Deep Learning (CS 470, CS 570)

Module 4, Lecture 1: Introduction to Convolutional Neural Network

Convolutional Neural Network (CNN)

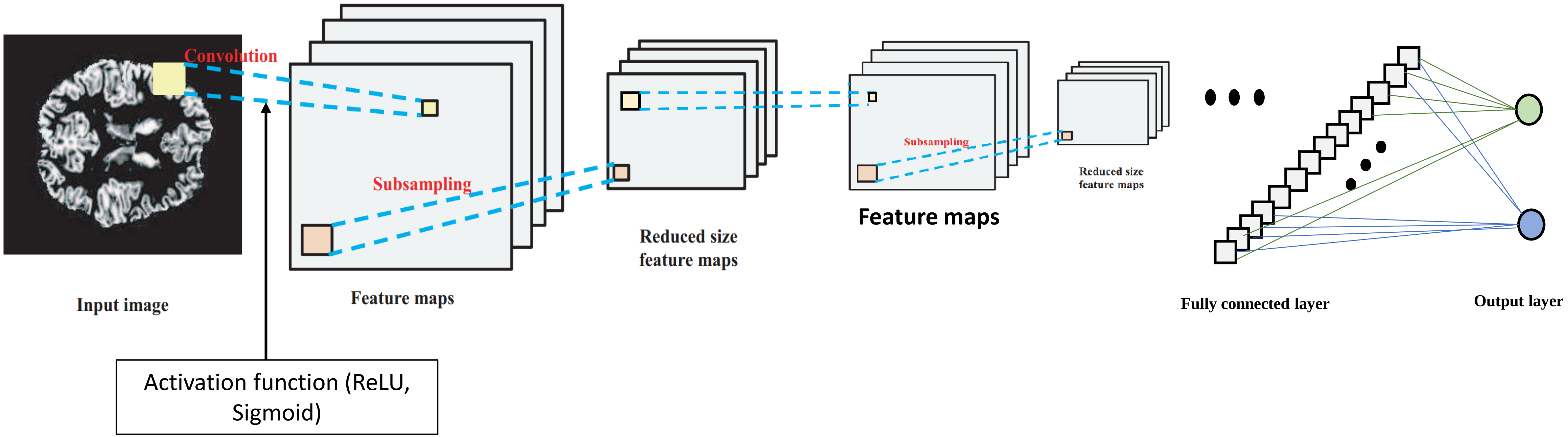
CNN is a specific neural network architecture that vastly used for machine learning operations on image/video data. Any data that has spatial relationship can be a good candidate for CNN.

Architecture of CNN consist of convolutional layer, activation layer, pooling layer, and fully connected layer.

Advantage over MLP: there are several advantages of CNN over MLP for image classification. We will discuss these after we understand the basic structure of a CNN.

Please read CNN vs MLP for image classification

CNN: Architecture



Convolutional filters/kernel: The small matrix used for convolution operation

Feature map: Convolved image produced after the convolutional operation

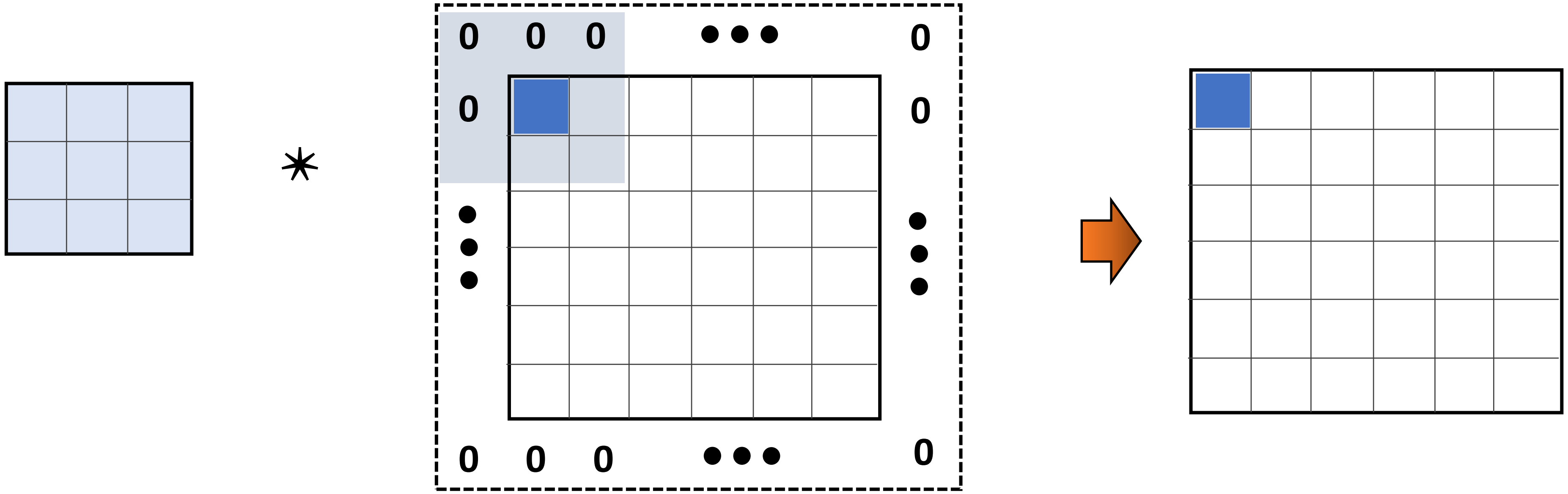
Convolutional layer: Contains a set of convolutional filters. The input image/feature maps are convolved with each of the filters.

