SC310005 Artificial Intelligence

Lecture 10: Deep Learning (Part 1)

teerapong.pa@chula.ac.th

Reference:

1. https://pytorch.org/tutorials/beginner/basics/quickstart_tutorial.html
2. http://introtodeeplearning.com/
3. https://www.simplilearn.com/tutorials/deep-learning-tutorial/introduction-to-deep-learning
4. https://www.geeksforgeeks.org/introduction-deep-learning/
5. https://stanford.edu/~shervine/teaching/cs-230/cheatsheet-convolutional-neural-networks

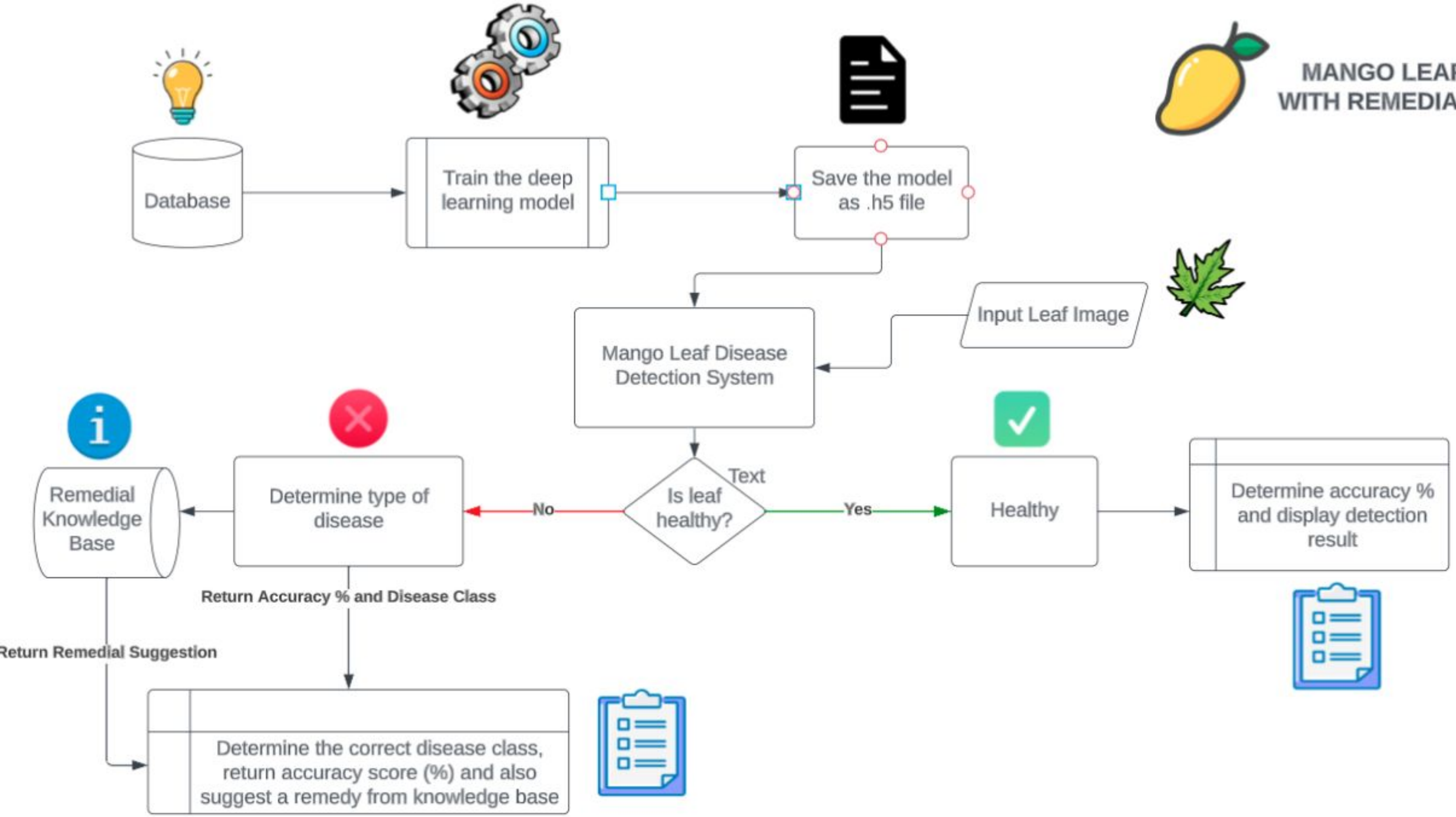
Artificial Intelligence Projects

มหาวิทยาลัยขอนแก่น
KHON KAEN UNIVERSITY

TensorFlow

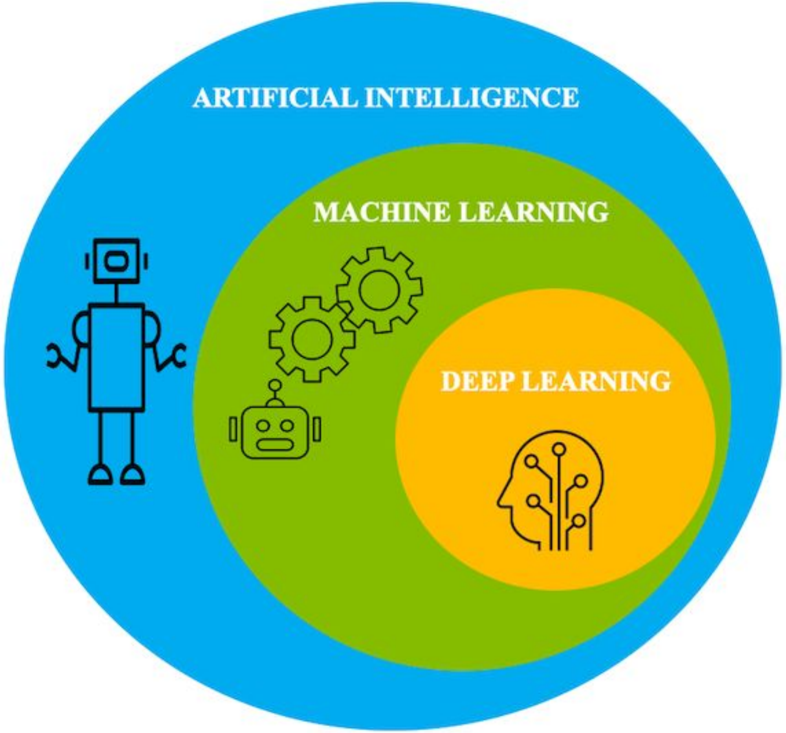
Mango🥭 Leaf🍃🍂 Disease Dataset

MANGO LEAF DISEASE DETECTION WITH REMEDIAL SUGGESTION SYSTEM


Database
Train the deep learning model
Save the model as .h5 file
Input Leaf Image
Mango Leaf Disease Detection System
Is leaf healthy?
Text
No
Yes
Healthy
Determine accuracy % and display detection result
Determine type of disease
Remedial Knowledge Base
Return Accuracy % and Disease Class
Return Remedial Suggestion
