Introduction to Machine Learning Applications

Spring 2023

Deep learning

Minor Gordon

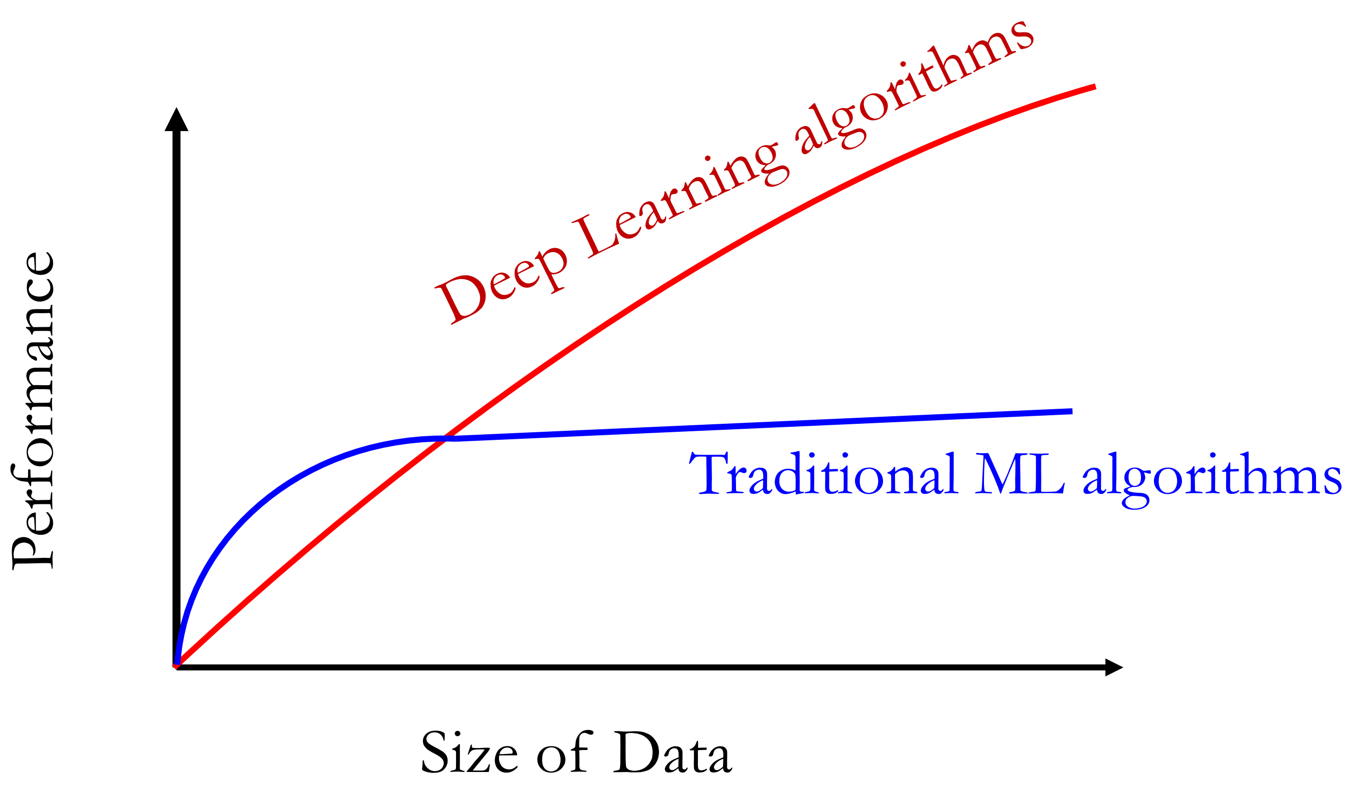
gordom6@rpi.edu

Rensselaer

Deep Learning

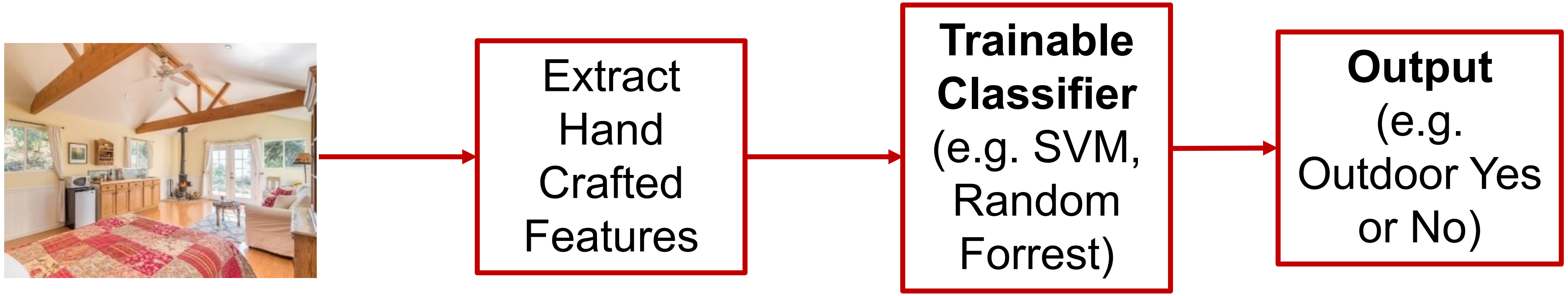
- Neural networks are universal function approximators.
- Deep learning combines large neural network architectures along with innovative training algorithms (Convolutions, RNN, GANS, Autoencoders, etc.) along with (typically) large datasets to build predictive models
- Because of size of data and complexities of operations dedicated hardware (GPU) is often required

Performance vs Sample Size



Traditional Supervised Learning

- Traditional pattern recognition models work with hand crafted features and relatively simple trainable classifiers.

