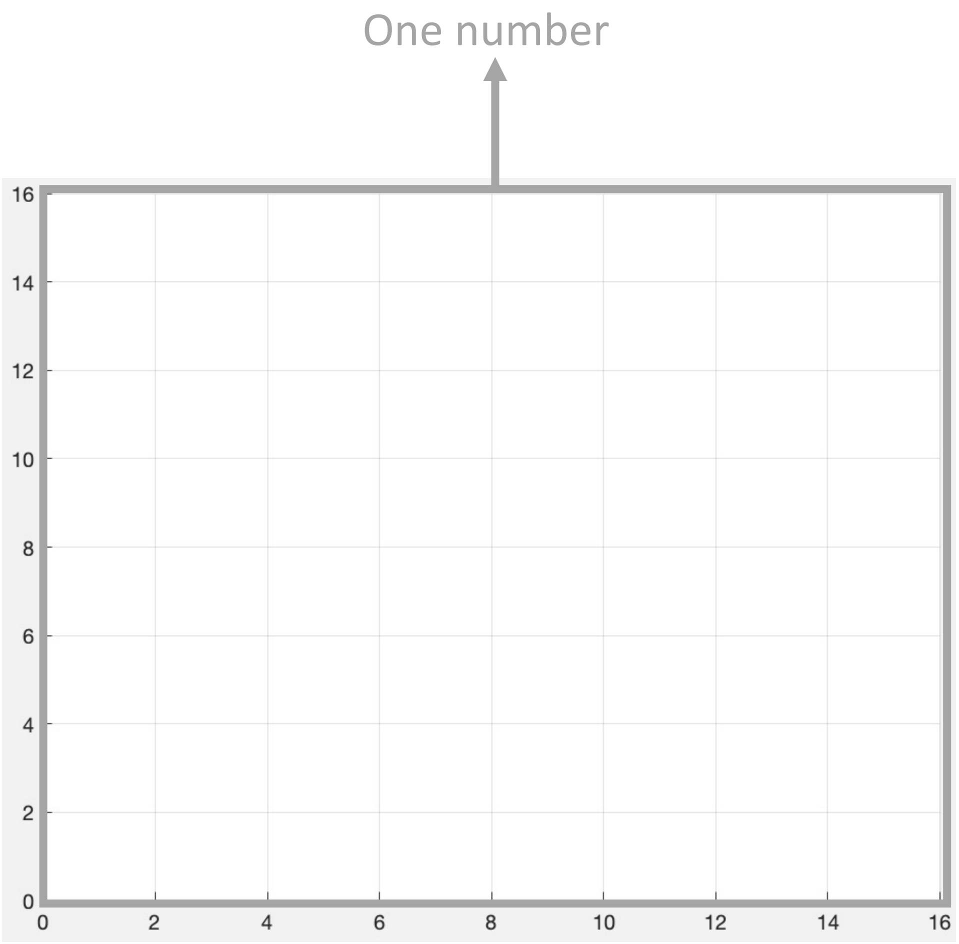
- Spatial pyramid pooling



Input resolution issue

- Spatial pyramid pooling

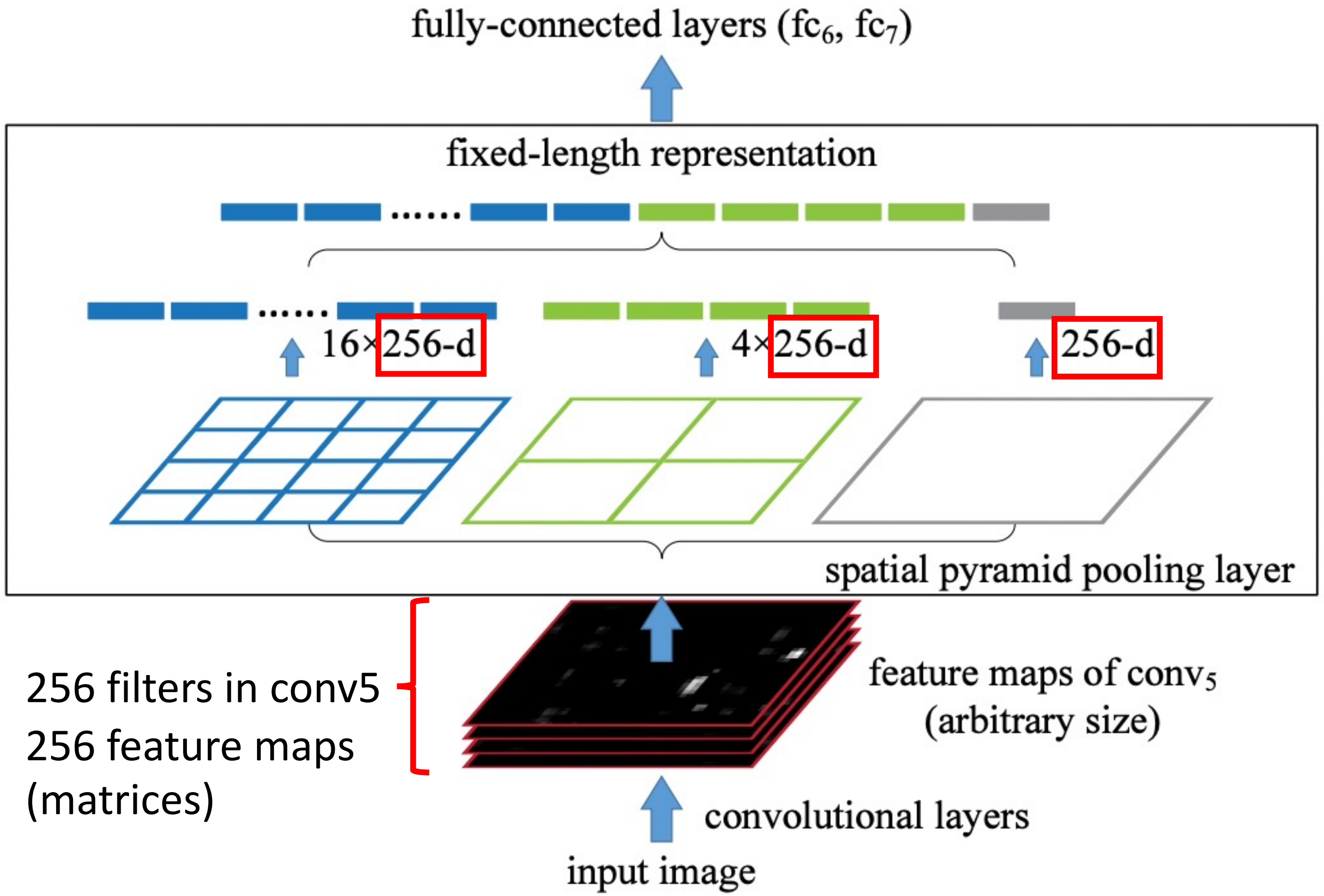


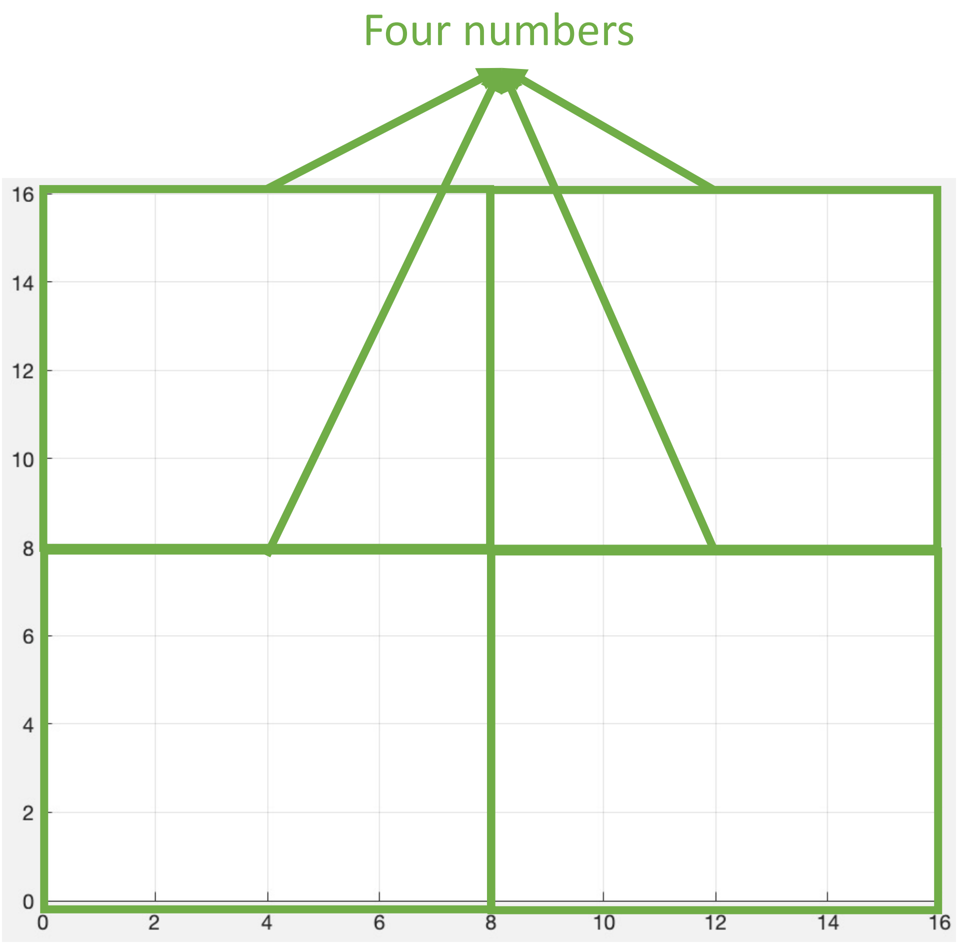


Some pooling (max/average)

Input resolution issue

