Data ^{X}

About Me:

Introduction To Convolution Neural Networks

Ikhlaq Sidhu, Chief Scientist, SCET/IEOR, UC Berkeley
Sana Iqbal, GSI, Data-X, UC Berkeley

Review: A simple Feed Forward Neural Net

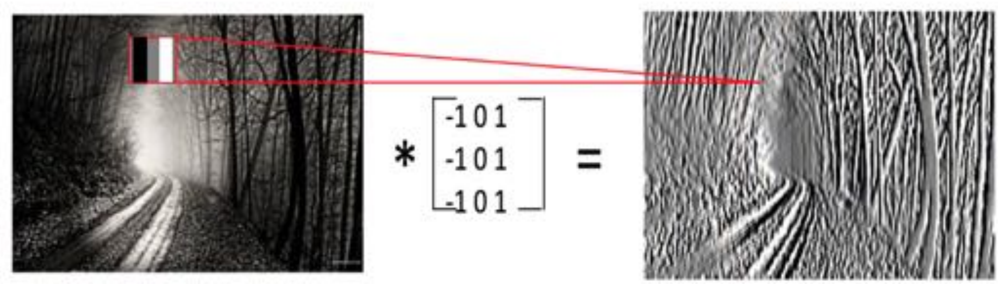
1. Accepts the Input.
2. Creates different combinations of weighted input
3. Induces Non-linearity in the input using Activations.
4. Repeats 1,2,3 in hidden layers.
5. Gives the Output.

Review: Convolution functions in Image Processing

- Convolution functions applied between an image matrix and a filter matrix gives us information about the spatial distribution of the image features.

- Using different types of filters affects the information extracted from an image- blur, edges, texture.

Convolutional of Two Signals



$$f[x,y] * g[x,y] = \sum_{n_1=-\infty}^{\infty} \sum_{n_2=-\infty}^{\infty} f[n_1,n_2] \cdot g[x-n_1, y-n_2]$$

elementwise multiplication and sum

Before We Go On

X is an image

h is filter (or kernal)

What is X * h?

$$\begin{bmatrix} -1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

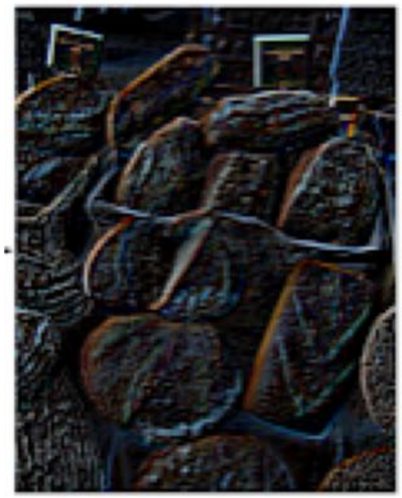
Before We Go On

X is an image

h is filter (or kernal)

$$\begin{bmatrix} -1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

What is X * h?



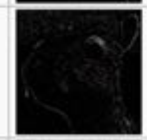
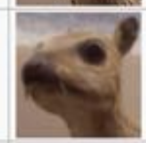
Answer:

1. It is another image
2. Depending on the kernal, it highlights specific features

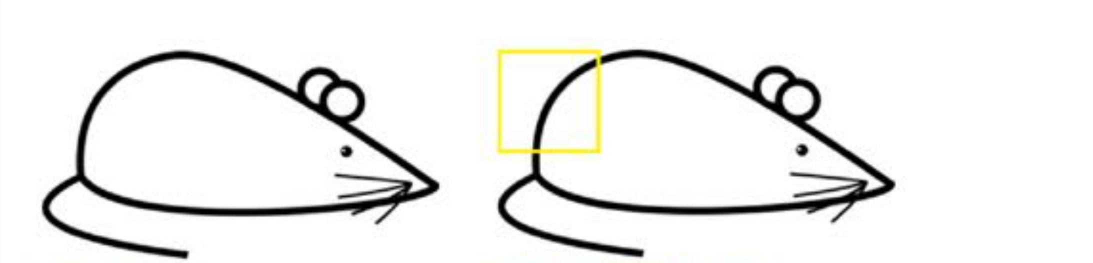
Many Operations with Convolution

These filter are more typical for image processing, but not feature extraction.

In CNNs, we are looking to extract features

Operation	Filter	Convolved Image
Identity	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	
Edge detection	$\begin{bmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix}$	
	$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$	
	$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$	
Sharpen	$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$	
Box blur (normalized)	$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$	
Gaussian blur (approximation)	$\frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$	

Feature Extraction



Original image

Visualization of the filter on the image

Visualization of the receptive field

0	0	0	0	0	0	30
0	0	0	0	50	50	50
0	0	0	20	50	0	0
0	0	0	50	50	0	0
0	0	0	50	50	0	0
0	0	0	50	50	0	0
0	0	0	50	50	0	0

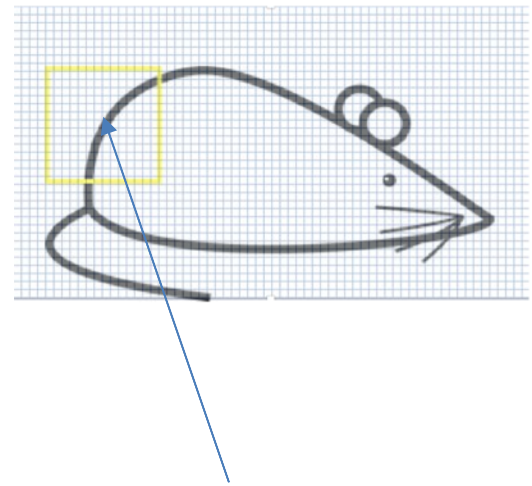
Pixel representation of the receptive field

*

0	0	0	0	0	30	0
0	0	0	0	30	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	0	0	0	0

Pixel representation of filter

Multiplication and Summation = (50*30)+(50*30)+(50*30)+(20*30)+(50*30) = 6600 (A large number!)