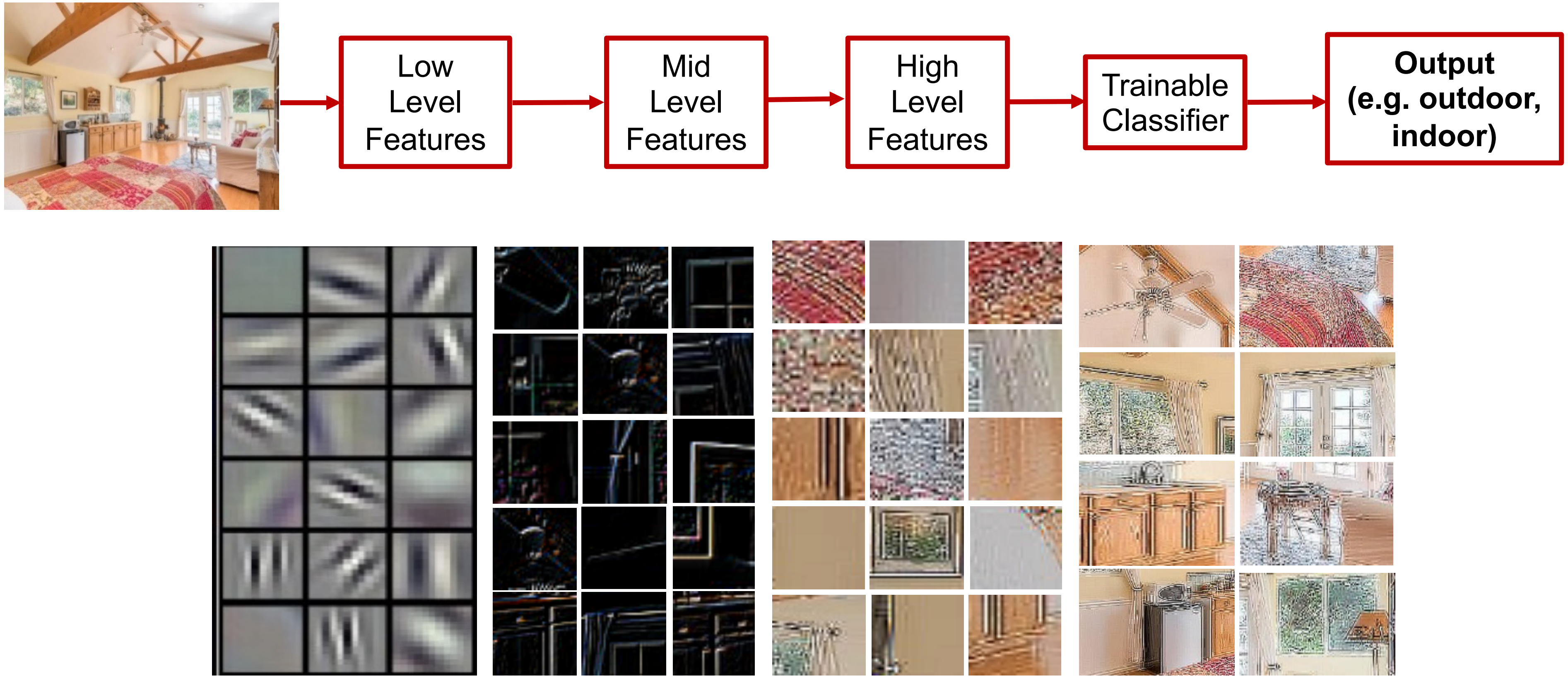


Limitations

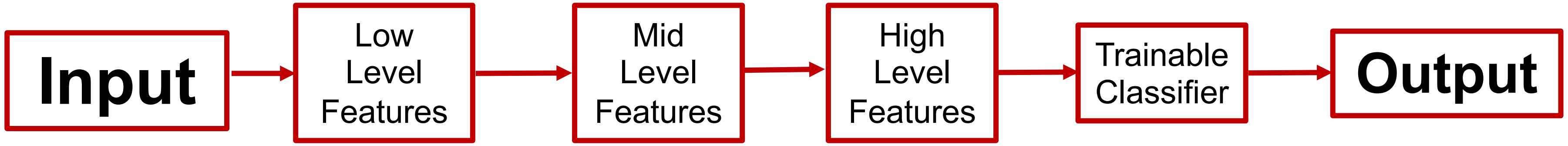
- Very tedious and costly to develop hand-crafted features
- The hand-crafted features are usually highly dependent on one application.

Deep Learning

- Deep learning has a **built-in automatic multi-stage feature learning process** that learns rich hierarchical representations (i.e., features).



Deep Learning



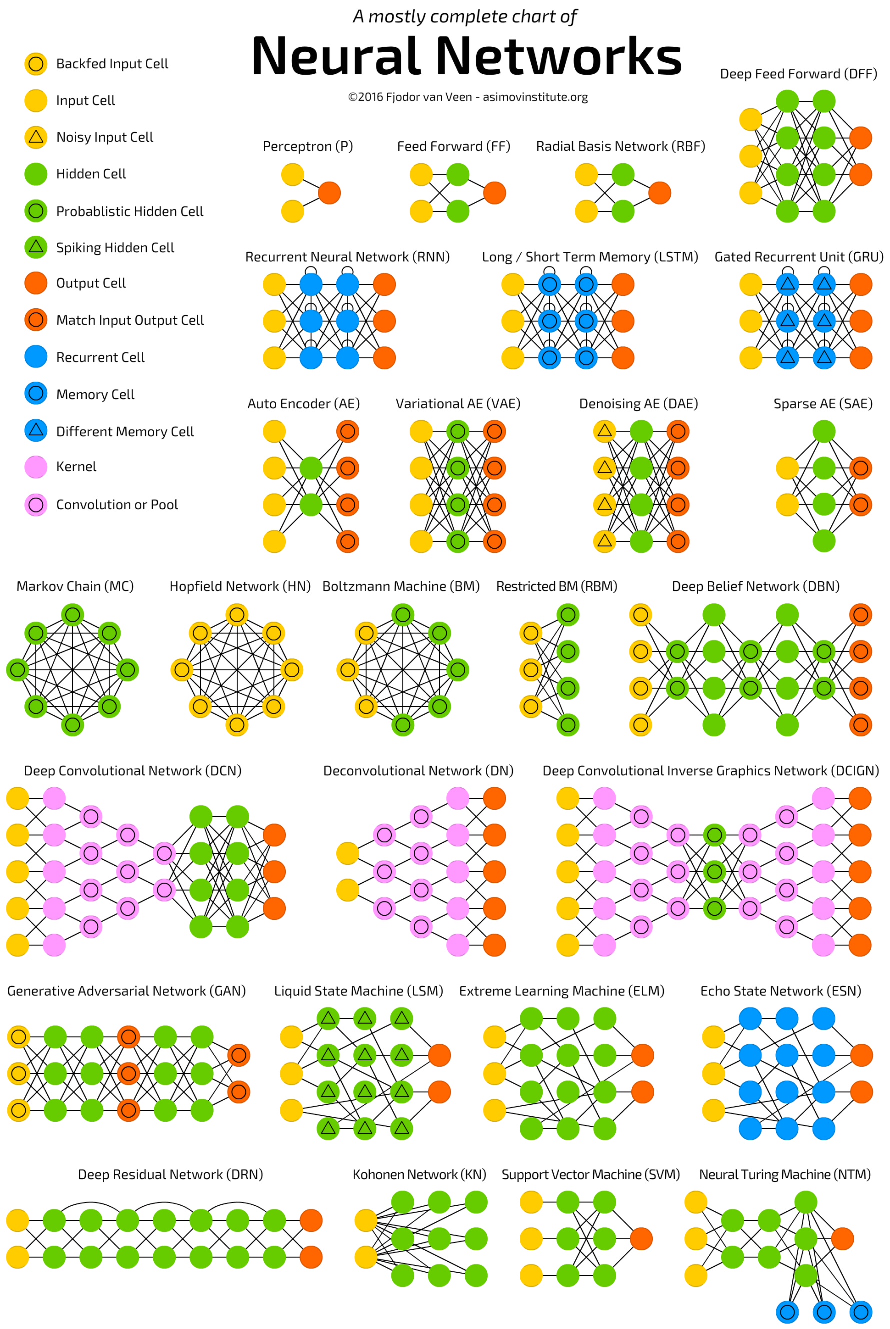
- Image

  Pixel → Edge → Texture → Motif → Part → Object

- Text

  Character → Word → Word-group → Clause → Sentence → Story

- Each module in Deep Learning transforms its input representation into a higher-level one, in a way similar to human cortex.



