TRAINING SESSION ON CONVOLUTIONAL NEURAL NETWORK

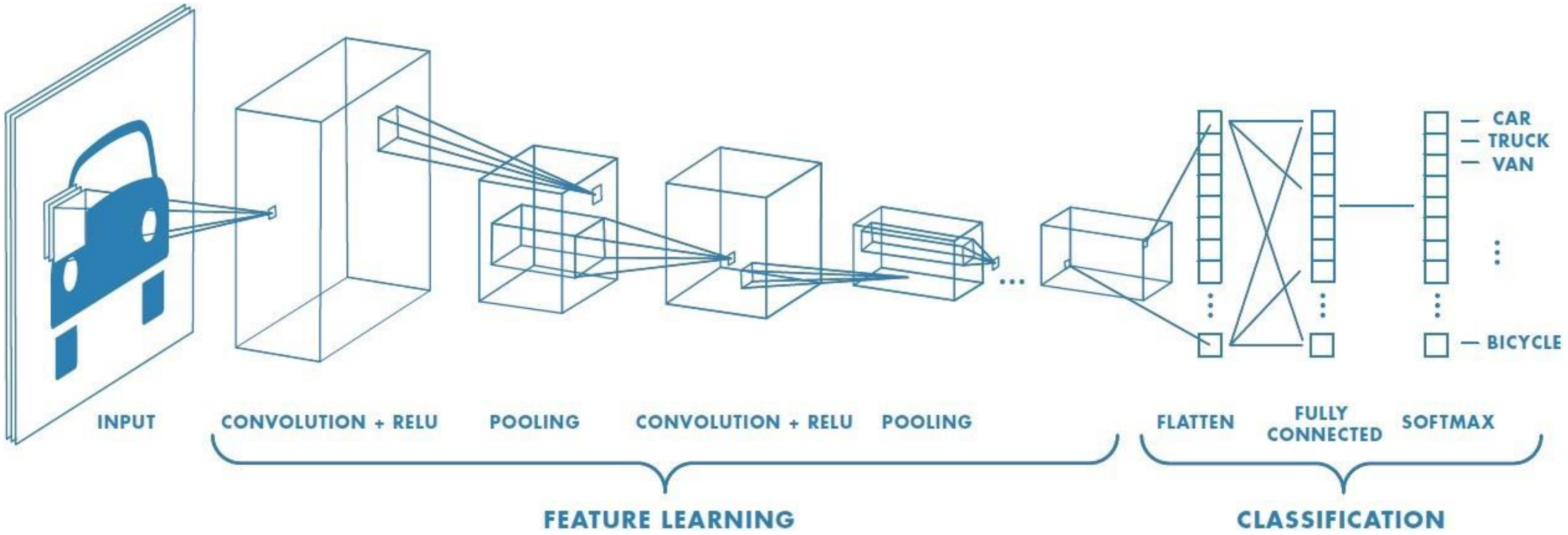
Presented by:

Dr. Bunil Kumar Balabantaray
Assistant Professor

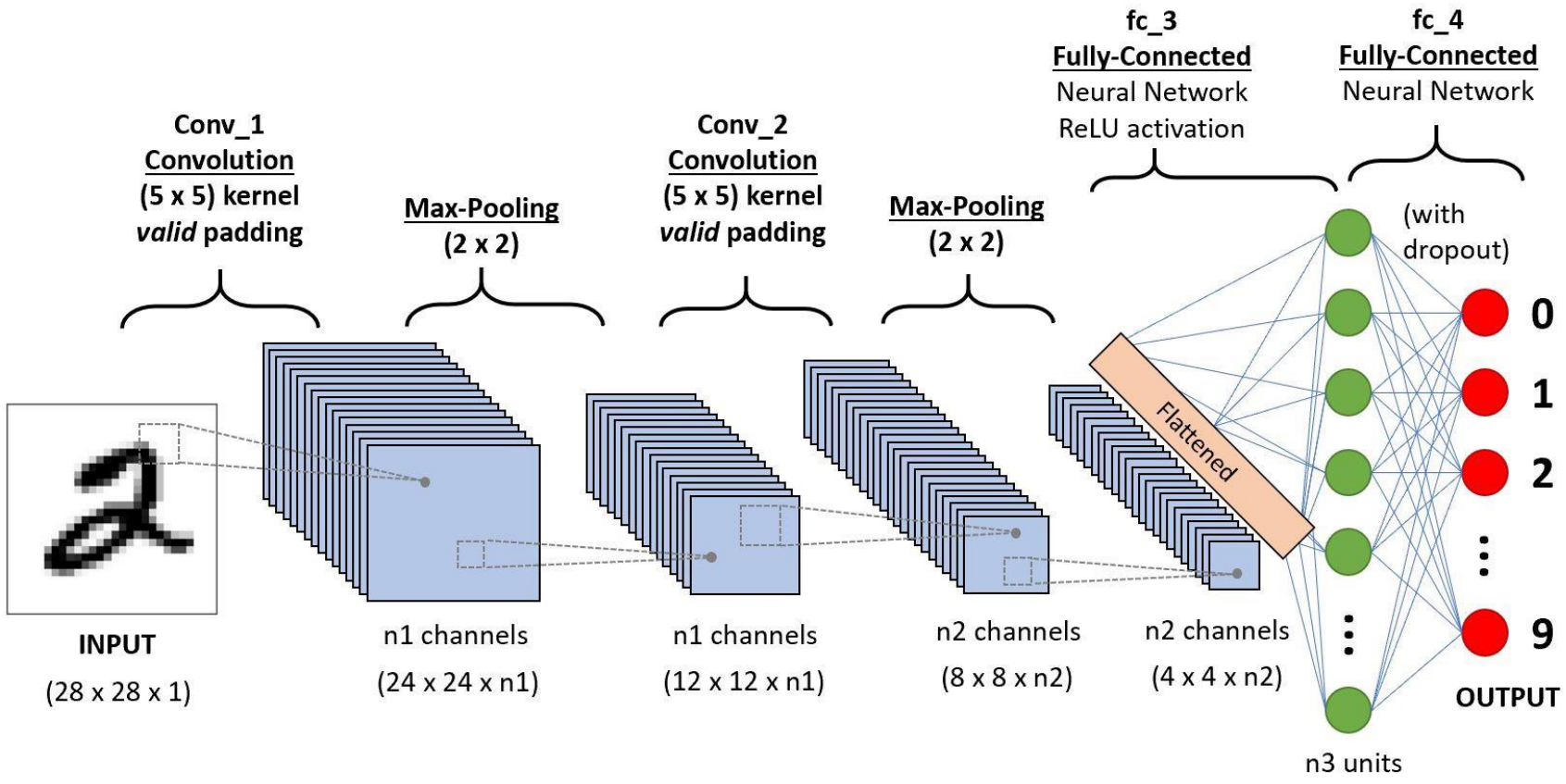
Department of Computer Science and Engineering,
NATIONAL INSTITUTE OF TECHNOLOGY MEGHALAYA

