EVD 3

CONVOLUTIONAL NEURAL NETWORKS

JEROEN VEEN

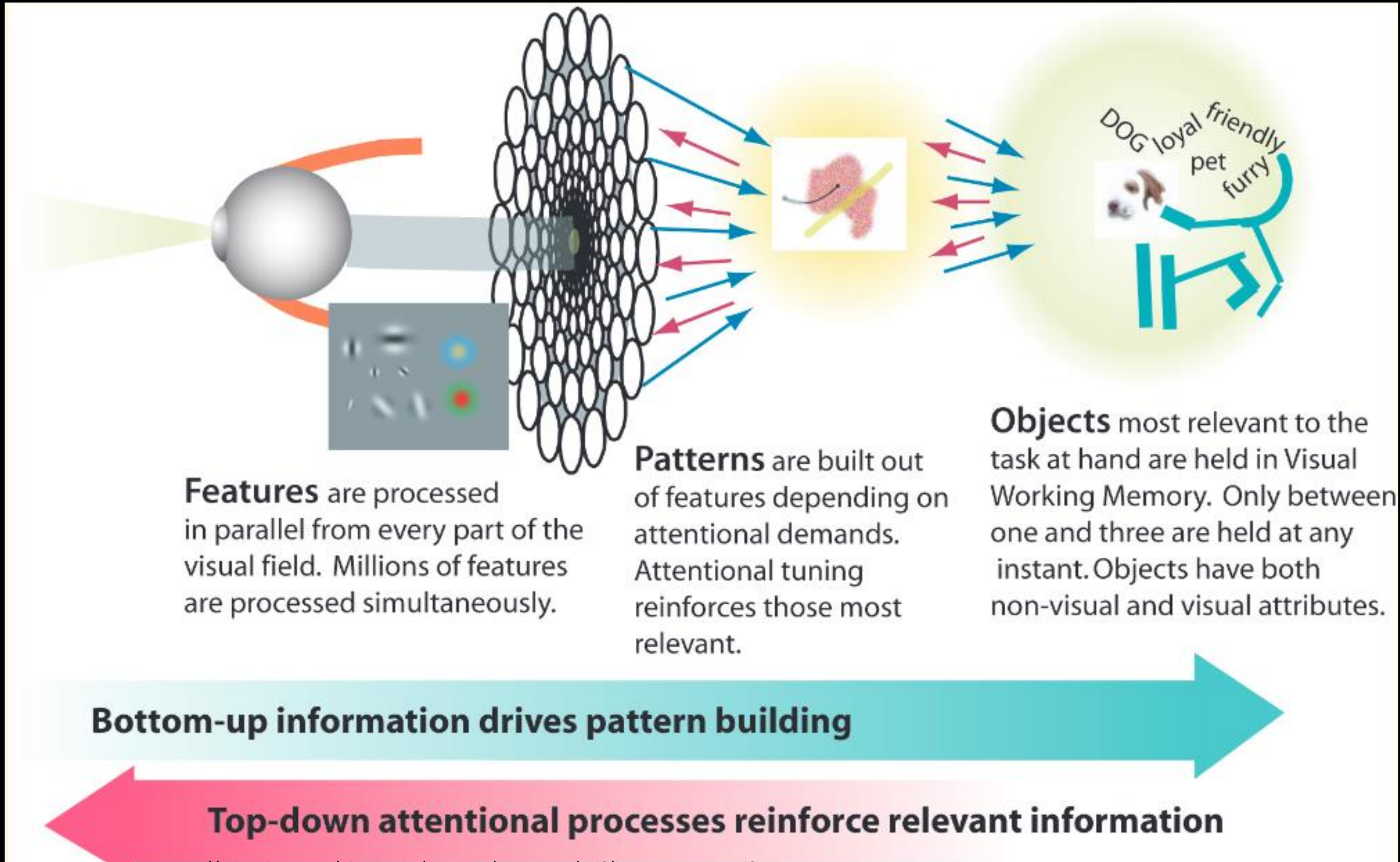
HAN_UNIVERSITY OF APPLIED SCIENCES

AGENDA

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

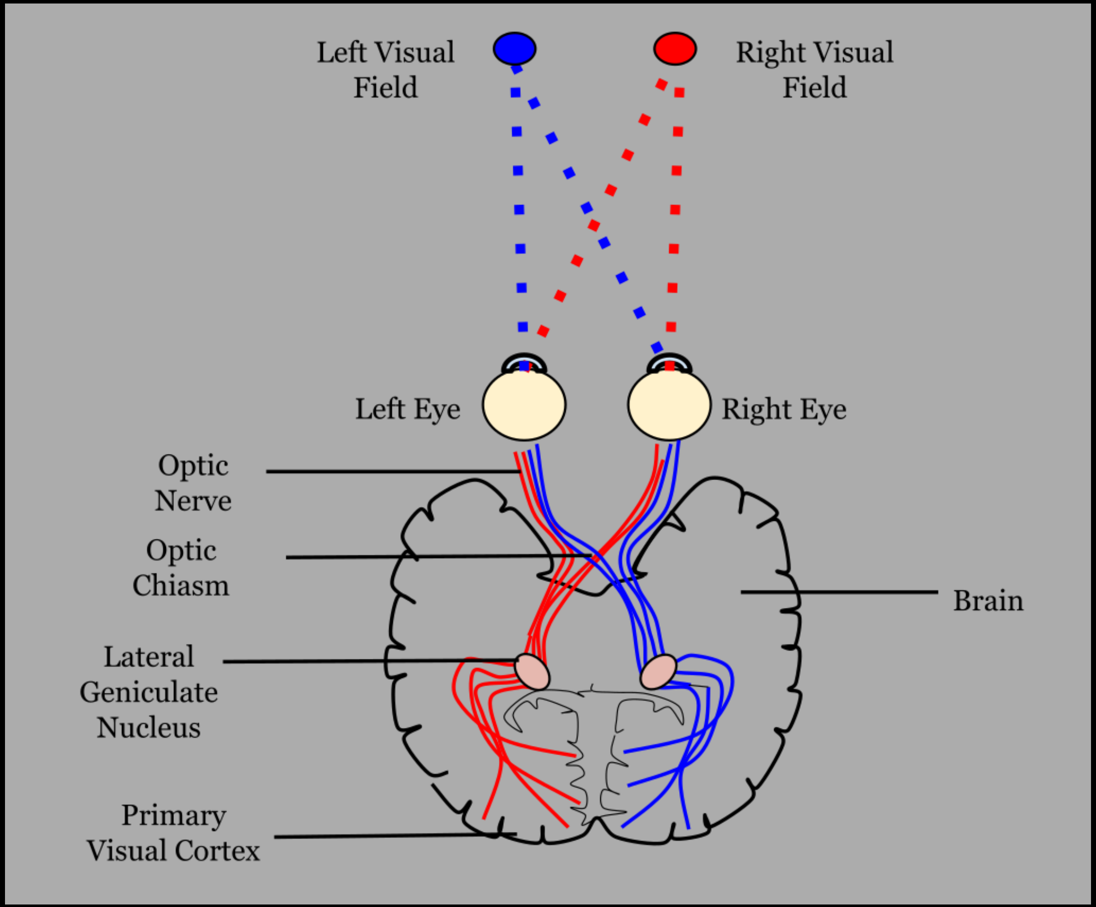
HAN_UNIVERSITY
OF APPLIED SCIENCES

INSPIRATION FROM THE VISUAL CORTEX

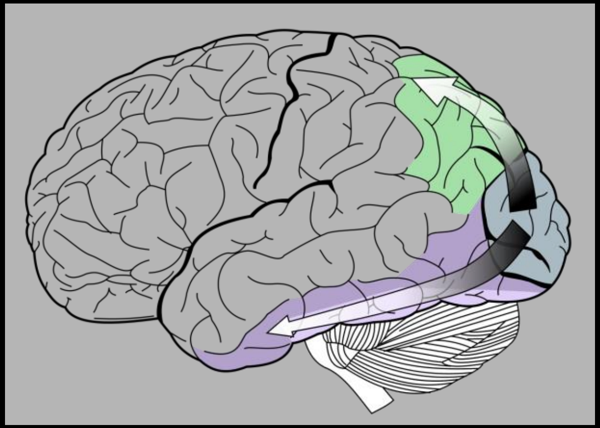


Source: https://cslu.ohsu.edu/~bedricks/courses/conj_610/pdf/lec_2_part_2.pdf

INSPIRATION FROM THE VISUAL CORTEX

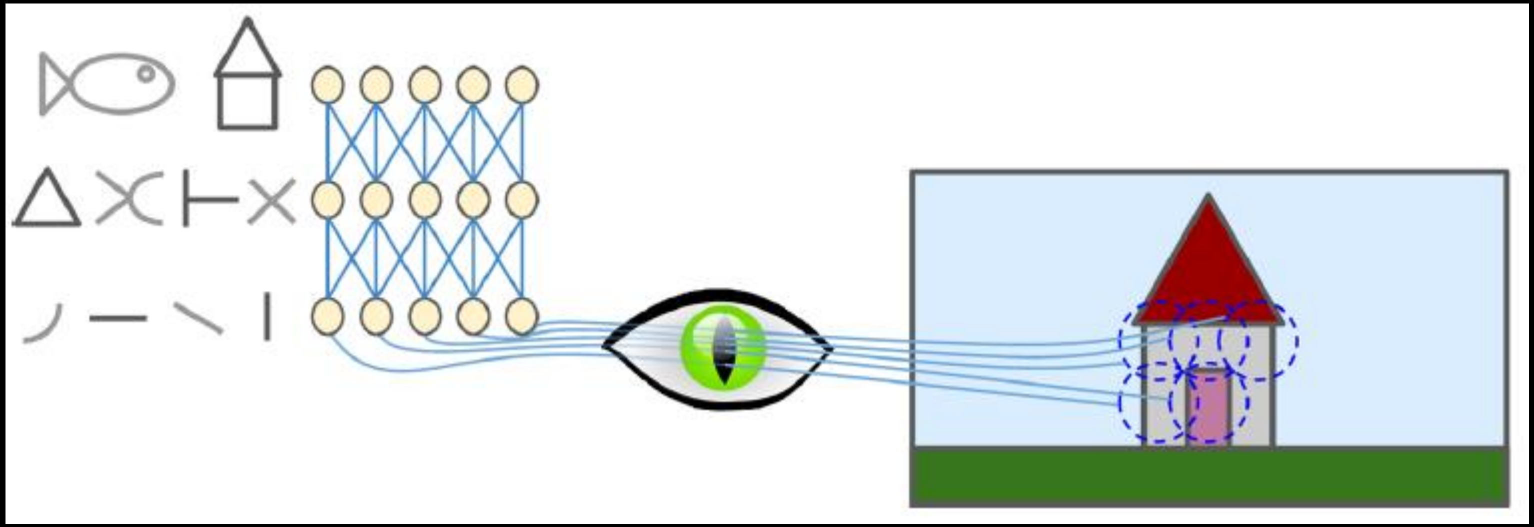


Source: https://en.wikipedia.org/wiki/Visual_cortex



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