Many different types of neural networks, with types based on type of problem

Source: http://www.asimovinstitute.org/neural-network-zoo/

Supervised Convolutional Neural Network

Why Convolutional Neural Networks?

A CNN is a deep, feed-forward artificial neural network that has successfully been applied to analyzing visual imagery.

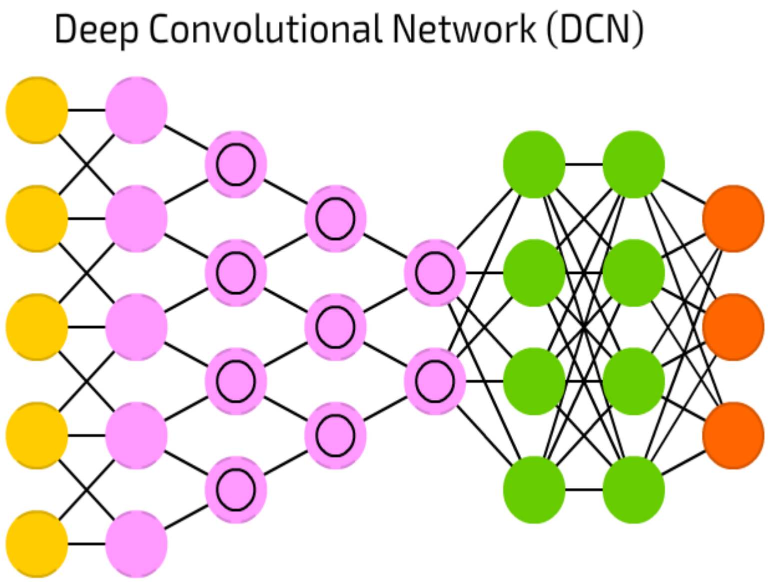
ImageNet Challenge

IMAGENET

- 1,000 object classes (categories).
- Images:
  - 1.2 M train
  - 100k test.

Convolutional Networks

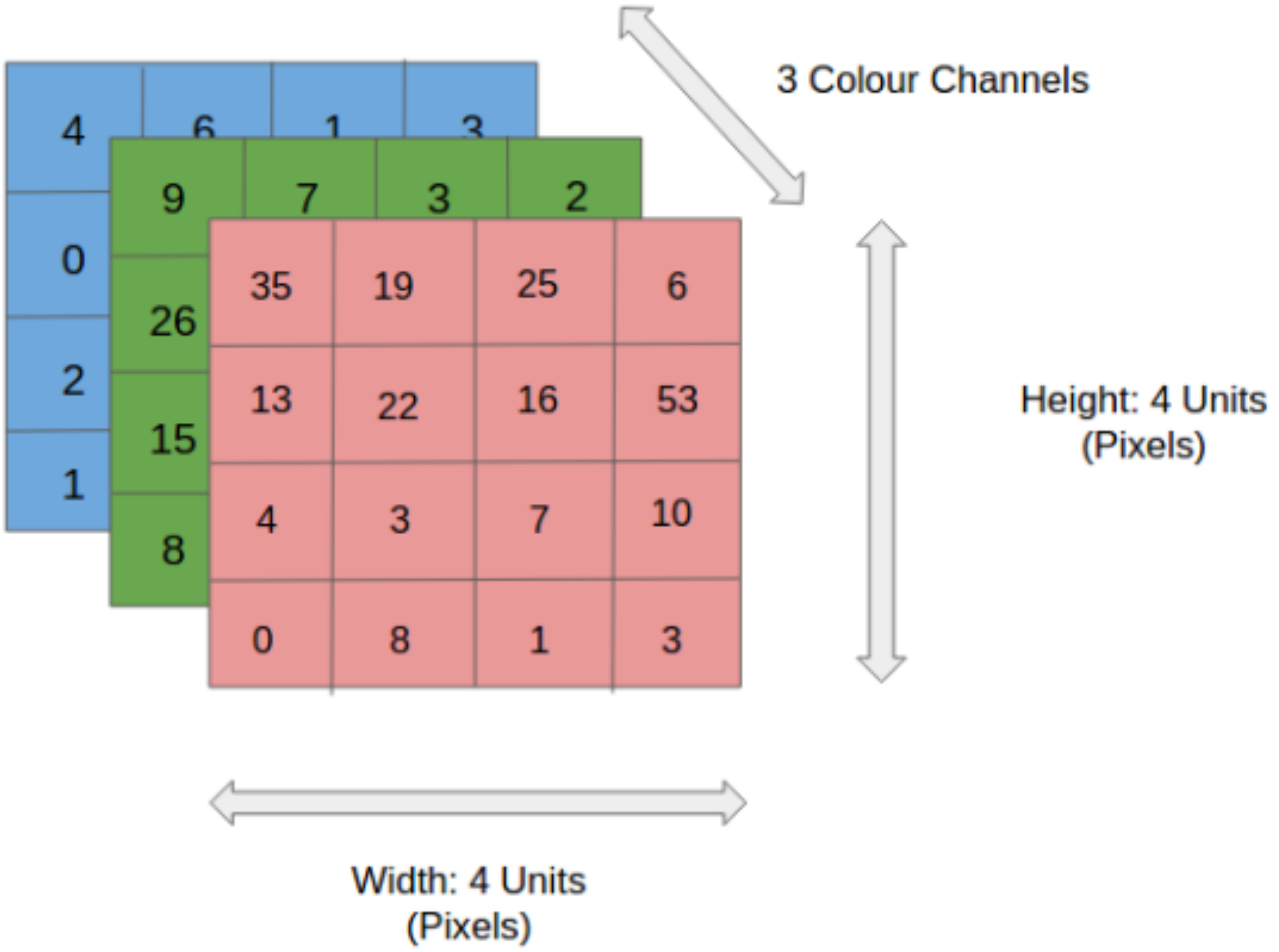
- Image processing
- Sentence Classification



Convolutional Neural Networks

- **Output:** Binary, Multinomial, Continuous, Count
- **Input:** fixed size, can use padding to make all images same size.
- **Architecture:** Choice is ad hoc
  - requires experimentation.
- **Optimization:** Backward propagation
  - hyper parameters for very deep model can be estimated properly only if you have billions of images.
    - Use an architecture and trained hyper parameters from other papers (Imagenet or Microsoft/Google APIs etc)