Pooling: The pooling layer responsible for down sampling operation on feature map. This layer used a pooling kernel.

Fully connected layer: A layer of neurons where each neuron has a weighted connection with each cell of the last feature maps.

Convolution Operation:2D

- **Image convolution: border pixels**

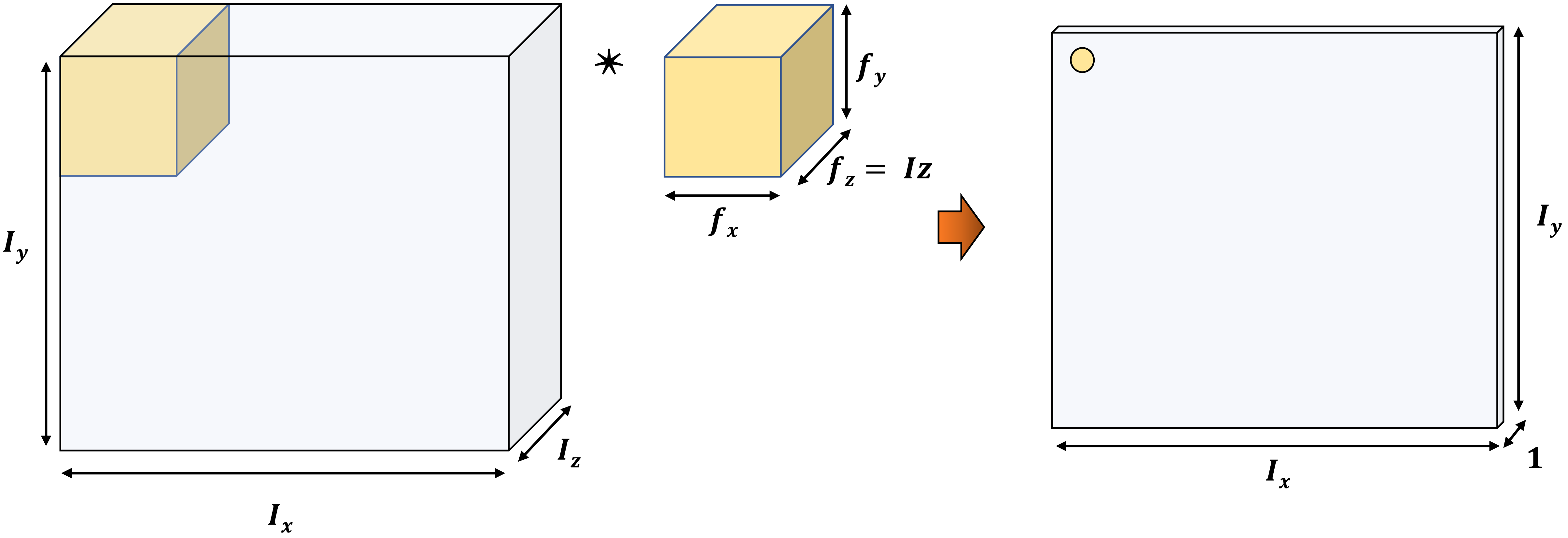


Images are generally zero padded or other extrapolation techniques are used before performing convolution.

- **Convolution layer: hyperparameters**
  - Number of filters
  - Dimension of filters
  - Stride length
  - Padding

We discussed 1D and 2D convolution operations in module 1. Remember, 2D convolution required a filter, a data matrix (often an image) and some hyperparameters such as filter size, stride lengths, padding lengths, that define how the convolution will be performed. For a CNN, we have one extra hyperparameter that is the number of filters in a convolutional layer. Furthermore, a CNN often need use 3D convolution which is graphically illustrated in the next few slides. We will see example of 3D convolution in our next topic when we will show example CNN architectures.