Anmol Gautam, Swastik Jena (Member of AI Research Group,NIT Meghalaya)

Introducing Convolutional Neural Networks



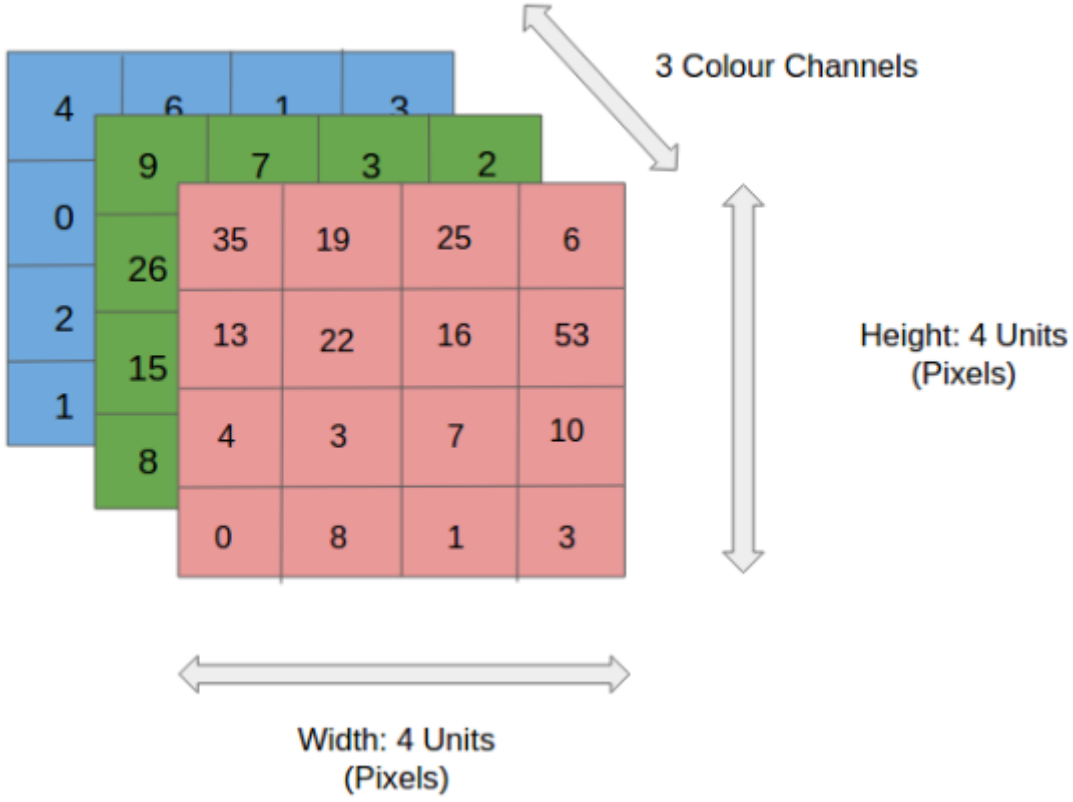
Introducing Convolutional Neural Networks



Introducing Convolutional Neural Networks

- preserve the spatial structure of the problem
- by learning internal feature representations using small squares of input data
- specifically designed to process pixel data.
- Convolutional layers learn local patterns using filters/kernels
- They are translation invariant
- They learn spatial hierarchies
- Input and output is a tensor called feature map

