LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

Introduction to Deep Learning

Chapter 4: CNN: Architecture

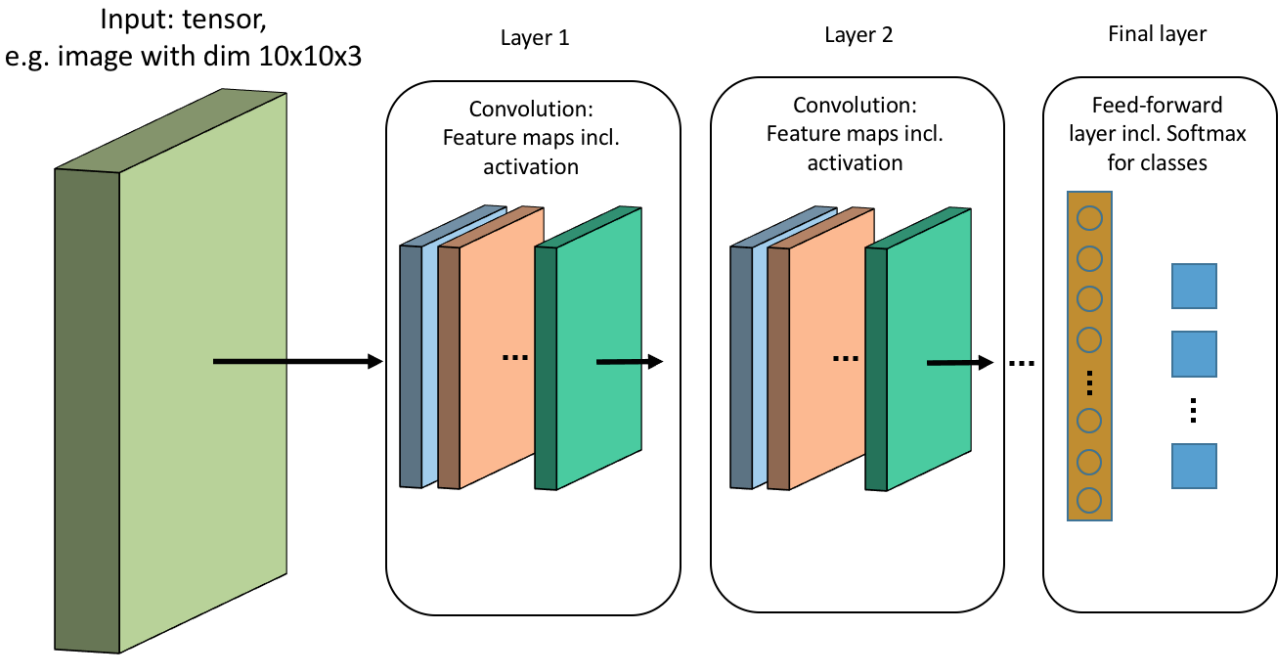
Bernd Bischl

Department of Statistics – LMU Munich

WS 2021/2022

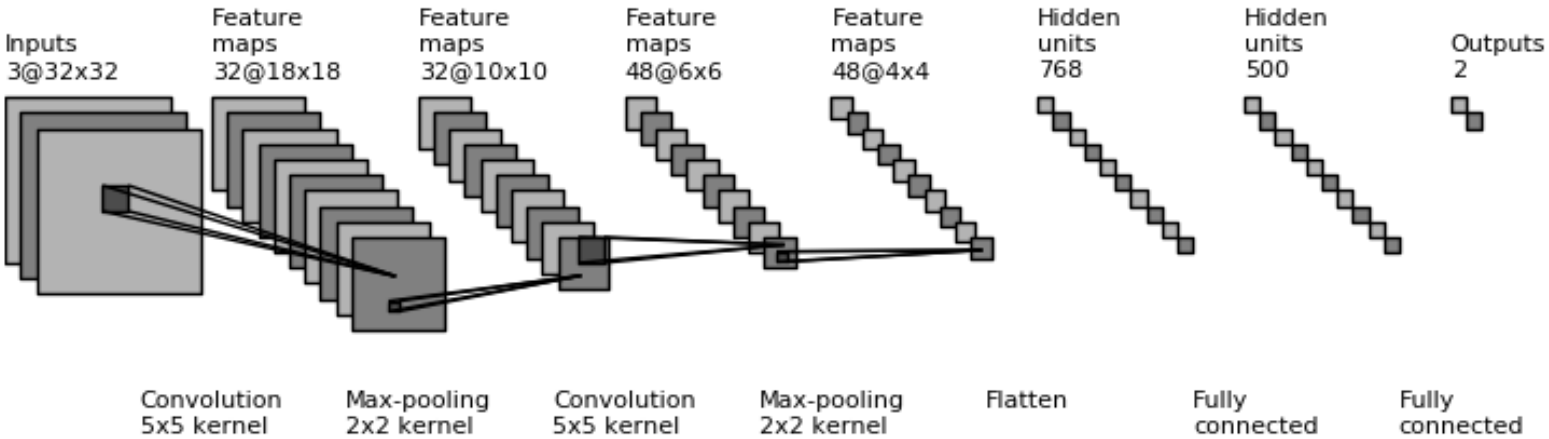
Different Perspectives of CNNs

CNNS - PERSPECTIVE I



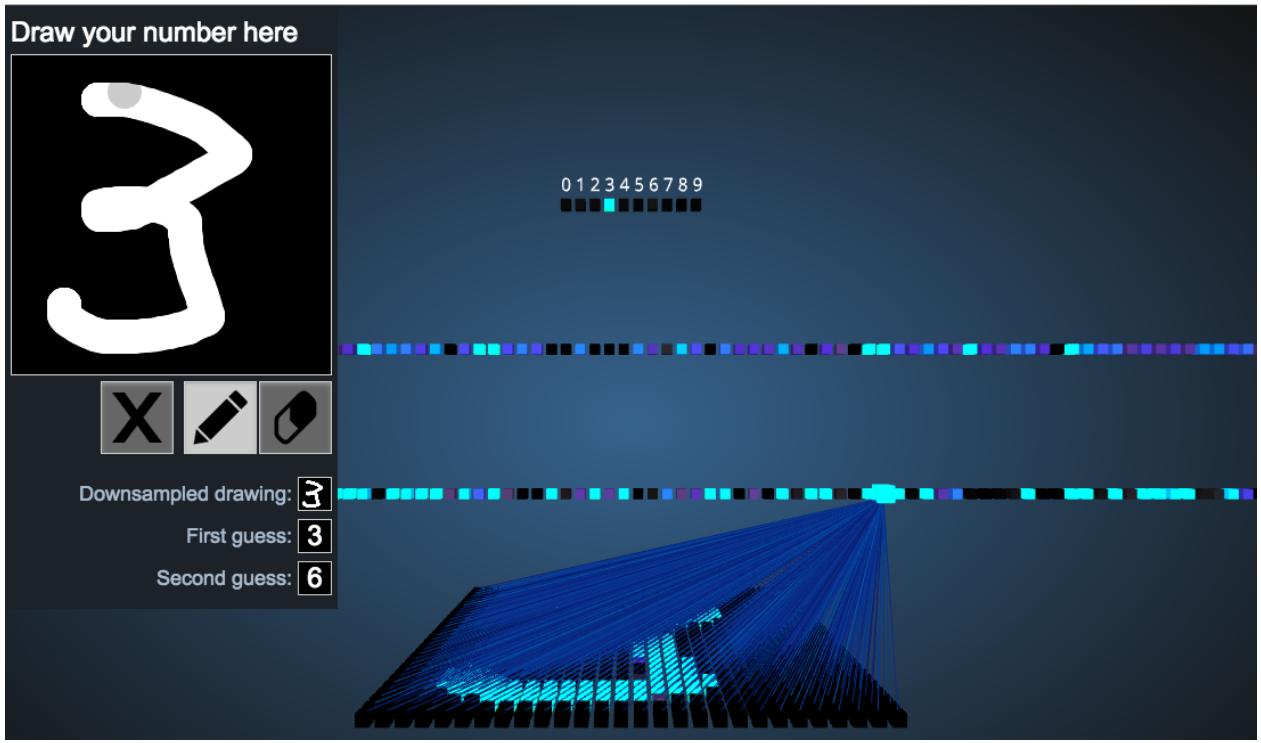
- Schematic architecture of a CNN.