LOCAL RECEPTIVE FIELDS

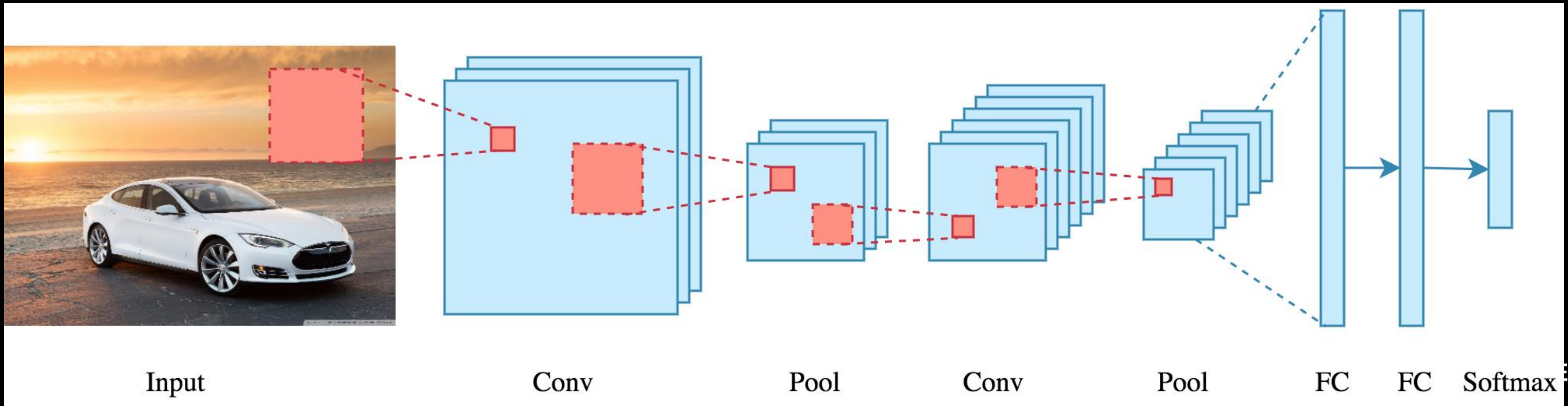


Source: Géron, ISBN: 9781492032632

- Each neuron has a different receptive field.

INTRODUCING CNNS

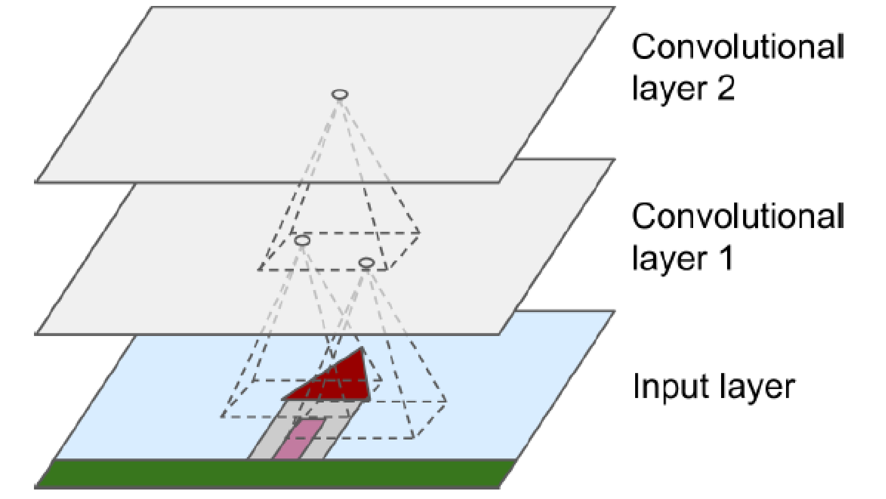
- Emulate the visual cortex
- Exploit strong spatially local correlation present in natural images
- Adding convolutional layers and pooling layers to ANN
- Developed in the late 1980s and then forgotten about due to the lack of processing power



Source: https://towardsdatascience.com/applied-deep-learning-part-4-convolutional-neural-networks-584bc134c1e2