Value of convolution at this spot
is large because the sum of products

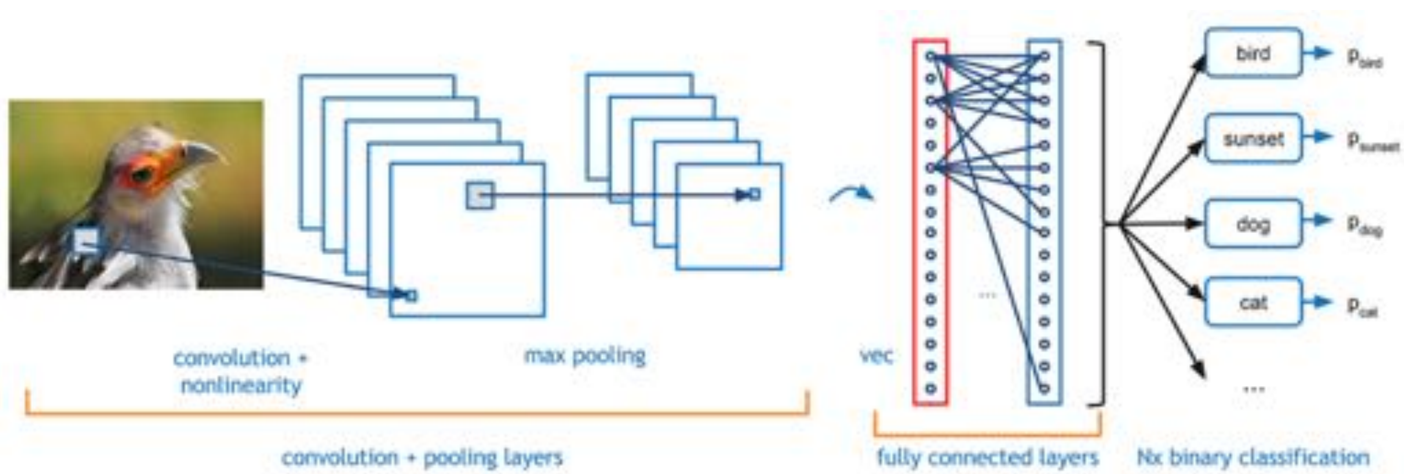
Think of this like a "moving dot product" of the kernel across the original image

https://adeshpande3.github.io/adeshpande3.github.io/A-Beginner%27s-Guide-To-Understanding-Convolutional-Neural-Networks/

Convolution Neural Networks

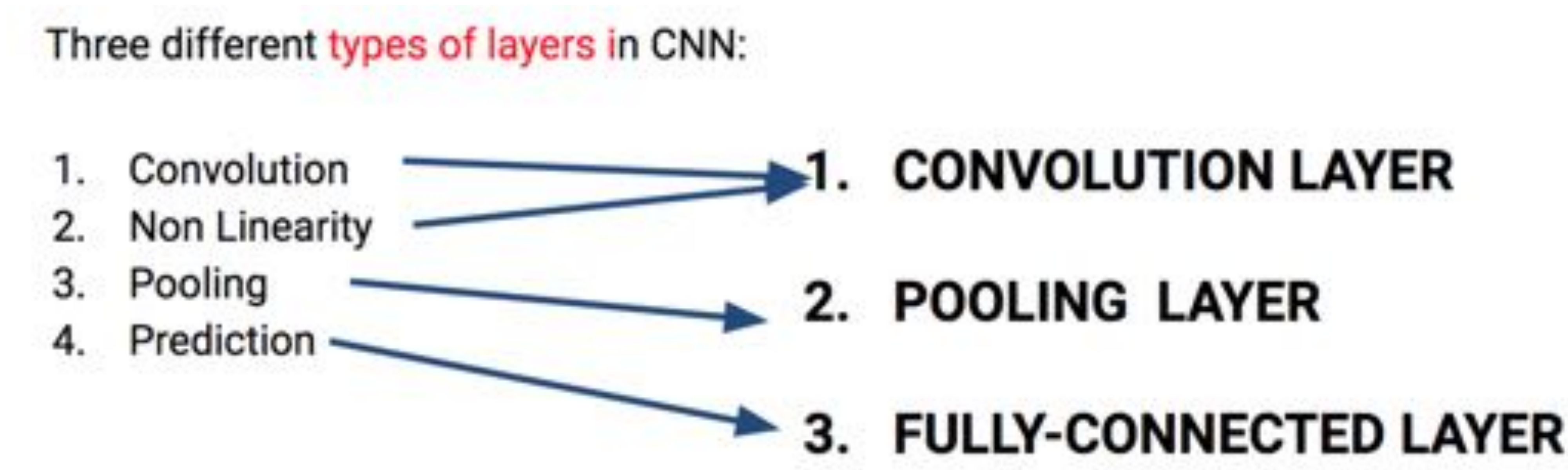
So there are four main operations in a basic CNN:

1. Convolution
2. Non Linearity (ReLU, tanh, sigmoid)
3. Pooling or Down-Sampling of features
4. Prediction



https://adeshpande3.github.io/adeshpande3.github.io/A-Beginner%27s-Guide-To-Understanding-Convolutional-Neural-Networks/

Convolution Neural Networks



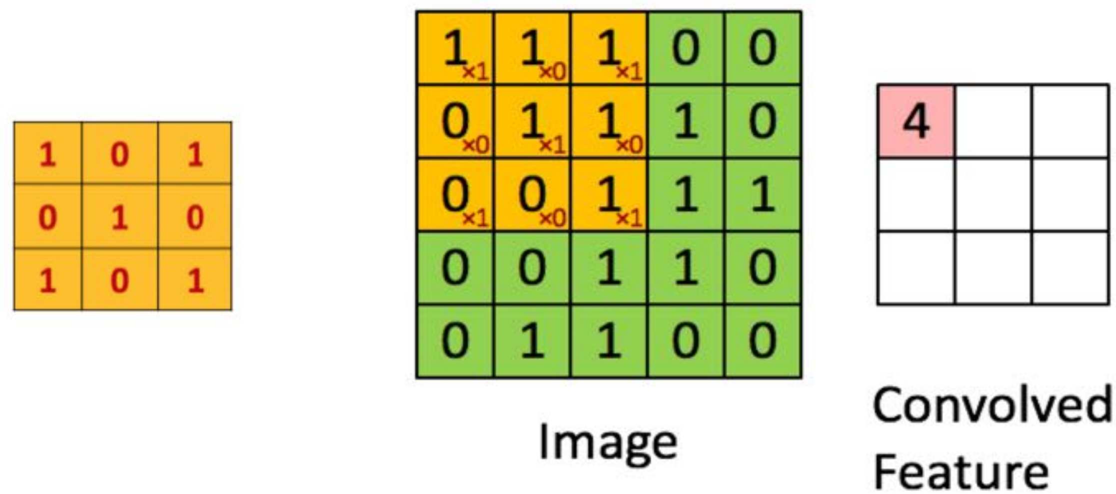
1. CONVOLUTION LAYER

A. This layer has the filters that we want to apply on our image
B. Each filter also called kernel has same depth as the channel of the image

For an image with one channel the below filter works as:

Sizes:
Image of h x w
Kernel of f x g

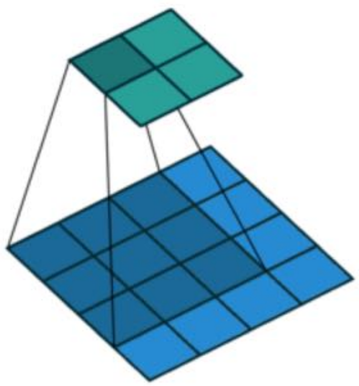
New image: (h-f+1) x (w-g+1)



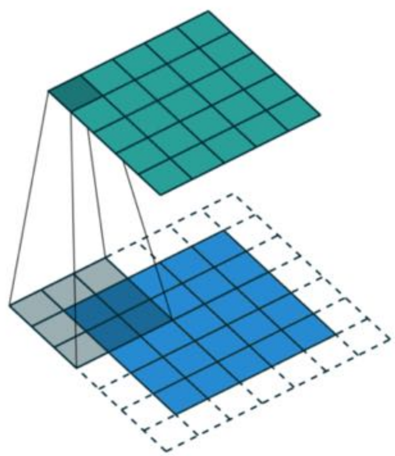
1. CONVOLUTION LAYER

Hyperparamters that can be tuned to change complexity and size of extracted features

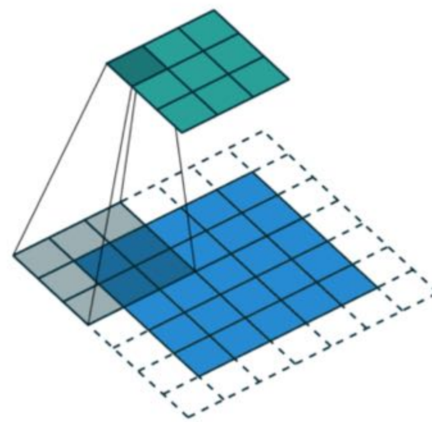
Filter Size, **Stride** (step size of the filter), **Zero-padding** (to maintain dimensions)



- Stride 1
- No Zero padding
- Kernel size 3x3
- Input size 4x4
- Output size 2x2



- Stride 1
- Zero padding 1
- Kernel size 3x3
- Input size 5x5
- Output size 5x5



- Stride 2
- Zero padding 1
- Kernel size 3x3
- Input size 5x5
- Output size 3x3

$$n_{out} = \left\lfloor \frac{n_{in} + 2p - k}{s} \right\rfloor + 1$$

n_{in}: number of input features
n_{out}: number of output features
k: convolution kernel size
p: convolution padding size
s: convolution stride size