Image Data


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

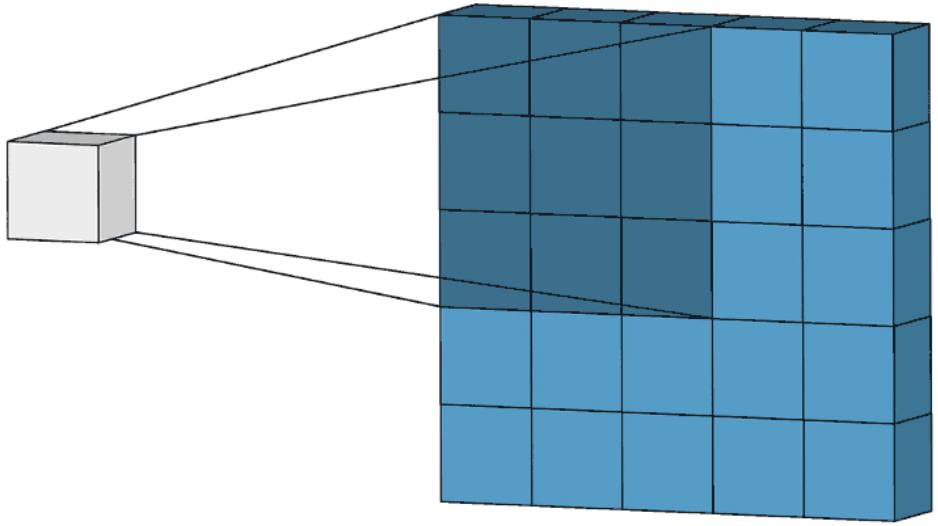
Building Blocks Of a CNN

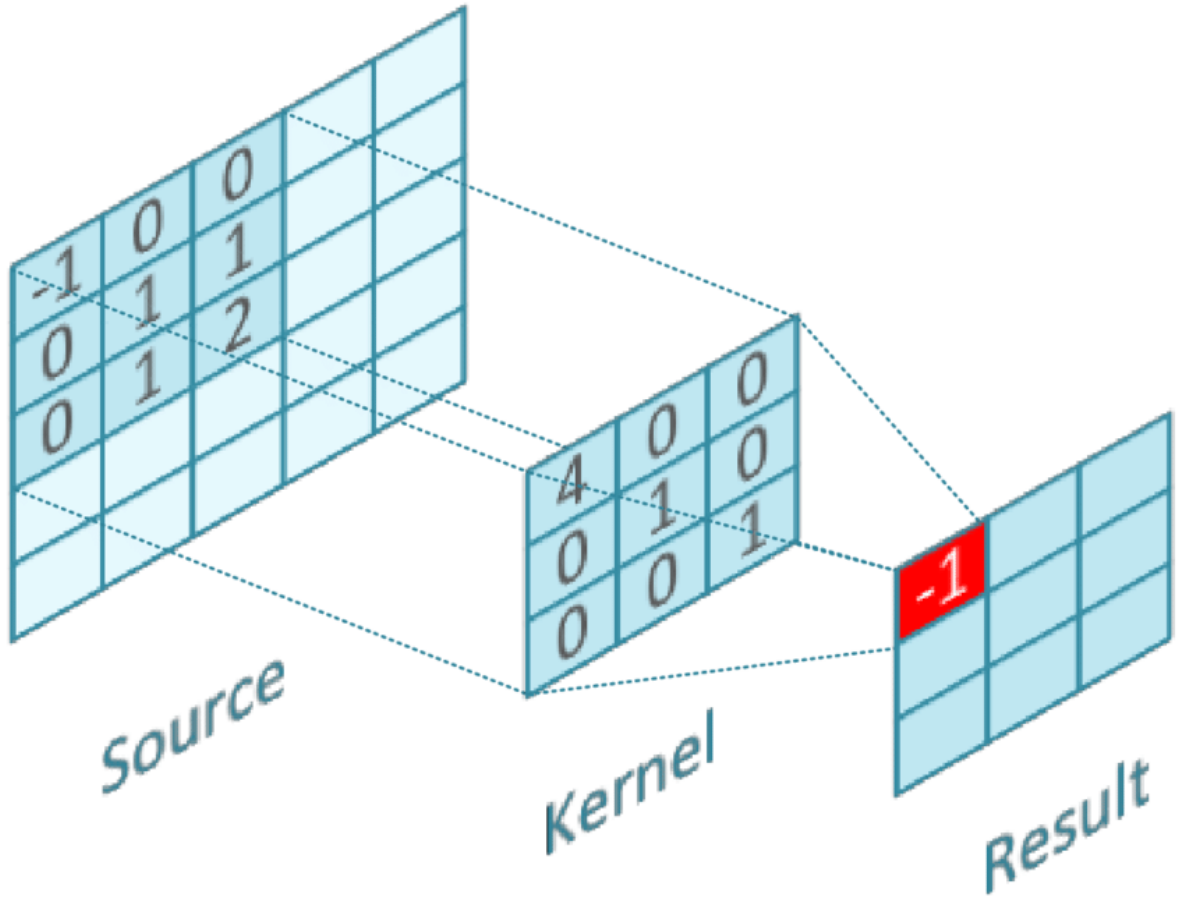
- Convolutional Layers
- Pooling Layers
- Fully-Connected Layers