Input Image

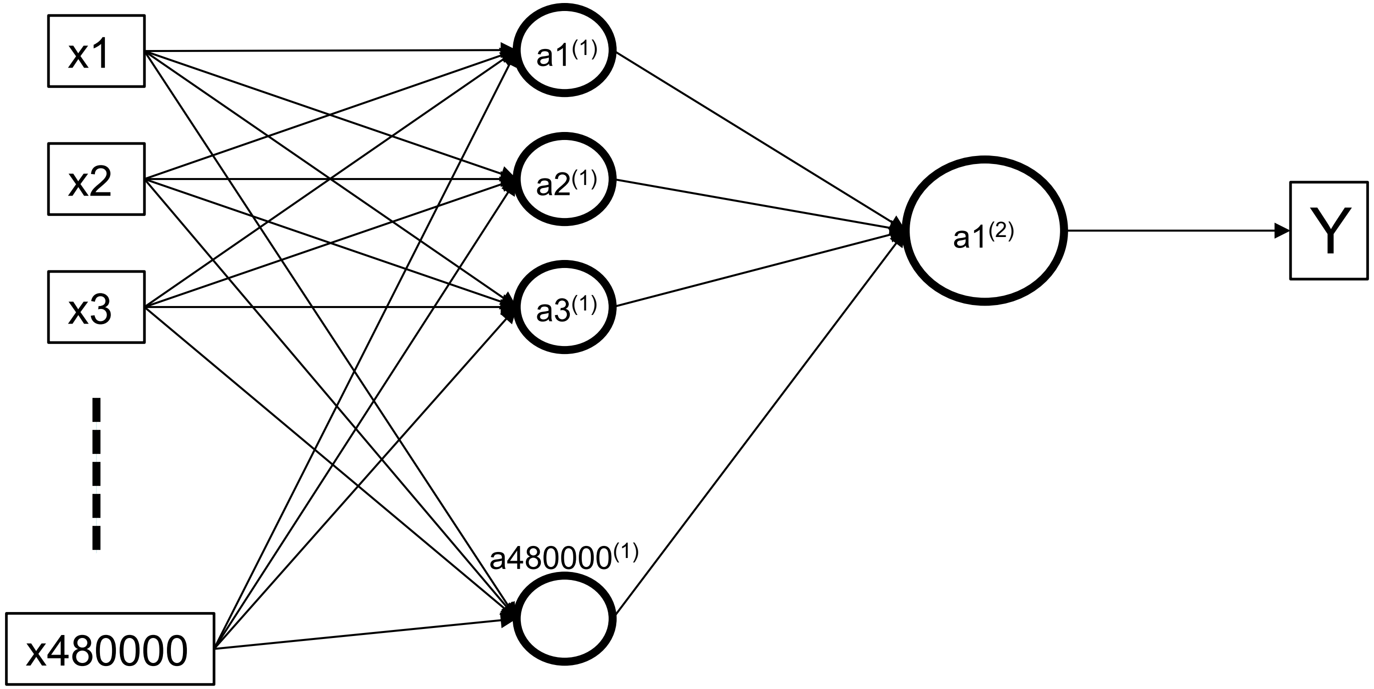


4
6
1
3
0
2
1
9
7
3
2
26
15
8
35
19
25
6
13
22
16
53
4
3
7
10
0
8
1
3
3 Colour Channels
Height: 4 Units
(Pixels)
Width: 4 Units
(Pixels)

Input an Image to a Neural Network

400 X 400 X 3

400px x 400px
x 3 color
channels



Input

Hidden Layers

Output

Number of Parameters

480000*480000 + 480000 +1 = **approximately 230 Billion !!!**

480000*1000 + 1000 +1 = approximately **480 million !!!**

Convolutional Layers

- Filter  $\begin{pmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{pmatrix}$

Input Image

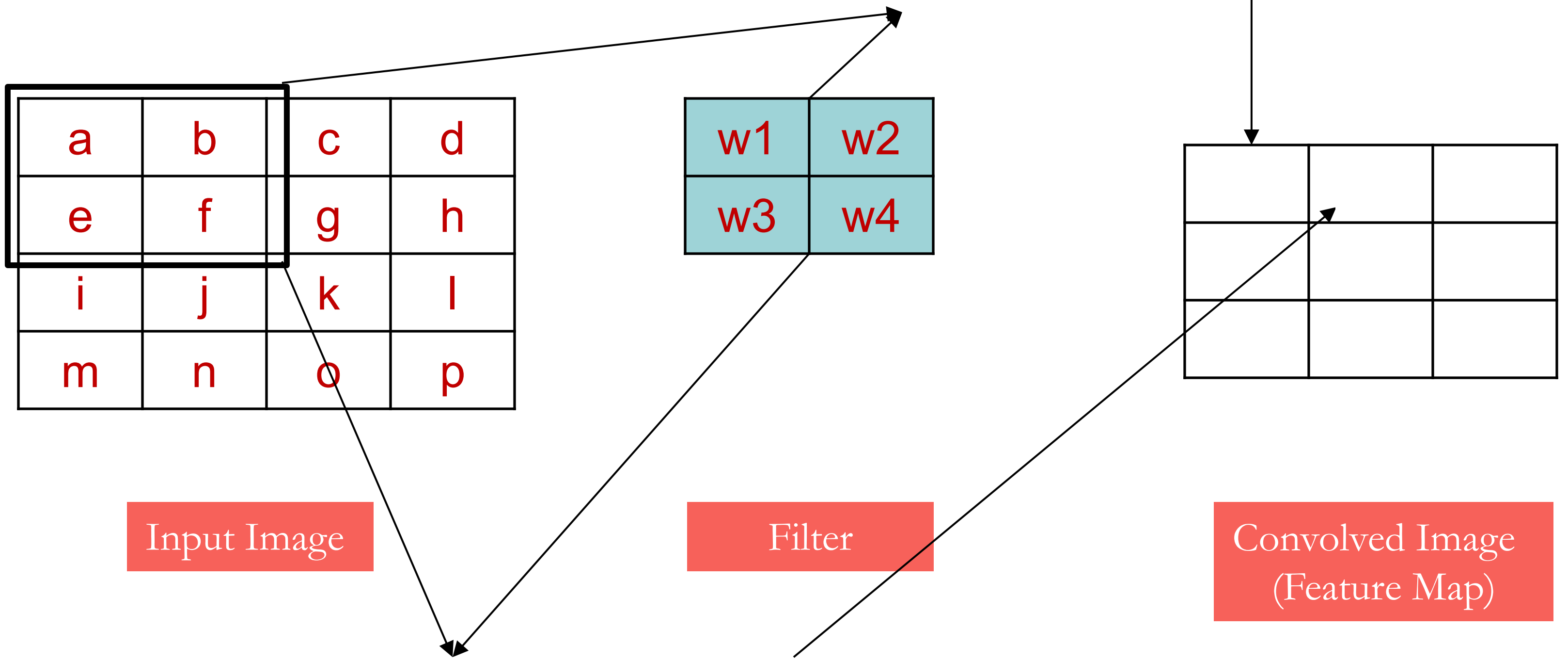
Convoluted Image

- Inspired by the neurophysiological experiments conducted by Hubel and Wiesel 1962.