- Spatial pyramid pooling

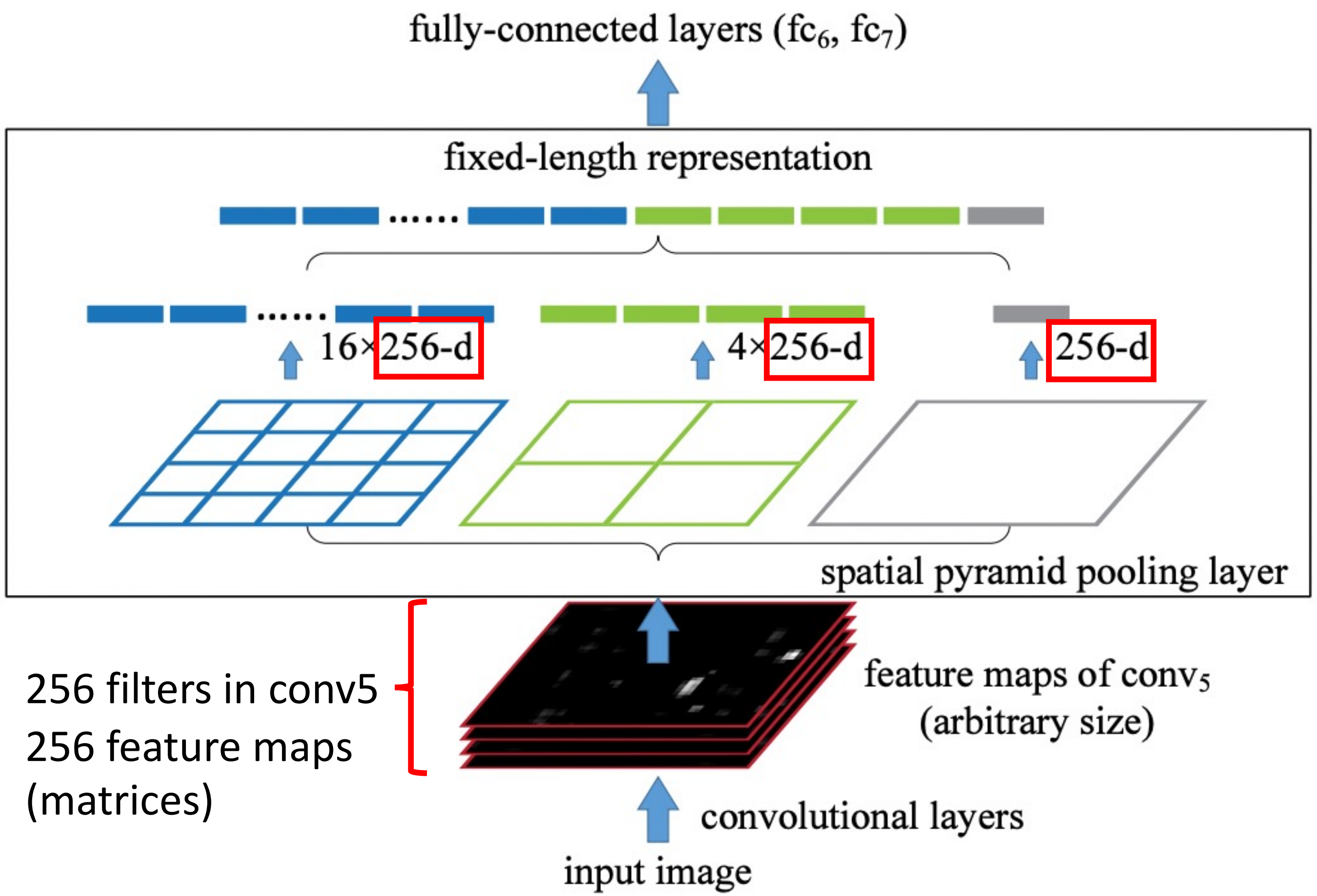




Some pooling (max/average)

Input resolution issue

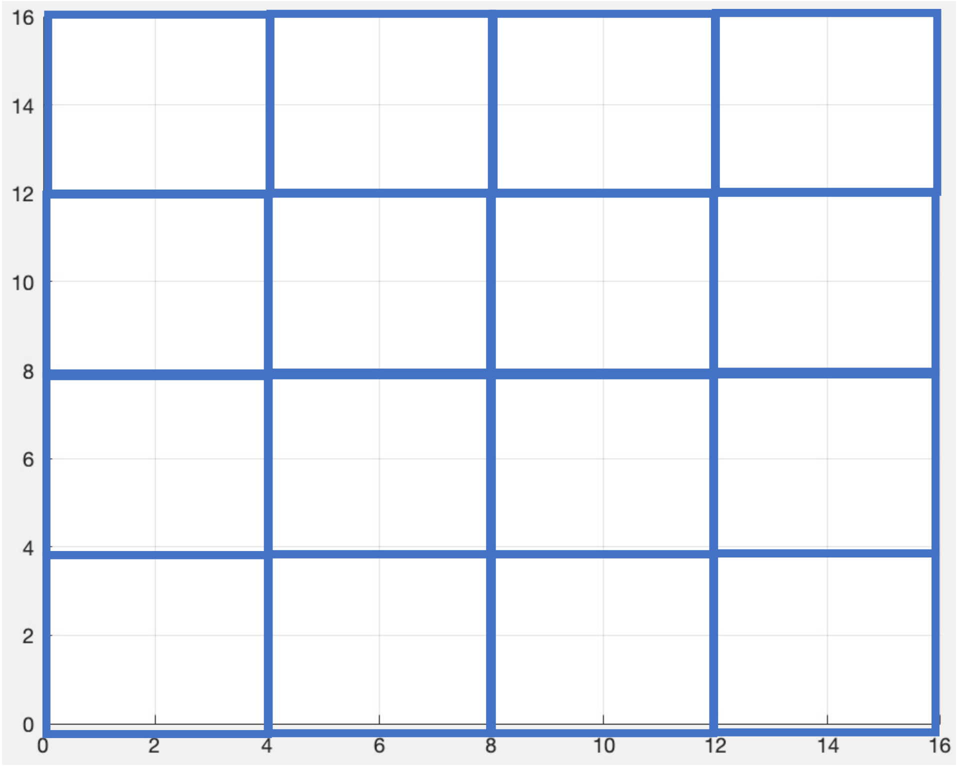
- Spatial pyramid pooling



256 filters in conv5

256 feature maps (matrices)