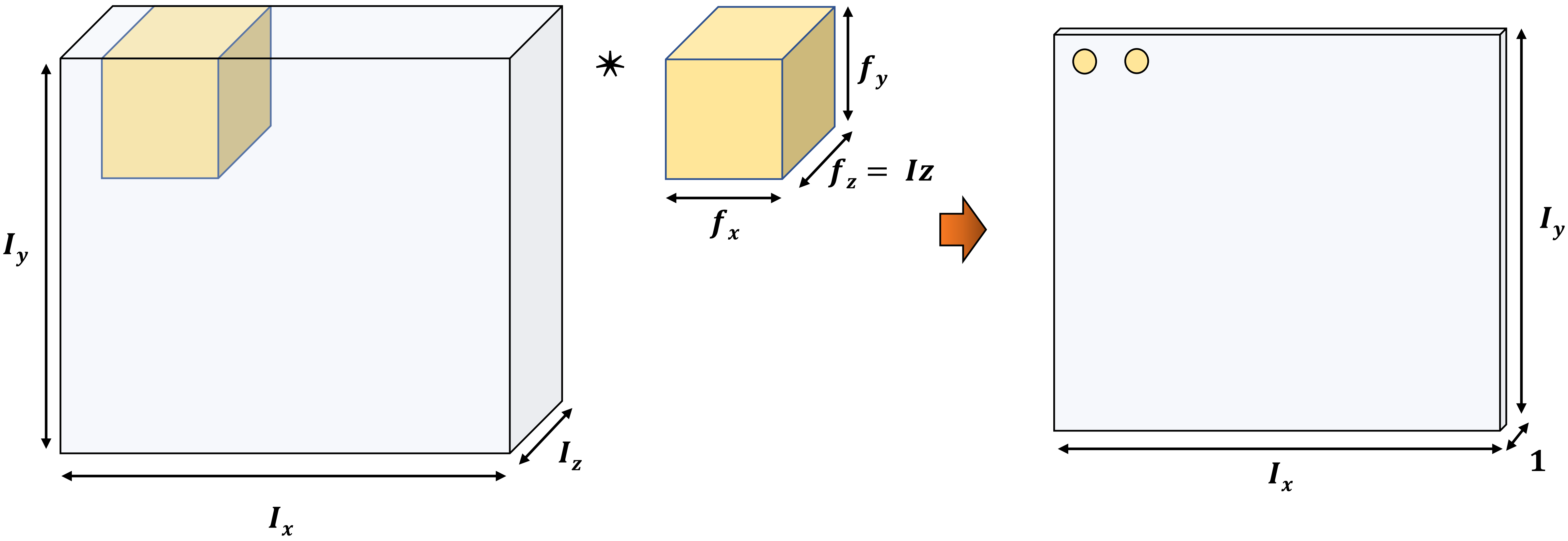
Convolution Operation:3D



3D convolution uses a 3D filter where filter depth and feature maps/Image depth is the same, i.e. $f_z = Iz$.

Each convolution operation produce only a scalar value. Therefore, convolved feature map/image has depth 1.

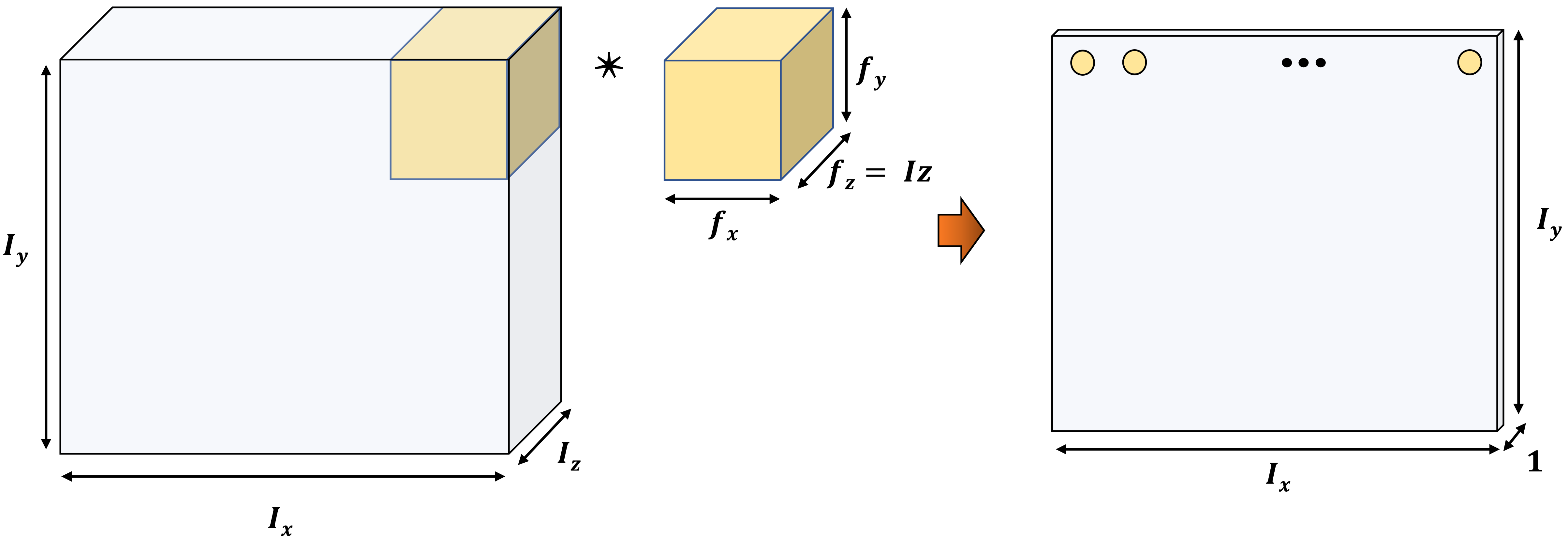
Convolution Operation:3D



3D convolution uses a 3D filter where filter depth and feature maps/Image depth is the same, i.e. $f_z = Iz$.

Each convolution operation produce only a scalar value. Therefore, convolved feature map/image has depth 1.

