

EVD 3

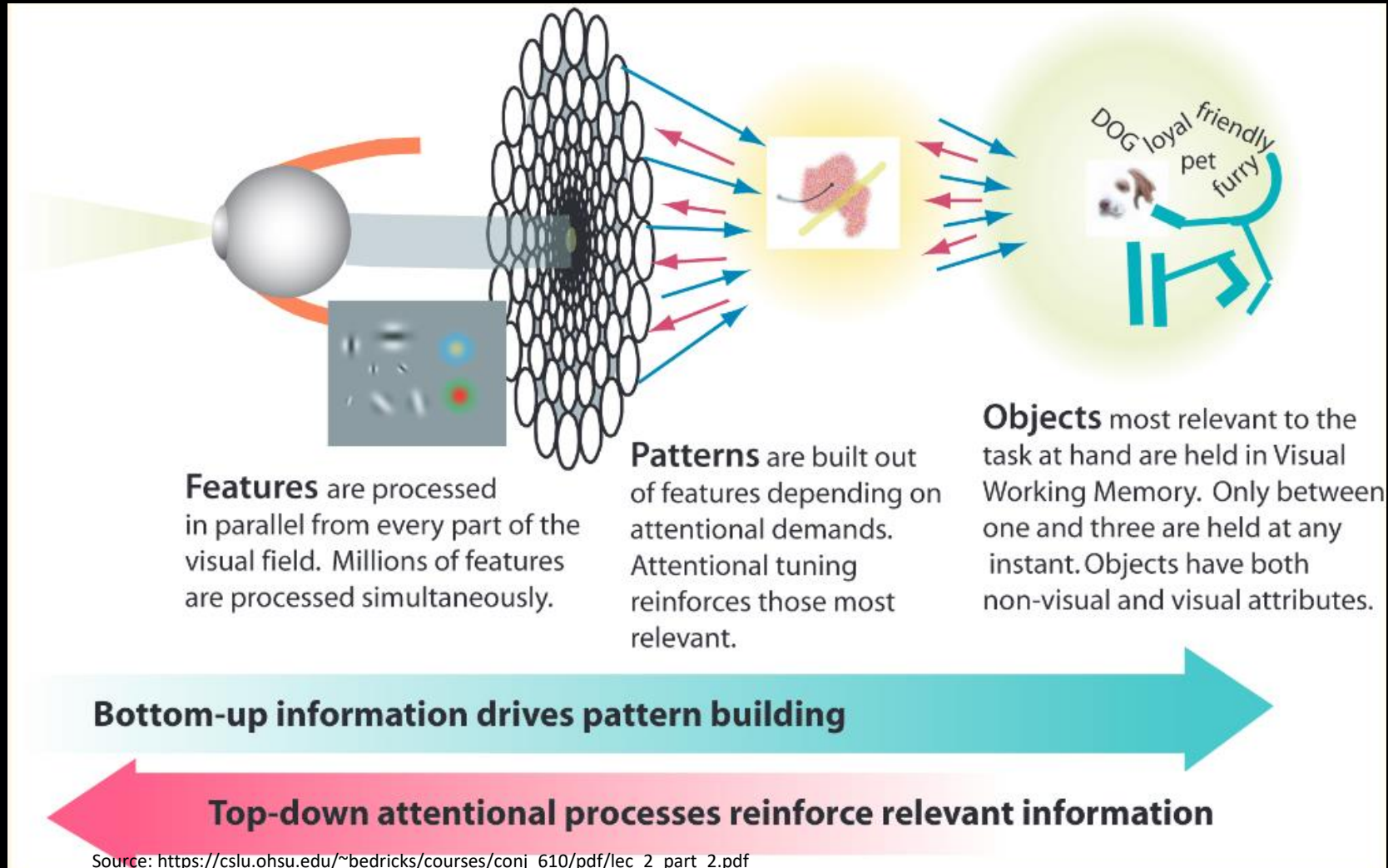
CONVOLUTIONAL NEURAL NETWORKS

JEROEN VEEN

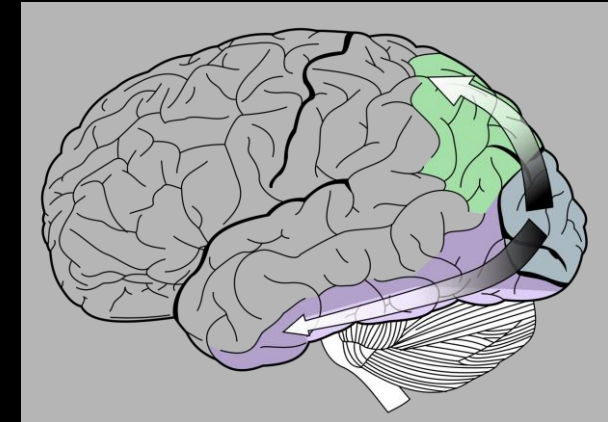
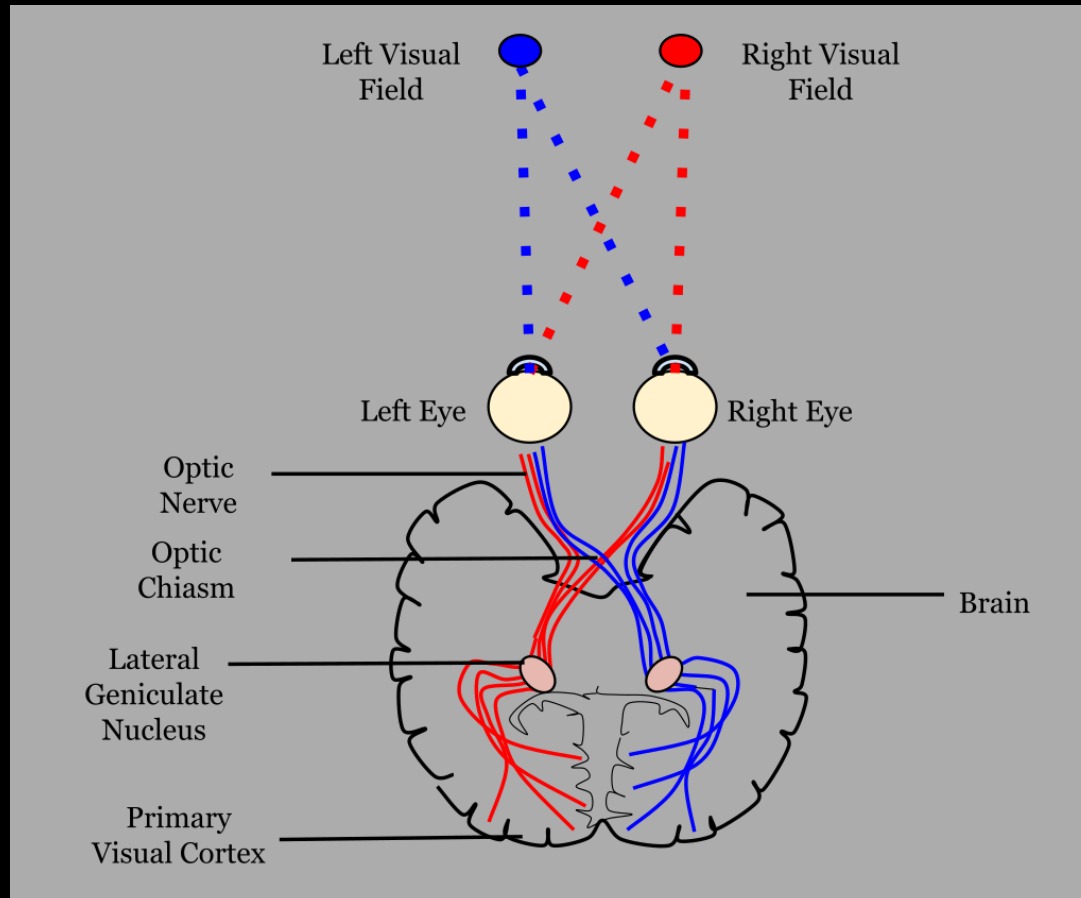
AGENDA

- Inspiration from the visual cortex
- CNN vs MLP
- Recap convolution
- Convolutional layer
- Pooling layer
- CNN architecture