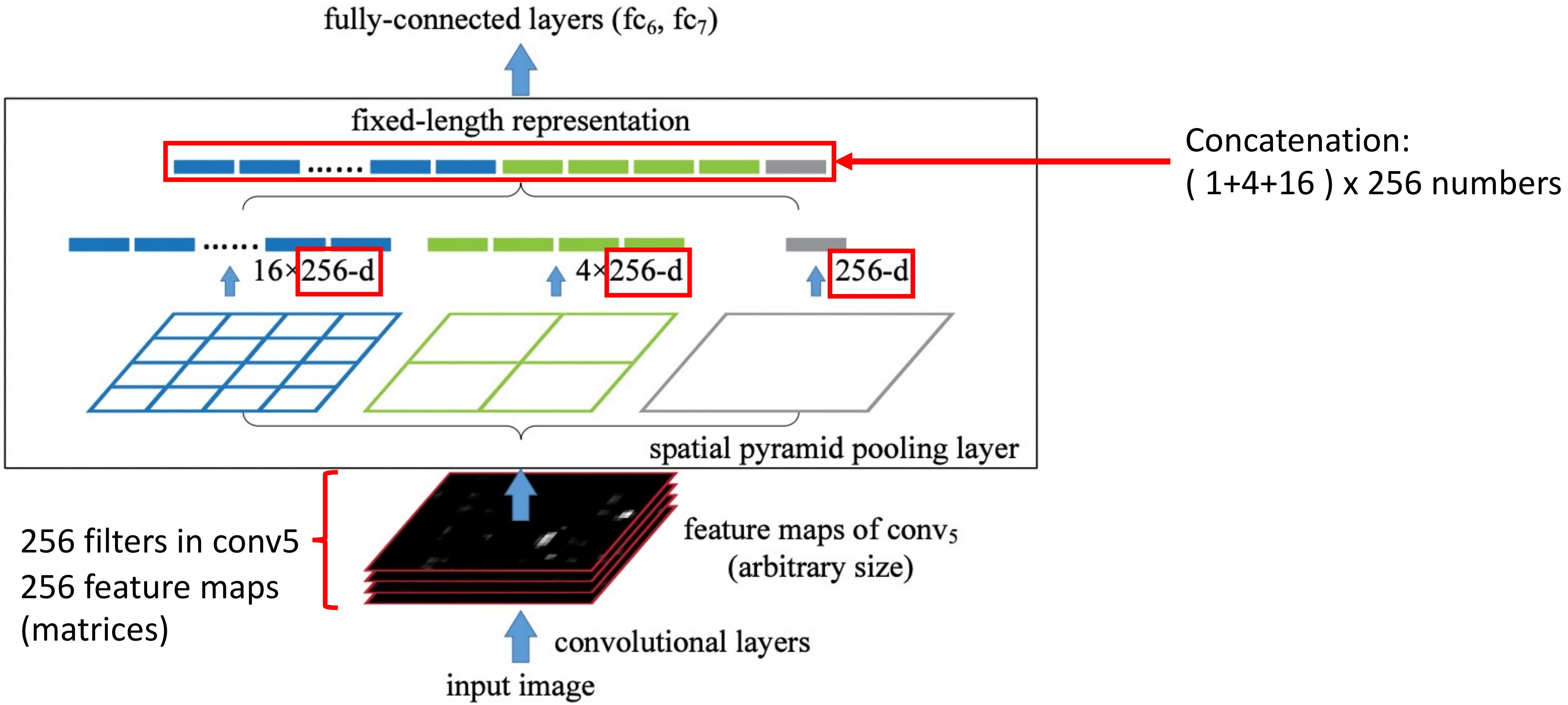
16 numbers



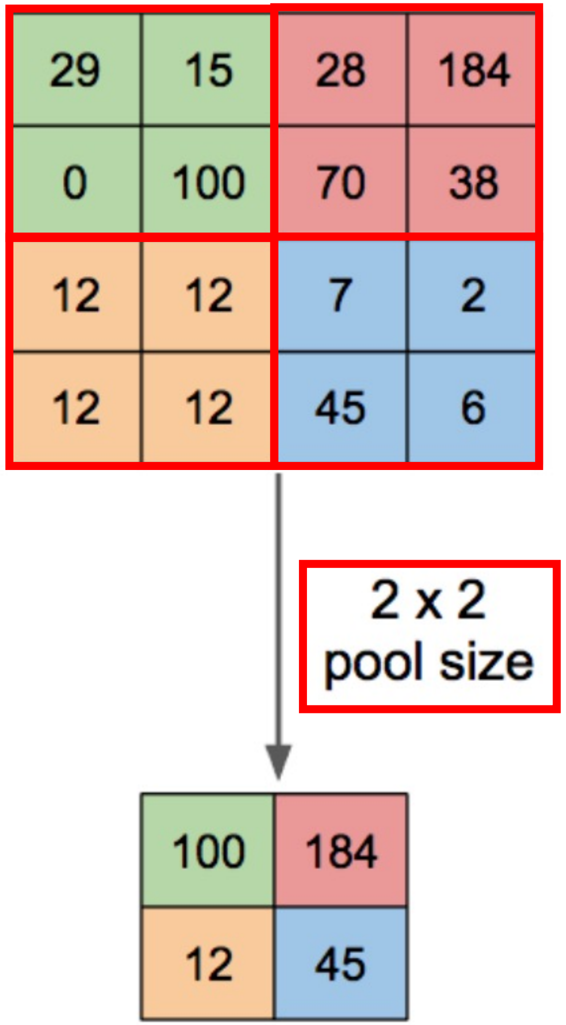
Some pooling (max/average)