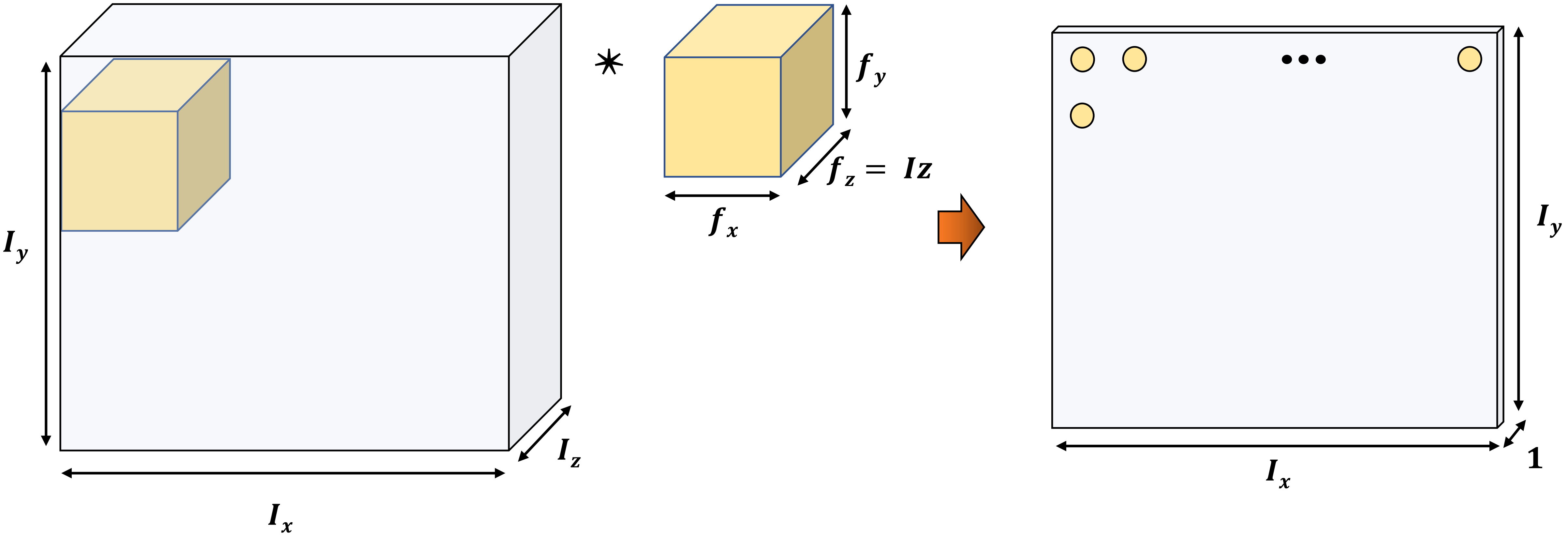
Convolution Operation:3D



3D convolution uses a 3D filter where filter depth and feature maps/Image depth is the same, i.e. $f_z = Iz$.

Each convolution operation produce only a scalar value. Therefore, convolved feature map/image has depth 1.

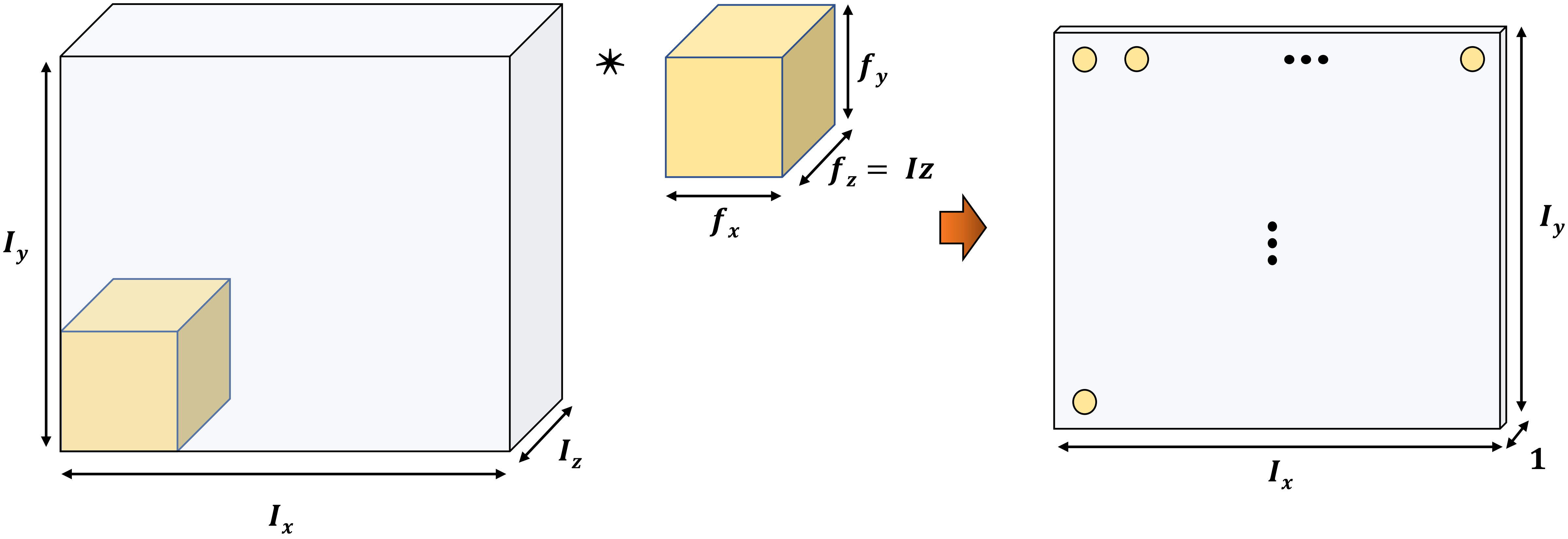
Convolution Operation:3D



3D convolution uses a 3D filter where filter depth and feature maps/Image depth is the same, i.e. $f_z = Iz$.

Each convolution operation produce only a scalar value. Therefore, convolved feature map/image has depth 1.