INSPIRATION FROM THE VISUAL CORTEX



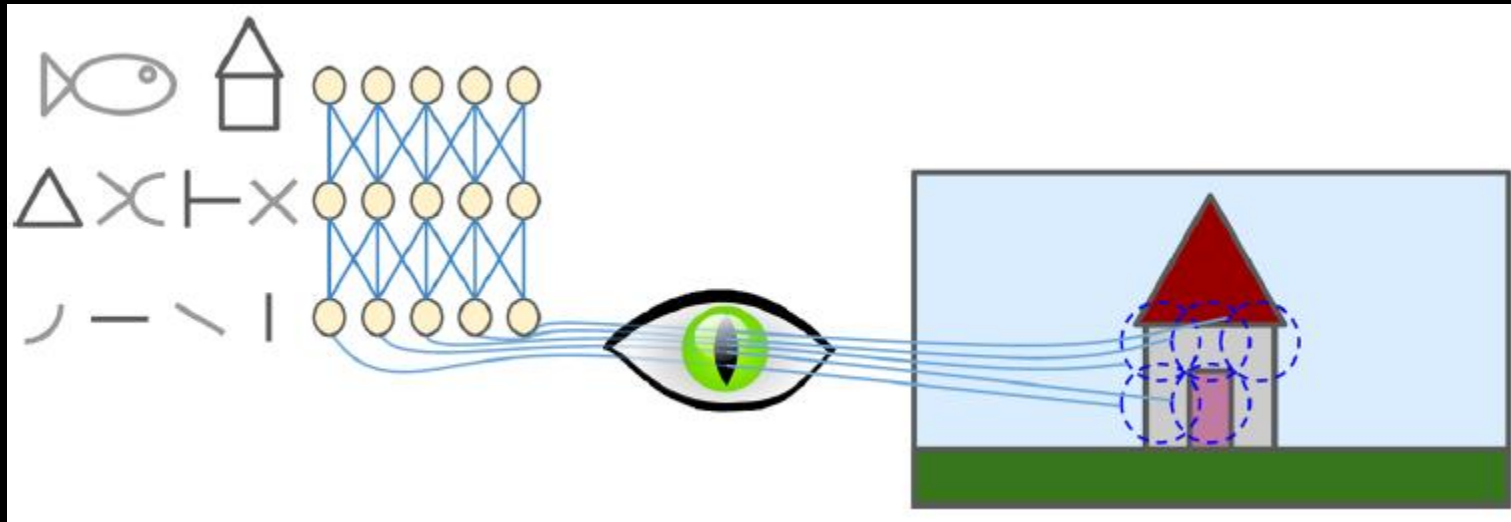
INSPIRATION FROM THE VISUAL CORTEX



Source: By Selket - I (Selket) made this from File:Gray728.svg, CC BY-SA 3.0, <https://commons.wikimedia.org/w/index.php?curid=1679336>

Source: https://en.wikipedia.org/wiki/Visual_cortex

LOCAL RECEPTIVE FIELDS

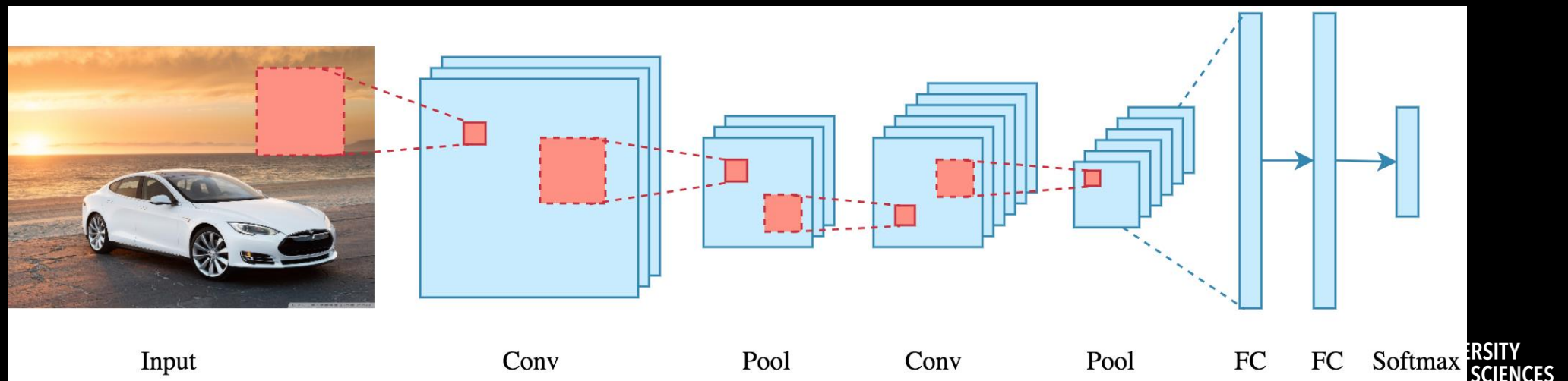


Source: Géron, ISBN: 9781492032632

- Each neuron has a different receptive field.

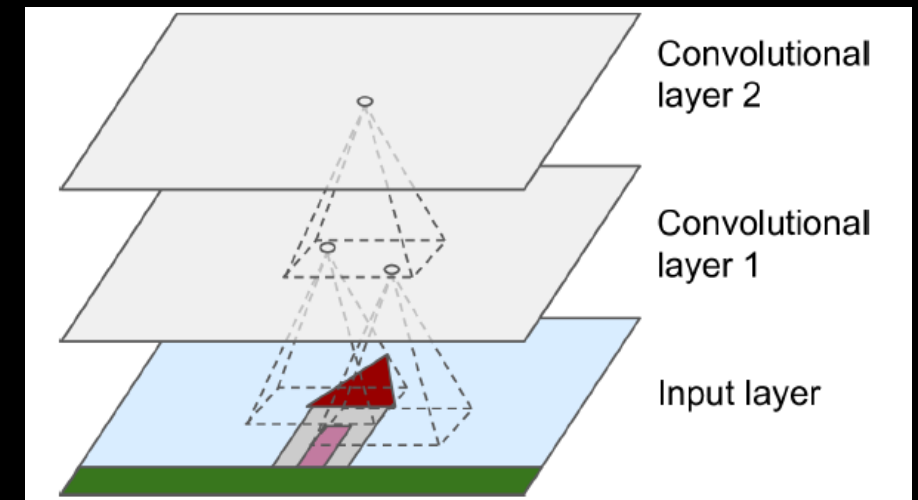
INTRODUCING CNNs

- Emulate the visual cortex
- Exploit strong spatially local correlation present in natural images
- Adding convolutional layers and pooling layers to ANN
- Developed in the late 1980s and then forgotten about due to the lack of processing power