Convolutional Layers

- Filters
- Feature Maps
- Padding
- Stride

Convolutional Operation

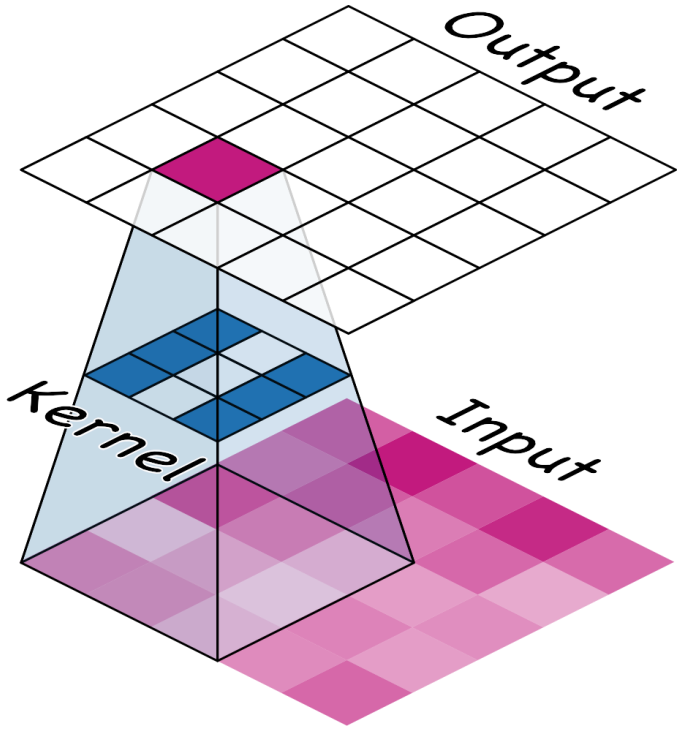




Kernels / Weights / Filters

- The weights a convnet learns during training are primarily contained in its convolutional layers. These weights we call kernels.

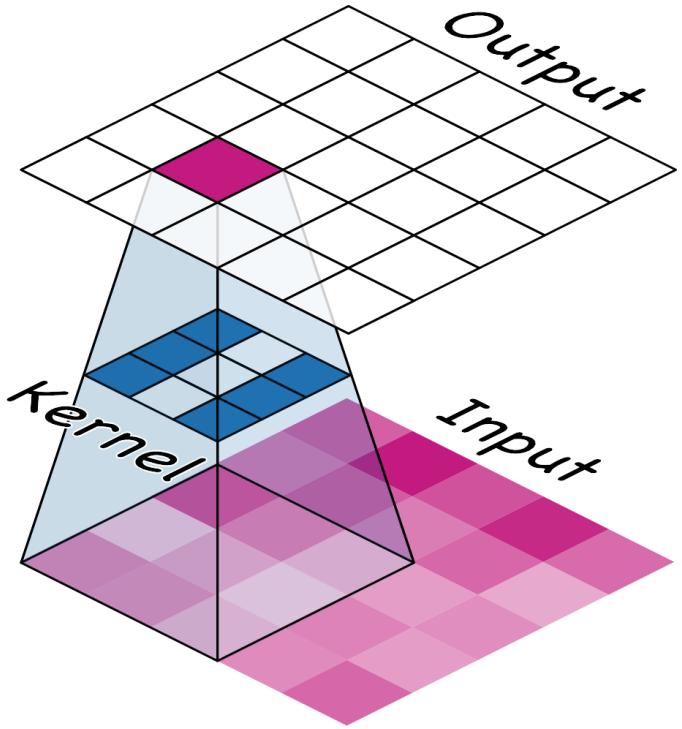
-1	2	-1
-1	2	-1
-1	2	-1



Kernels / Weights / Filters

- The weights a convnet learns during training are primarily contained in its convolutional layers. These weights we call kernels.