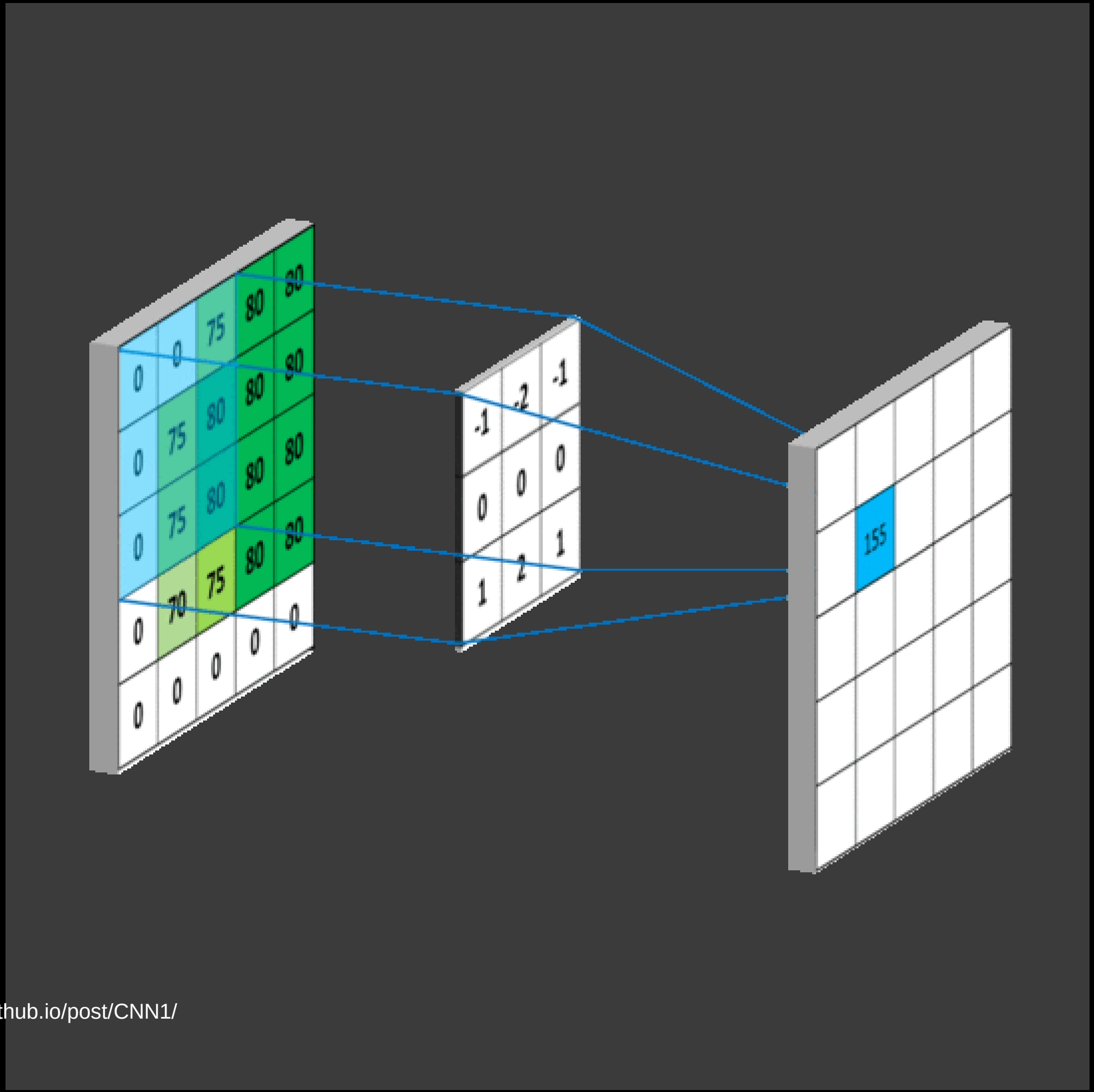
LAYERS, LAYERS, LAYERS

- **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



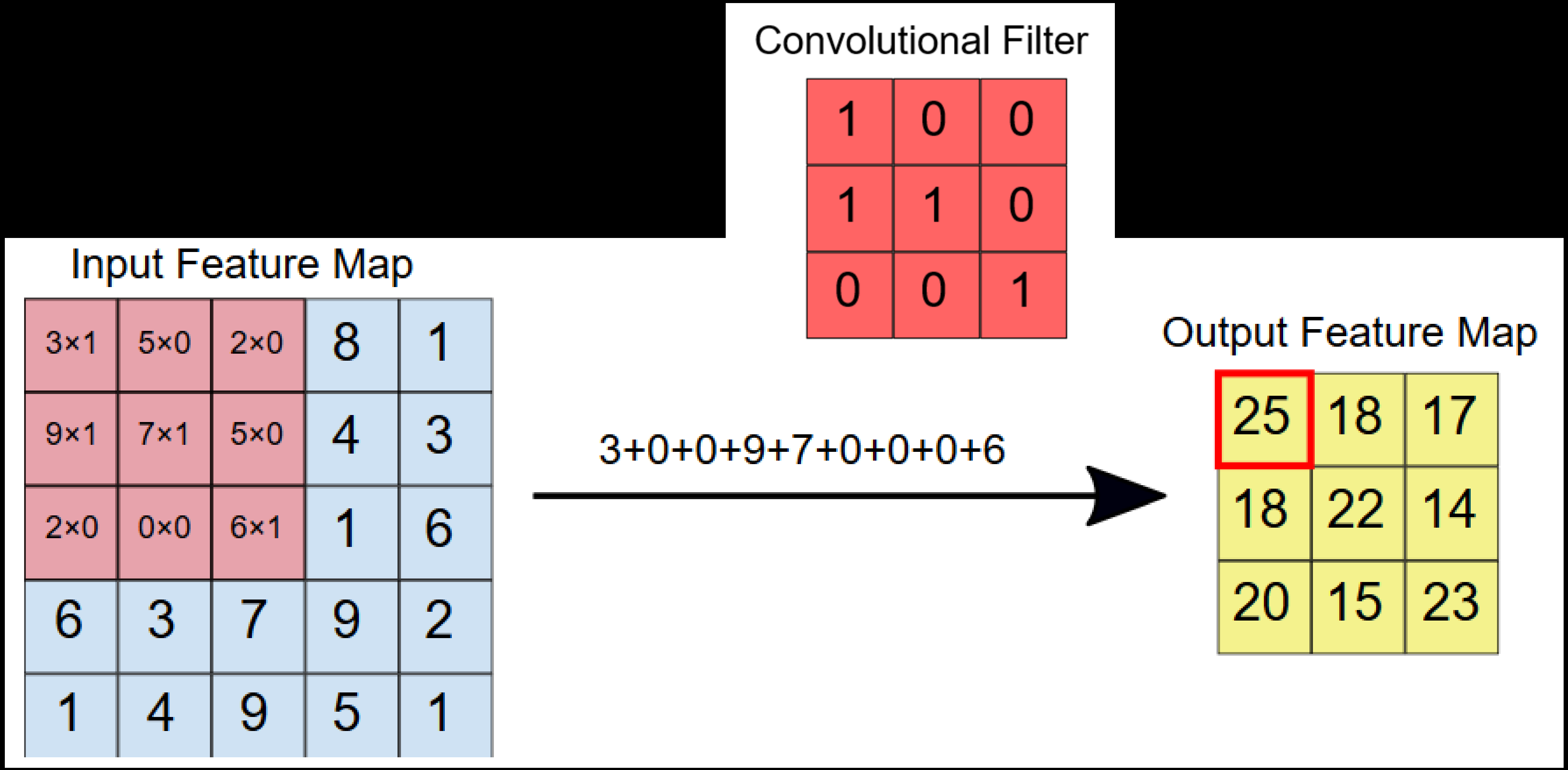
Source: Géron, ISBN: 9781492032632

RECAP CONVOLUTION



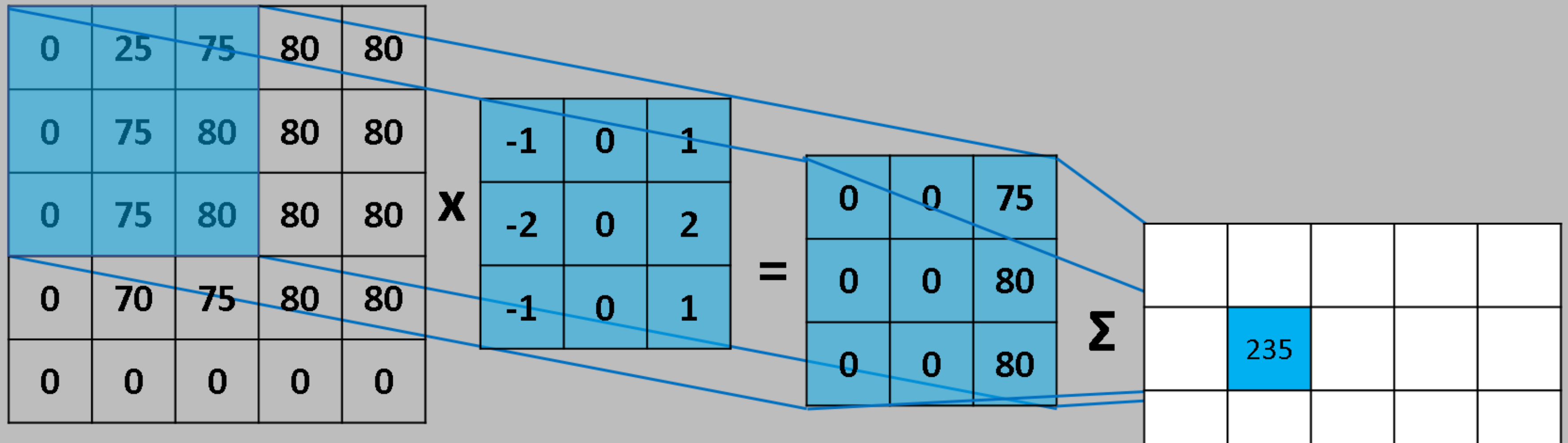
Source: https://mlnotebook.github.io/post/CNN1/

EXAMPLE

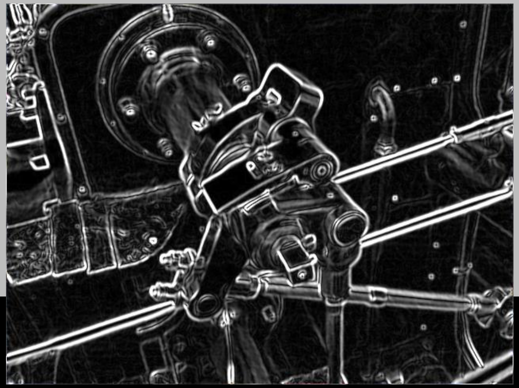


Source: https://developers.google.com/machine-learning/practica/image-classification/convolutional-neural-networks