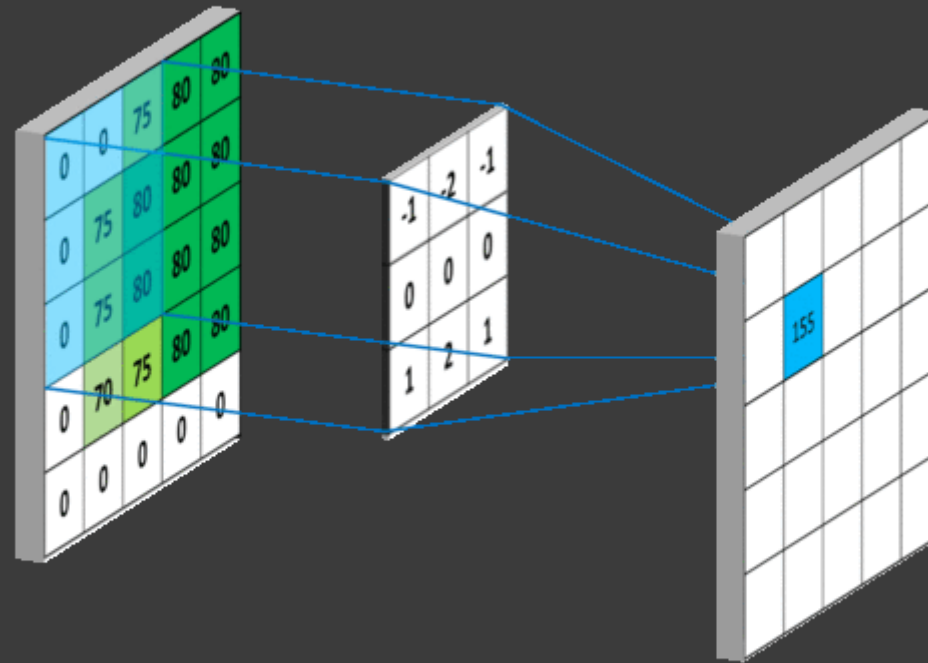


LAYERS, LAYERS, LAYERS

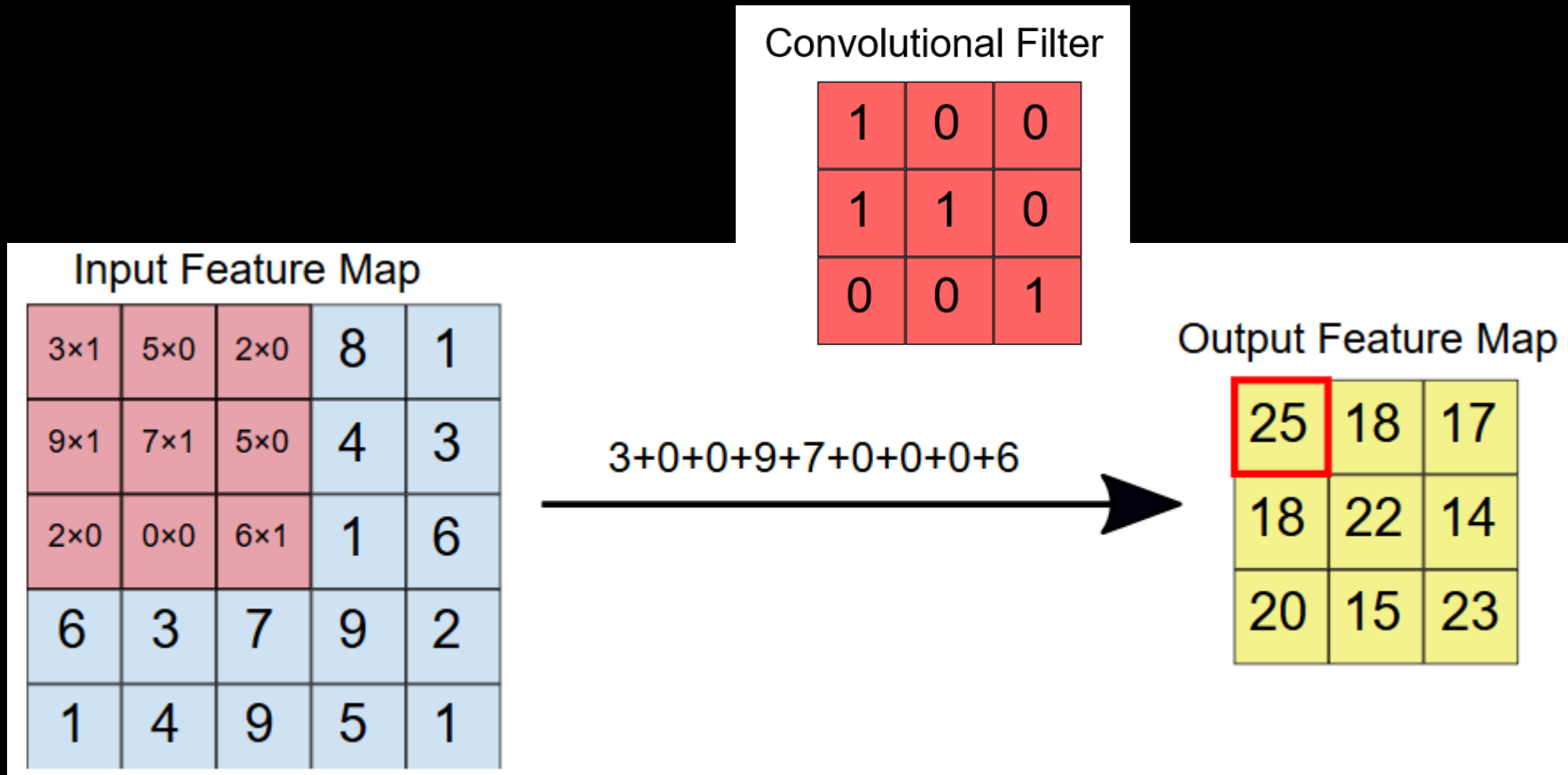
- **Input** layer: rearranged image
- **Convolutional** layer (CONV)
- **Activation** layer (ACT) or non-linear activation function (ReLU)
- **Pooling** layer (POOL) to shrink feature image
- **Fully-connected** aka dense layer (FC) to map features to classes
- **Batch-normalization** layer (BN) to reduce volatility of learning rate
- **Dropout** layer (DO) to reduce overfitting
- **Output** layer for classification