- The input tensor is convolved by different filters yielding different feature maps (coloured) in subsequent layers.
- A dense layer connects the final feature maps with the softmax-activated output neurons.

CNNS - PERSPECTIVE II



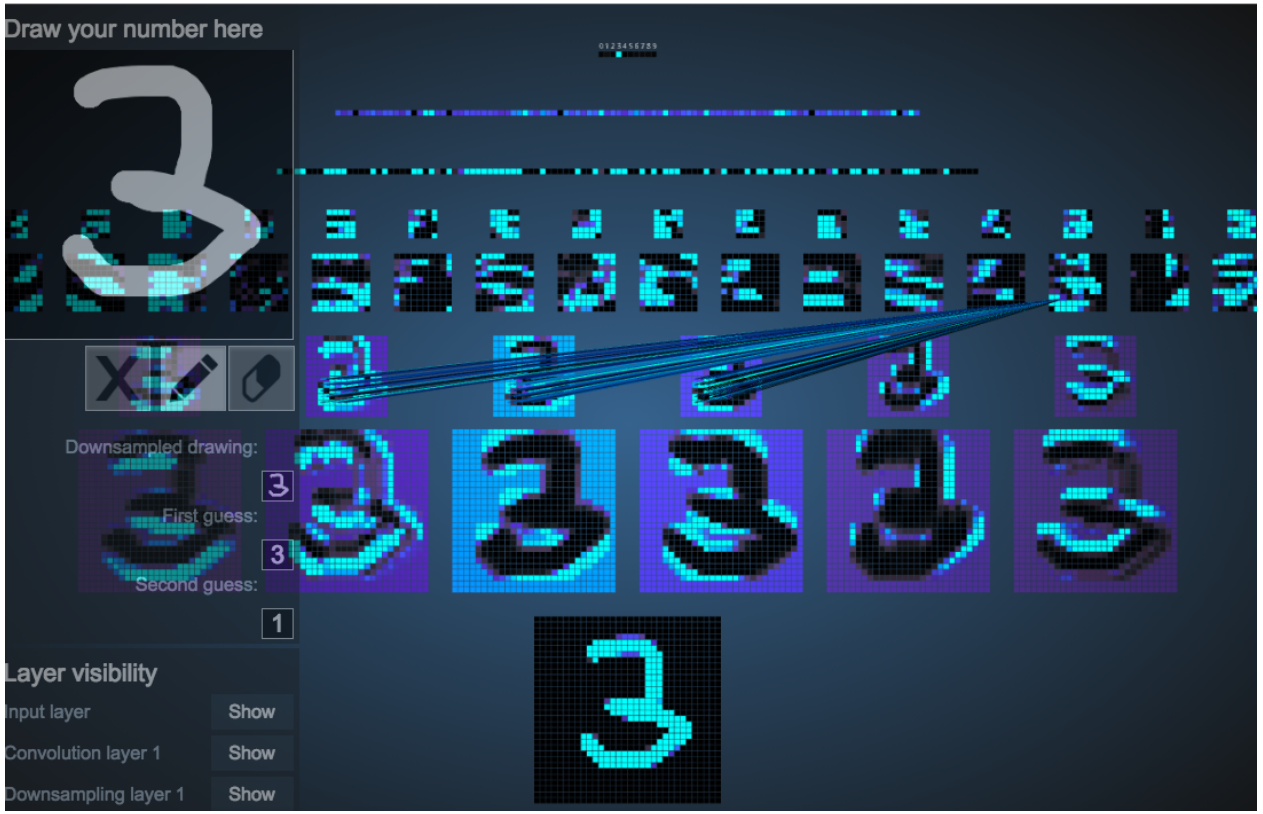
- Flat view of a CNN architecture for a classification problem.
- Consists of 2 CNN layers that are each followed by max-pooling, then flattened and connected with the final output neurons via a dense layer.