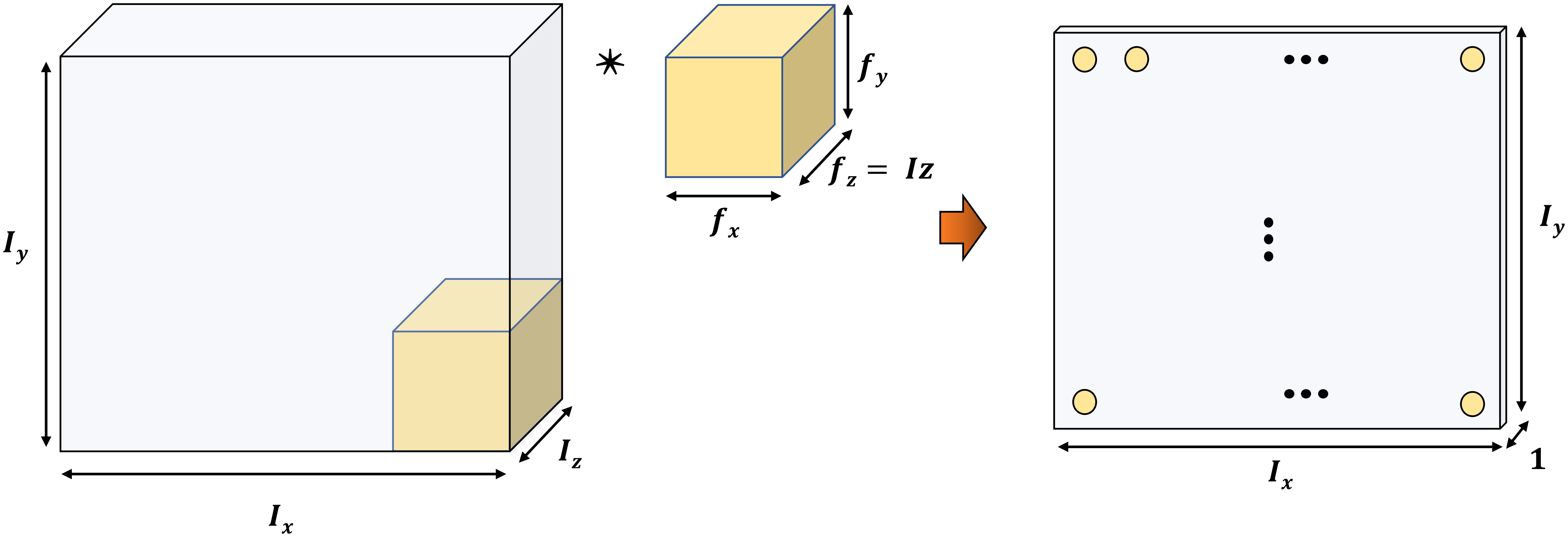
Convolution Operation:3D



3D convolution uses a 3D filter where filter depth and feature maps/Image depth is the same, i.e. $f_z = Iz$.

Each convolution operation produce only a scalar value. Therefore, convolved feature map/image has depth 1.

Convolution Operation:3D



3D convolution uses a 3D filter where filter depth and feature maps/Image depth is the same, i.e. $f_z = Iz$.

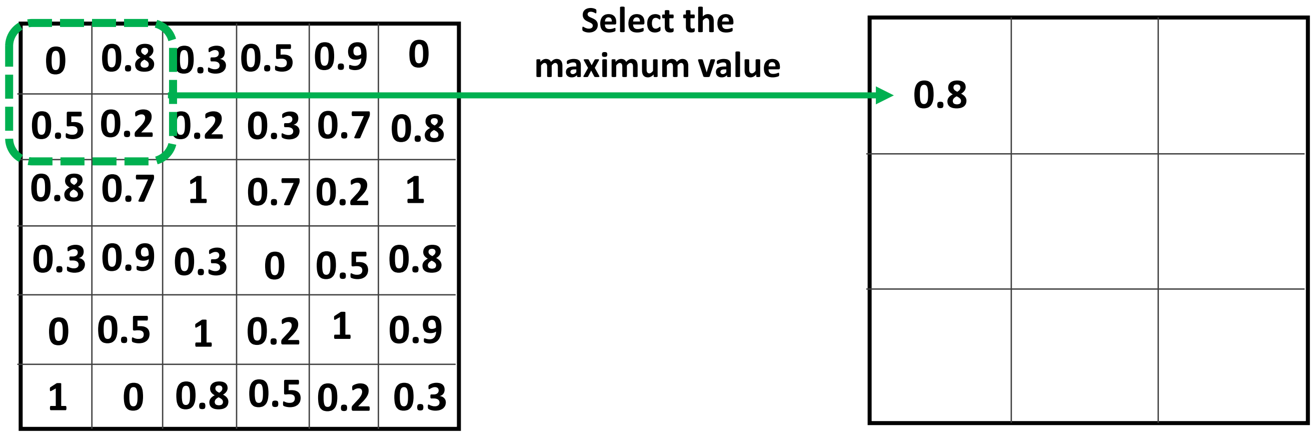
Each convolution operation produce only a scalar value. Therefore, convolved feature map/image has depth 1.

Pooling Operation

Sampling: Reduce the feature size

Output: Lower resolution version of the input feature map

Another important CNN operation is pooling that produce a number for a selected region of feature values. Thus, the process reduce the size of the feature map. The process of producing one number depends of which type of pooling method is used. Some common pooling types are max. pooling and average pooling.