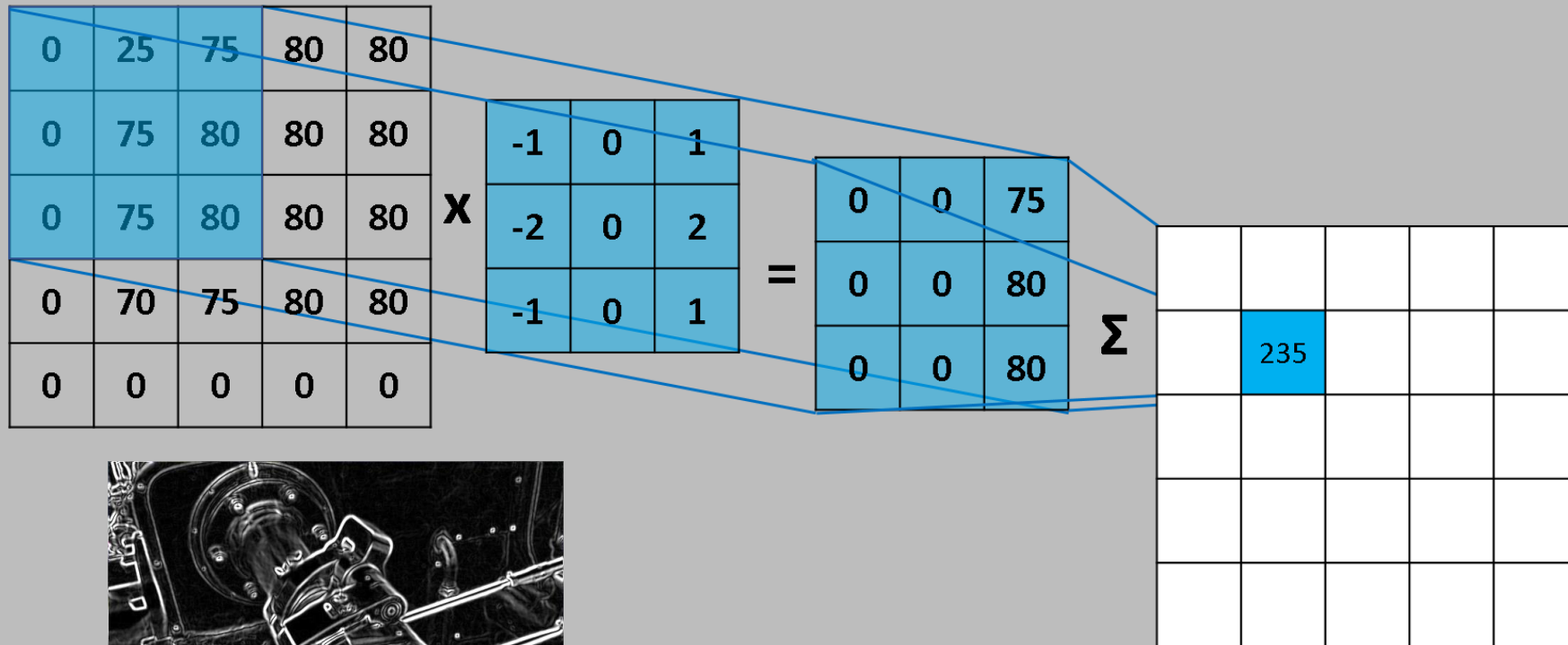
RECAP CONVOLUTION



EXAMPLE



EXAMPLE: VERTICAL SOBEL KERNEL



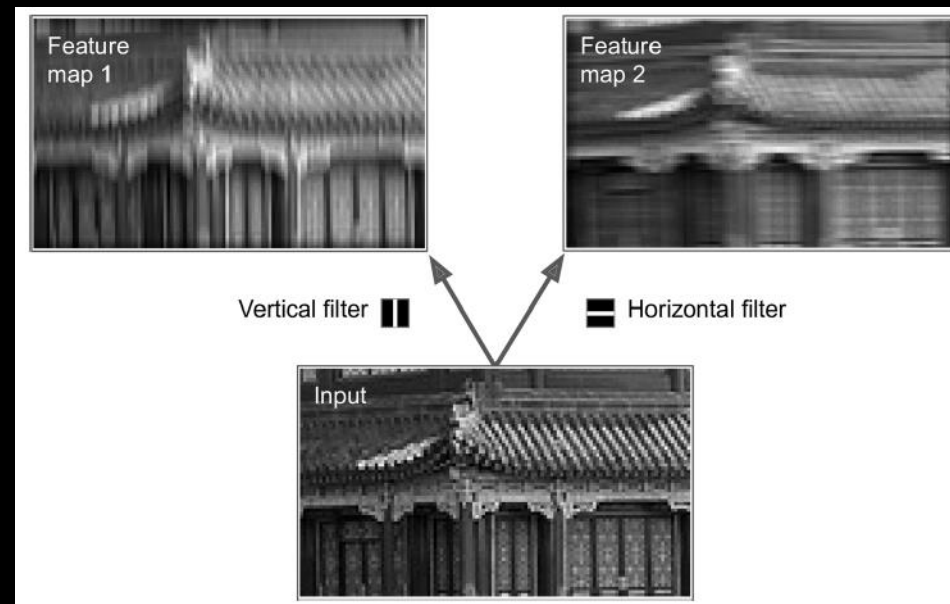
Source: <https://mlnotebook.github.io/post/CNN1/>



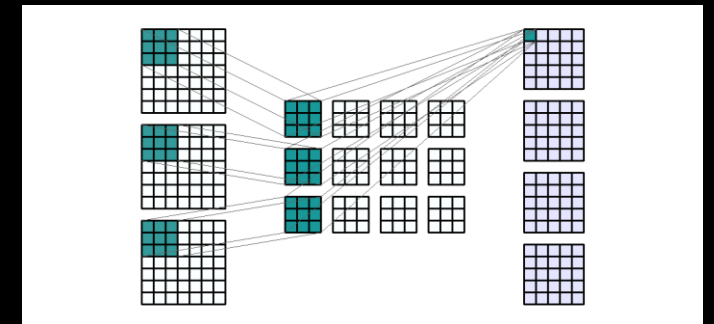
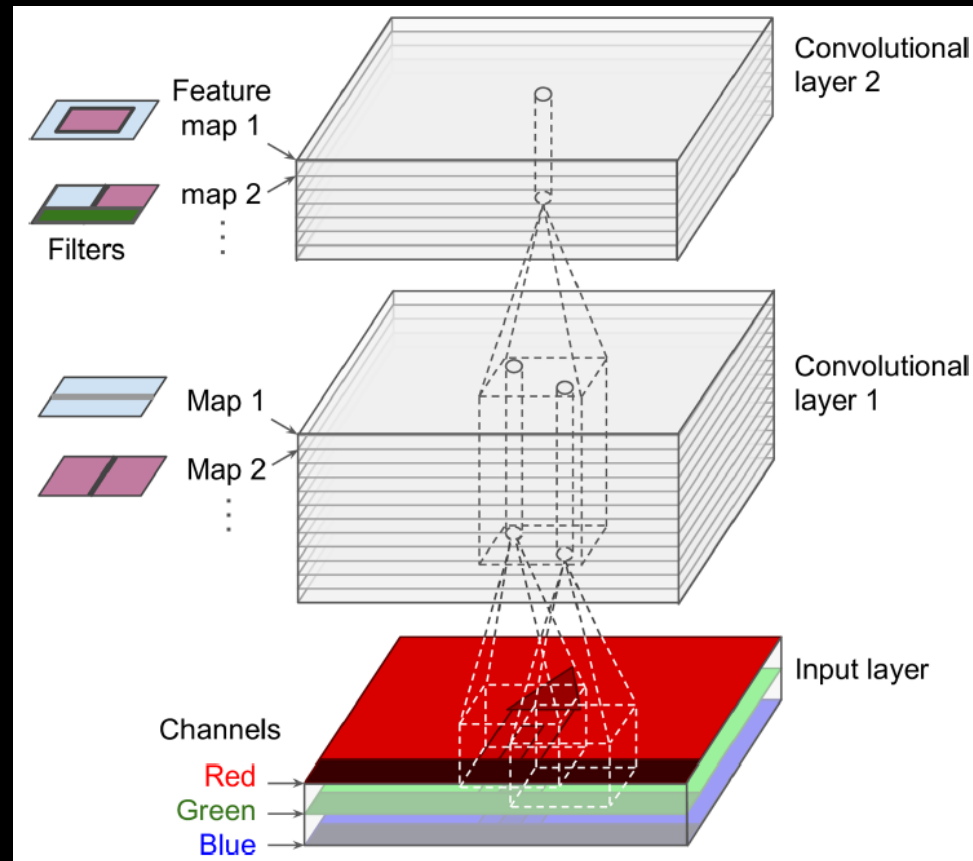
By Simpsons contributor, CC BY-SA 3.0,
<https://commons.wikimedia.org/w/index.php?curid=8904663>

CONVOLUTIONAL KERNELS

- Aka convolutional matrix, filter
- Areas are highlighted that activate the filter the most



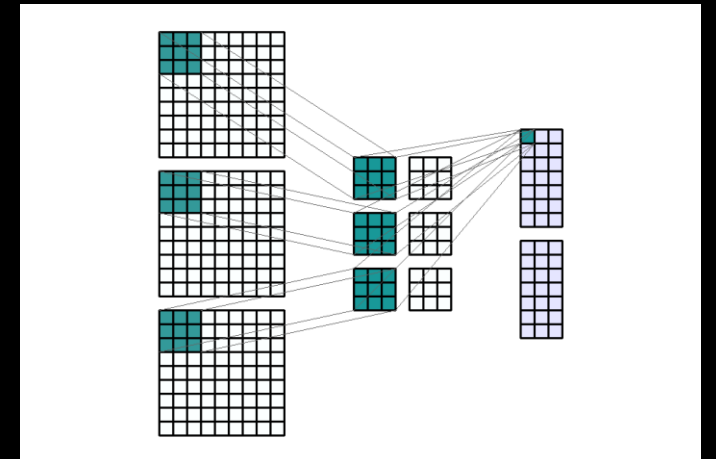
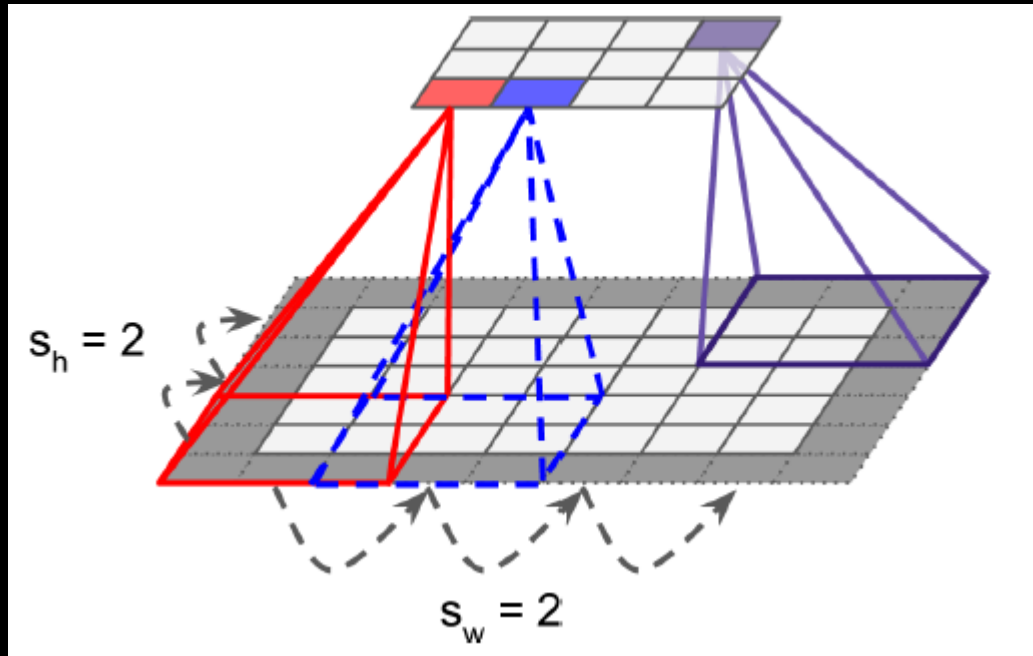
STACKING MULTIPLE FEATURE MAPS