CNNS - PERSPECTIVE III



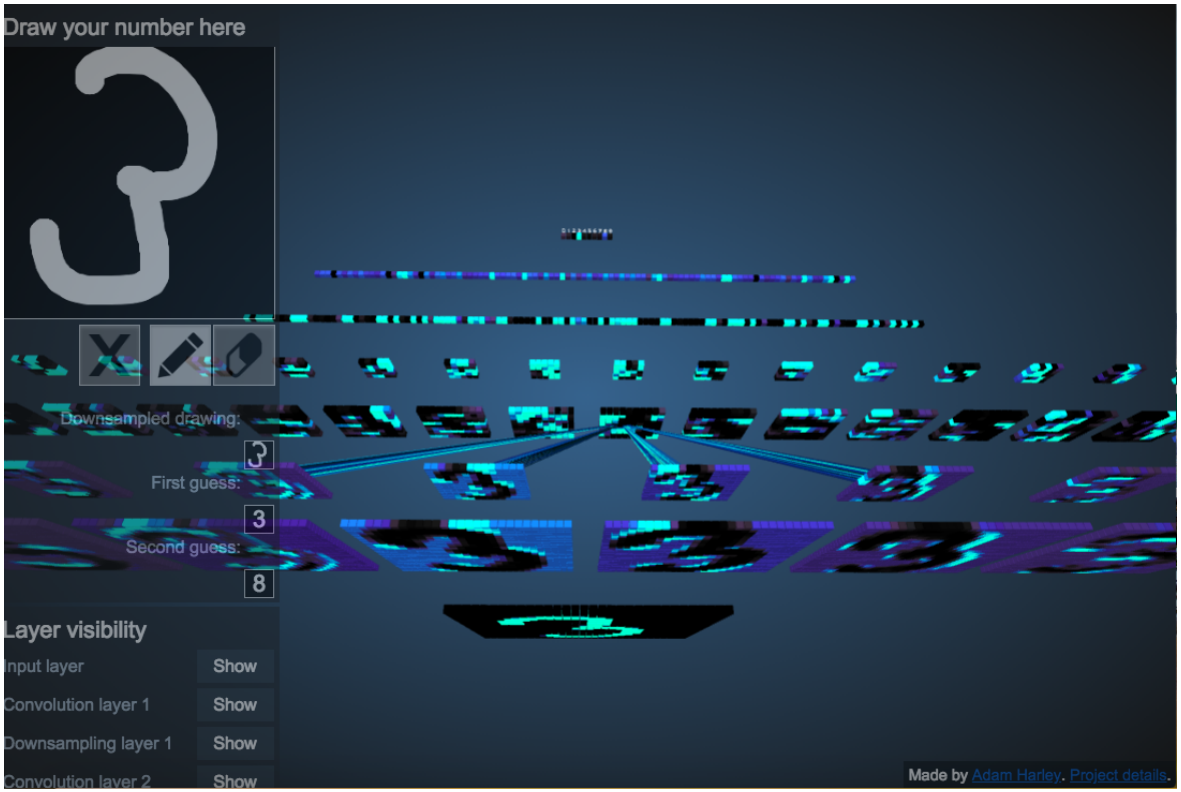
- Awesome interactive visualization (by Adam Harley) ▸ Click here.
- Vanilla 2-layer densely-connected net on MNIST data for input digit 3.
- Each neuron in layer 1 is connected to each of the input neurons.

CNNS - PERSPECTIVE III



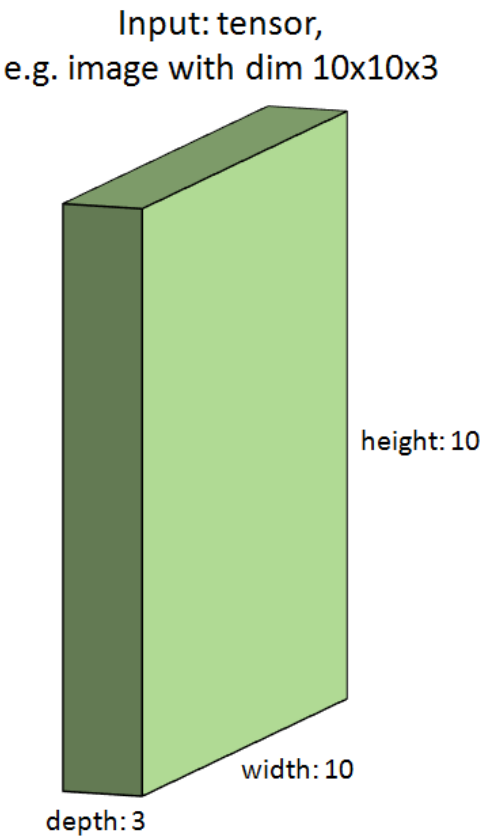
- Front view on 2-layer CNN with Pooling and final dense layer on MNIST data for input digit 3.
- Each neuron in the second CNN layer is connected to a patch of neurons from each of the previous feature maps via the convolutional kernel.

CNNS - PERSPECTIVE III



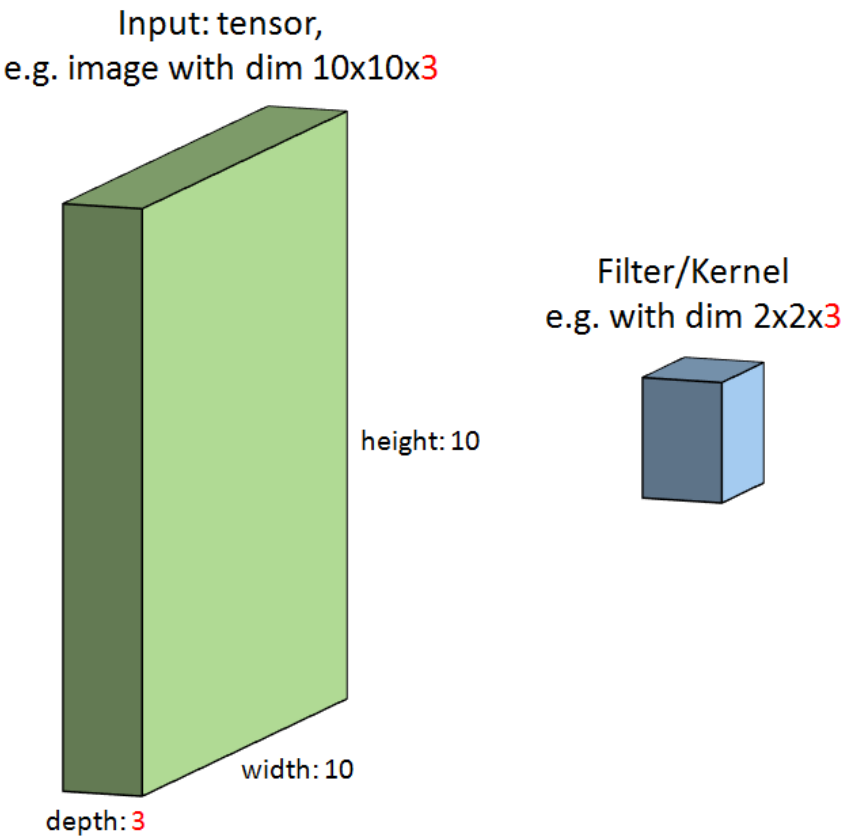
- Bottom view on 2-layer CNN with Pooling and final dense layer on MNIST data for input digit 3.
- Each neuron in the second CNN layer is connected to a patch of neurons from each of the previous feature maps via the convolutional kernel.