EXAMPLE: VERTICAL SOBEL KERNEL



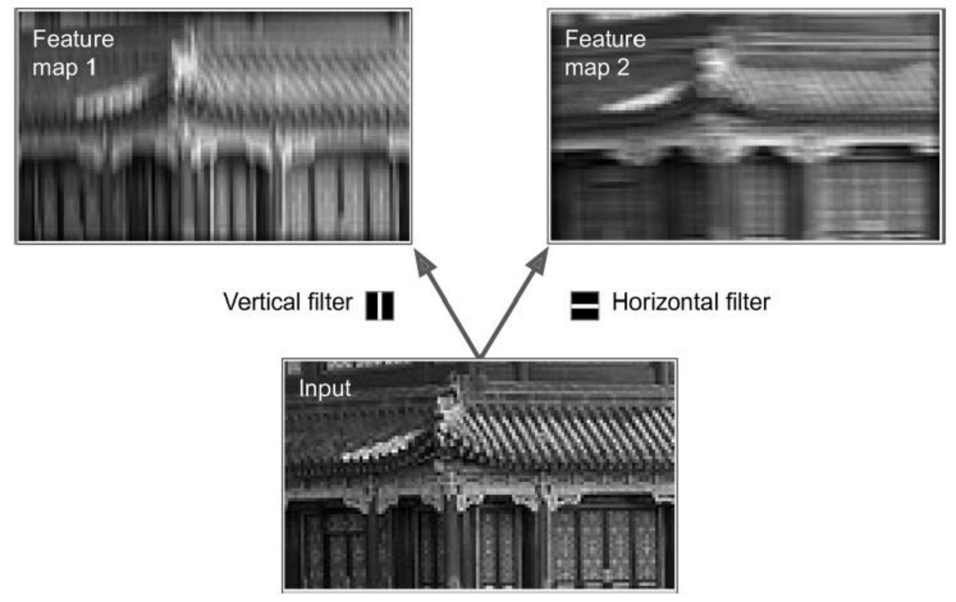
Source: https://mlnotebook.github.io/post/CNN1/



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

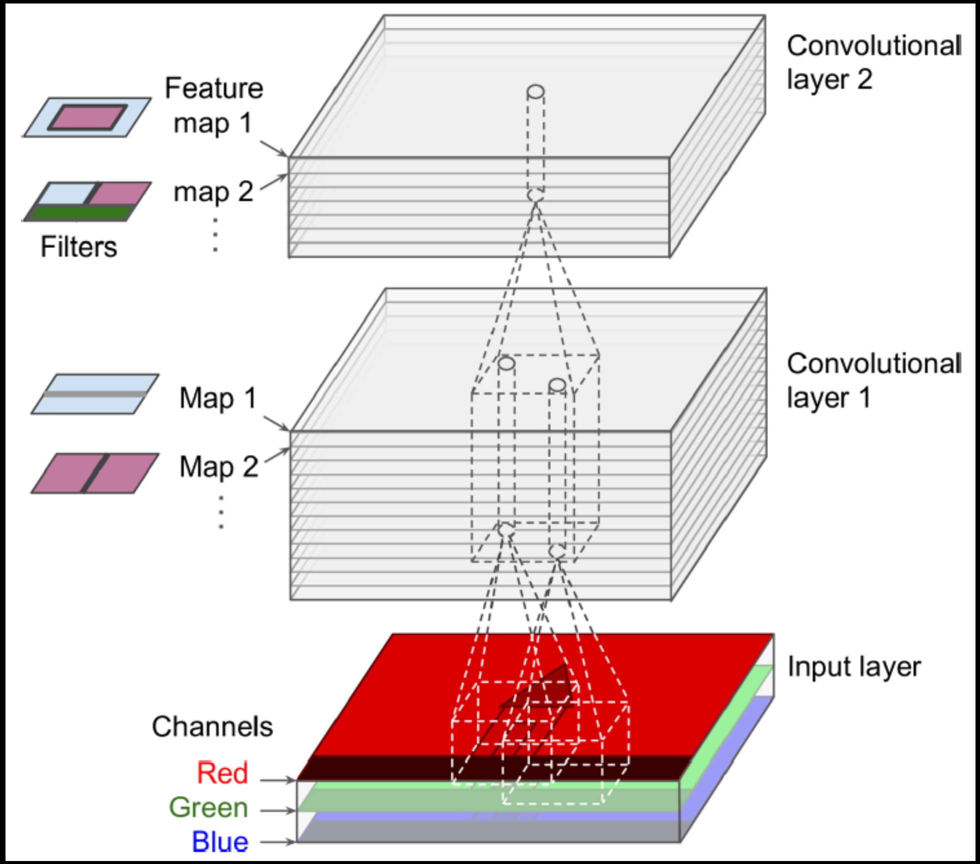
CONVOLUTIONAL KERNELS

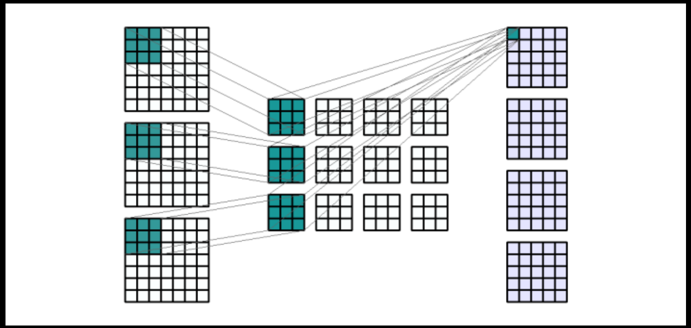
- Aka convolutional matrix, filter
- Areas are highlighted that activate the filter the most



Source: Géron, ISBN: 9781492032632

STACKING MULTIPLE FEATURE MAPS

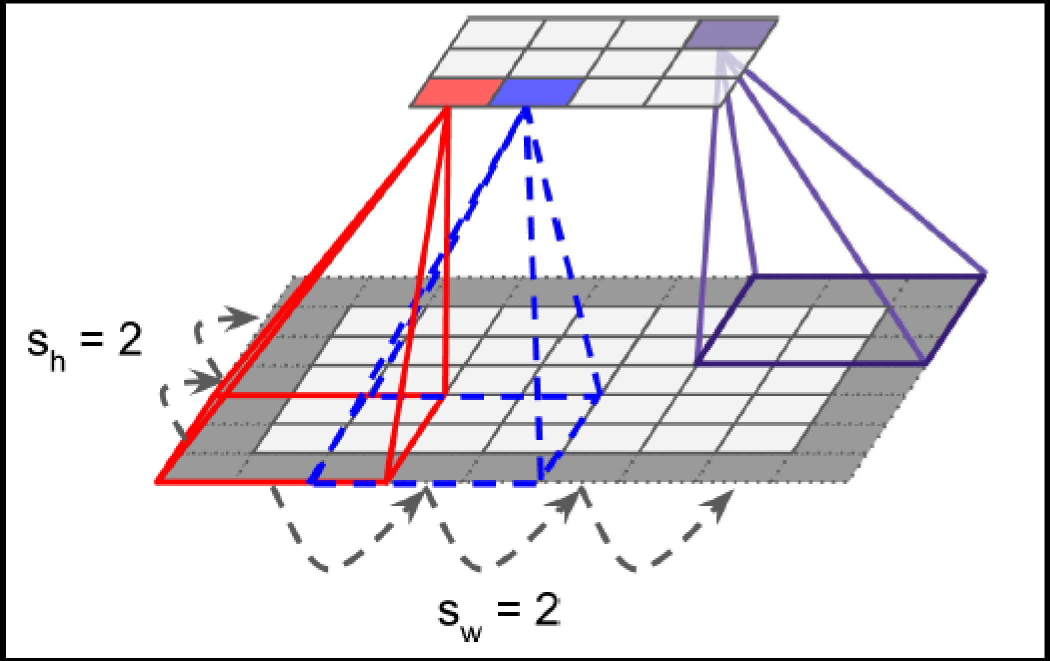


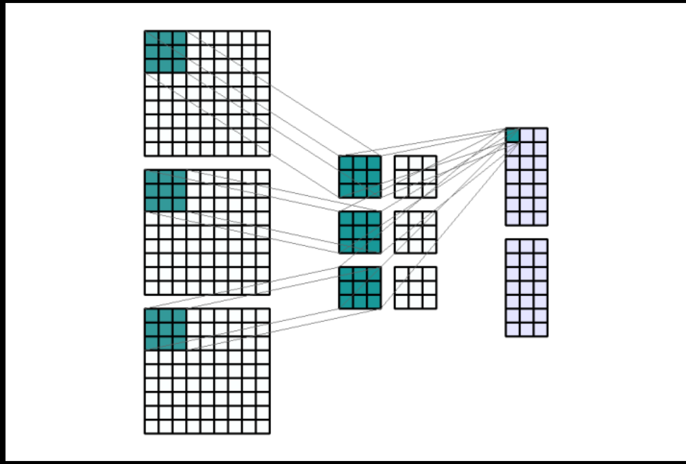


Source : Géron, ISBN: 9781492032632, https://towardsdatascience.com/conv2d-to-finally-understand-what-happens-in-the-forward-pass-1bbaafb0b148