-1	2	-1
-1	2	-1
-1	2	-1



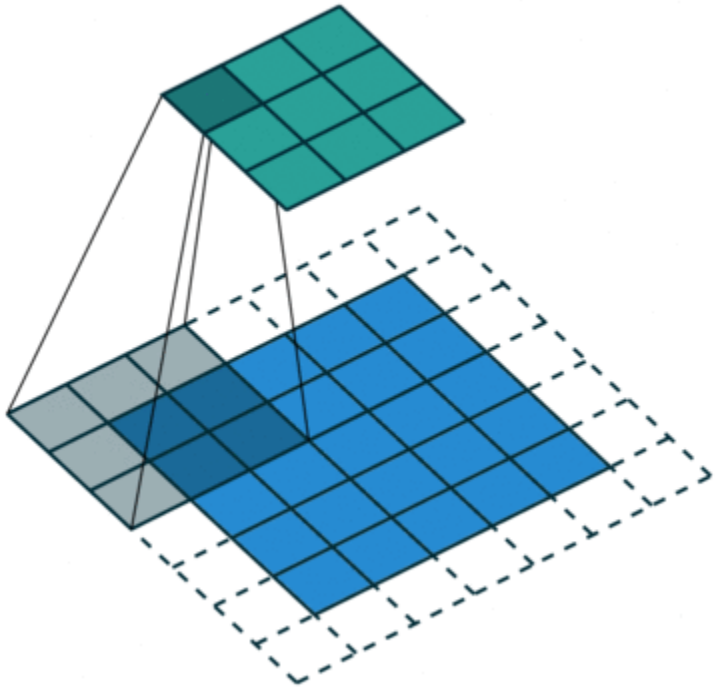
Kernels / Weights / Filters

-1	-1
1	1

-1	-2	-1
0	0	0
1	2	1

-2	-1	0
-1	1	1
0	1	2

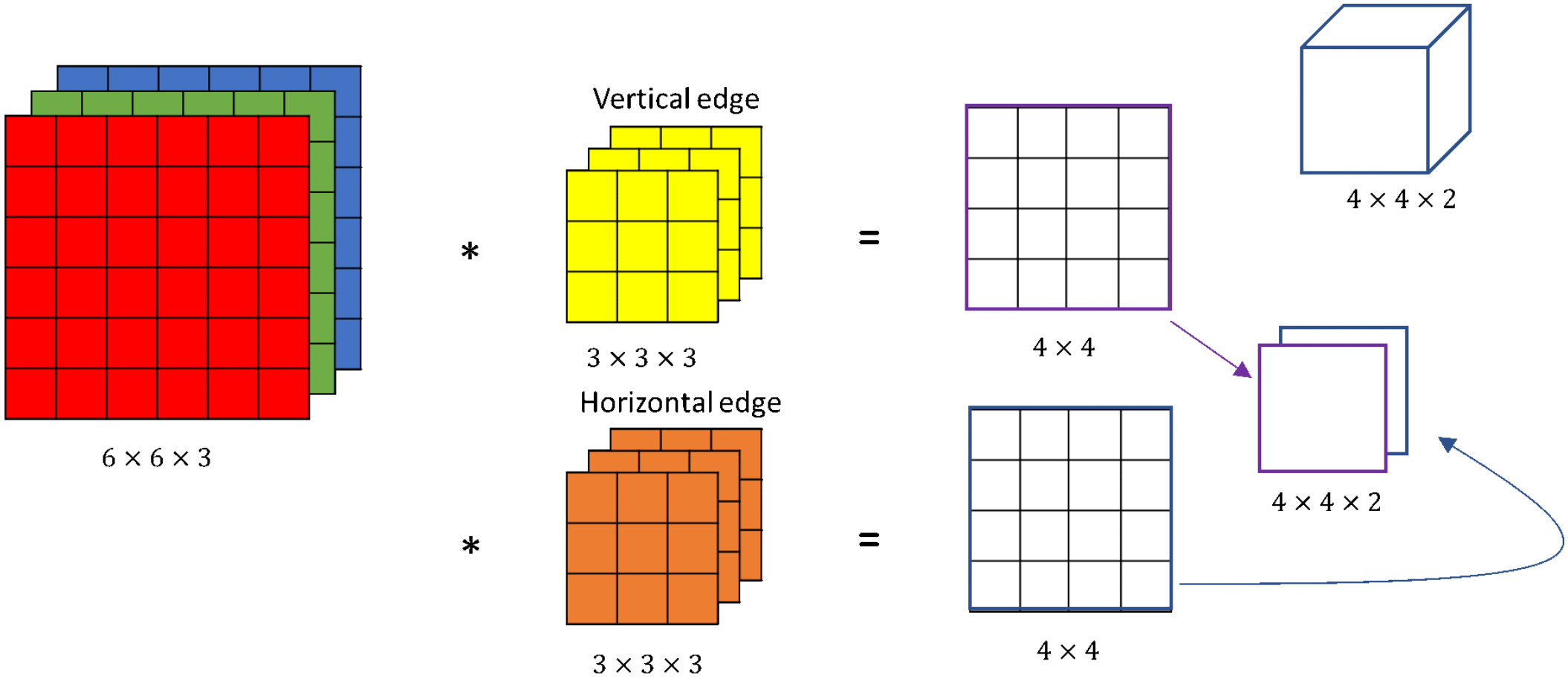
Padding



Stride

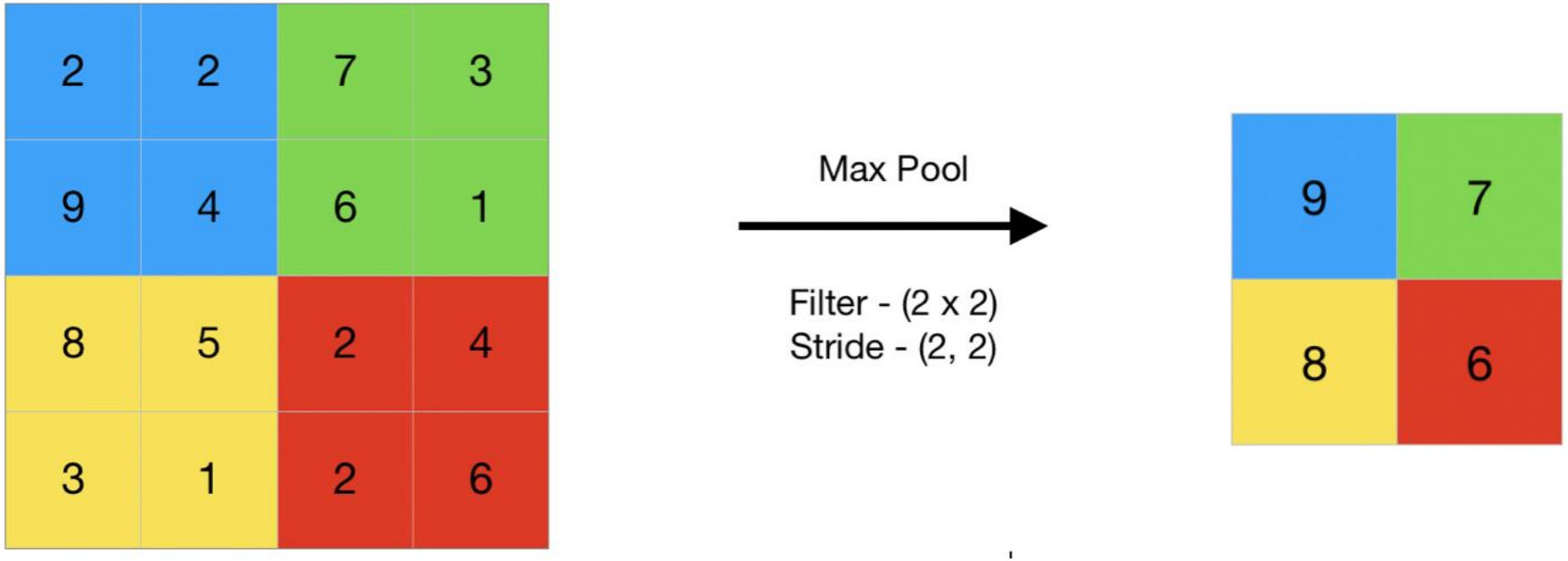
- Stride is a component of convolutional neural networks tuned for the compression of images.