CNNS - ARCHITECTURE



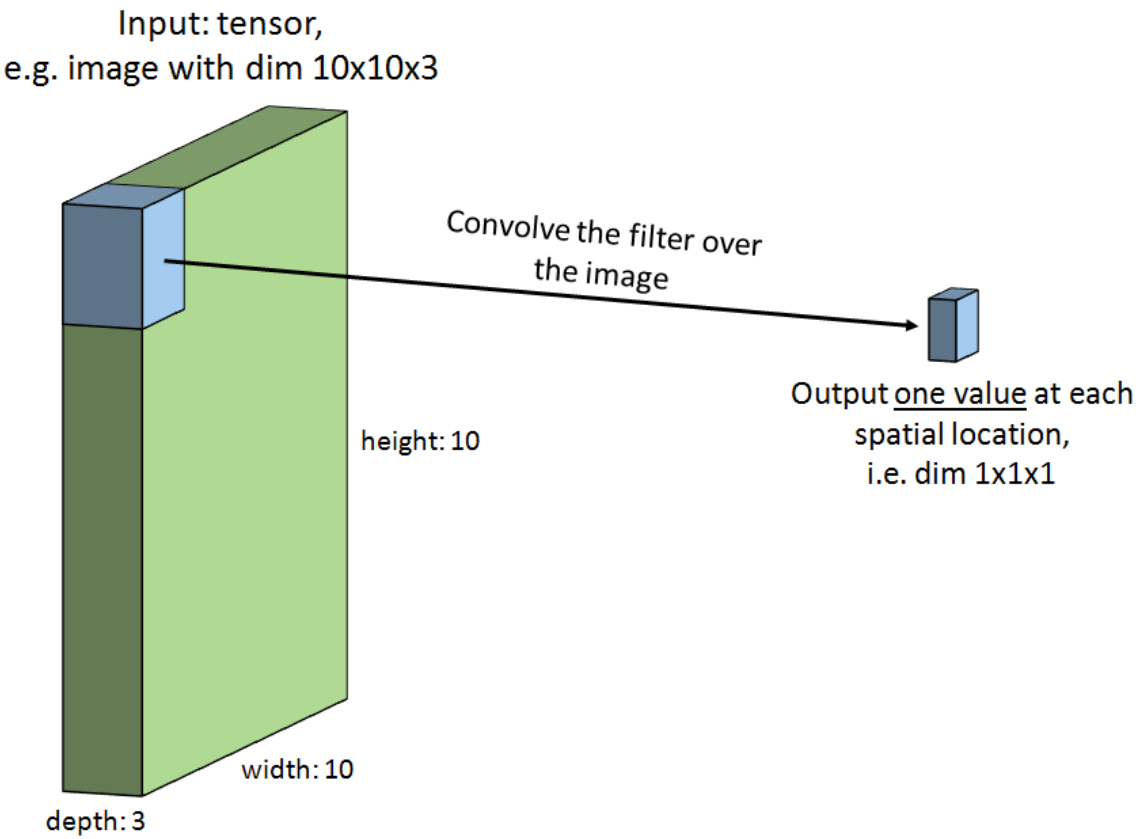
- Suppose we have the following input tensor with dimensions $10 \times 10 \times 3$.

CNNS - ARCHITECTURE



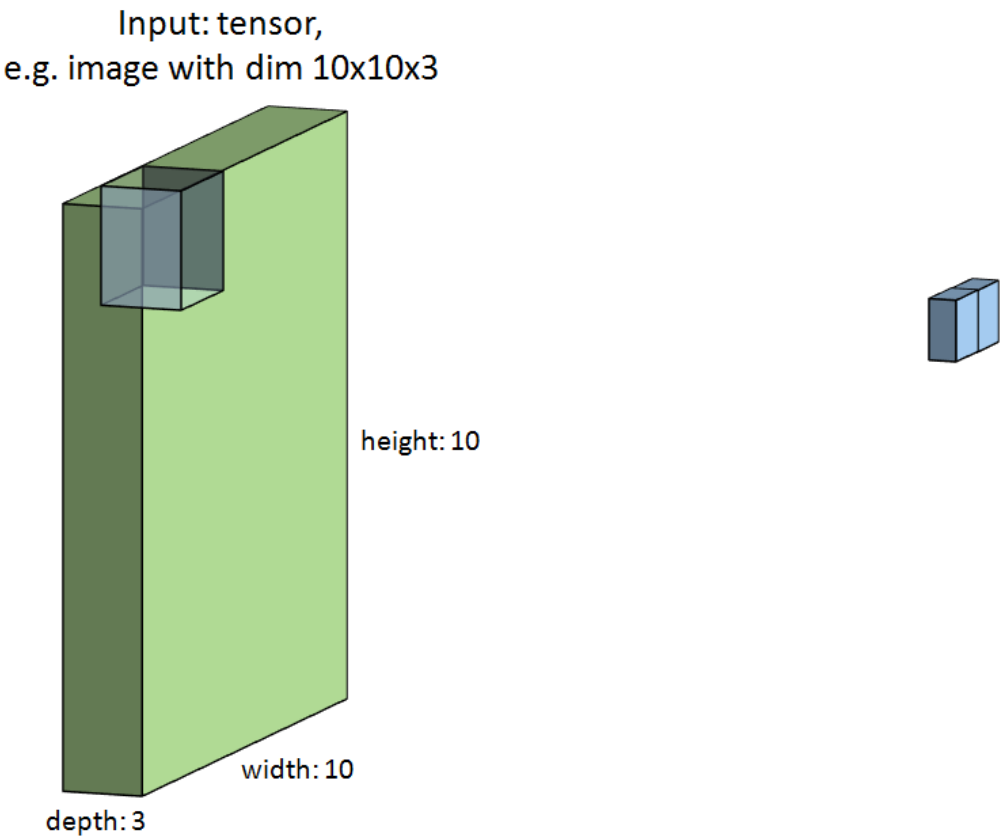
- We use a filter of size 2.