n_in = n_out **when**

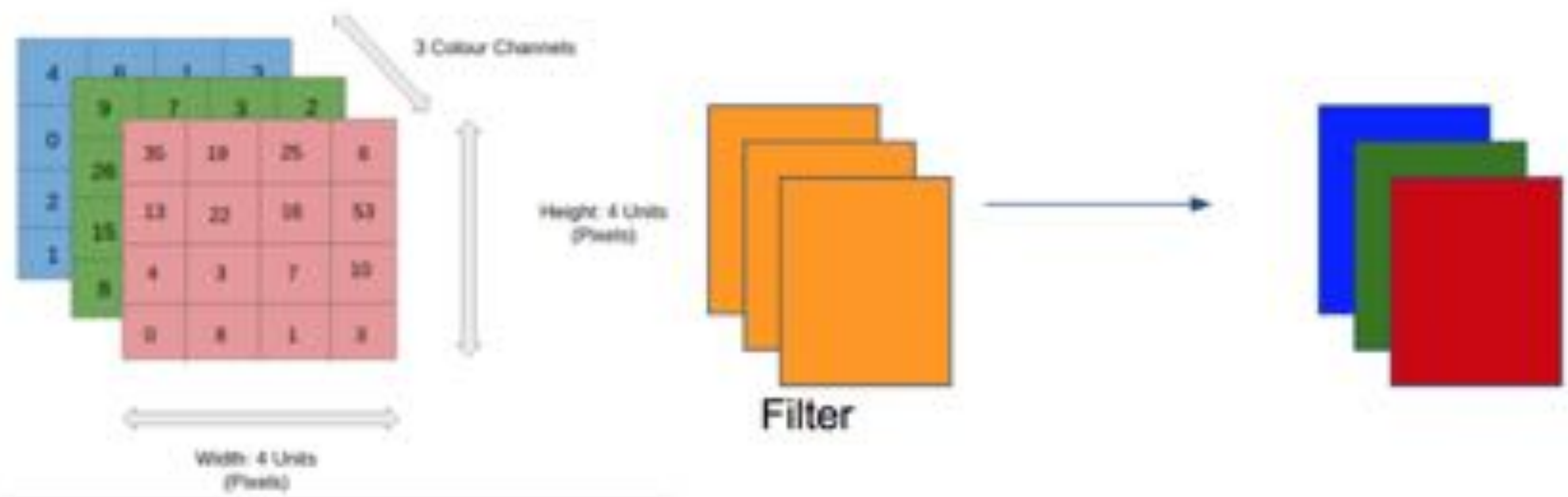
$$Zero\ Padding = \frac{(K-1)}{2}$$

(K = kernel size)