- Stride is a parameter of the filter that modifies the amount of movement over the image.

Convolution on RGB images



Pooling Layers

- Average Pooling
- Max Pooling
- Etc.



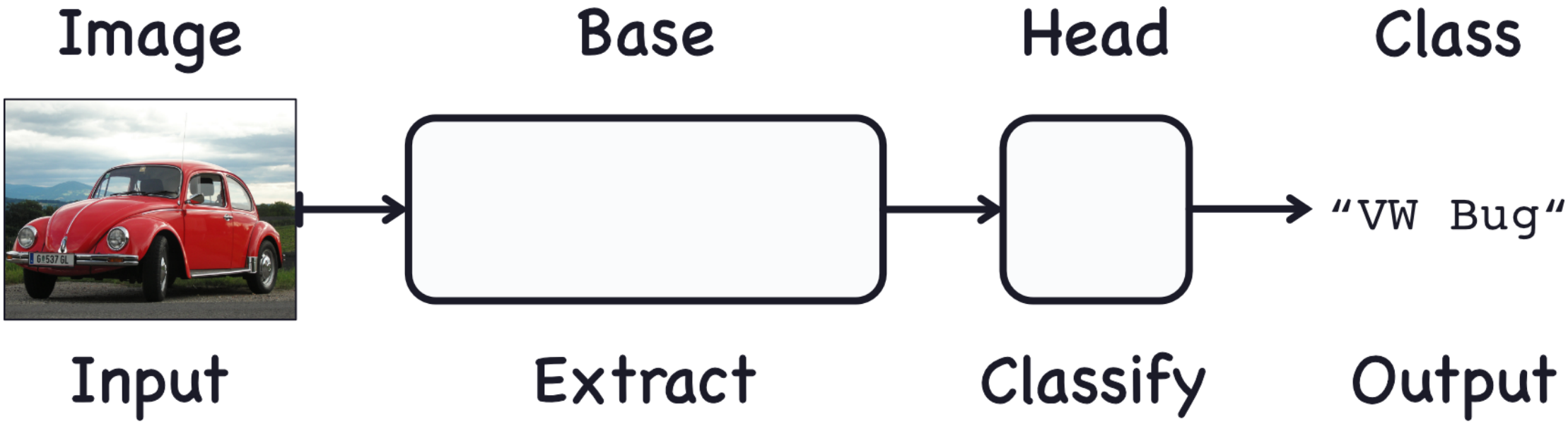
Fully Connected Layers

- Feed forward layers
- Non linearity
- After feature extraction used to create non linear combinations for predictions