0	0.8	0.3	0.5	0.9	0
0.5	0.2	0.2	0.3	0.7	0.8
0.8	0.7	1	0.7	0.2	1
0.3	0.9	0.3	0	0.5	0.8
0	0.5	1	0.2	1	0.9
1	0	0.8	0.5	0.2	0.3

Select the maximum value

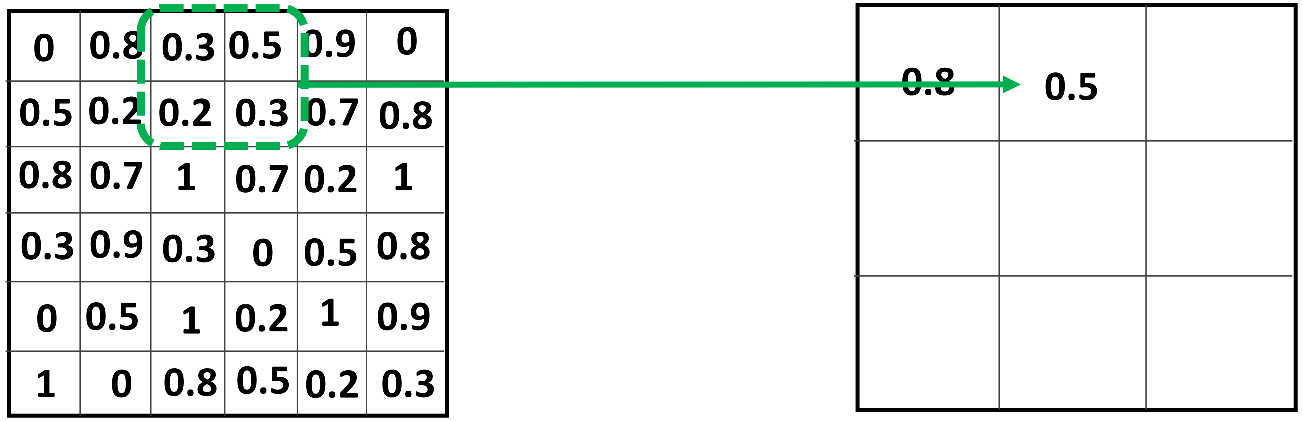
0.8		

Max pooling: select maximum value of a region

Pooling Operation

Sampling: Reduce the feature size

Output: Lower resolution version of the input feature map



0	0.8	0.3	0.5	0.9	0
0.5	0.2	0.2	0.3	0.7	0.8
0.8	0.7	1	0.7	0.2	1
0.3	0.9	0.3	0	0.5	0.8
0	0.5	1	0.2	1	0.9
1	0	0.8	0.5	0.2	0.3

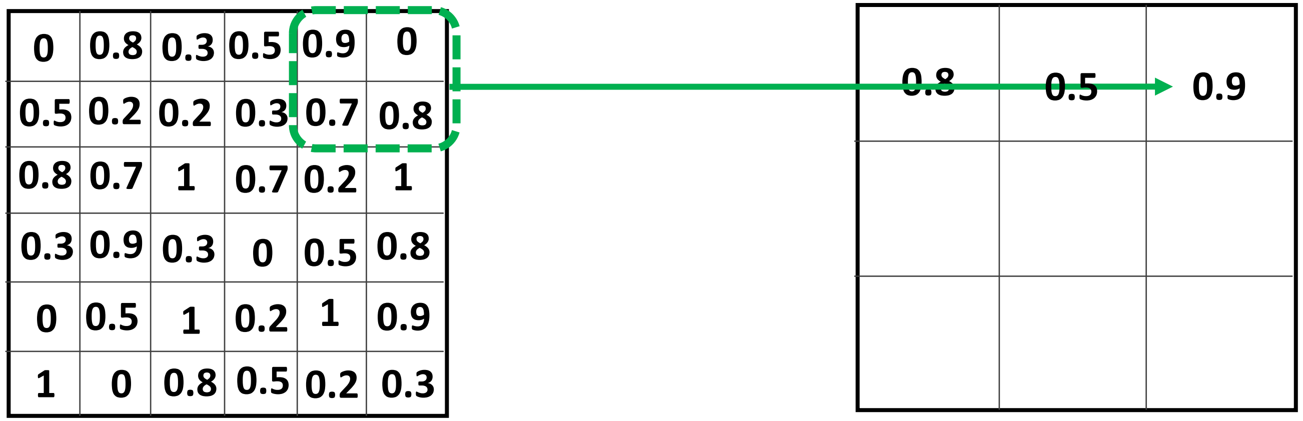
0.8	0.5	

Max pooling: select maximum value of a region

Pooling Operation

Sampling: Reduce the feature size

Output: Lower resolution version of the input feature map



0	0.8	0.3	0.5	0.9	0
0.5	0.2	0.2	0.3	0.7	0.8
0.8	0.7	1	0.7	0.2	1
0.3	0.9	0.3	0	0.5	0.8
0	0.5	1	0.2	1	0.9
1	0	0.8	0.5	0.2	0.3