Source : Géron, ISBN: 9781492032632, <https://towardsdatascience.com/conv2d-to-finally-understand-what-happens-in-the-forward-pass-1bbaafb0b148>

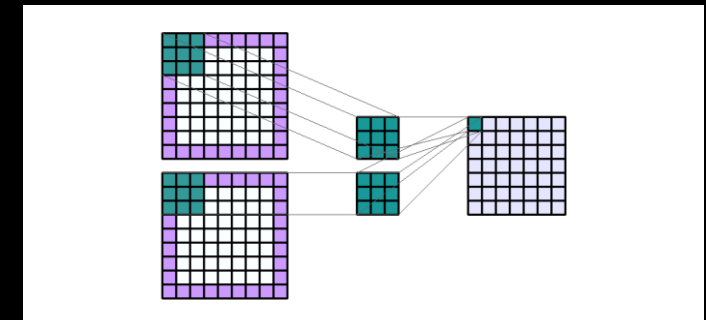
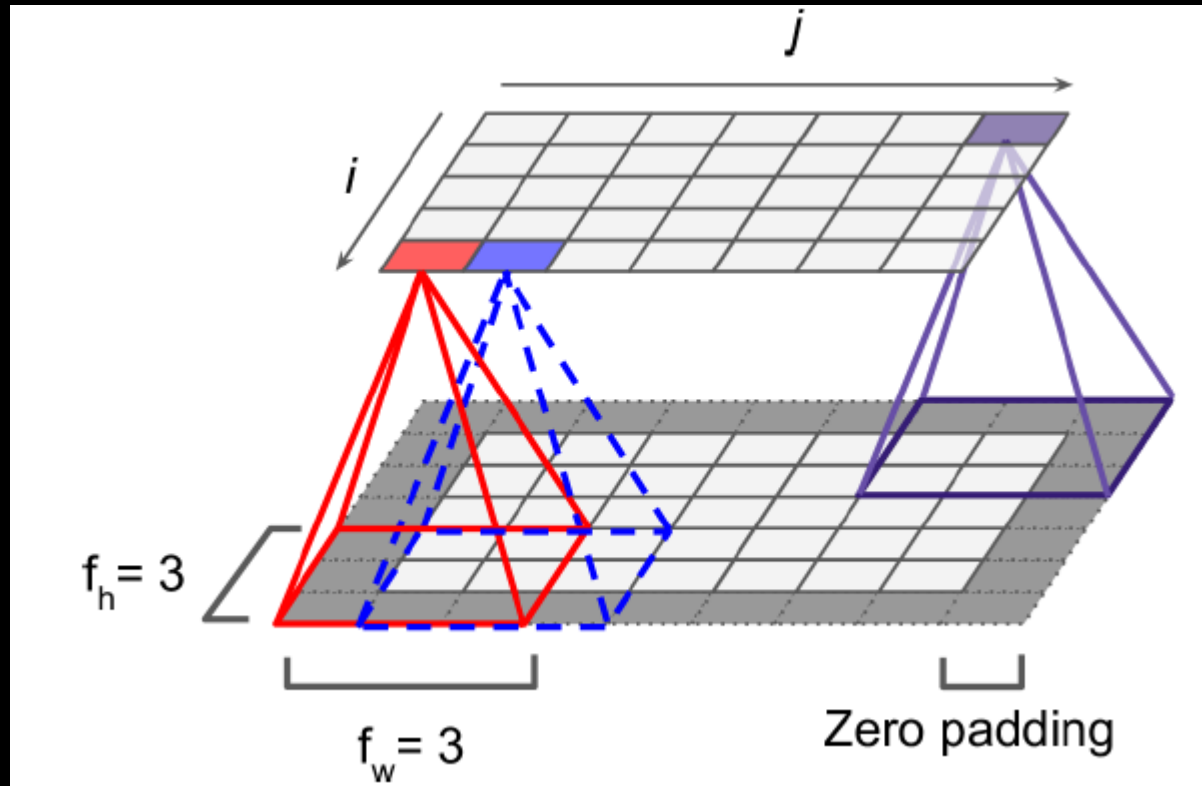
STRIDE

- Reduce hidden-layer nodes

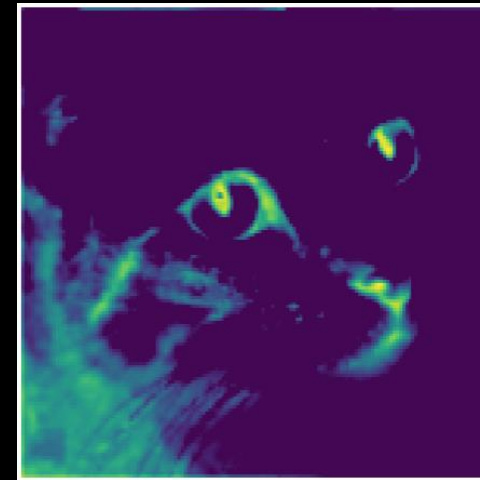
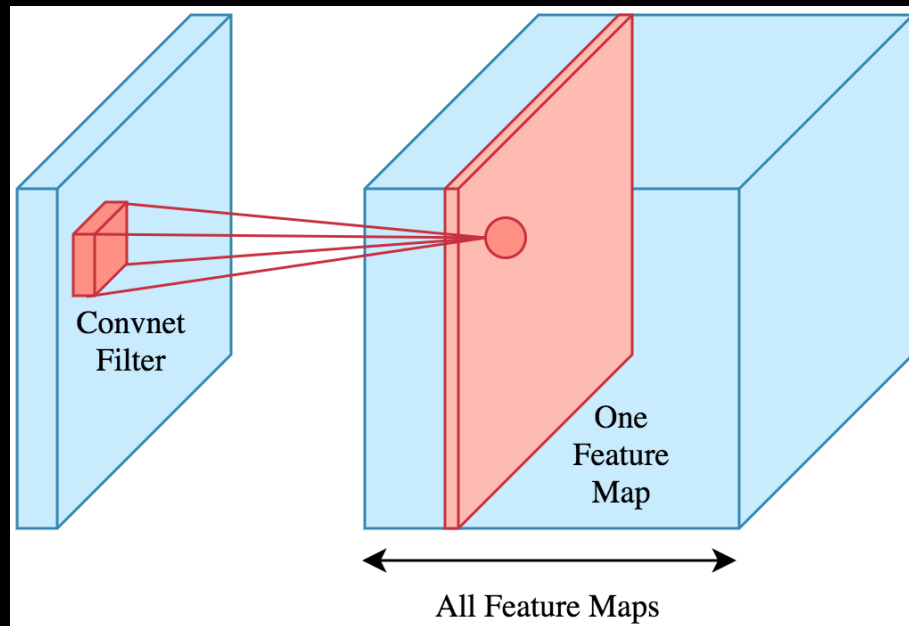