Fully Connected Layers

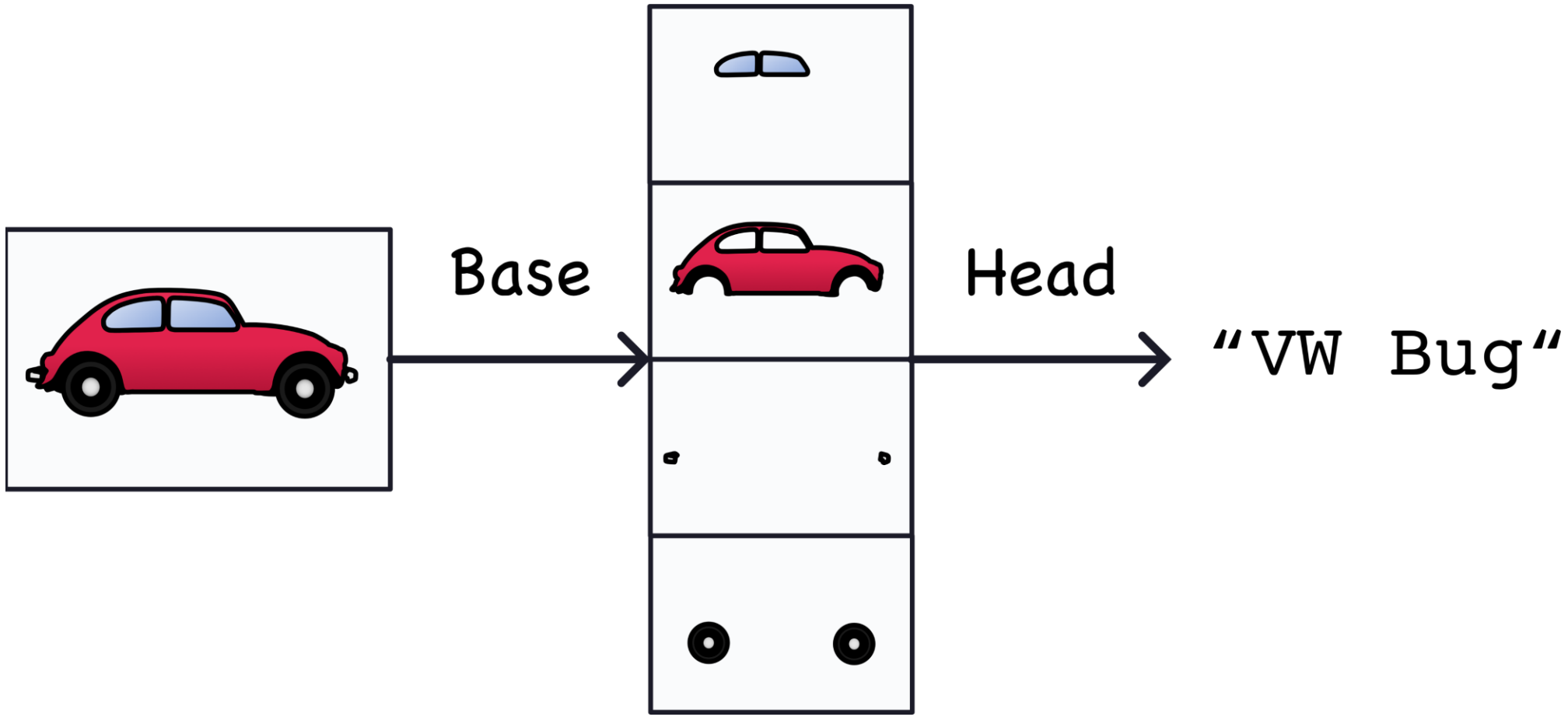
- Feed forward layers
- Non linearity
- After feature extraction used to create non linear combinations for predictions