STRIDE

- Reduce hidden-layer nodes

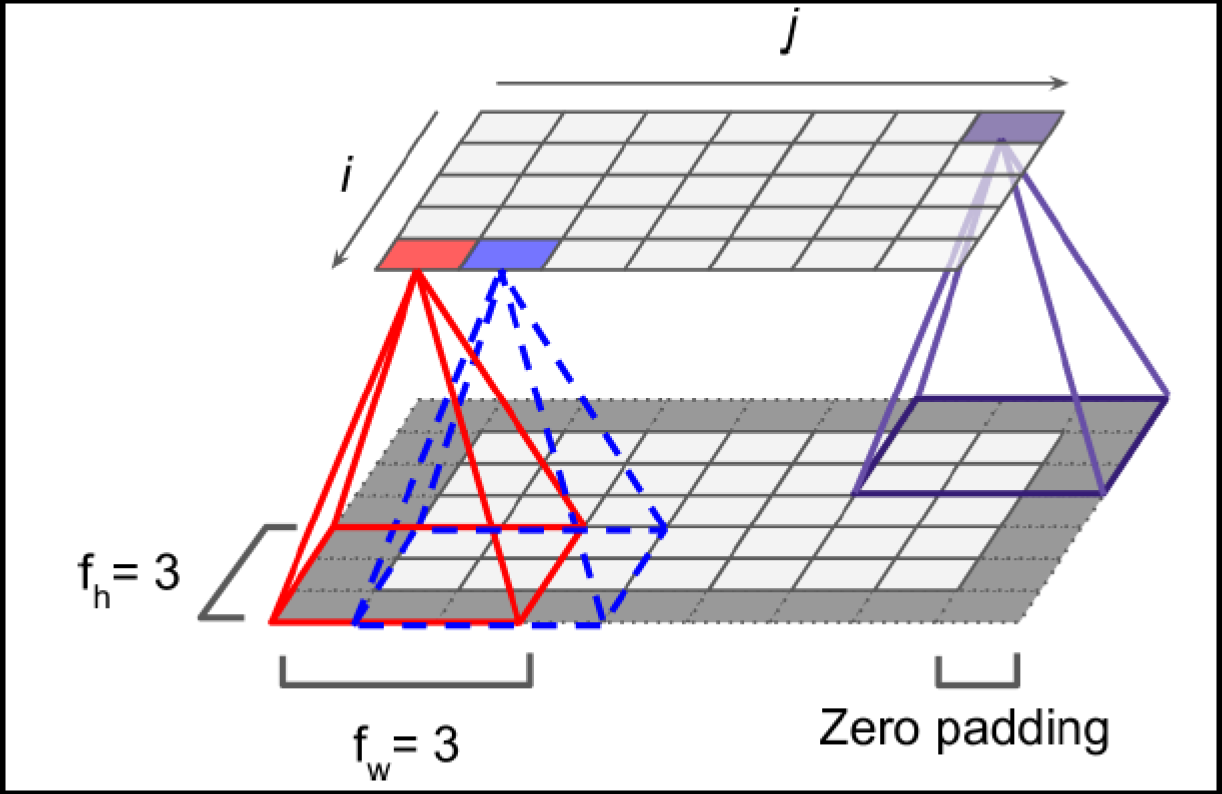


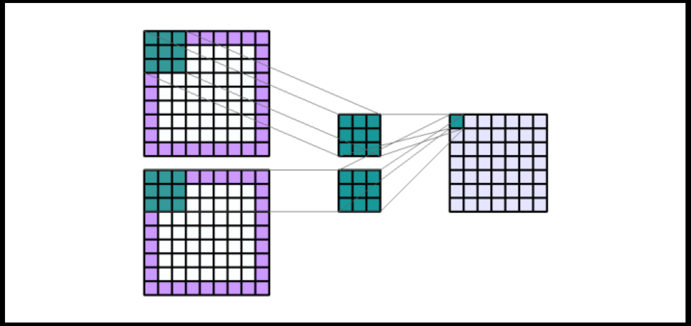


Source : Géron, ISBN: 9781492032632, https://towardsdatascience.com/conv2d-to-finally-understand-what-happens-in-the-forward-pass-1bbaafb0b148

ZERO-PADDING

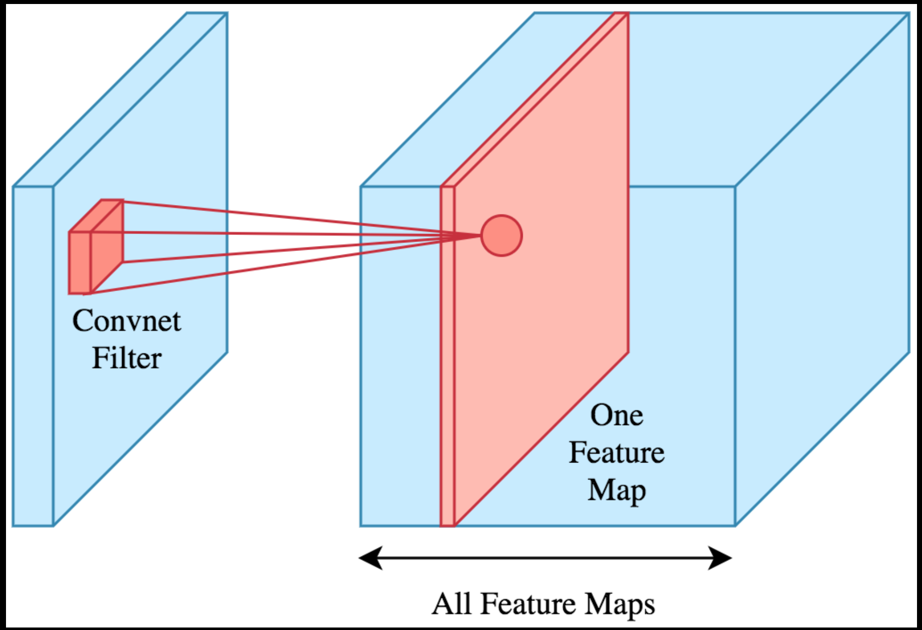
- Preserve image size

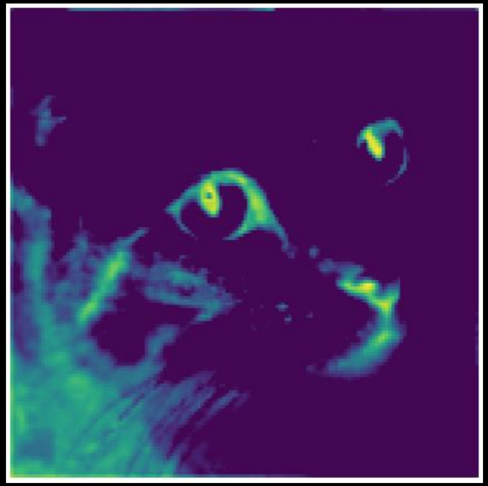




Source : Géron, ISBN: 9781492032632, https://towardsdatascience.com/conv2d-to-finally-understand-what-happens-in-the-forward-pass-1bbaafb0b148

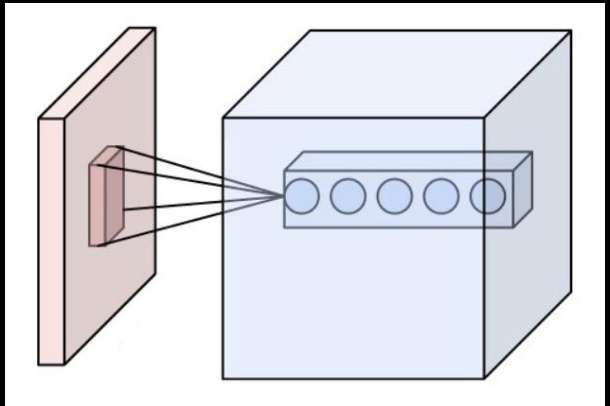
VISUALIZING FEATURE MAPS





CONVOLUTIONAL LAYER SUMMARY

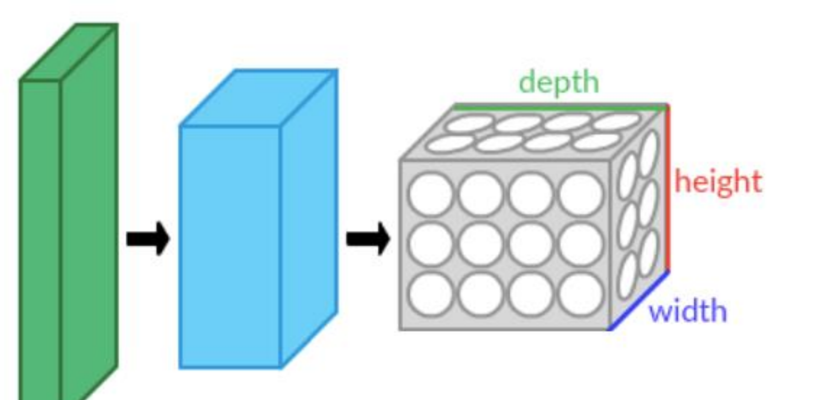
- Local connectivity
- Shared weights
- 3D volumes of neurons,



nr of neurons = $\underbrace{\overbrace{W}^{\text{Input volume}}-\overbrace{K}^{\text{kernel size}}+\overbrace{2P}^{\text{zero padding}}}_{}\over{\underbrace{S}_{\text{stride}}}$ $+1$

(per layer)