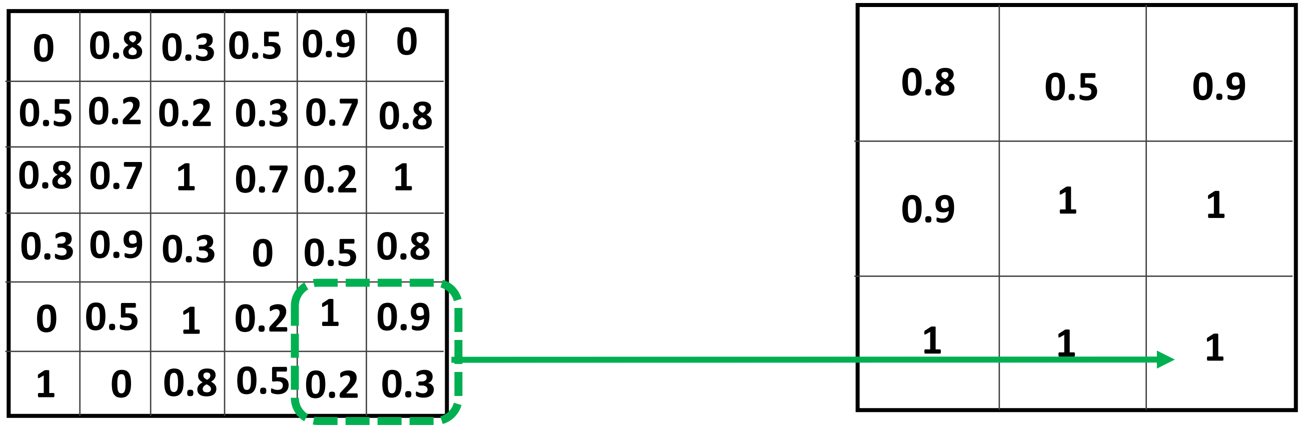
0.8	0.5	0.9

Max pooling: select maximum value of a region

Pooling Operation

Sampling: Reduce the feature size

Output: Lower resolution version of the input feature map



0	0.8	0.3	0.5	0.9	0
0.5	0.2	0.2	0.3	0.7	0.8
0.8	0.7	1	0.7	0.2	1
0.3	0.9	0.3	0	0.5	0.8
0	0.5	1	0.2	1	0.9
1	0	0.8	0.5	0.2	0.3

0.8	0.5	0.9
0.9	1	1
1	1	1

Max pooling: select maximum value of a region

Average pooling: select average value of a region. A variation of pooling technique

Hyperparameters: filter size (here 2x2), stride (here 2), type of pooling (max or avg. pooling)

CNN vs MLP: Image Classification

Trainable parameters:

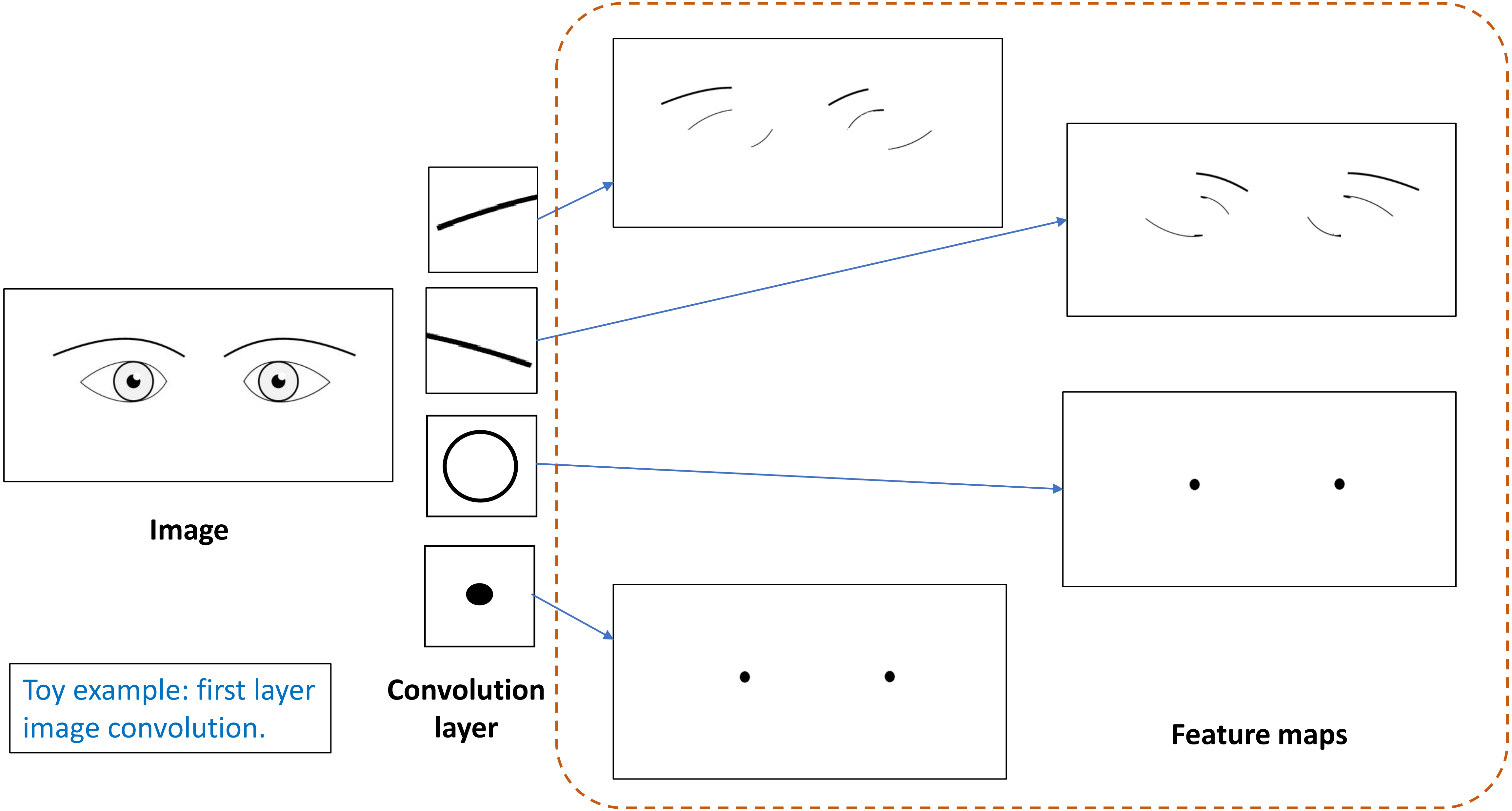
- For image pixels MLP requires a input neuron/node, compared to that CNN reuses a fixed number of filters over the spatial region
- Training more parameters are difficult
- A model with a higher number of parameters requires more data for generalized learning