Input resolution issue

- Spatial pyramid pooling

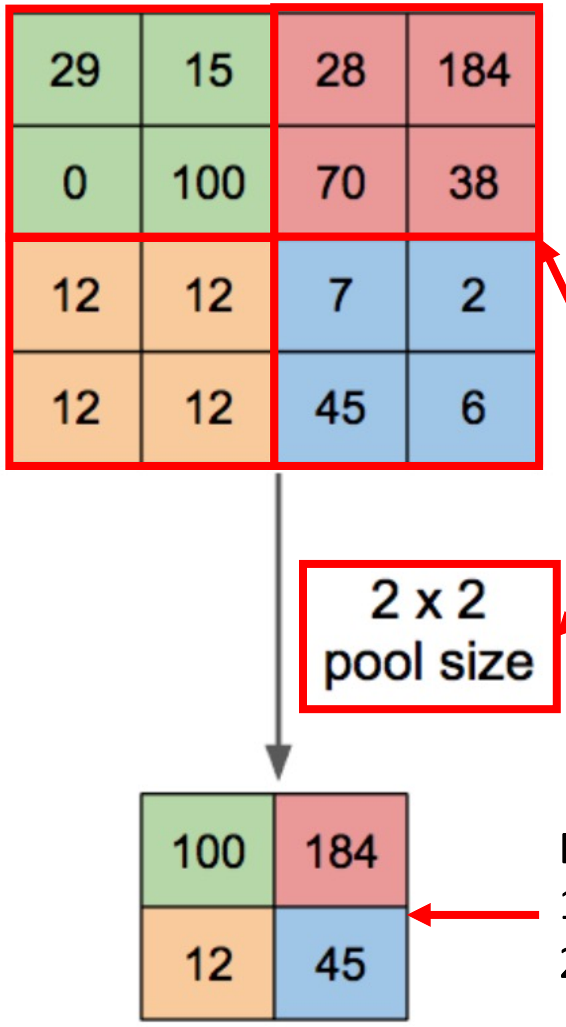


Input resolution issue



Max pooling

Input resolution issue



Input dimension

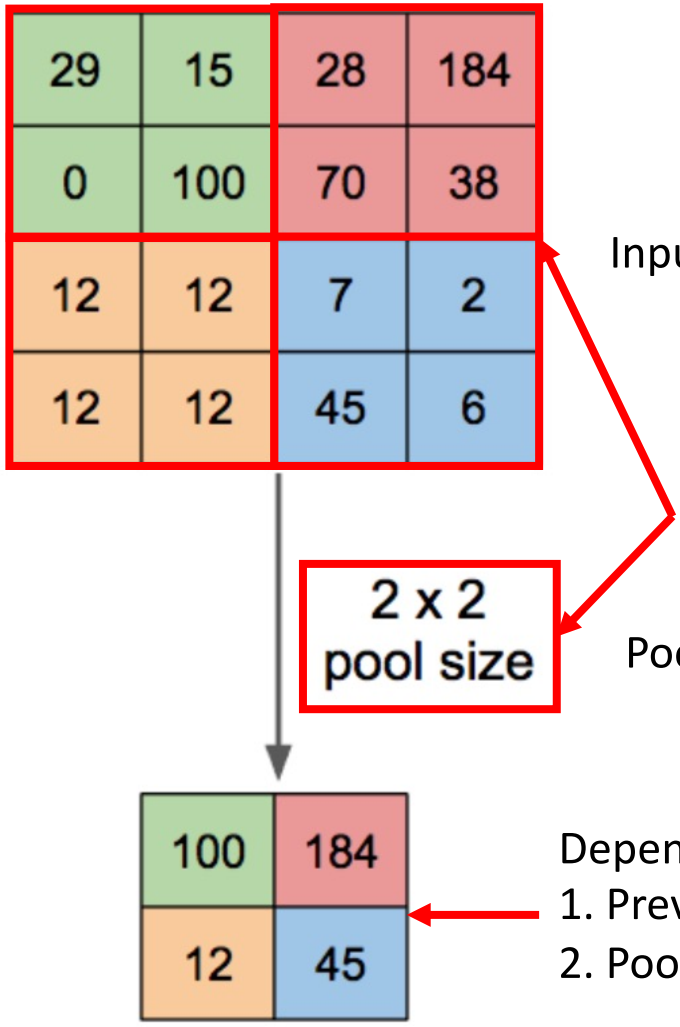
Predefined

Pool size

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

Max pooling

Input resolution issue



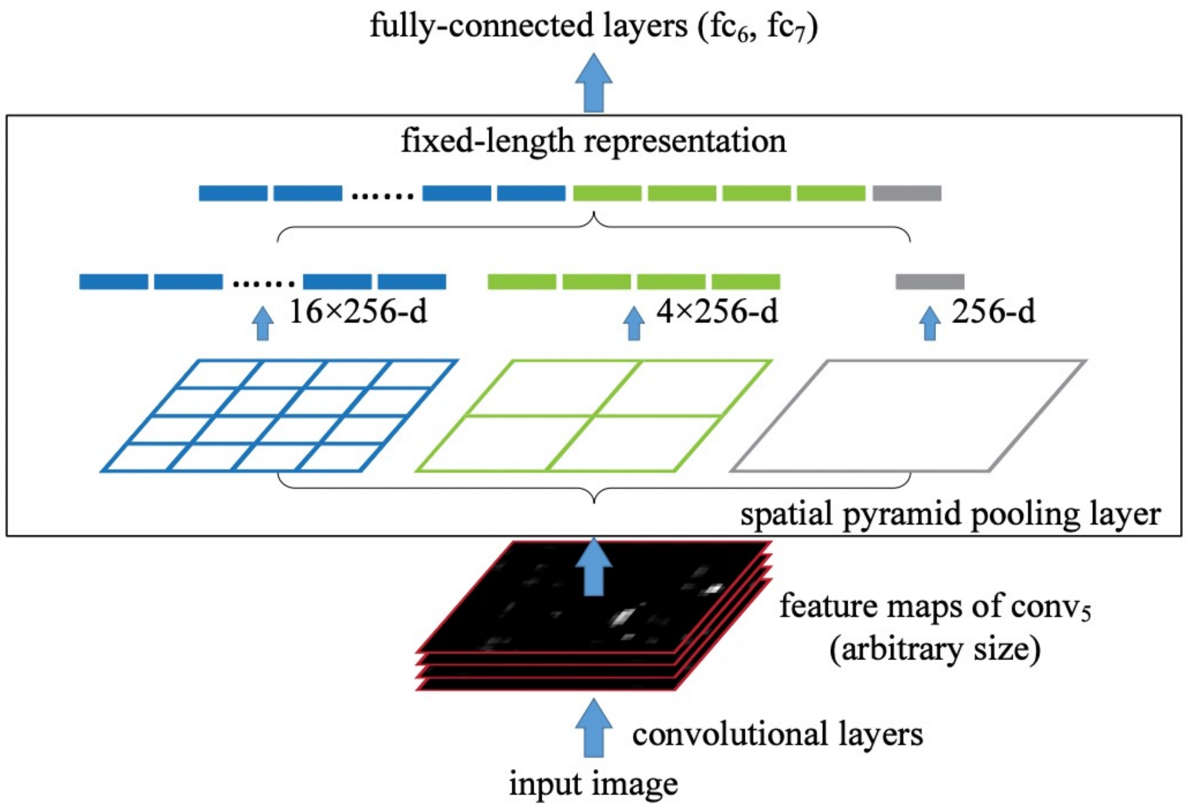