ZERO-PADDING

- Preserve image size



VISUALIZING FEATURE MAPS

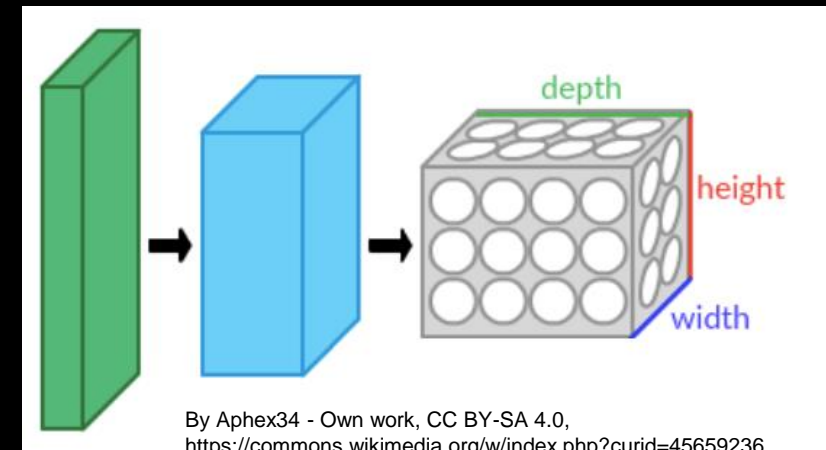
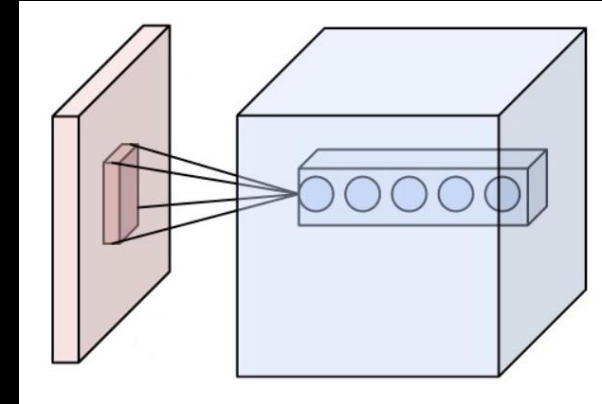


CONVOLUTIONAL LAYER SUMMARY

- Local connectivity
- Shared weights
- 3D volumes of neurons,

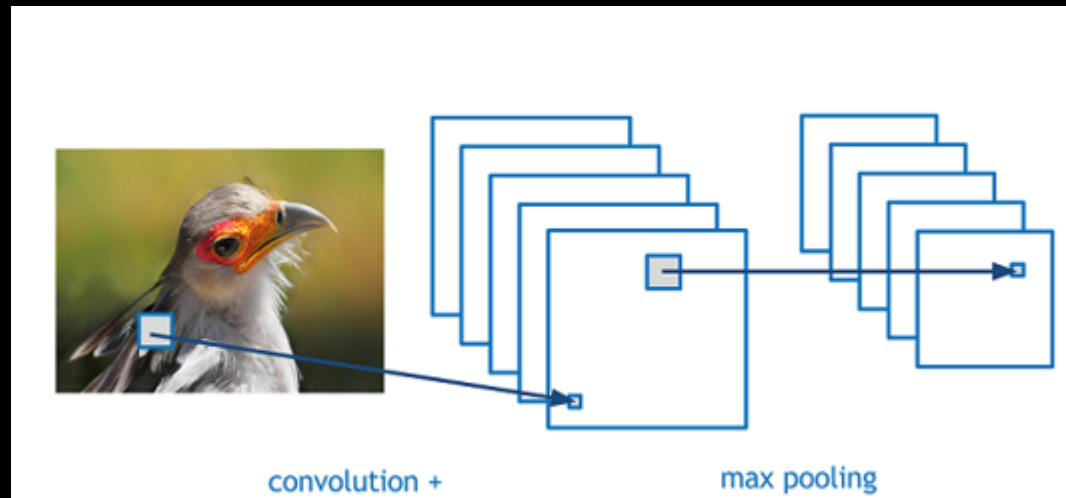
$$\text{nr of neurons (per layer)} = \frac{\text{Input volume} - \text{kernel size} + 2 \times \text{zero padding}}{\text{stride}} + 1$$

Should be an integer number!



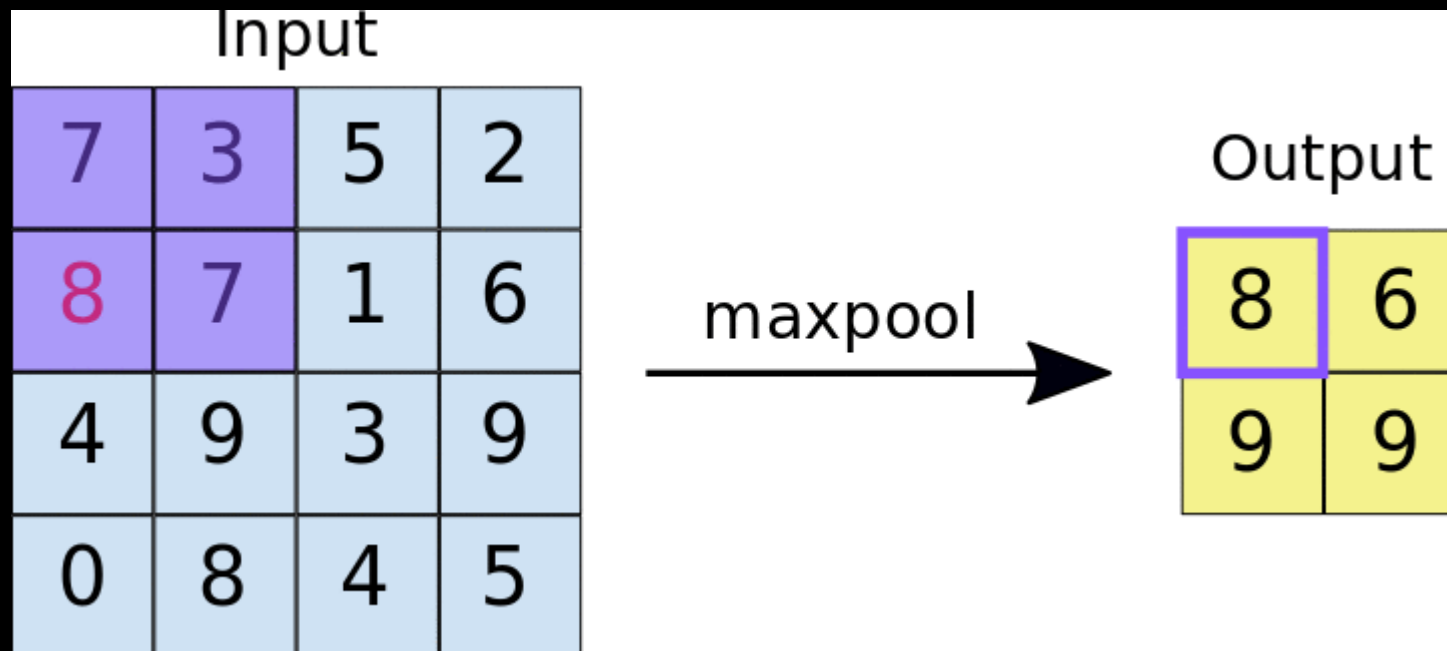
POOLING LAYER

- Non-linear down sampling
- Reduce number of dimensions of the feature map
- Find larger-scale detail than just edges and curves



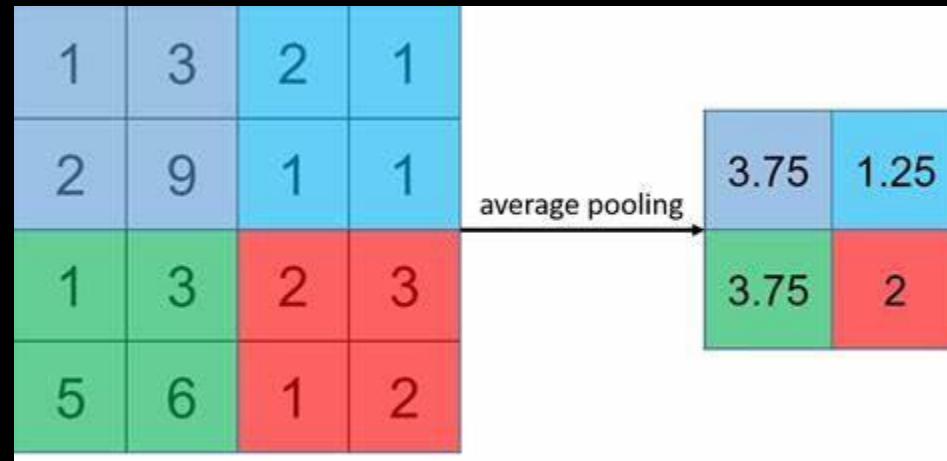
MAX POOLING

- Select maximum value from matrix (default size is 2 X 2)