Image Source: https://github.com/vdumoulin/conv_arithmetic

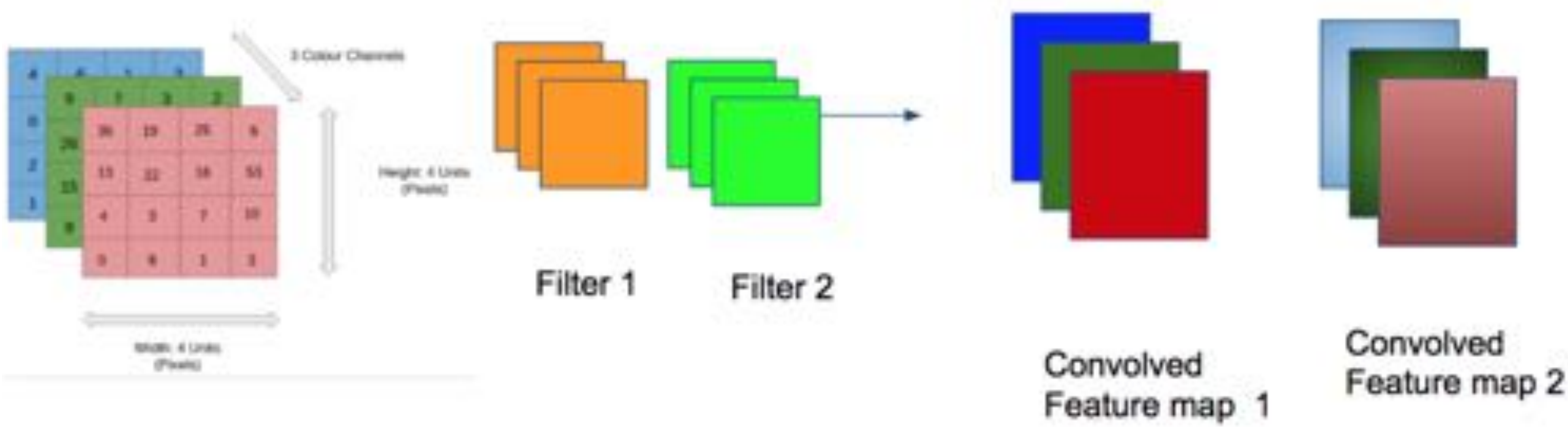
1. CONVOLUTION LAYER

For an image with 3 channels the there would be 1 filter for each channel

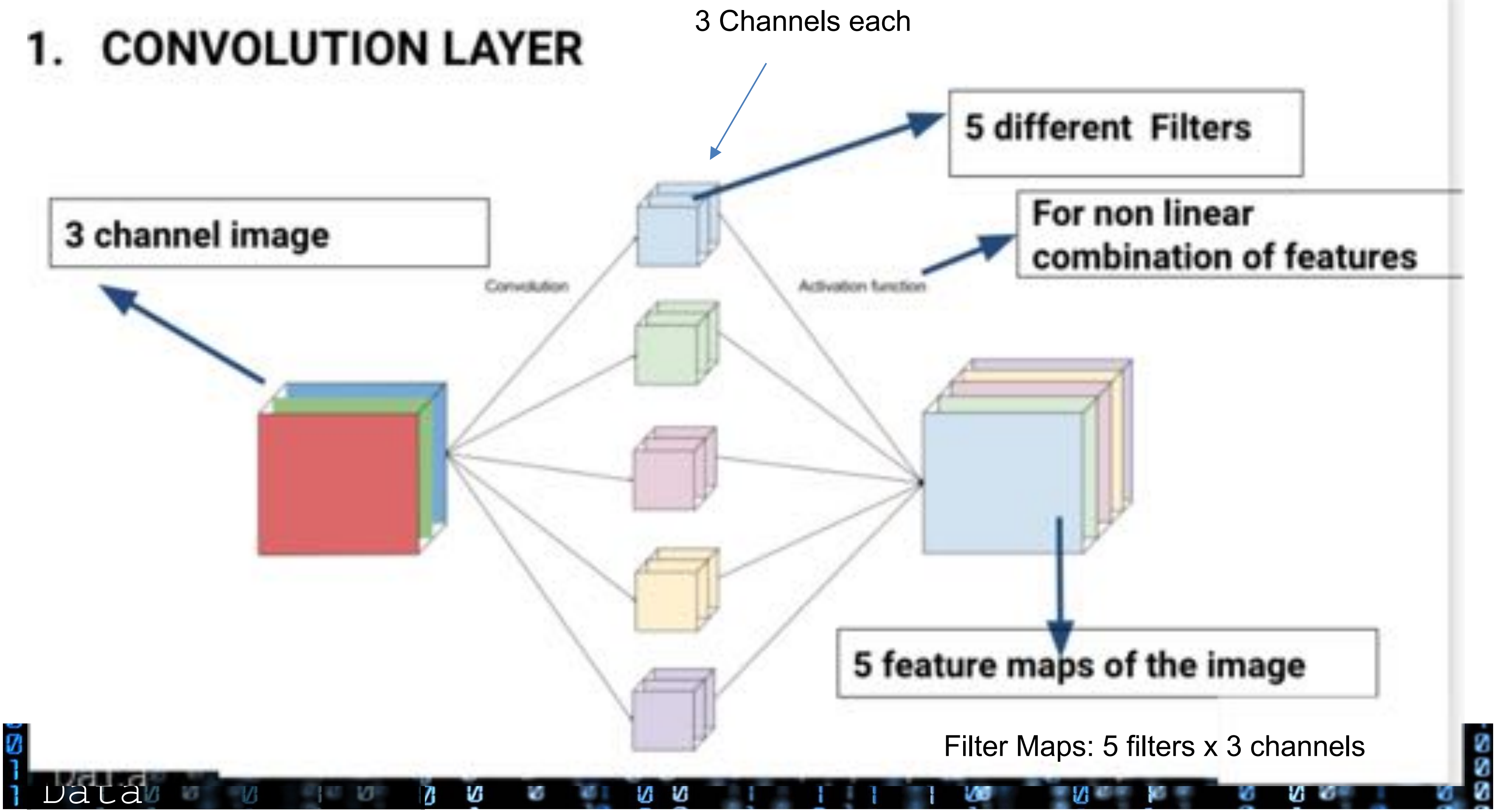


1. CONVOLUTION LAYER

For an image with 3 channels and multiple filters

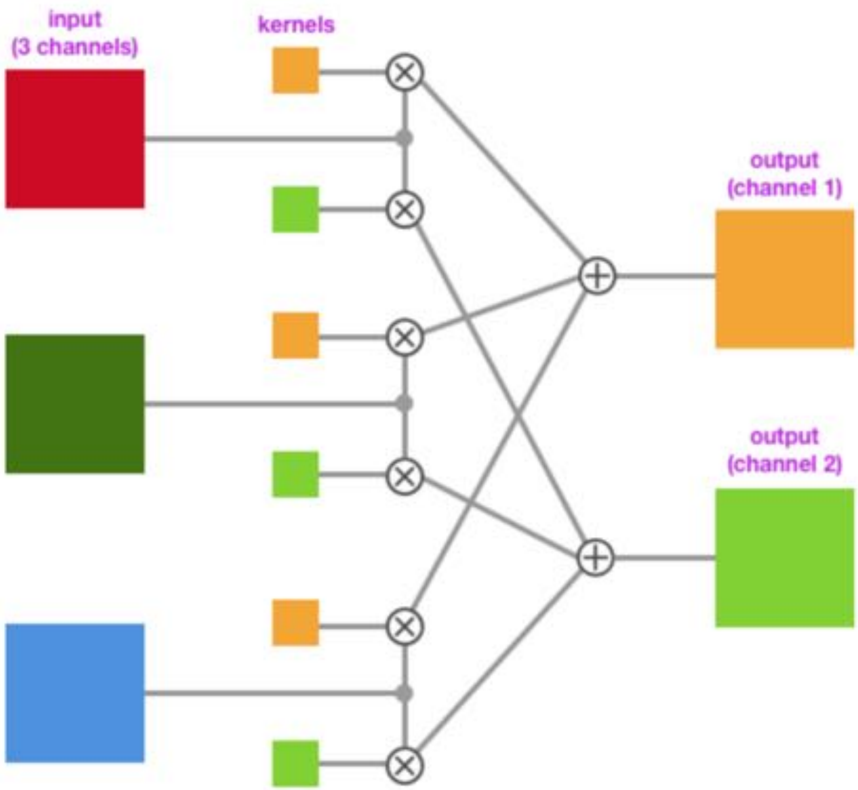


1. CONVOLUTION LAYER



1. CONVOLUTION LAYER

A detail: Its is possible to sum
the convolutions of 2 or more kernels



1) Convolution and Non-linearity Blocks First

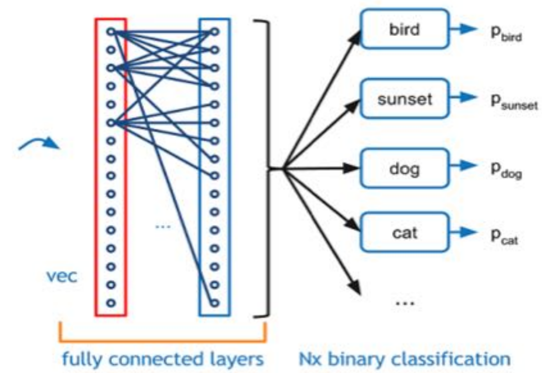
2) Then Fully Connected Layers leading to prediction

Image * → kernels → Feature Map (convolved Images) → ReLU (or other non-linearity → Max Pooling