Input dimension

Predefined

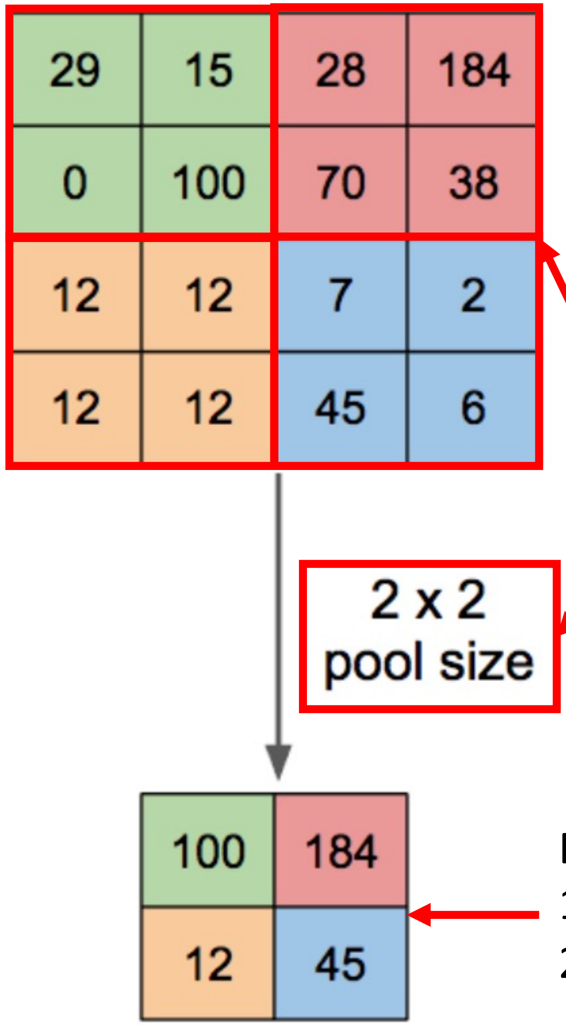
Pool size

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

Max pooling



Input resolution issue



Input dimension

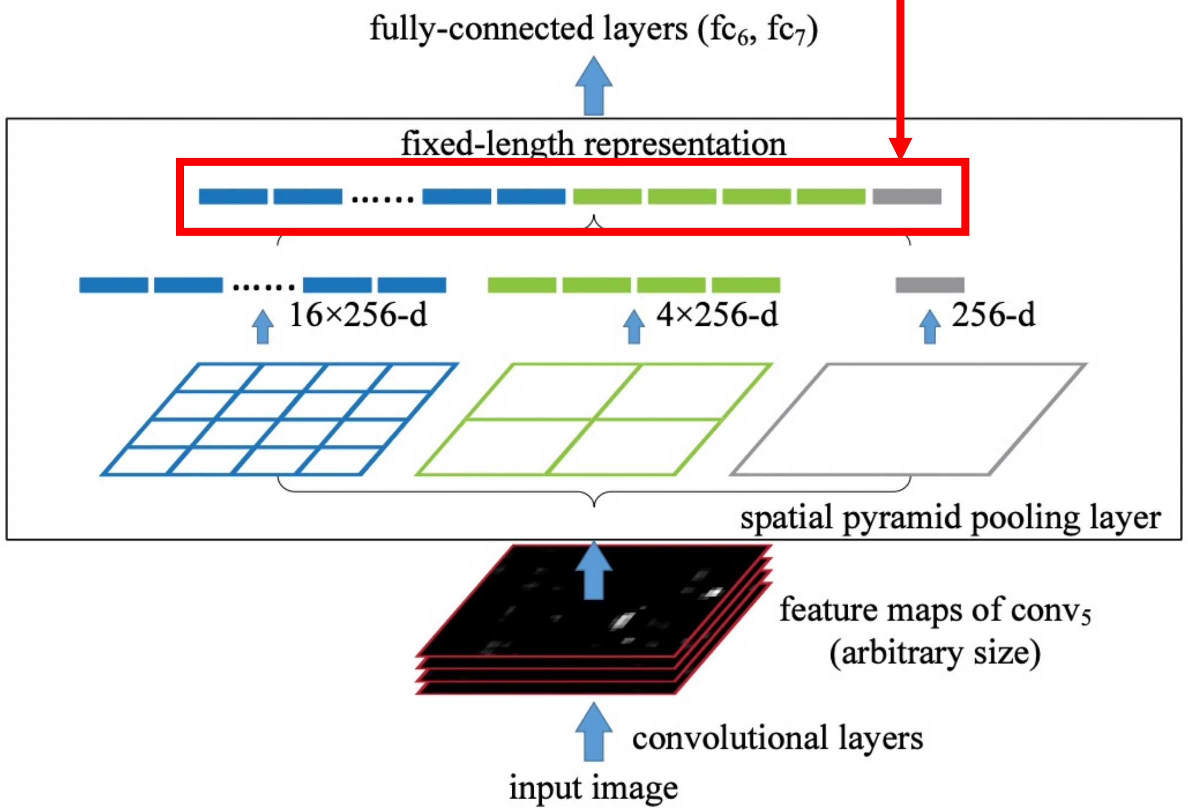
Predefined

Pool size

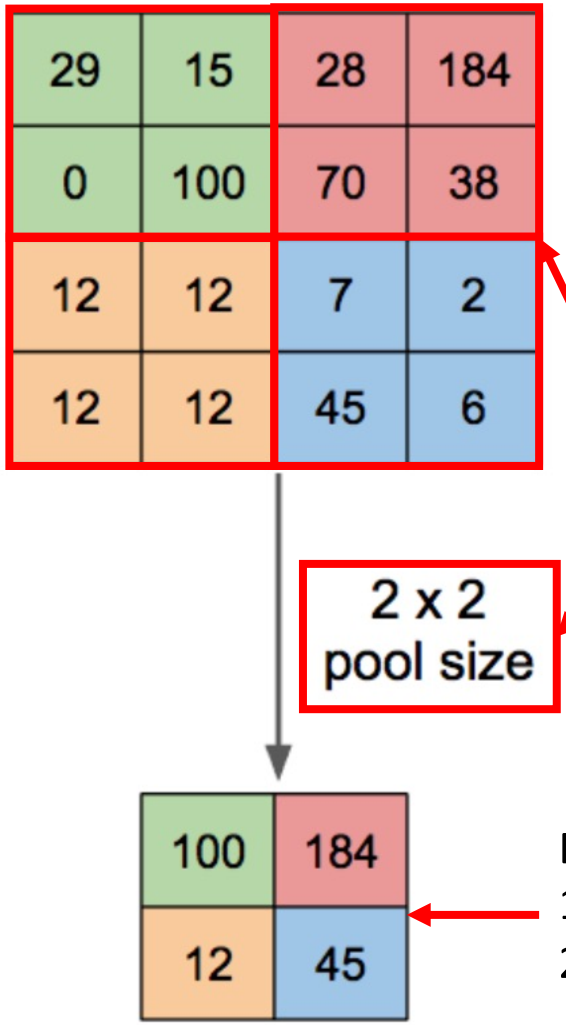
Depends on:
1. Previous feat map size
2. Pooling size