Convolutional Layers

- What is a convolution?

$$h1 = f(a * w1 + b * w2 + e * w3 + f * w4)$$

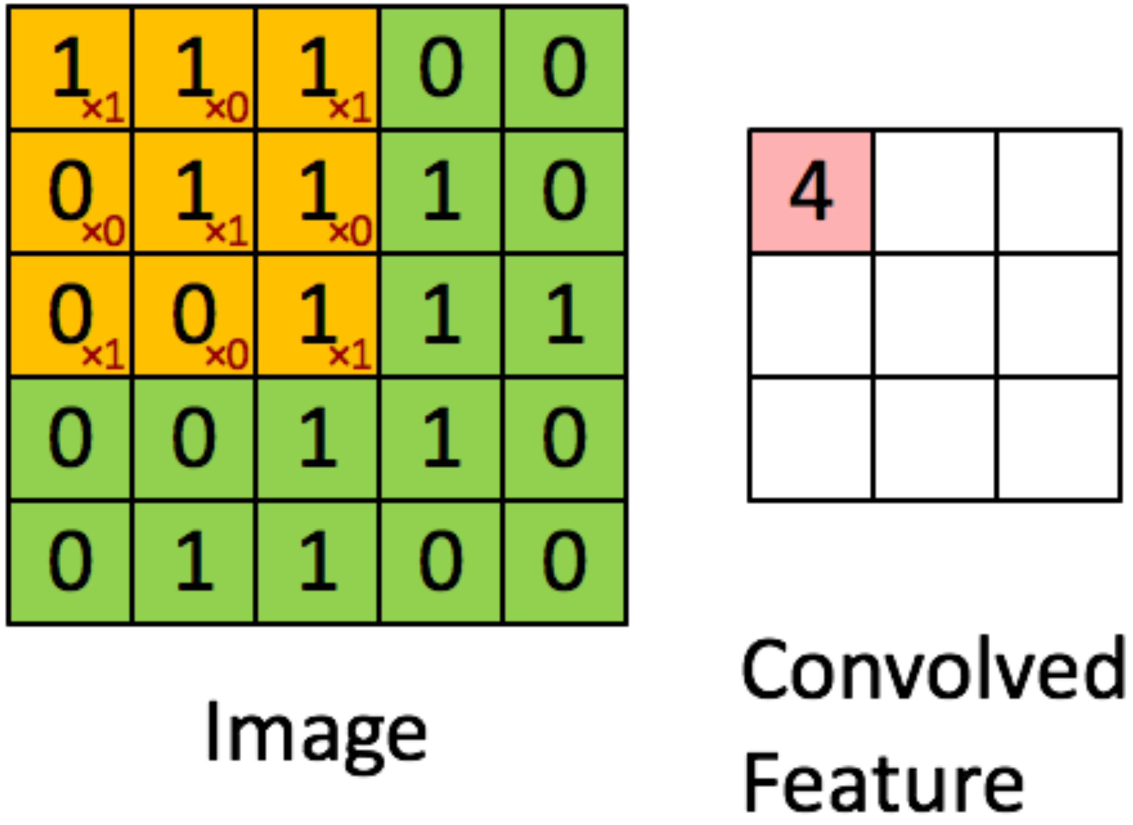


$$h2 = f(b * w1 + c * w2 + f * w3 + g * w4)$$

Number of Parameters for one feature map = 4

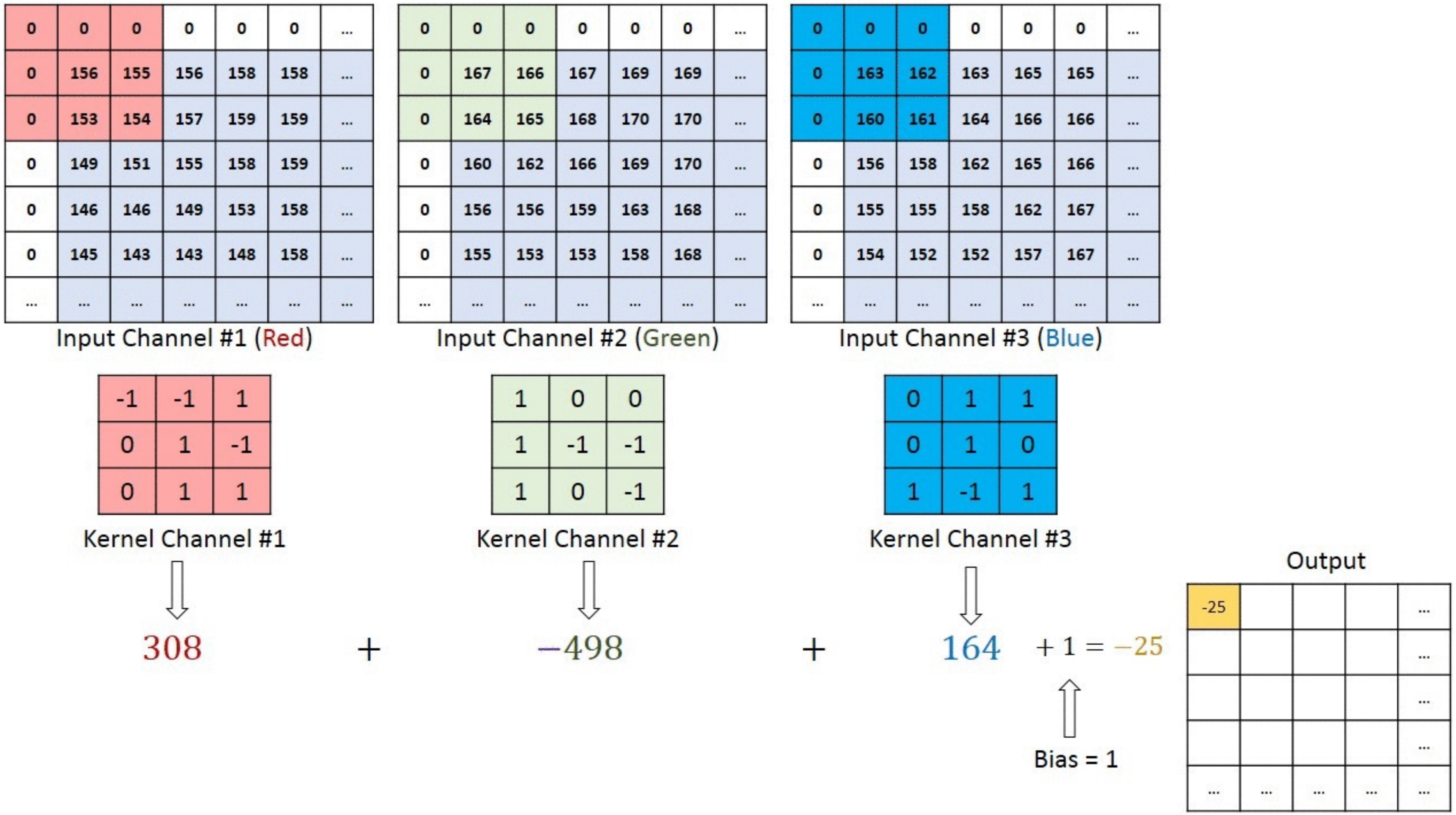
Number of Parameters for 100 feature map = 4*100