Spatial information:

- MLP flattens an image to a vector, CNN keeps the image structure intact
- MLP loses local spatial information when flattens the input
- CNN can treat local correlations in the input more effectively

Please read CNN vs MLP for image classification

CNN: Deeper Understanding



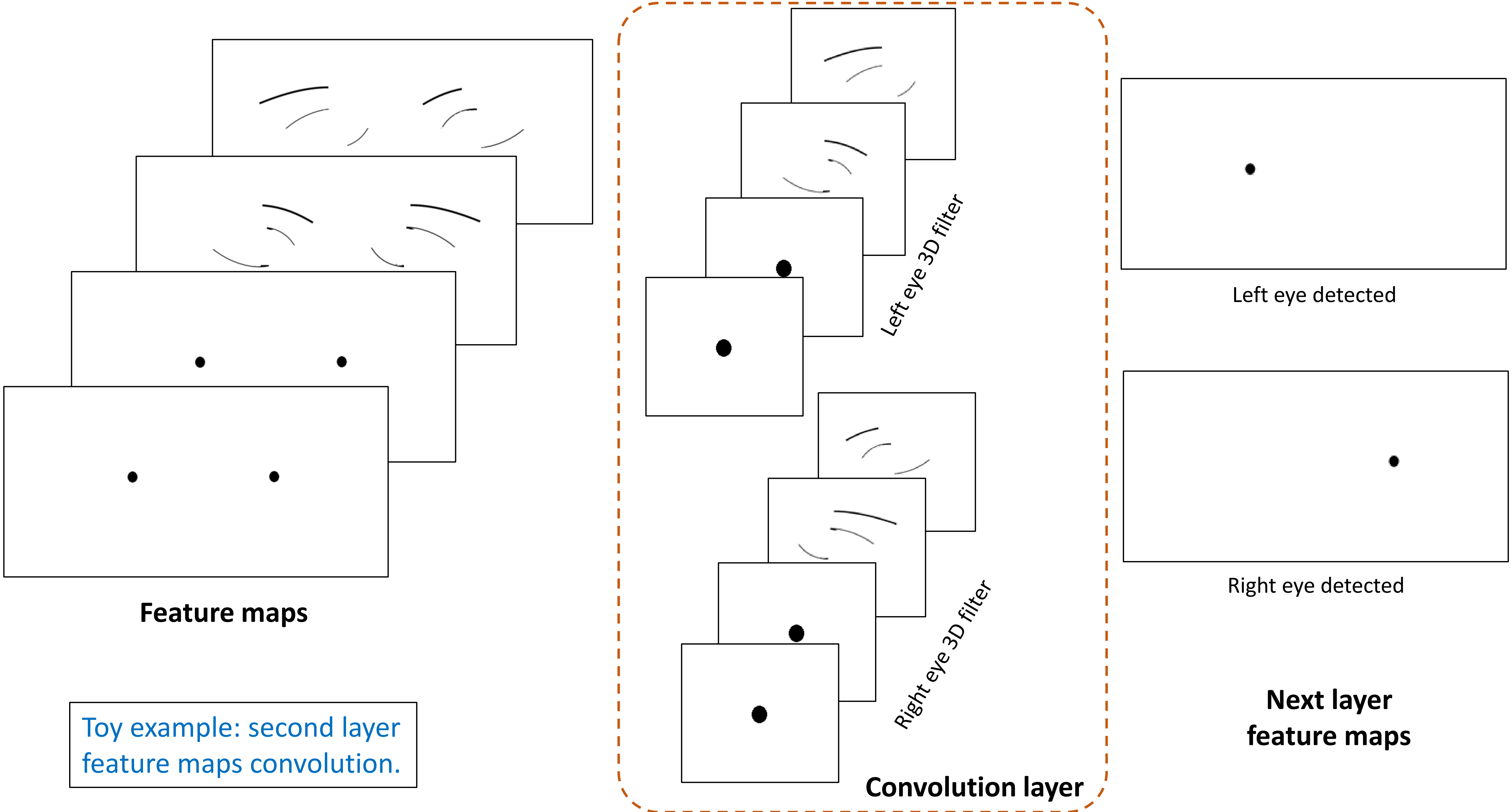
Image

Convolution layer

Feature maps

Toy example: first layer image convolution.

CNN: Deeper Understanding



Toy example: second layer feature maps convolution.

CNN: Deeper Understanding

A CNN learns classification by hierarchical learning of low level to high level image features. For example, a CNN trained for face detection might learn to detect feature like edges, color gradient etc. in the first layer, comparative complex features such as eyebrows, lips in the next layer, and so on. In the final layer (output layer), the CNN is able to combine the information from the previous layers for face detection. Again, this is just a toy example to explain the CNN functioning.

Activation layers in the CNN add the ability to fit a non-linear decision boundary

Pooling layers help to reduce the dimension of the feature maps. We will mainly concentrate on max. pooling. Polling layers help in the followings.

- Preserving information of features detected in the last layer while removing unimportant details
- Larger receptive field for the next convolutional layer facilitate higher level feature detection
- Provide invariance to small translations of the input

Additional Readings

A very intuitive introduction to CNN

CNN vs MLP for image classification