Convolutional Classifier



photograph by Dnalor 1 (Wikimedia Commons) CC-BY-SA 3.0

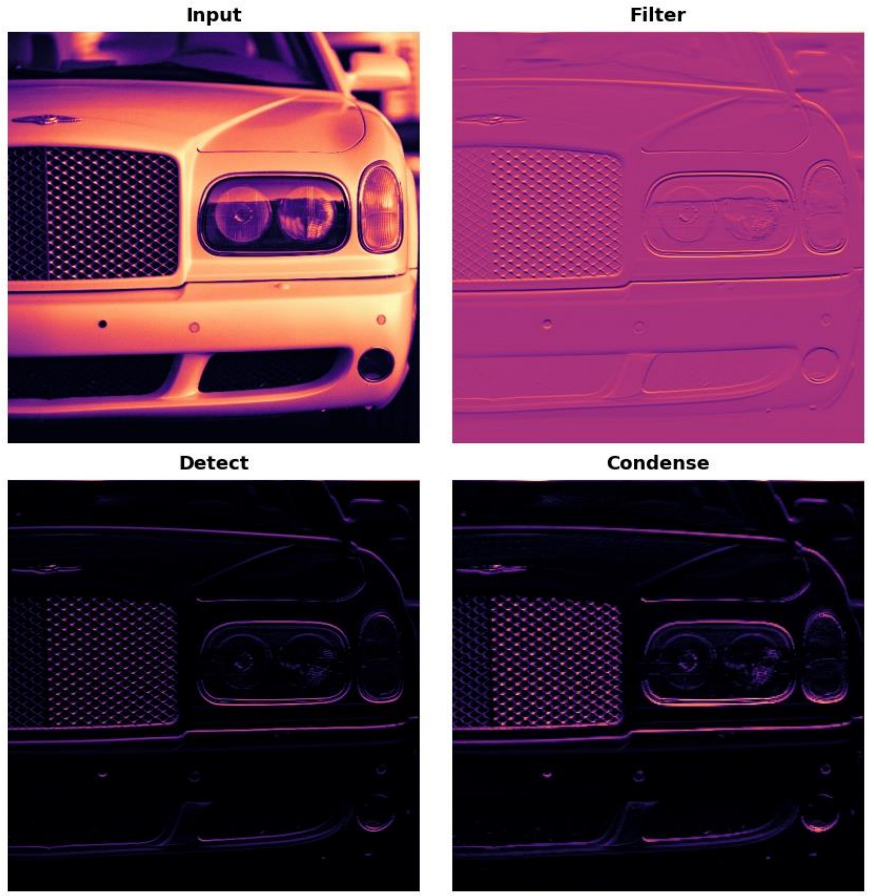
Convolutional Classifier



Feature Extraction

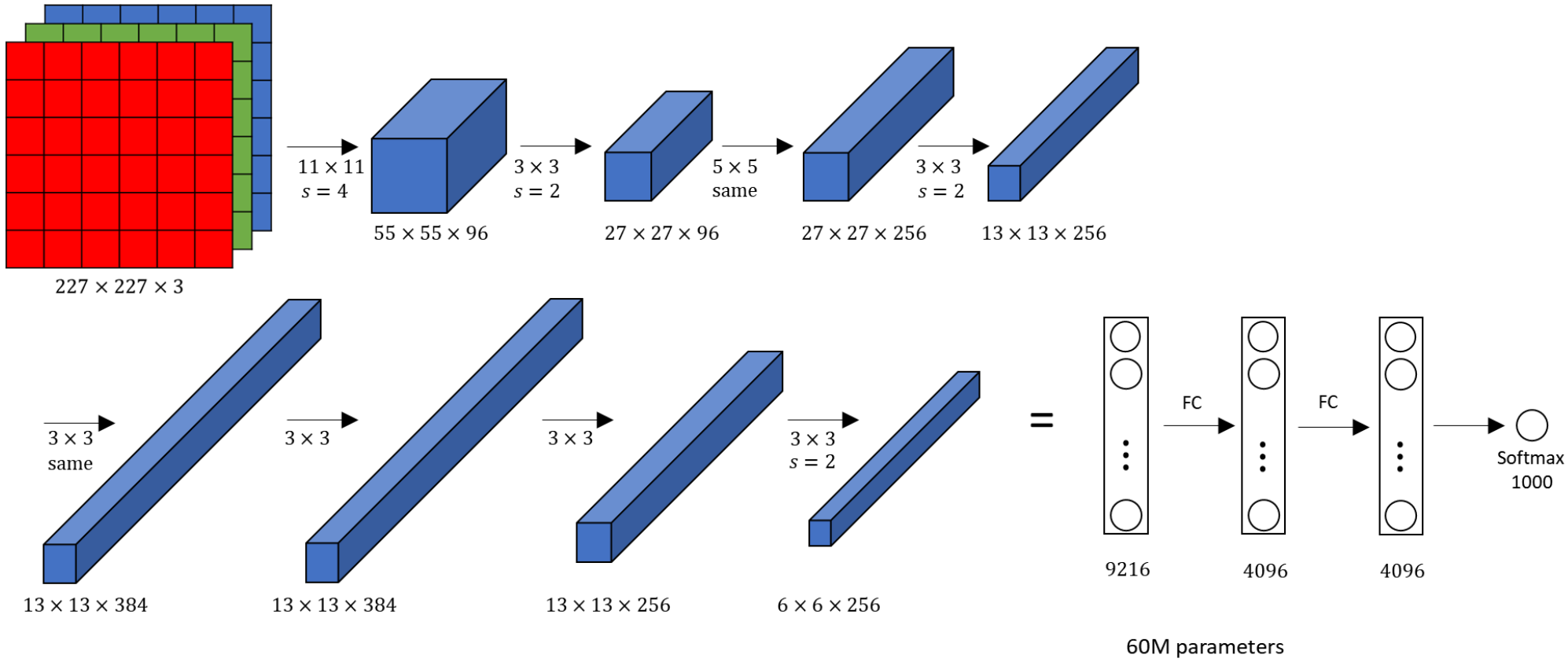
- The feature extraction performed by the base consists of three basic operations:
  - Filter an image for a particular feature (convolution)
  - Detect that feature within the filtered image (ReLU)
  - Condense the image to enhance the features (maximum pooling)

Feature Extraction



One Layer Of CNN

Alexnet


227 × 227 × 3
11 × 11
s = 4
55 × 55 × 96
3 × 3
s = 2
27 × 27 × 96
5 × 5
same
27 × 27 × 256
3 × 3
s = 2
13 × 13 × 256
3 × 3
same
13 × 13 × 384
3 × 3
13 × 13 × 384
3 × 3
13 × 13 × 256
3 × 3
s = 2
6 × 6 × 256
=
9216
FC
4096
FC
4096
Softmax
1000
60M parameters

Transfer Learning

- Instead of building a convnet from scratch if we can reuse the learned model.
- E.g. alexnet, VGG16, etc.
- they were trained on different dataset (IMAGENET)
- Lets explore this

Case 1

- Extract Features from the input
- Use just before we reach flatten layer

Case 2

- Freeze initial few layers
- Only allow training on last few layers on new dataset by modifying them or using another classifier all together

Case 3

- Tune the complete model
- Keep learning rate very small

Thank You