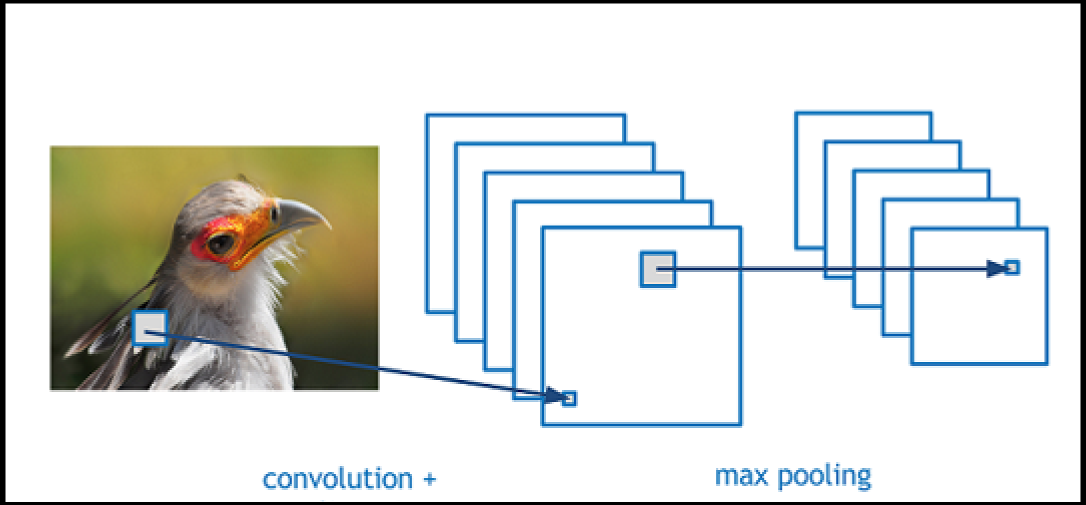
Should be an integer number!



By Aphex34 - Own work, CC BY-SA 4.0,
https://commons.wikimedia.org/w/index.php?curid=45659236

POOLING LAYER

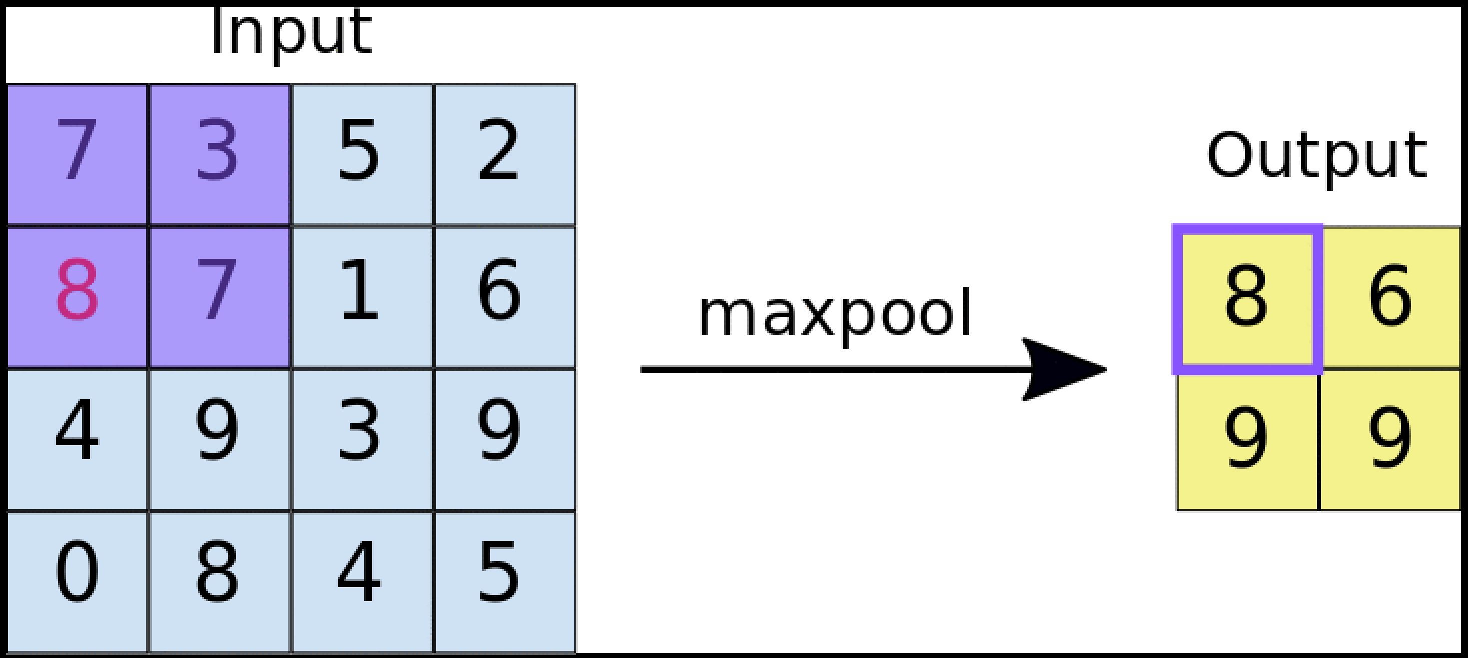
- Non-linear down sampling
- Reduce number of dimensions of the feature map
- Find larger-scale detail than just edges and curves



https://en.wikipedia.org/wiki/Convolutional_neural_network#Pooling_layer

MAX POOLING

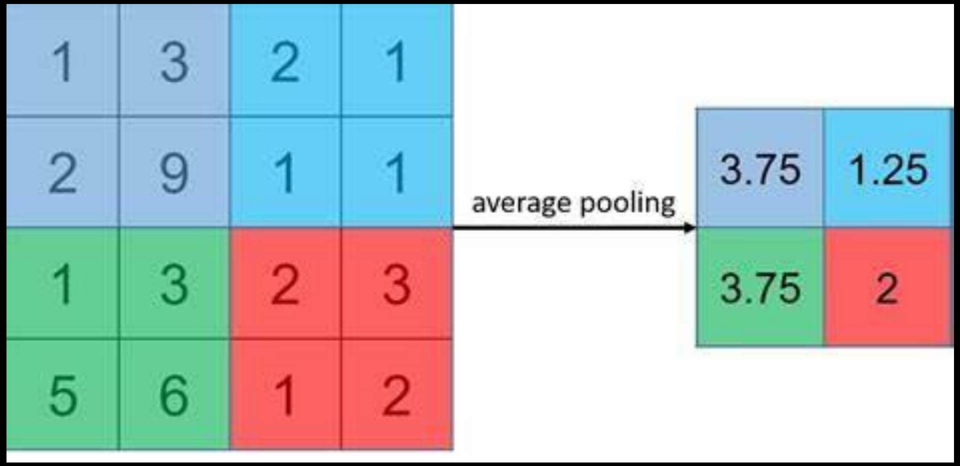
- Select maximum value from matrix (default size is 2 X 2)



HAN_UNIVERSITY
OF APPLIED SCIENCES

AVERAGE POOLING

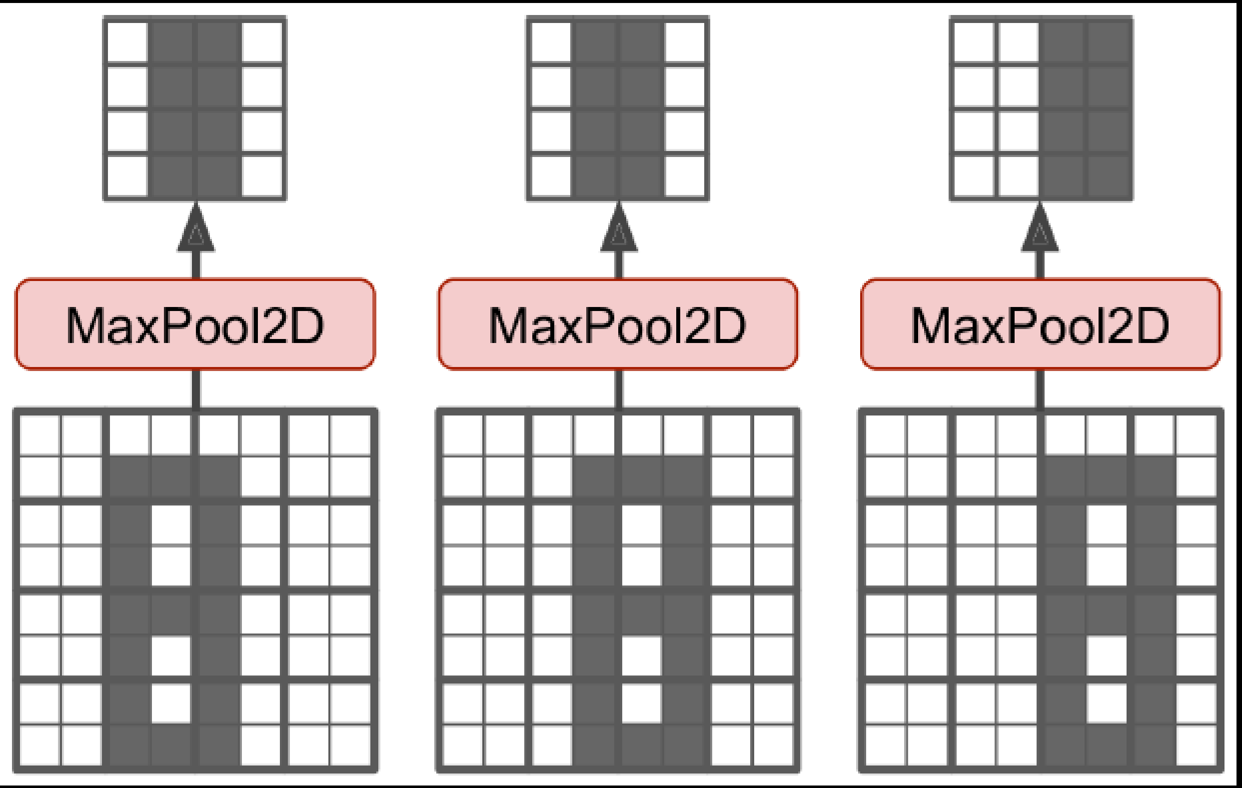
- Take average value from matrix (default size is 2 X 2)