Convolution w/ kernels

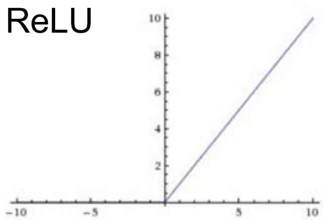
- ReLU
- Tanh
- Sigmoid

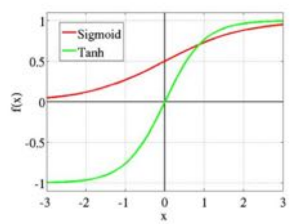
Down-sampling



Can repeat in multiple layers

ReLU



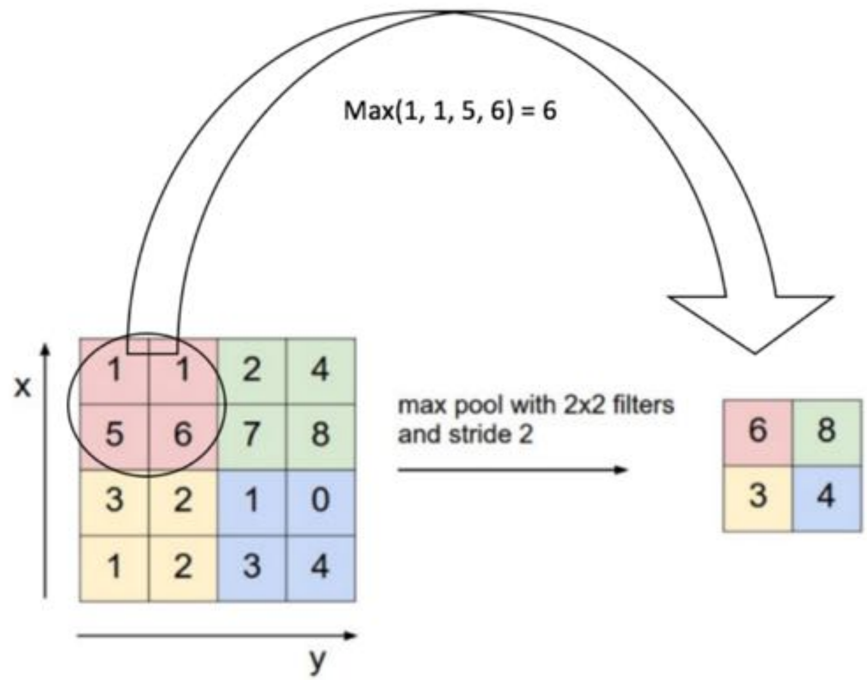


DataX

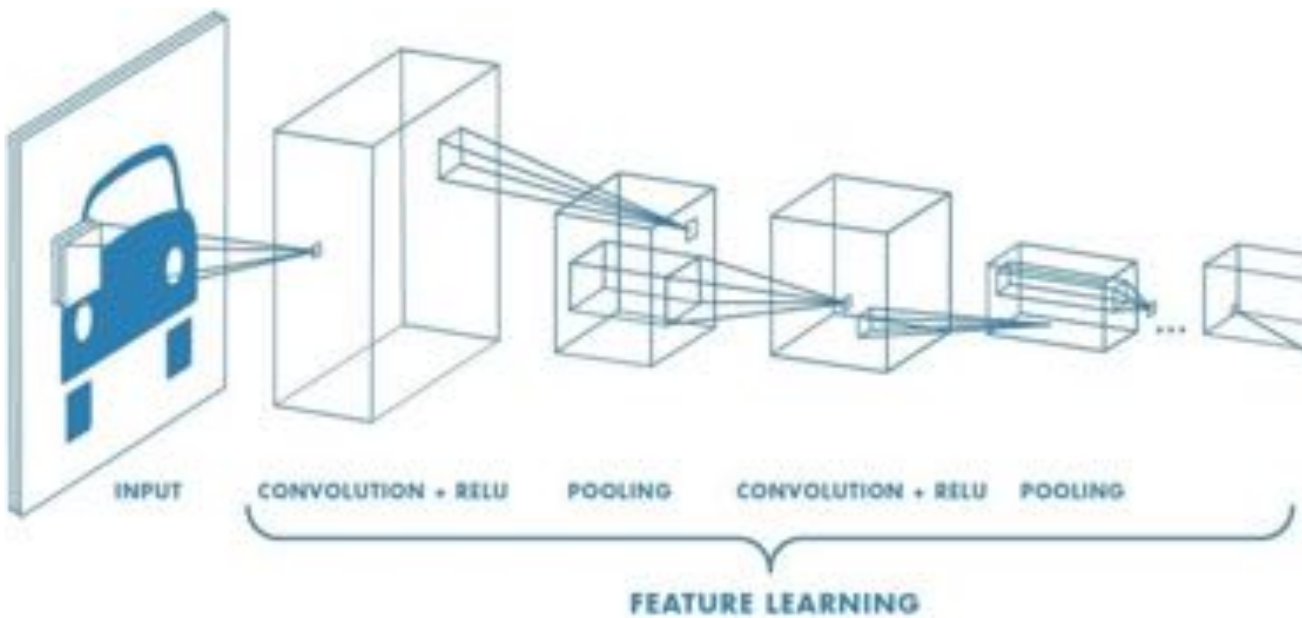
2. POOLING LAYER

1. Reduces the spatial dimensions (Width x Height) of the Input Volume - ‘down-sampling’

2. Usually max or average pooling is done.



Putting the CNN Together

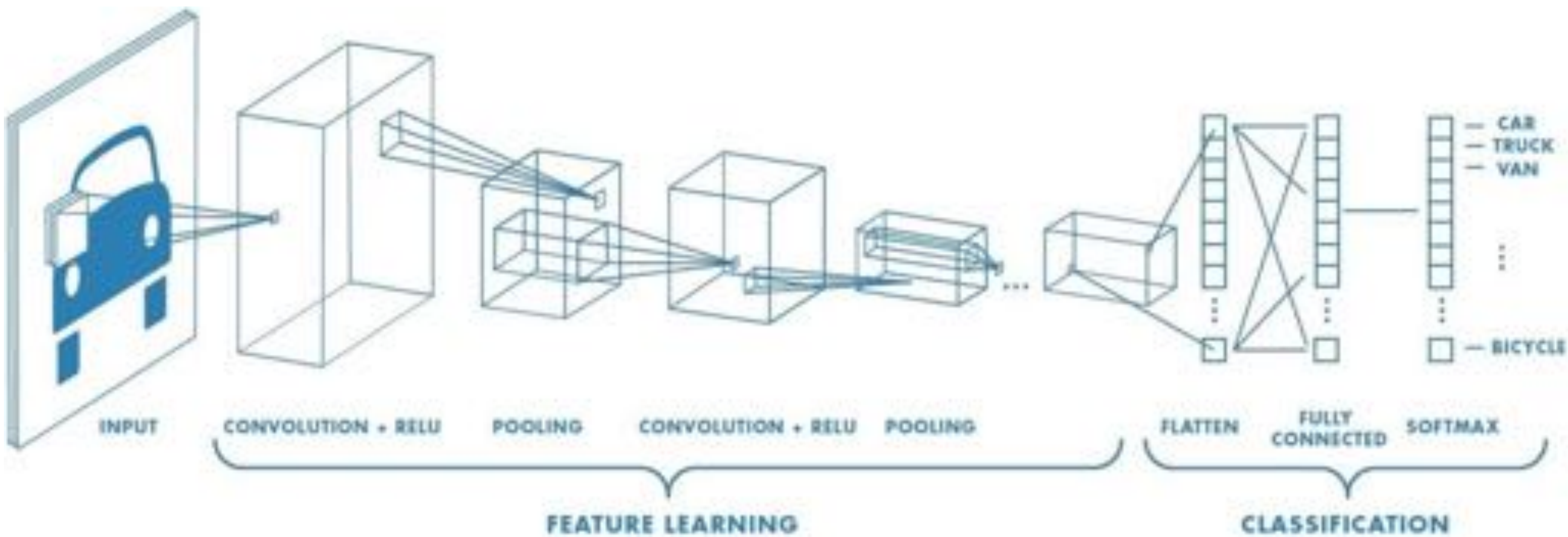


https://www.mathworks.com/discovery/convolutional-neural-network.html

Pooling:

- Reduces the spatial dimensions (Width x Height) of the Input Volume
  - 'down-sampling'
- It does not affect the depth dimension of the Volume.
- Usually placed after the Convolutional layer.
- The decrease in size leads to less computational overhead for the upcoming layers of the network