CNNS - ARCHITECTURE



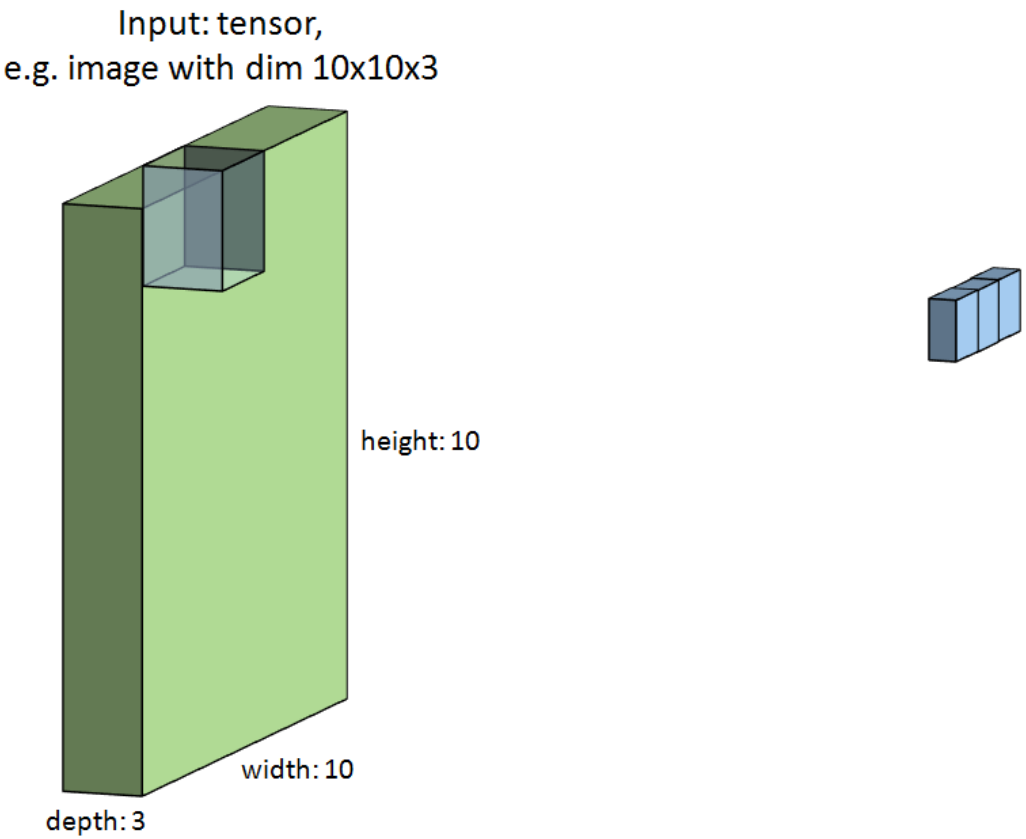
- Applying it to the first spatial location, yields one scalar value.

CNNS - ARCHITECTURE



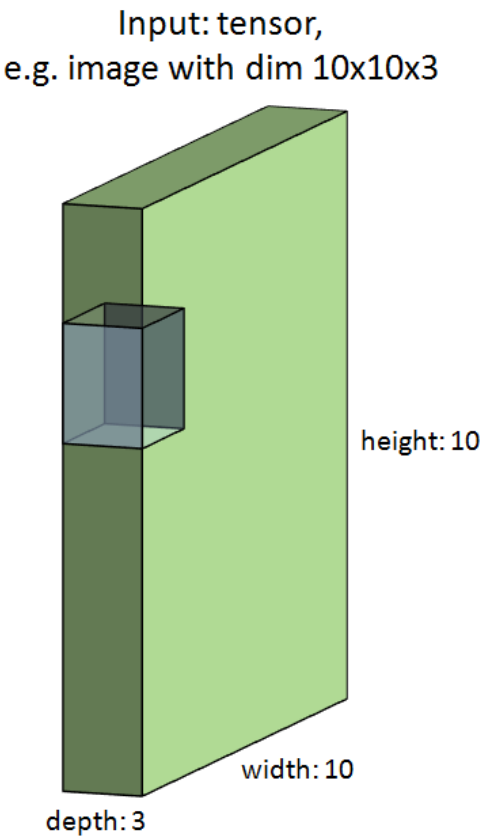
- The second spatial location yields another one..

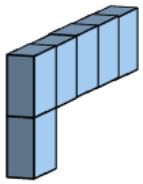
CNNS - ARCHITECTURE



- ...and another one...