AVERAGE POOLING

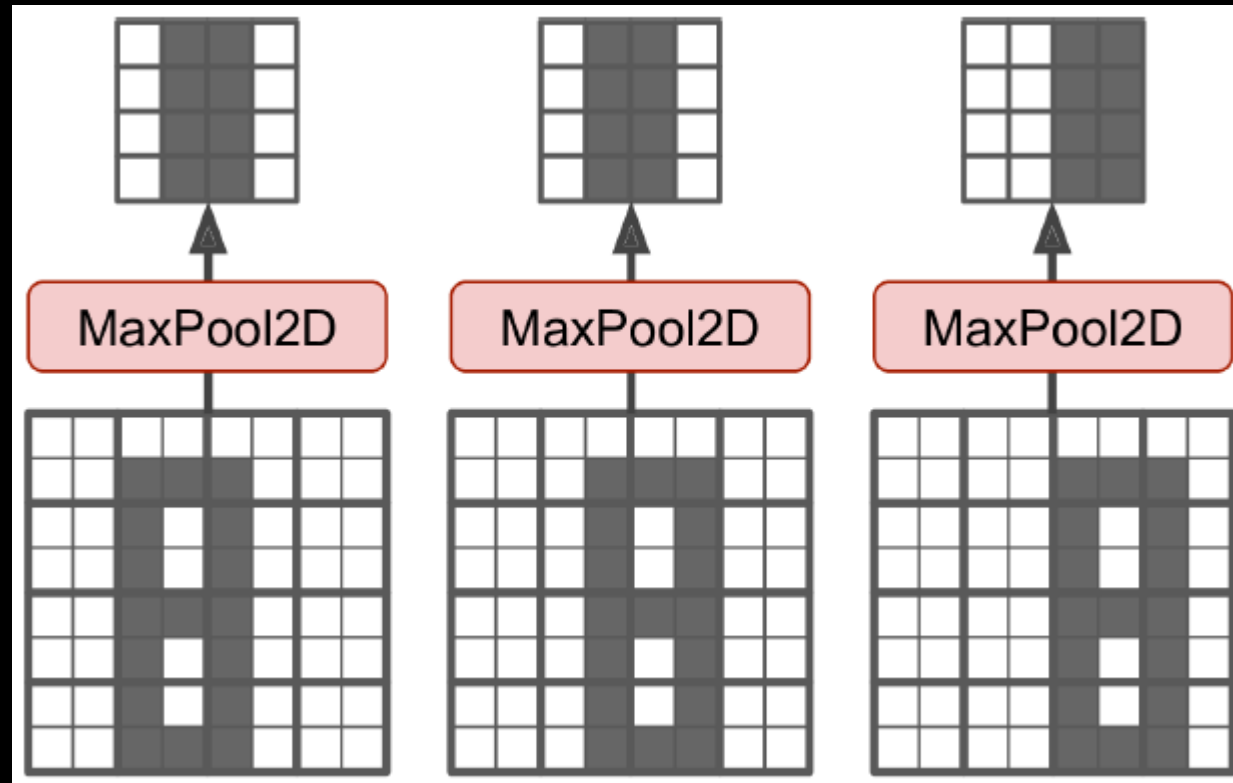
- Take average value from matrix (default size is 2 X 2)



Source: http://static.zybuluo.com/mShuaiZhao/85xogdnxjl4hq6hm3magreb/average_pooling.png

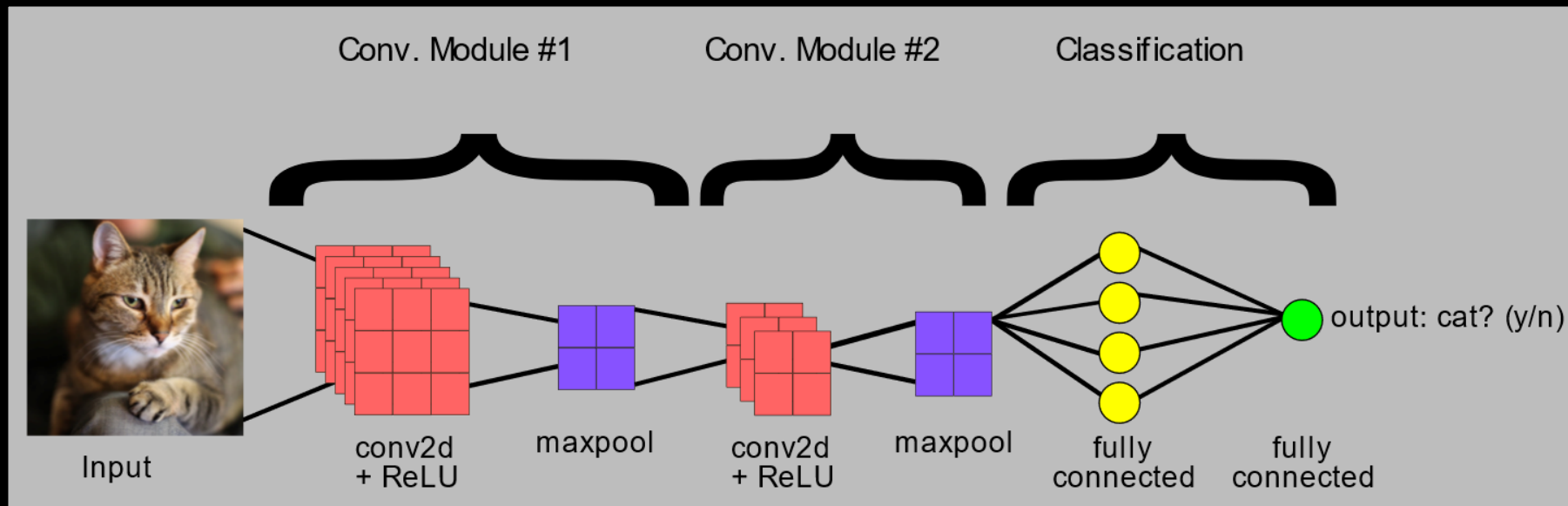
In most of the cases, max pooling is used because its performance is much better than average pooling.