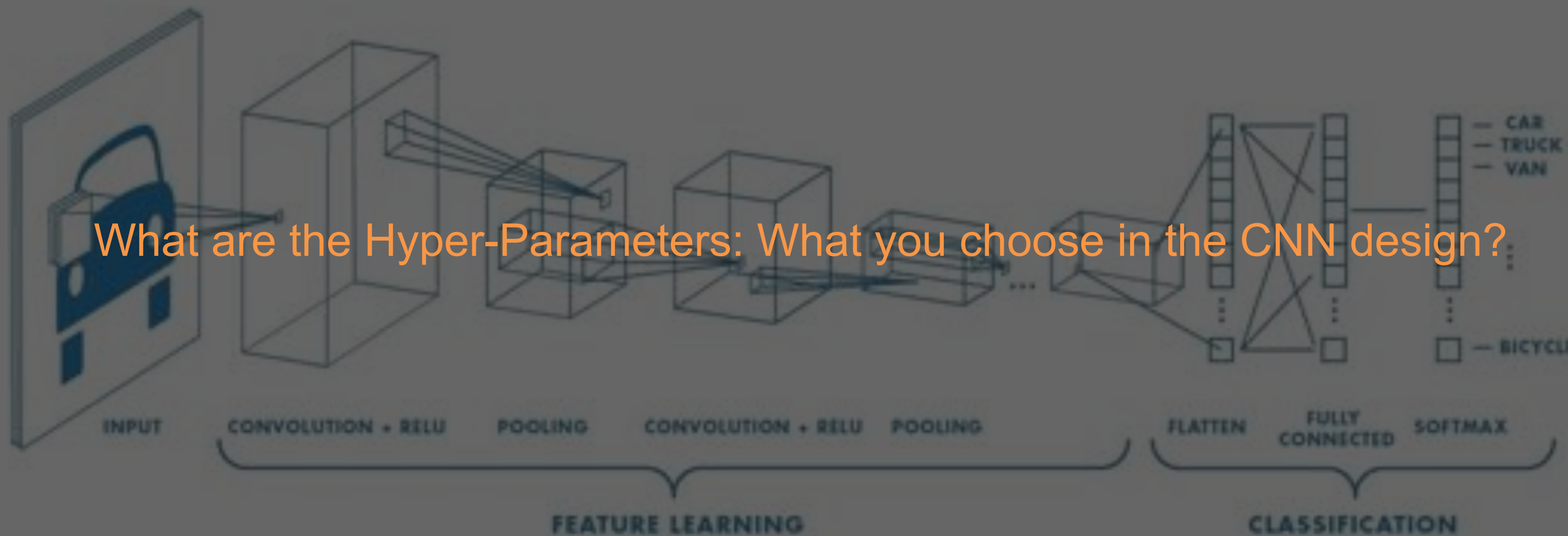
Putting the CNN Together with the Fully Connected Layer



https://www.mathworks.com/discovery/convolutional-neural-network.html

Putting the CNN Together with the Fully Connected Layer

What are the Parameters – What the Algorithm Chooses?



What are the Hyper-Parameters: What you choose in the CNN design?

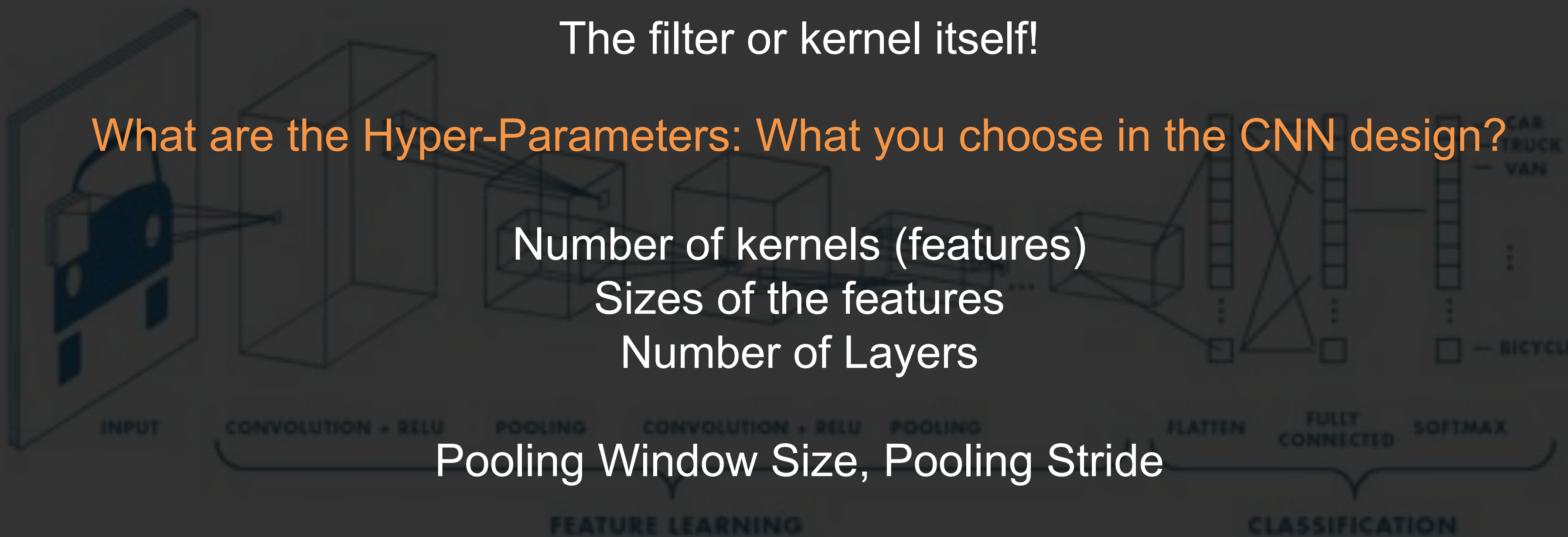
https://www.mathworks.com/discovery/convolutional-neural-network.html

Putting the CNN Together with the Fully Connected Layer

What are the Parameters – What the Algorithm Chooses?

The filter or kernel itself!

What are the Hyper-Parameters: What you choose in the CNN design?



Number of kernels (features)
Sizes of the features
Number of Layers

Pooling Window Size, Pooling Stride

Fully Connected Layer: Number of Neurons, Number of Outputs

https://www.mathworks.com/discovery/convolutional-neural-network.html

End of Section