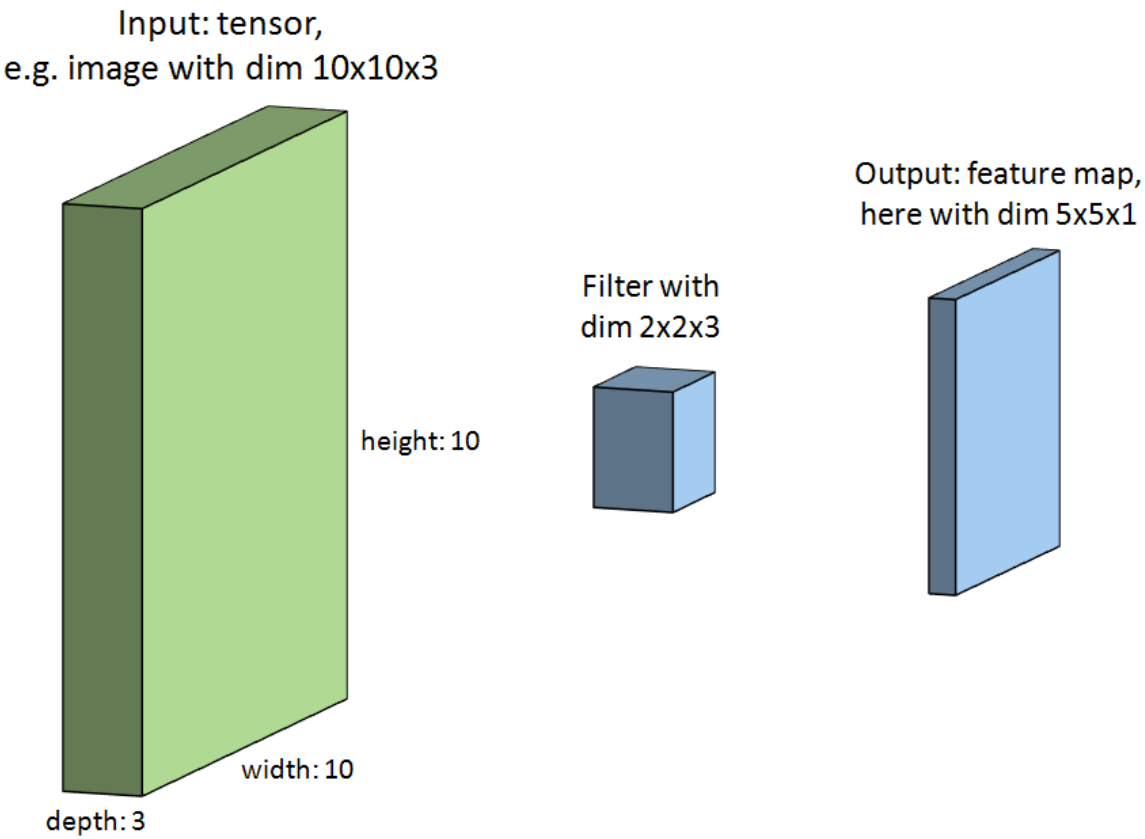
CNNS - ARCHITECTURE





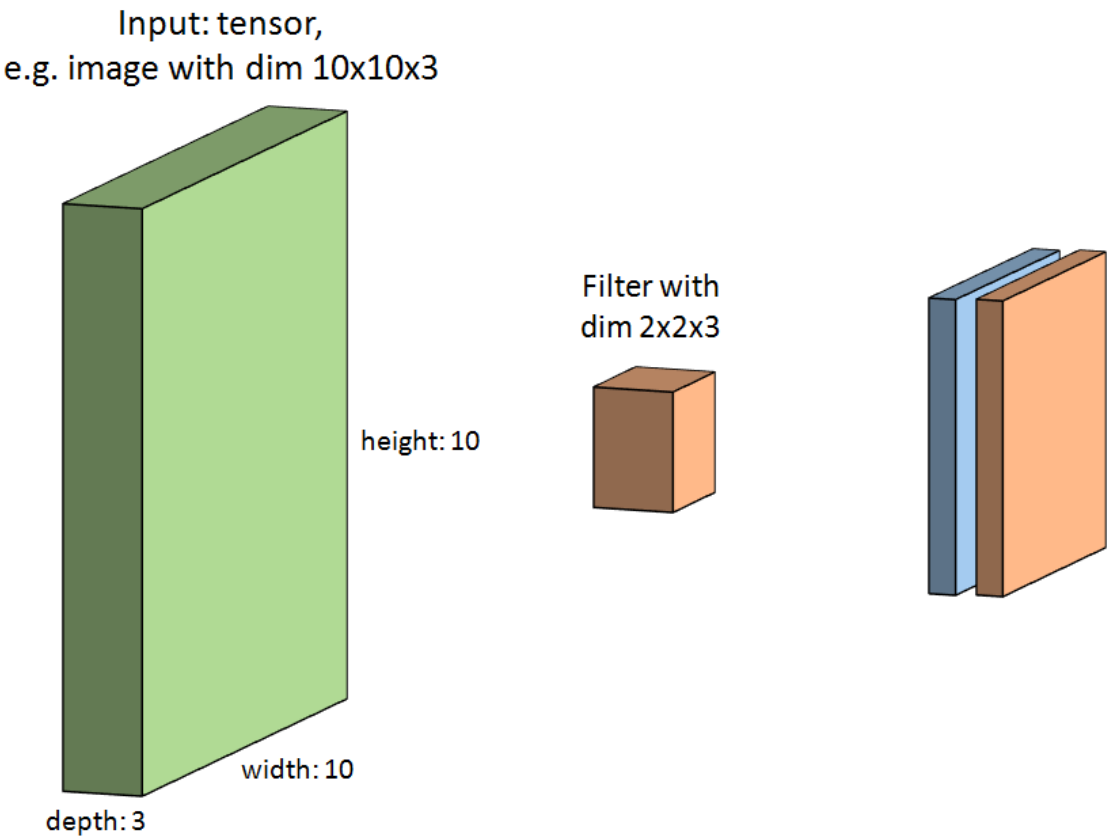
- ...and another one...

CNNS - ARCHITECTURE



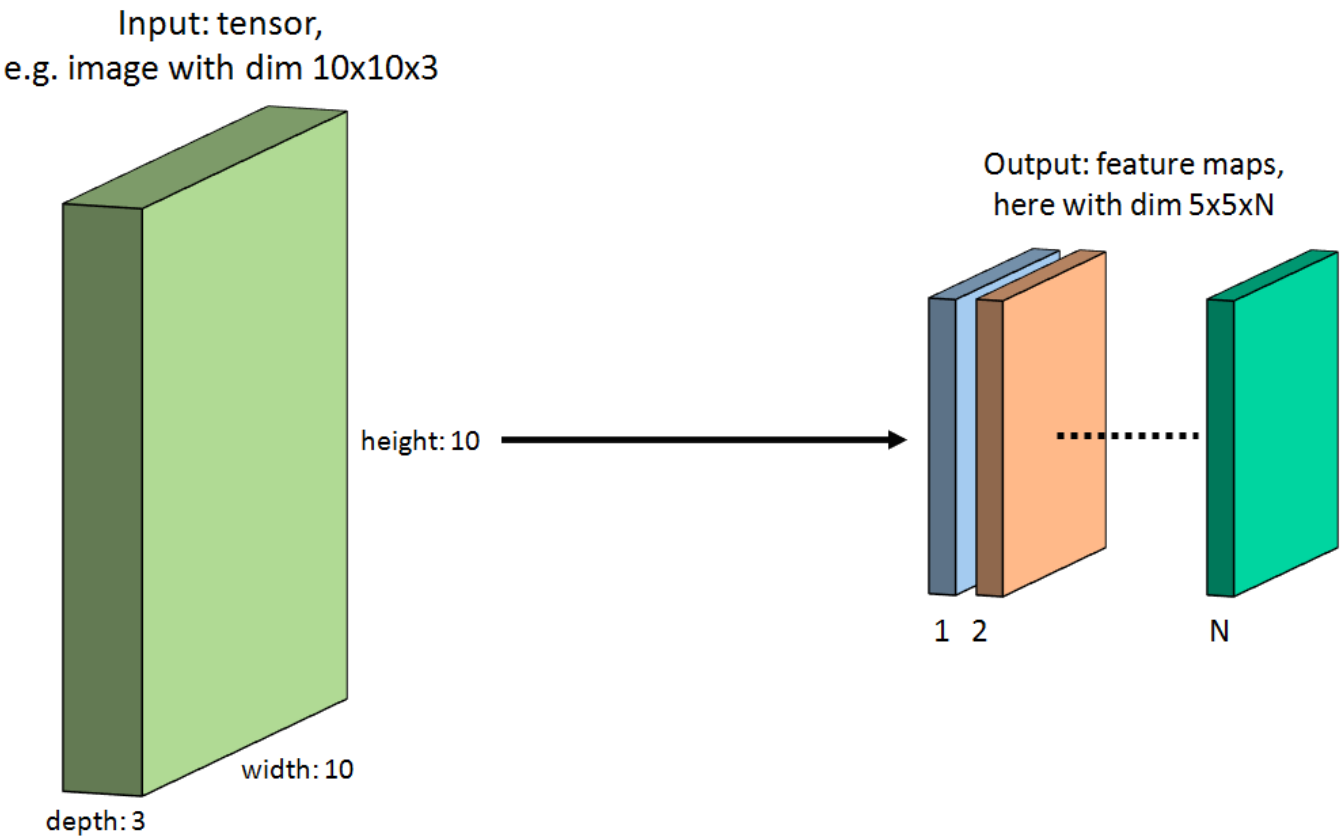
- Finally we obtain an output which is called feature map.