Max pooling

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



Input resolution issue



Input dimension

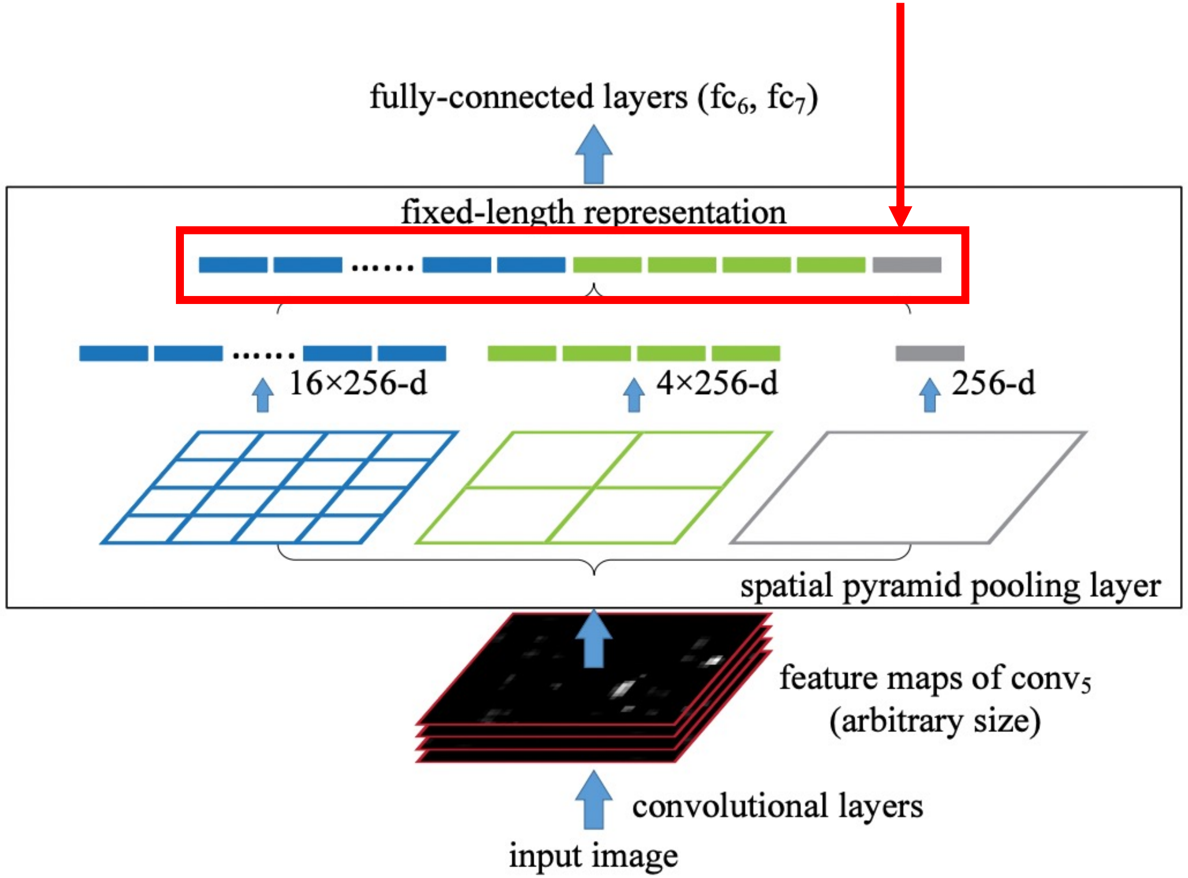
Predefined

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

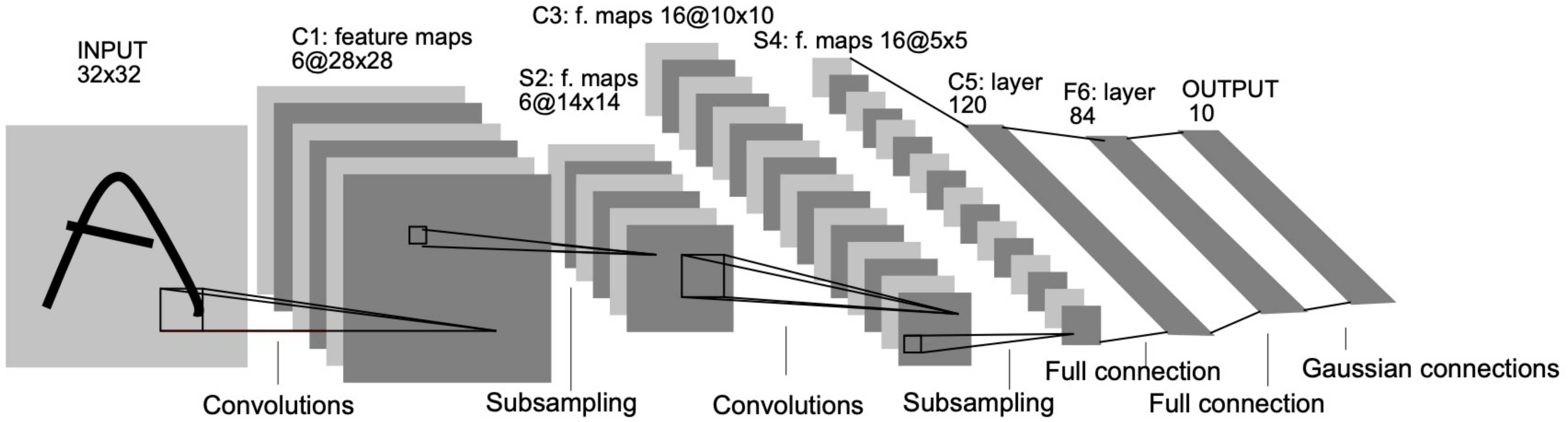


Fig. 1. Architecture of LeNet-5, a Convolutional Neural Network, here for digits recognition. Each plane is a feature map, i.e. a set of units whose weights are constrained to be identical.

Input resolution issue

- Global average pooling

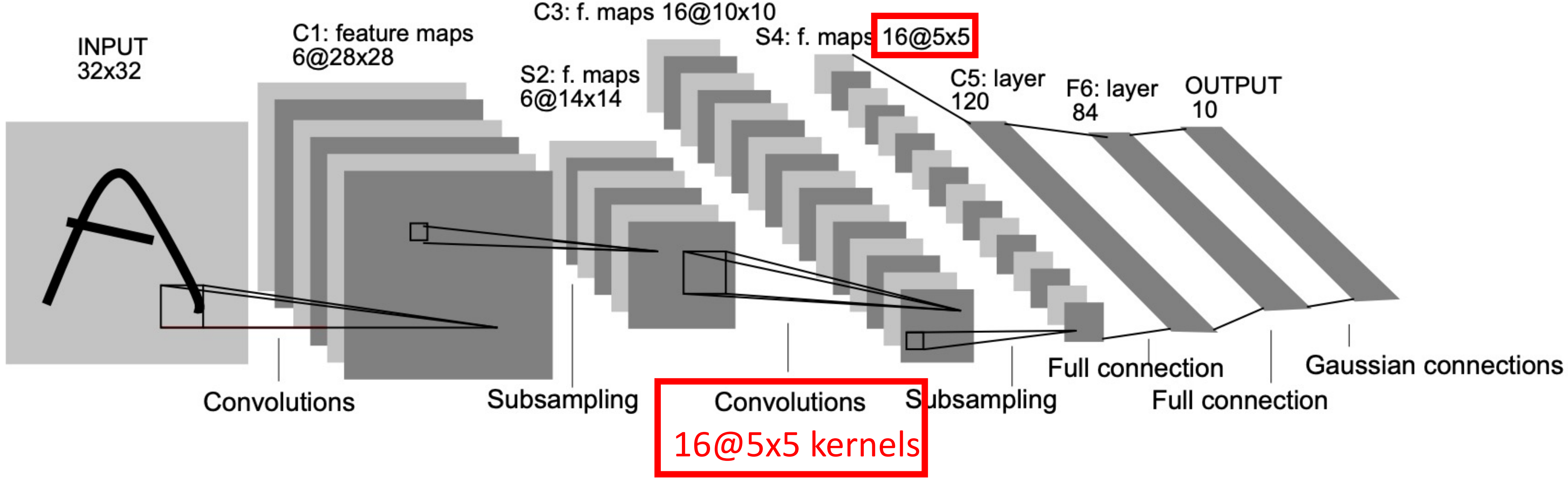


Fig. 1. Architecture of LeNet-5, a Convolutional Neural Network, here for digits recognition. Each plane is a feature map, i.e. a set of units whose weights are constrained to be identical.

Input resolution issue

- Global average pooling

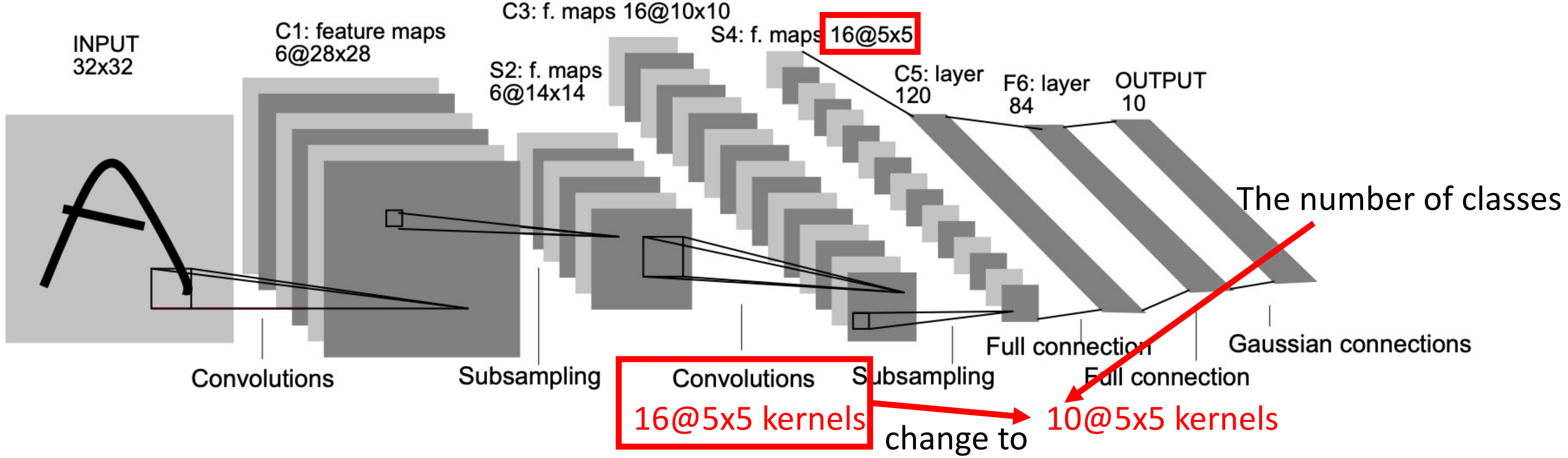


Fig. 1. Architecture of LeNet-5, a Convolutional Neural Network, here for digits recognition. Each plane is a feature map, i.e. a set of units whose weights are constrained to be identical.