Convolution Layer – The Kernel



Kernel/Filter, K =
1 0 1
0 1 0
1 0 1

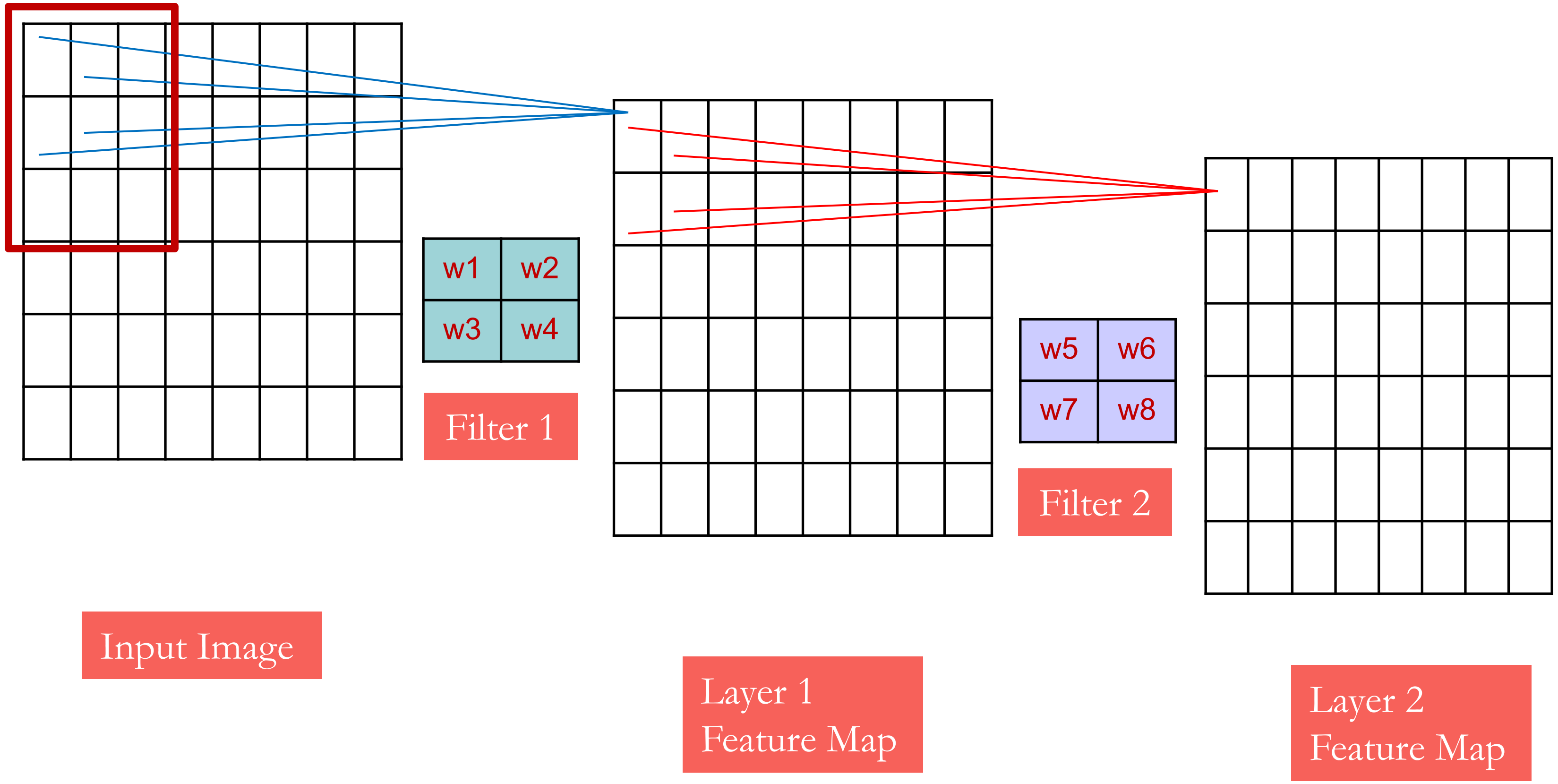
Convolution operation on a MxNx3 image



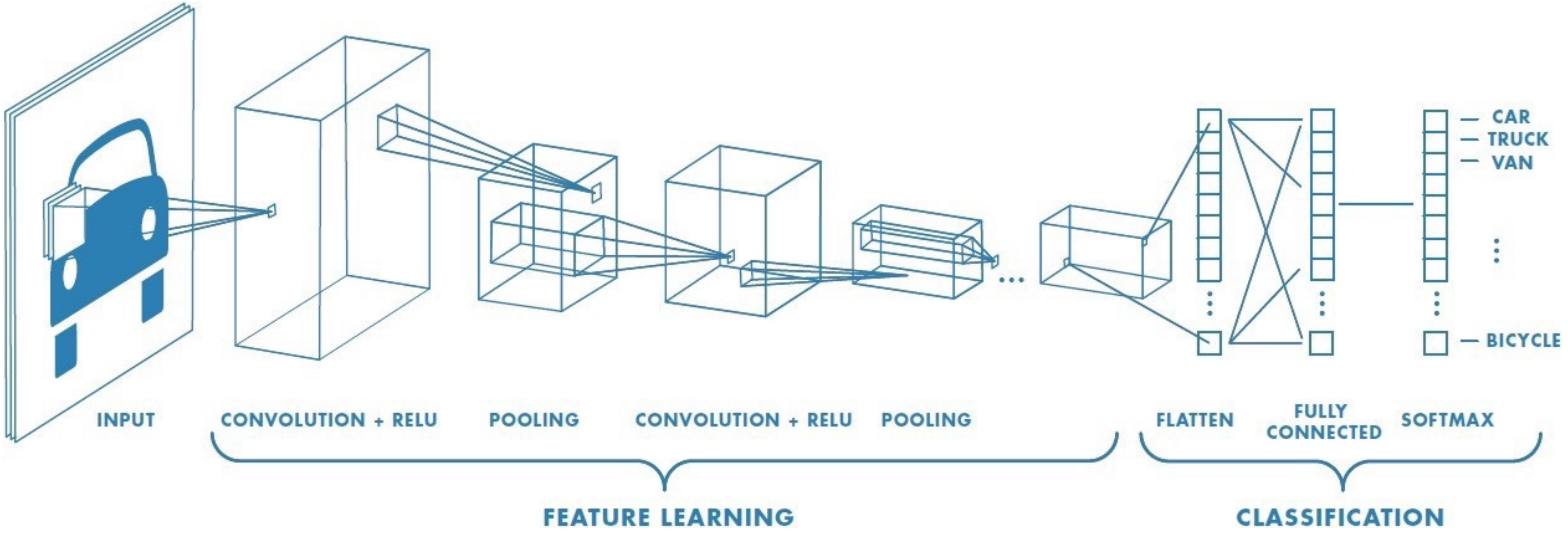
Convolution operation

- The objective of the Convolution Operation is to **extract the high-level features** such as edges, from the input image.
- ConvNets need not be limited to only one Convolutional Layer.
- Conventionally, the first ConvLayer is responsible for capturing low-level features such as edges, color, gradient orientation, etc.
- With added layers, the architecture adapts to high-level features as well.

Lower Level to More Complex Features



- In Convolutional neural networks, hidden units are only connected to local receptive field.



Credits: https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53

Pooling

- Max pooling: reports the maximum output within a rectangular neighborhood.
- Average pooling: reports the average output of a rectangular neighborhood.