CNNS - ARCHITECTURE



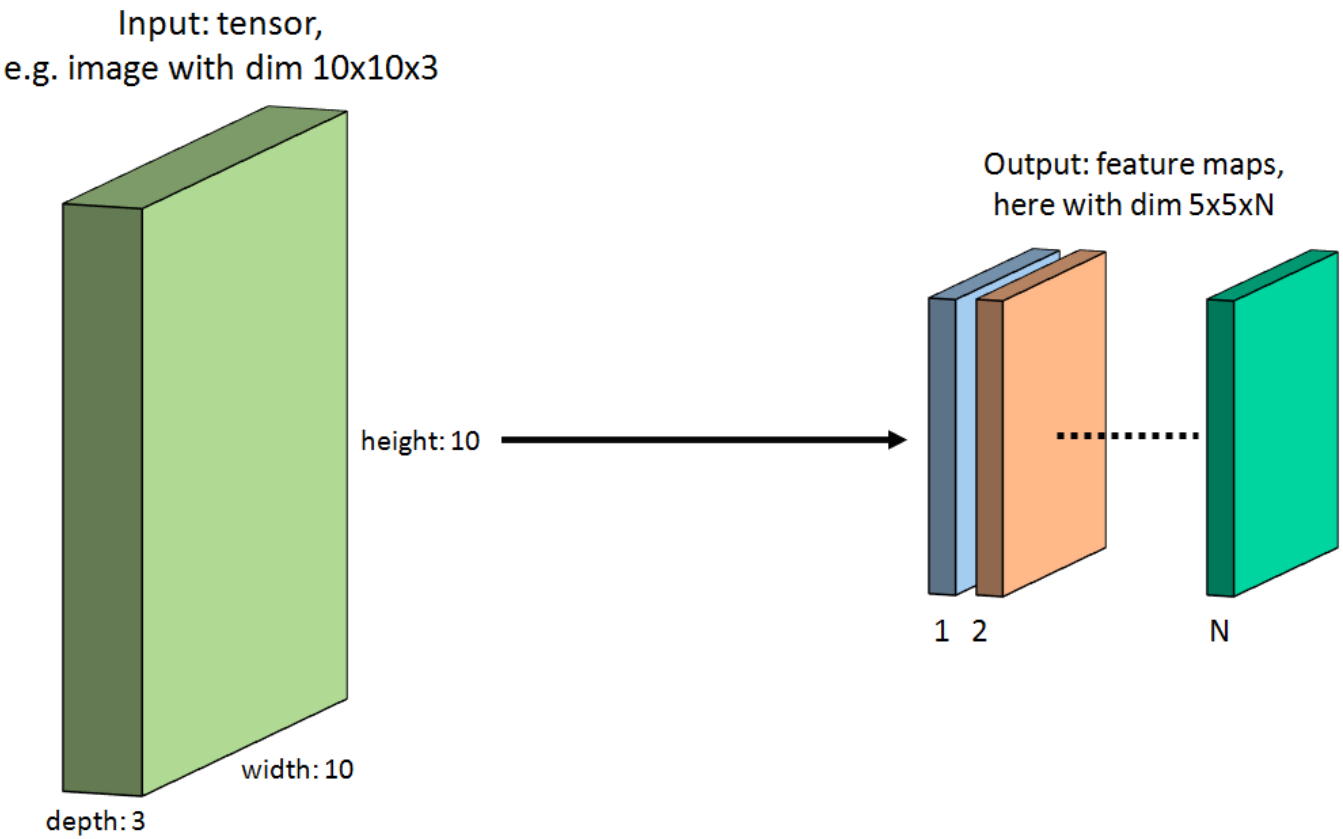
- We initialize another filter to obtain a second feature map.

CNNS - ARCHITECTURE



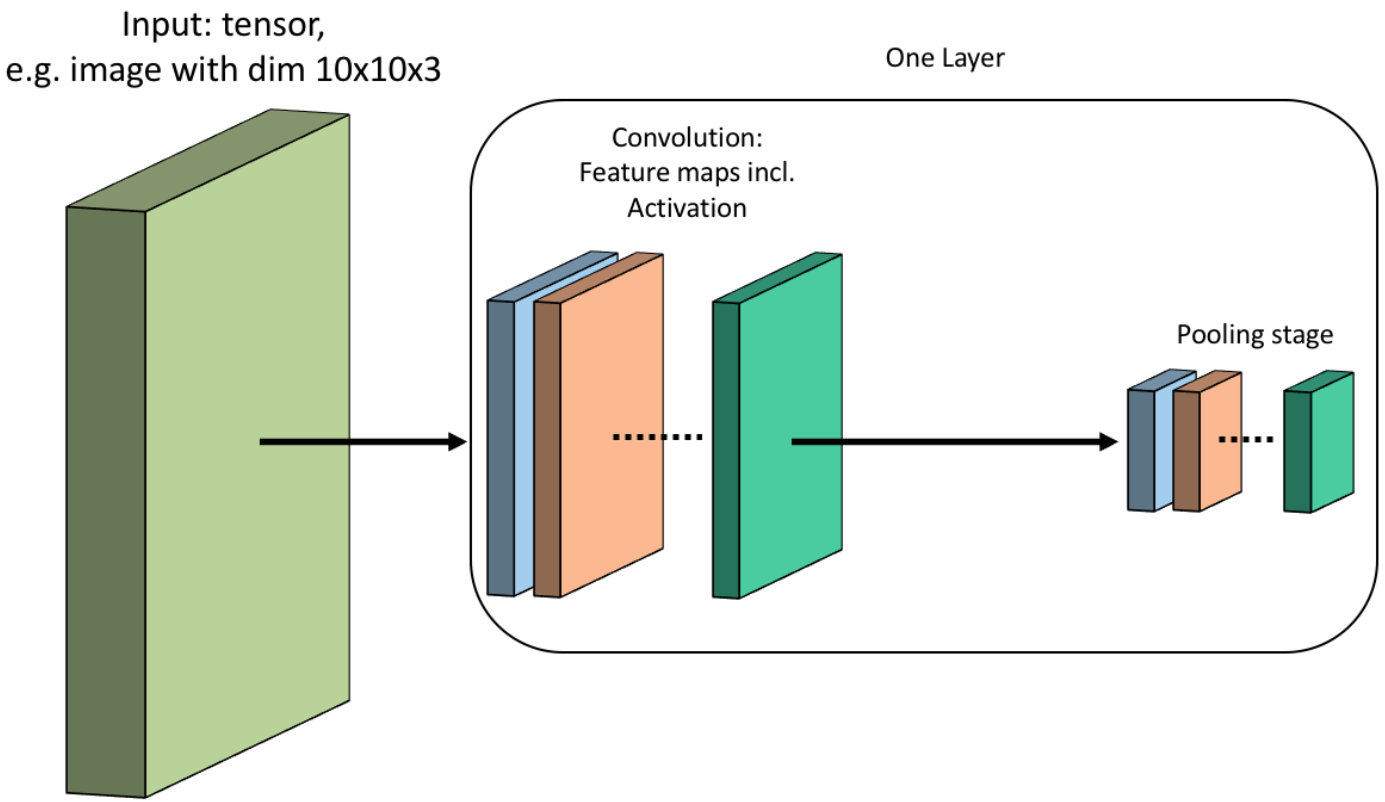
- All feature maps yield us a “new image” with dim $h \times w \times N$.