Input resolution issue

- Global average pooling

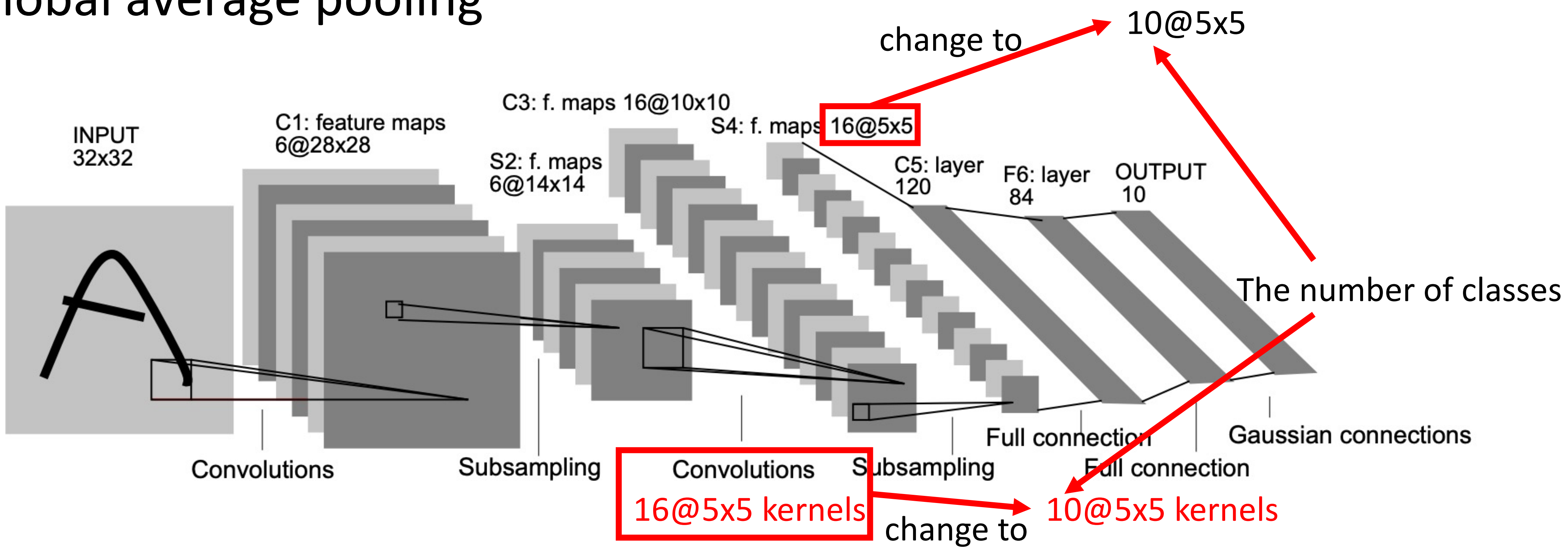


Fig. 1. Architecture of LeNet-5, a Convolutional Neural Network, here for digits recognition. Each plane is a feature map, i.e. a set of units whose weights are constrained to be identical.

Input resolution issue

- Global average pooling

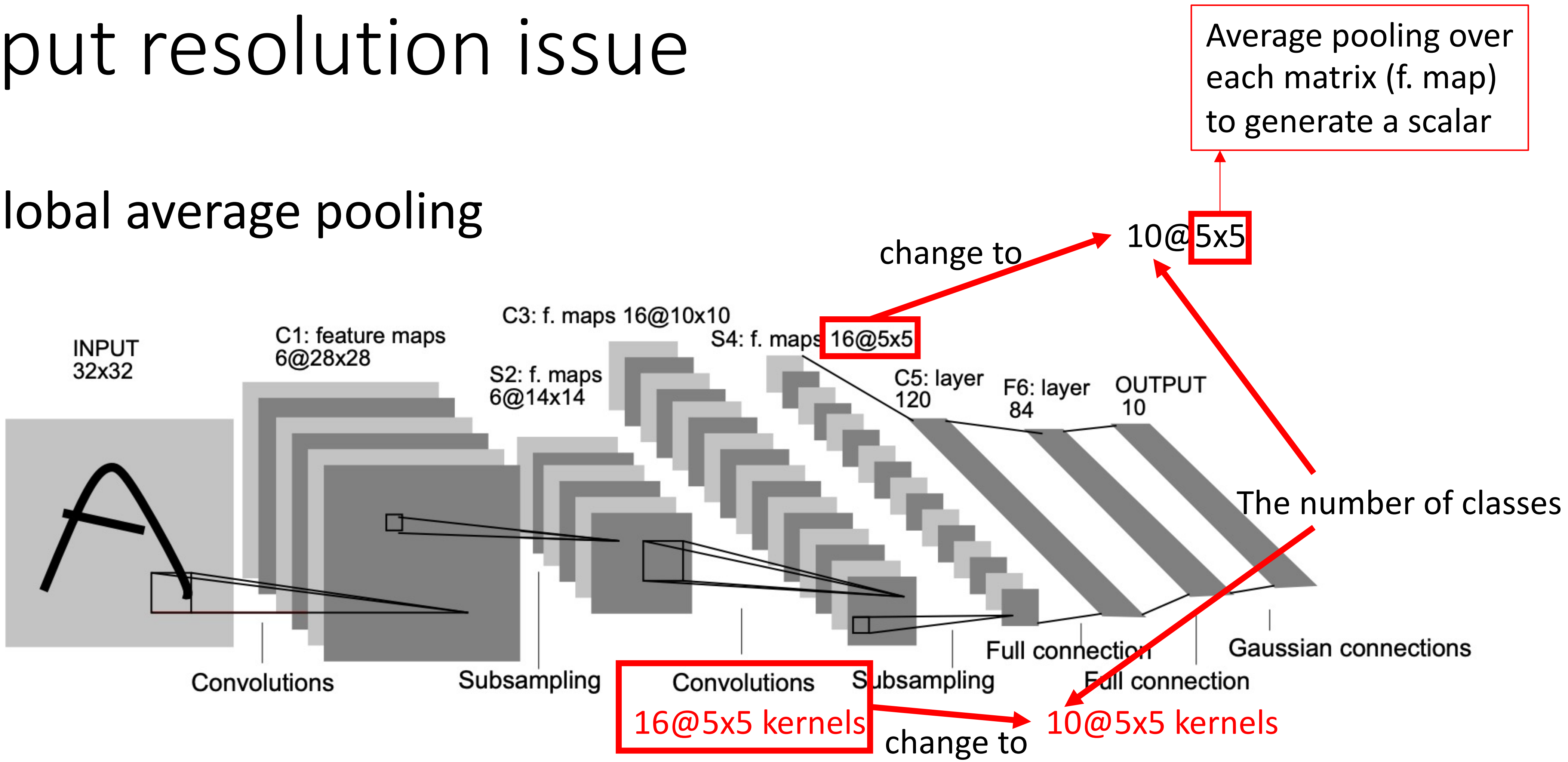


Fig. 1. Architecture of LeNet-5, a Convolutional Neural Network, here for digits recognition. Each plane is a feature map, i.e. a set of units whose weights are constrained to be identical.