Source:http://static.zybuluo.com/mShuaiZhao/85xogdnxjl4hqd6hm3magreb/average_pooling.png

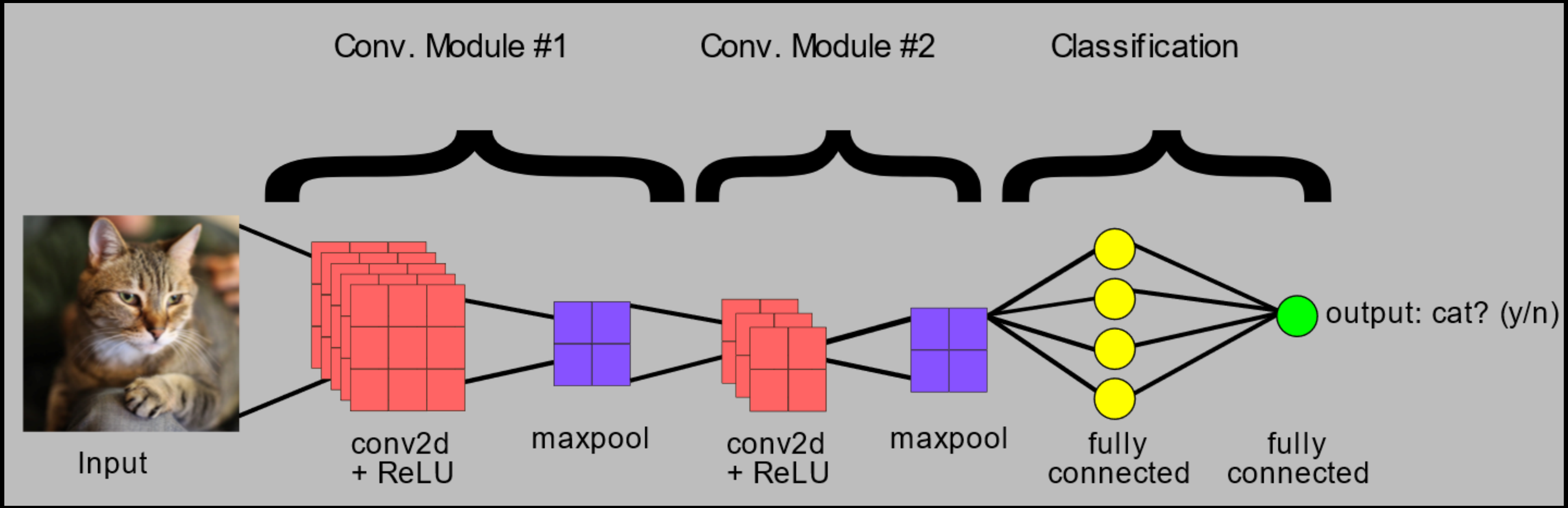
In most of the cases, max pooling is used because its performance is much better than average pooling.

INVARIANCE TO SMALL TRANSLATIONS



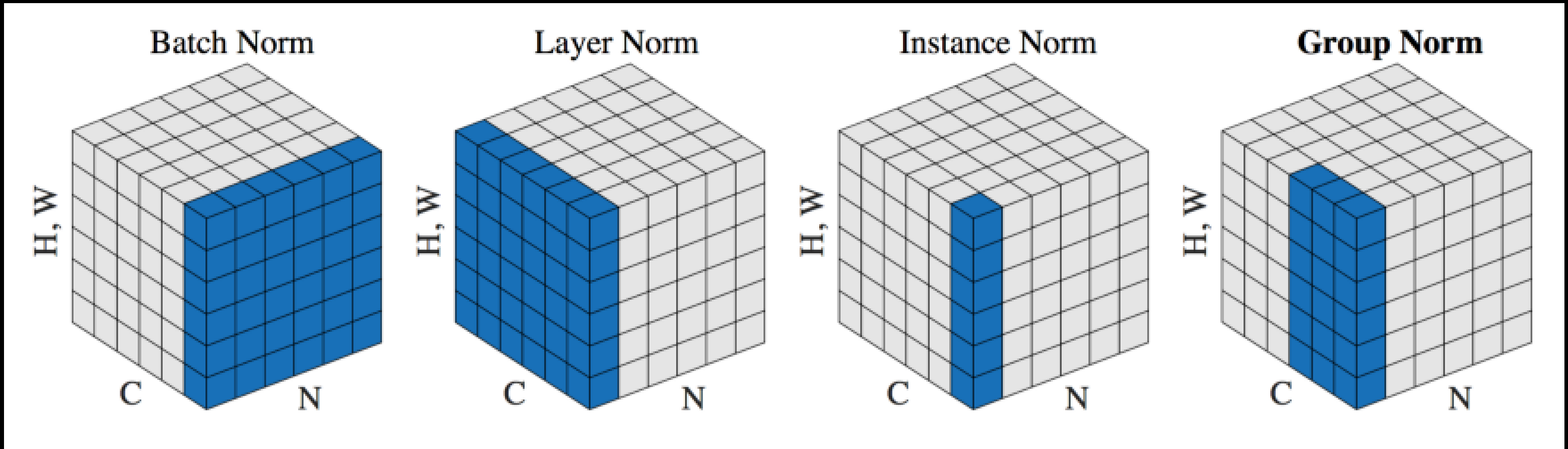
FULLY CONNECTED LAYERS

- Perform classification
- Softmax activation function



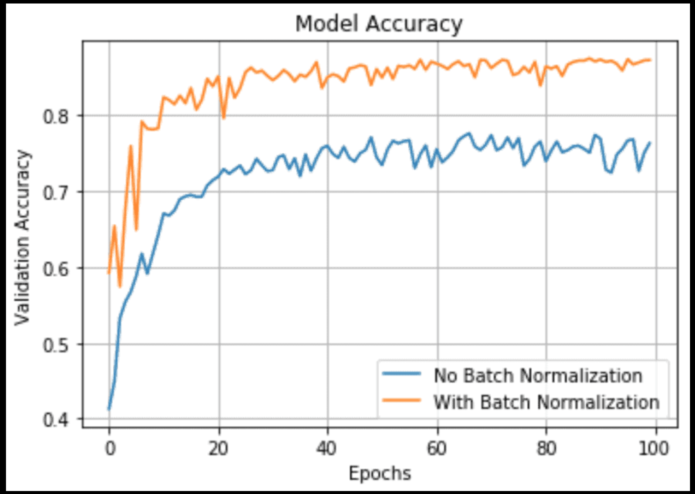
NORMALIZATION LAYER

- Stabilize gradients for optimal performance
- Recenter and rescale our data such that is between 0 and 1 or -1 and 1
- Normalize inputs to intermediate layers