1	3	5	3
4	2	3	1
3	1	1	3
0	1	0	4

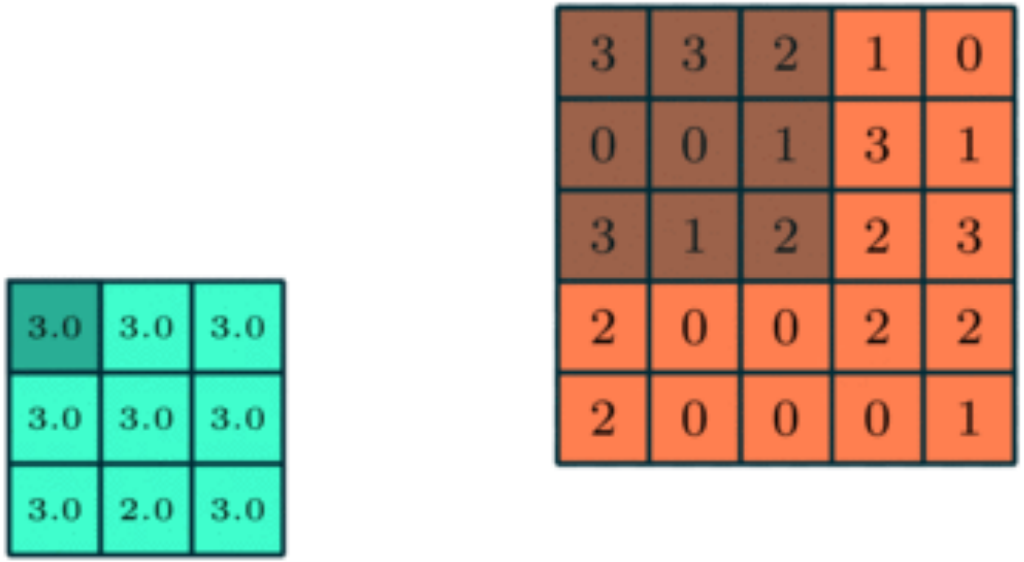
Input Matrix

MaxPool with 2X2 filter with stride of 2

4	5
3	4

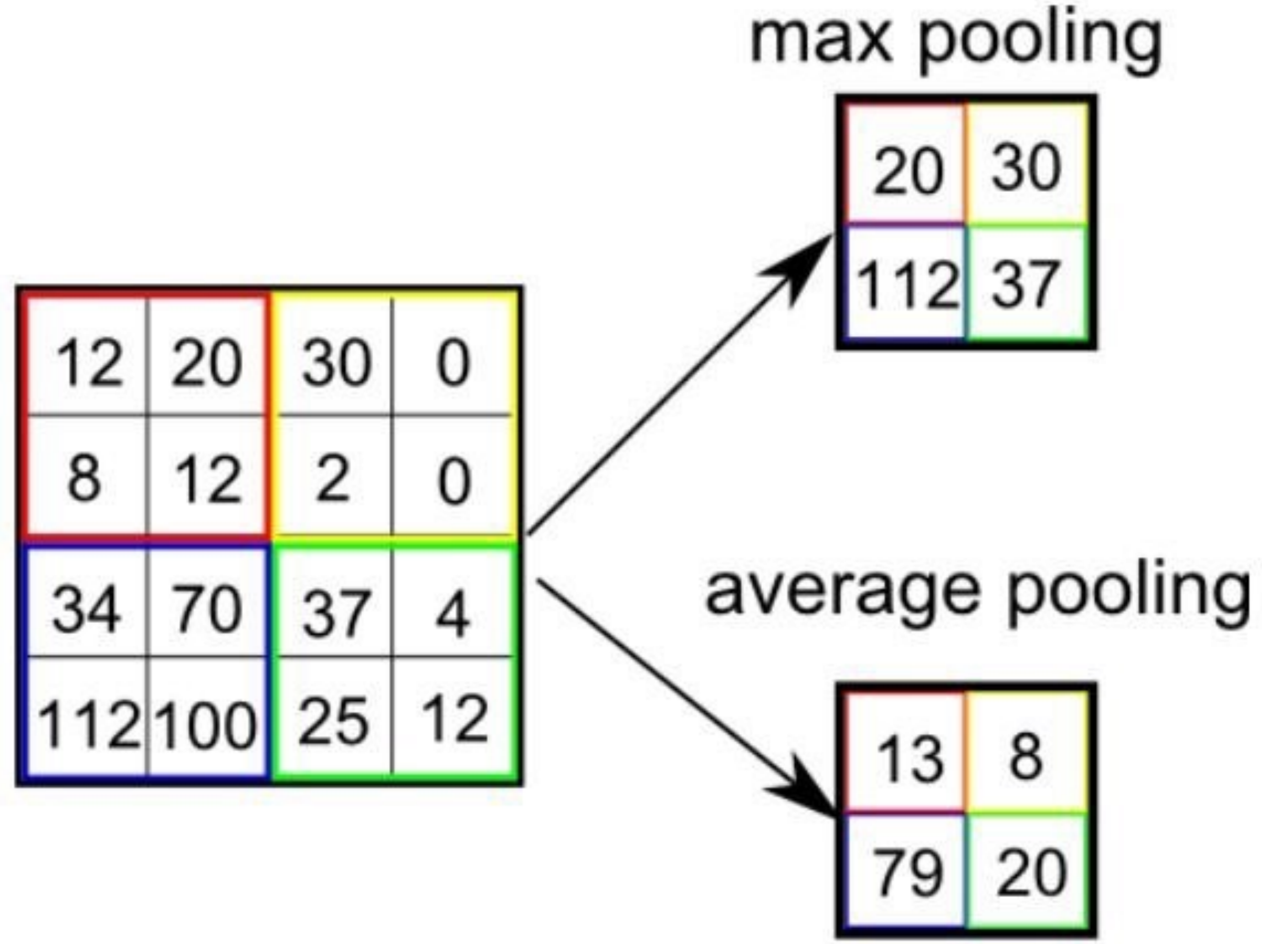
Output Matrix

Pooling layer

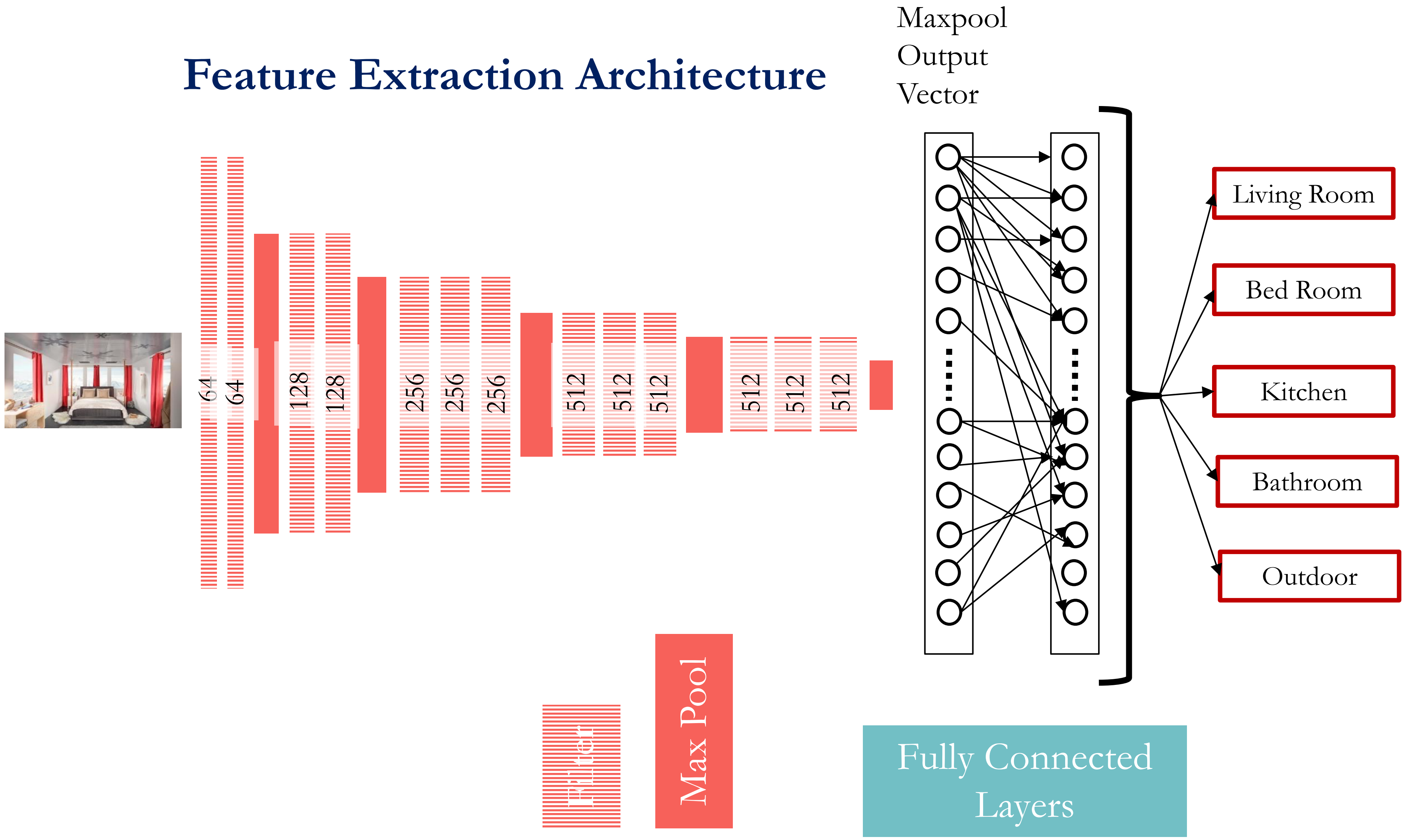


- **Max Pooling** returns the **maximum value** from the portion of the image covered by the Kernel.
- **Average Pooling** returns the **average of all the values** from the portion of the image covered by the Kernel.

Pooling example



Convolutional Neural Network



Batch normalization

- Main goal is to increase the stability of a neural network.
- This technique normalizes the output of a previous activation layer by subtracting the batch mean and dividing by the batch standard deviation.
- We can use higher learning rates because batch normalization makes sure that there’s no activation that’s gone really high or really low.
- Reduces overfitting
  - By using this approach, we can avoid dropout, and thus avoid losing information.

Pretrained Models and Transfer Learning