CNNS - ARCHITECTURE



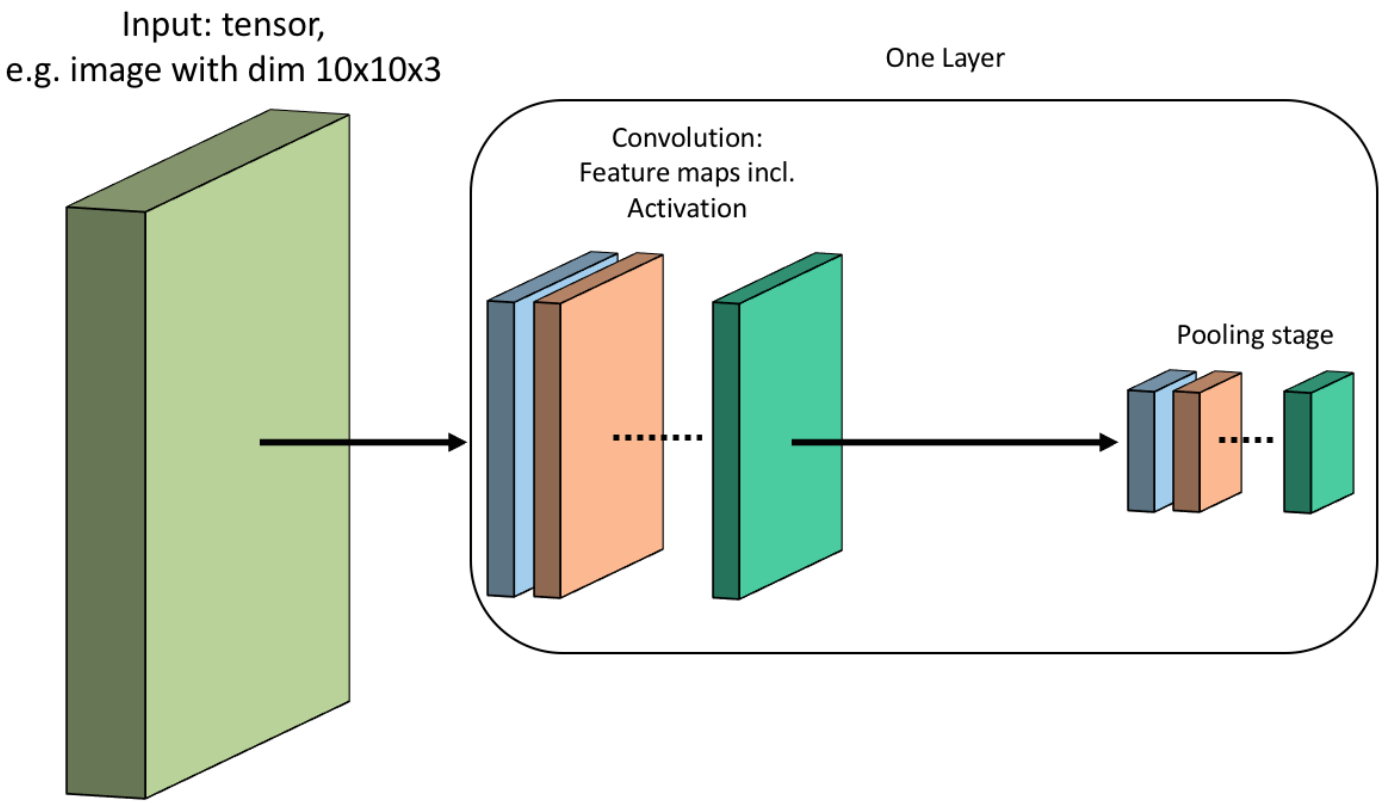
- We actually append them to a new tensor with depth = # filters.

CNNS - ARCHITECTURE



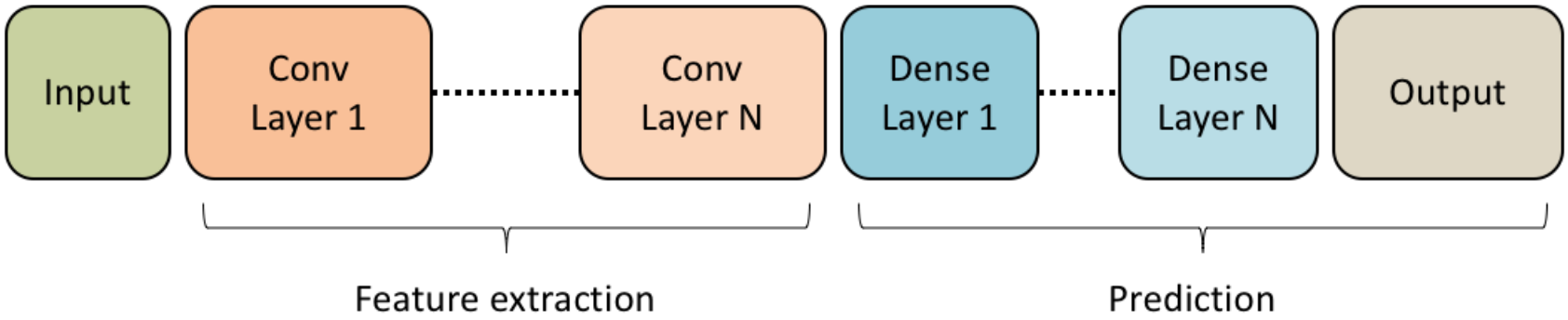
- All feature map entries will then be activated (e.g. via ReLU), just like the neurons of a standard feedforward net.

CNNS - ARCHITECTURE



- One may use pooling operations to downsample the dimensions of the feature maps.
- Pooling is applied on each feature map independently: the latter, blue block is the pooled version of the previous, blue feature map.

CNNS - ARCHITECTURE



- Many of these layers can be placed successively, to extract even more complex features.
- The feature maps are fed into each other sequentially. For instance, each filter from the second conv layer gets all previous feature maps from the first conv layer as an input. Each filter from the first layer extracts information from the input image tensor.
- The feature maps of the final conv layer are flattened (into a vector) and fed into a dense layer which, in turn, is followed by more dense layers and finally, the output layer.