BATCH NORMALIZATION

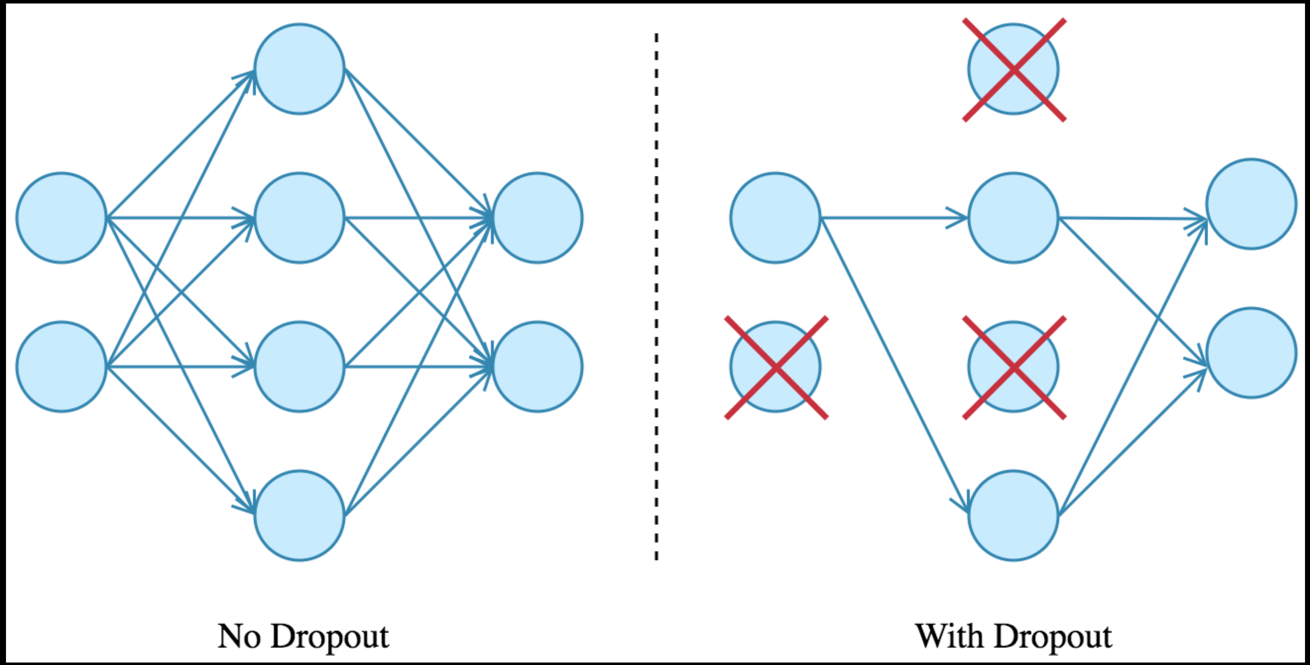
- Standardize the input of a layer across a single batch
- Often impractical to train on entire dataset



See e.g.
https://towardsdatascience.com/different-types-of-normalization-in-tensorflow-dac60396efb0

DROPOUT LAYERS

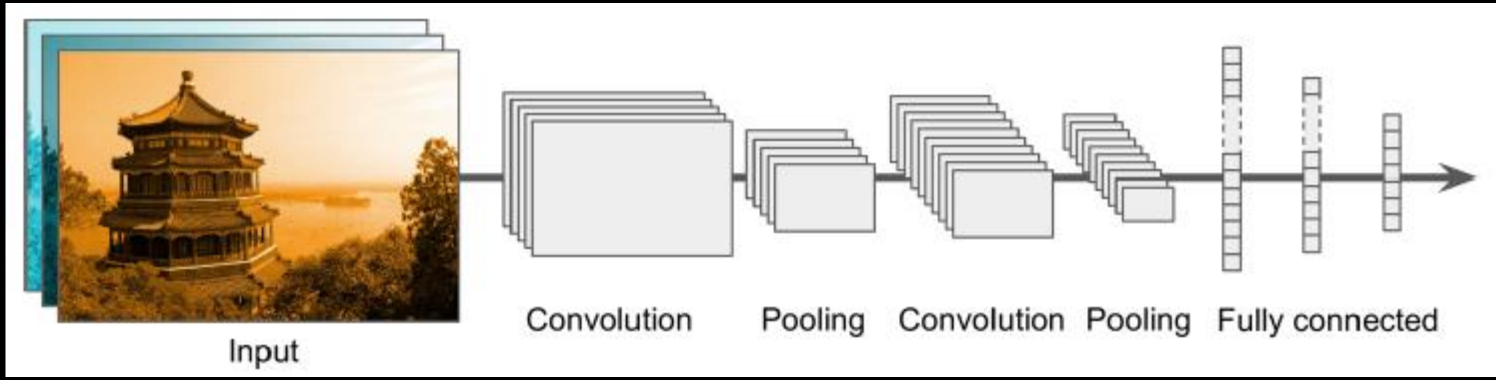
- Randomly removing units from the neural network during a training gradient step.
- Reduce overfitting



Source: https://towardsdatascience.com/applied-deep-learning-part-4-convolutional-neural-networks-584bc134c1e2

HAN_UNIVERSITY
OF APPLIED SCIENCES

TYPICAL CNN ARCHITECTURE



Source: Géron, ISBN: 9781492032632

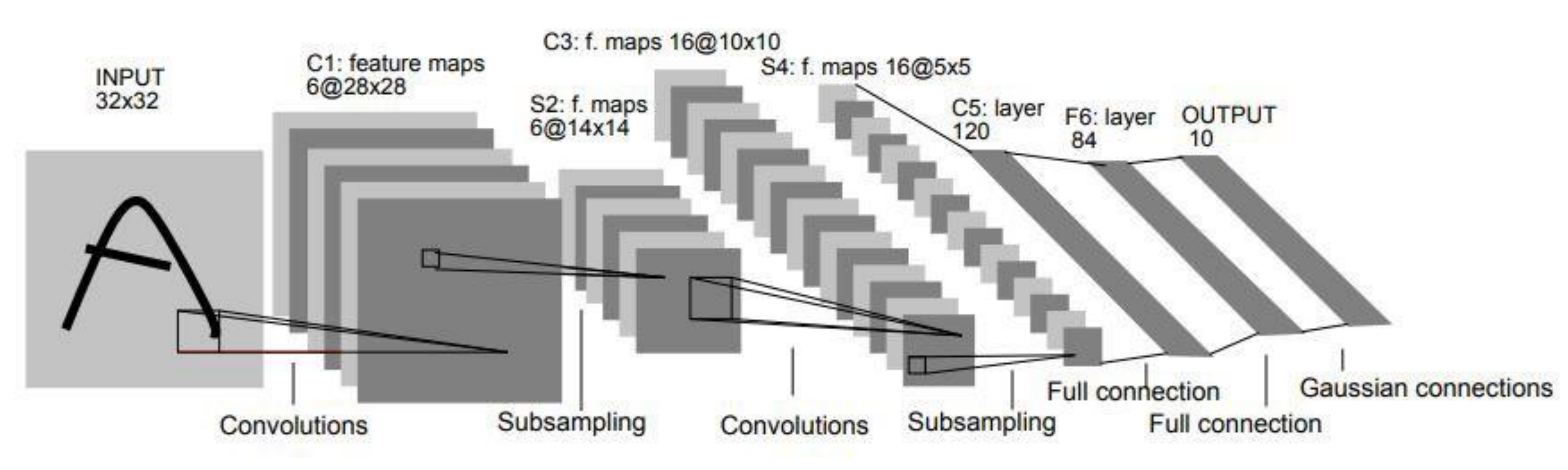
LEVERAGING PRETRAINED MODELS

- 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

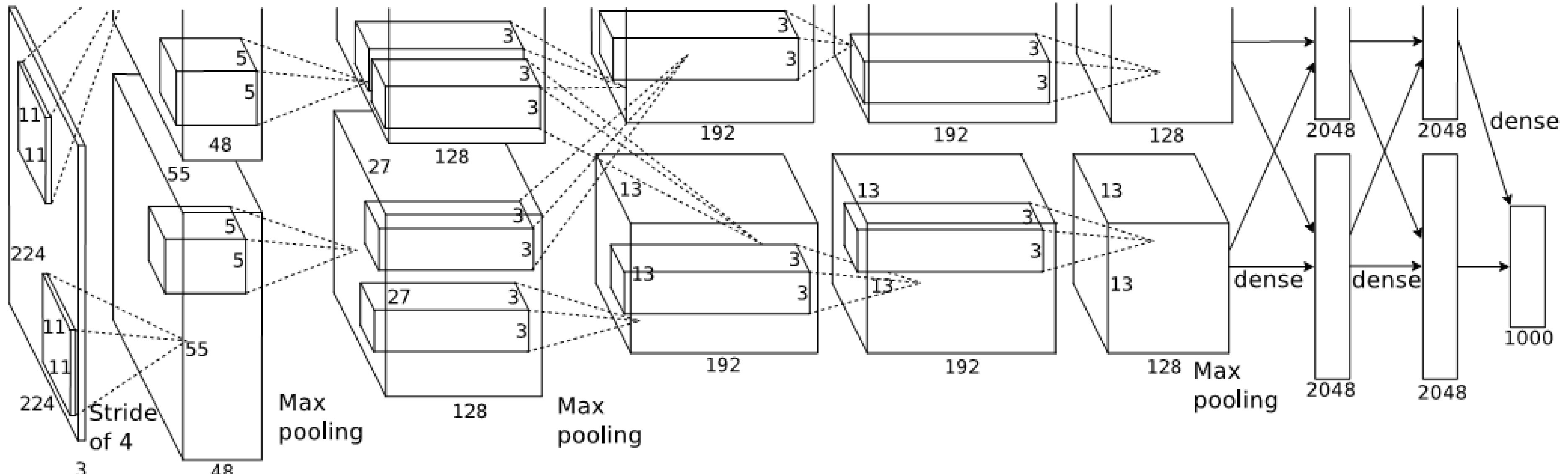
LENET



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

ALEXNET

- ImageNet competition



Source: Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. “Imagenet classification with deep convolutional neural networks.” Advances in neural information processing systems. 2012.

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

HAN_UNIVERSITY
OF APPLIED SCIENCES

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/

HAN_UNIVERSITY
OF APPLIED SCIENCES