Pretrained Models – Key to a Quick Start

- Possible to take already trained models and retrain the last layer to do something specific, even if the model has been trained on something general.

The Inception Model:
Deep Convolutional
Neural Network
Capable of identifying
1,000 object classes
trained from 100K
images.

ImageNet Challenge

IMAGENET

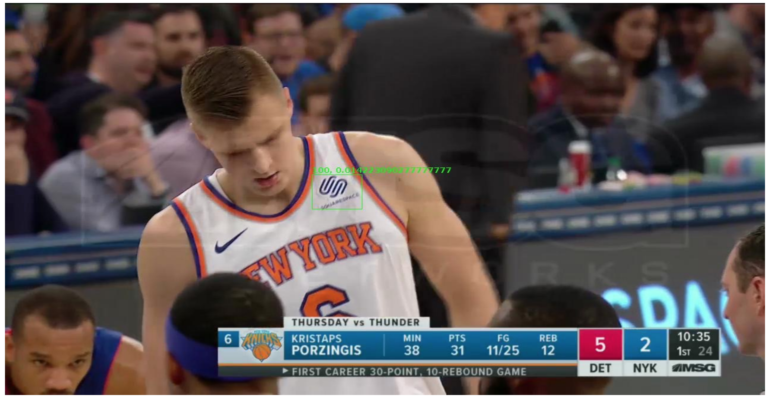
- 1,000 object classes (categories).
- Images:
  - 1.2 M train
  - 100k test.

https://github.com/tensorflow/models

Pretrained Models – Reusing Last Layer

- Let's say you need to identify Star Wars Characters to detect for Copyright violations.

- Or detect logos in NBA games



https://github.com/llSourcell/tensorflow_image_classifier

Transfer Learning

- Transfer learning is a machine learning method where a model developed for a task is reused as the starting point for a model on a second task.

Pre-trained Model Approach

1. **Select Source Model**
2. **Reuse Model.**
3. **Tune Model (Maybe retrain last layers)**