Input resolution issue

- Global average pooling

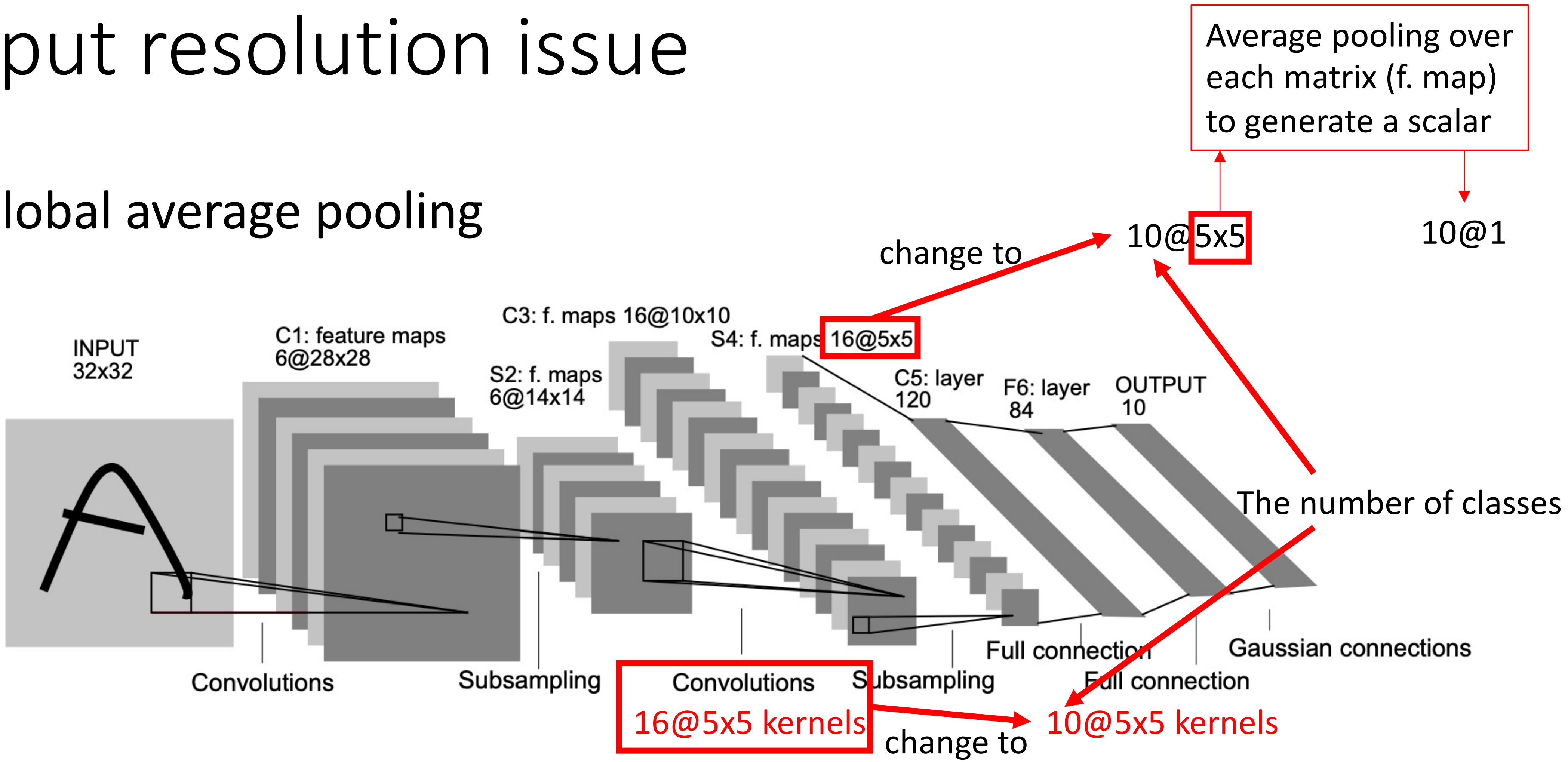


Fig. 1. Architecture of LeNet-5, a Convolutional Neural Network, here for digits recognition. Each plane is a feature map, i.e. a set of units whose weights are constrained to be identical.

Input resolution issue

- Global average pooling

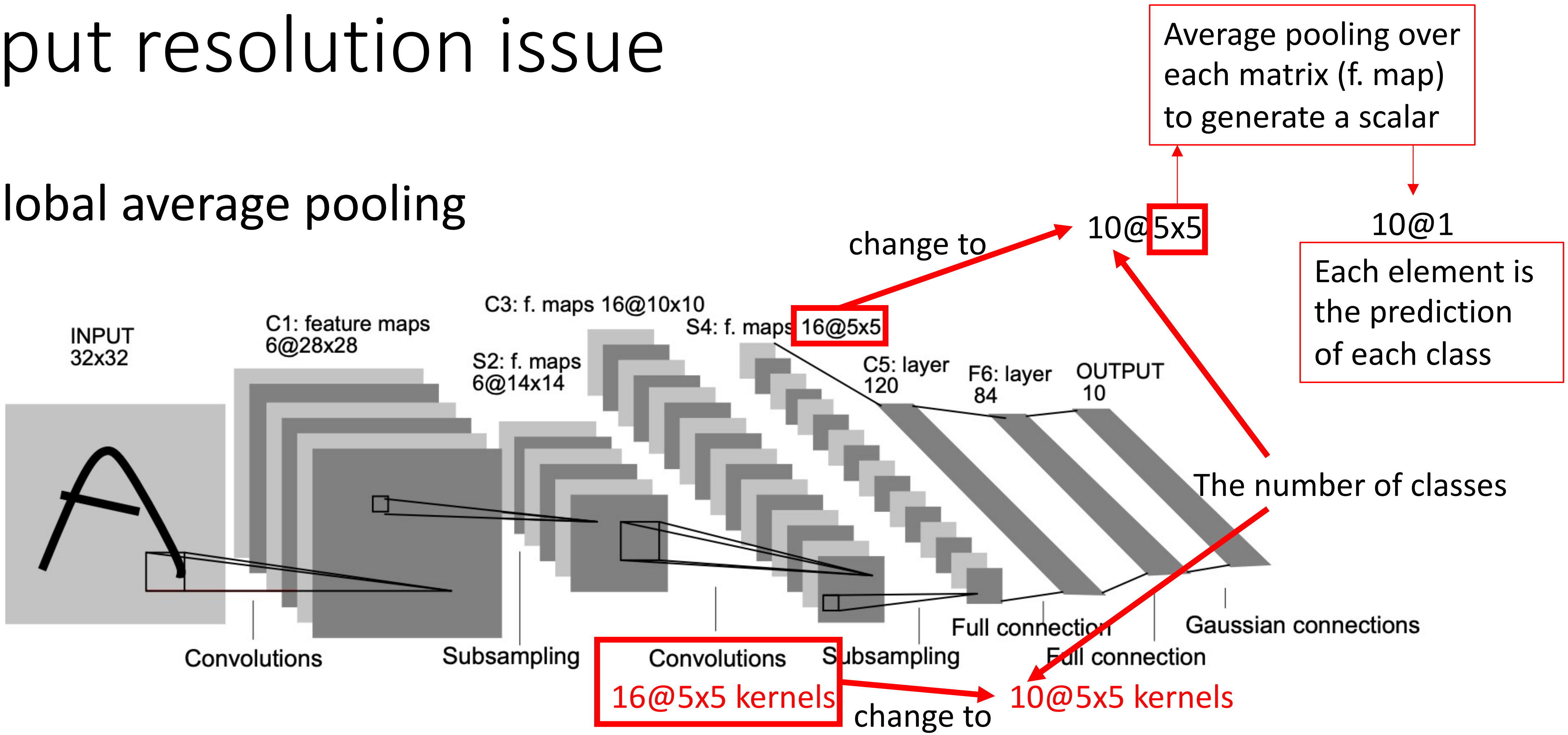


Fig. 1. Architecture of LeNet-5, a Convolutional Neural Network, here for digits recognition. Each plane is a feature map, i.e. a set of units whose weights are constrained to be identical.

References

- [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)