INVARIANCE TO SMALL TRANSLATIONS



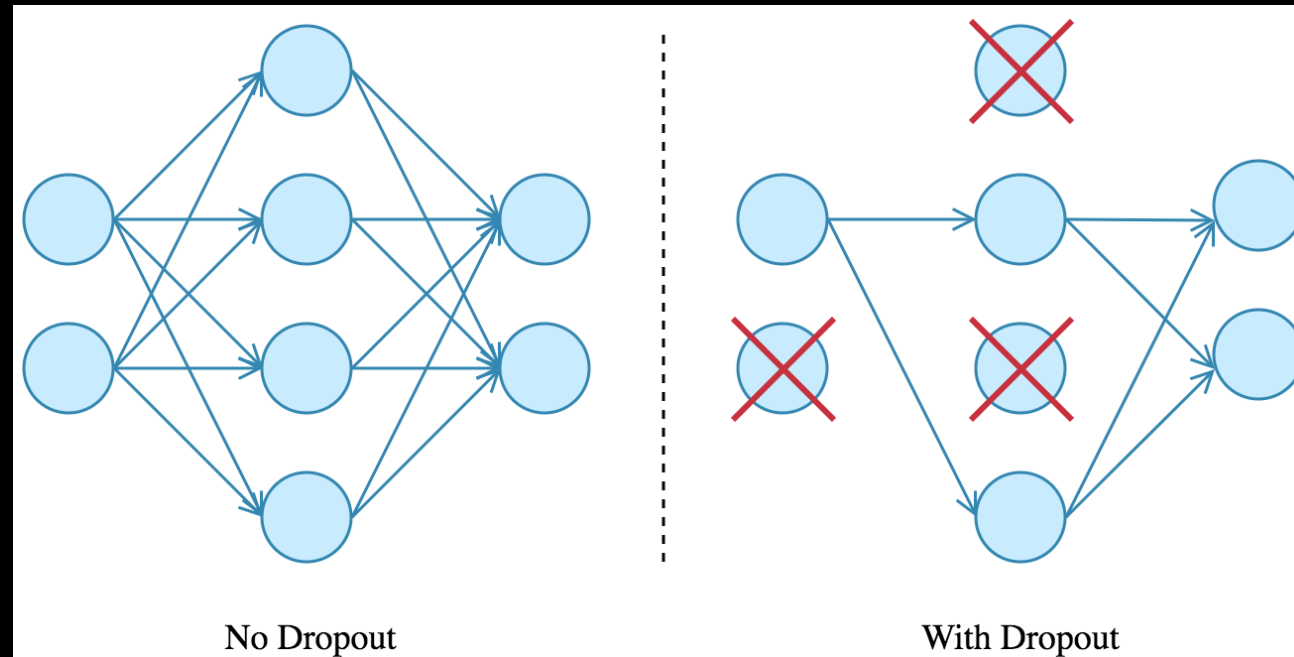
FULLY CONNECTED LAYERS

- Perform classification
- Softmax activation function

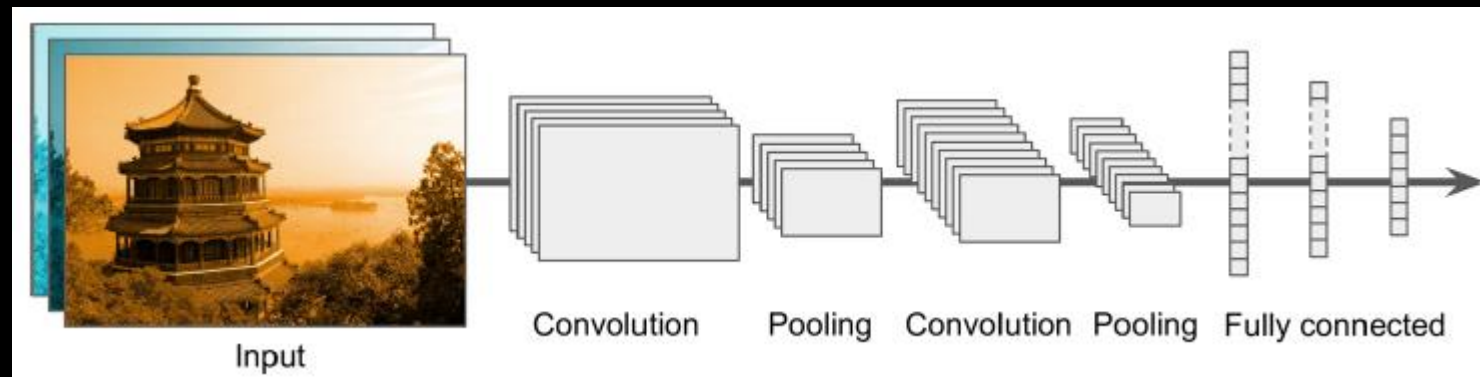


DROPOUT LAYERS

- Randomly removing units from the neural network during a training gradient step.
- Reduce overfitting



TYPICAL CNN ARCHITECTURE

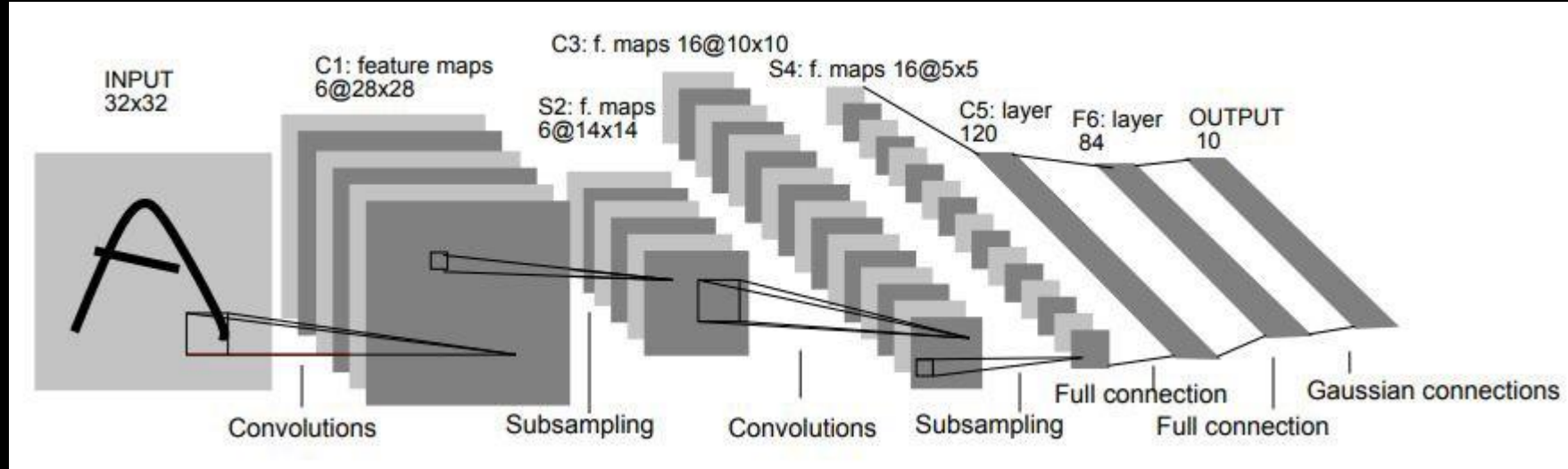


Source: Géron, ISBN: 9781492032632

LEVERAGING PRETRAINED MODELS

- Feature extraction
 - retrieving intermediate representations produced by the pretrained model
 - E.g. higher-level attributes such as color, texture, shape
- To increase performance when using feature extraction with a pretrained model, engineers often *fine-tune* the weight parameters applied to the extracted features.
- Common architectures are discussed by Géron

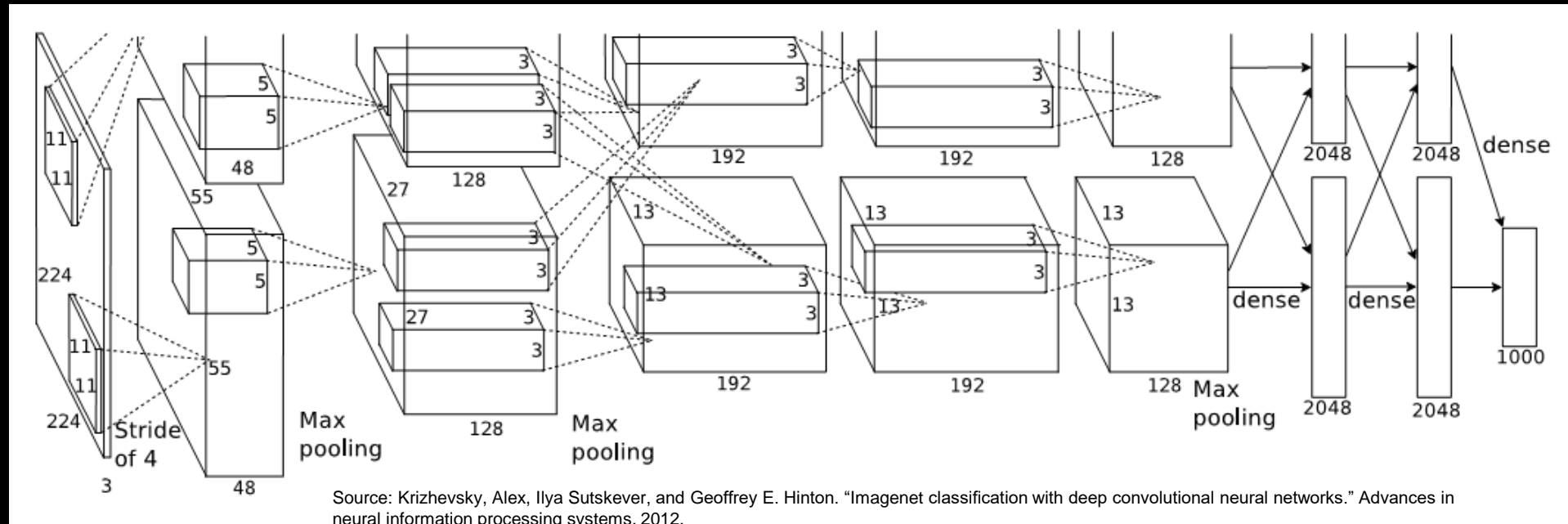
LENET



Source: LeCun, Yann, et al. "Gradient-based learning applied to document recognition." Proceedings of the IEEE 86.11 (1998): 2278–2324.

ALEXNET

- ImageNet competition



ZFNet (2013), GoogLeNet (2014), VGGNet (2014), ResNet (2015), DenseNet (2016) etc.

INSPIRATION

- <https://www.tensorflow.org/hub>
- <https://keras.io/api/applications/>
- <https://www.kaggle.com/>
- <https://google.github.io/mediapipe/>
- <https://developer.ibm.com/articles/transfer-learning-for-deep-learning/>