Determine the correct disease class, return accuracy score (%) and also suggest a remedy from knowledge base

AI vs Machine Learning vs Deep Learning

- **Artificial Intelligence (AI)** is the concept of creating smart intelligent machines.
- **Machine Learning (ML)** is a subset of artificial intelligence that helps you build AI-driven applications.
- **Deep Learning** is a subset of machine learning that uses vast volumes of data and complex algorithms to train a model.



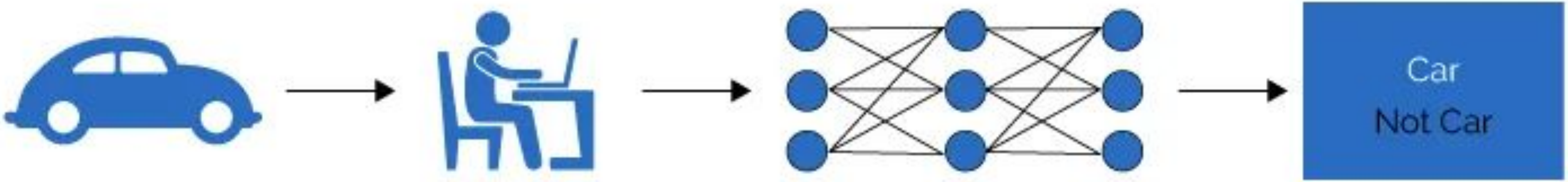
Applications of Artificial Intelligence

- Machine Translation such as Google Translate
- Self Driving Vehicles such as Google's Waymo
- AI Robots such as Sophia and Aibo
- Speech Recognition applications like Apple's Siri or OK Google

https://www.youtube.com/watch?v=kJoAcEI2PXQ&t=14s

Sophia is a humanoid robot with unique features developed by Hanson Robotics Limited

Machine Learning



Input

Feature extraction

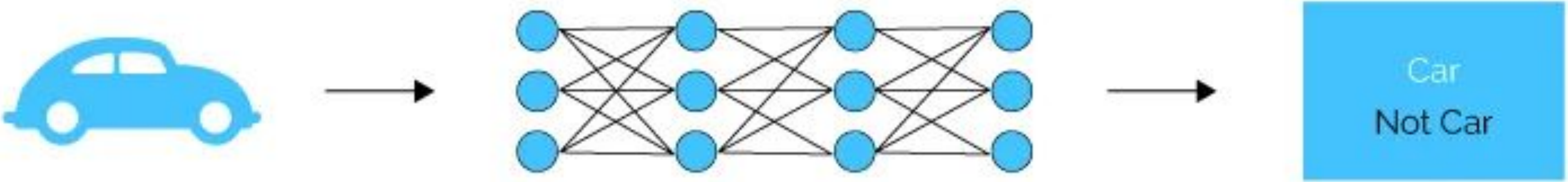
Classification

Car

Not Car

Output

Deep Learning



Input

Feature extraction + Classification

Car

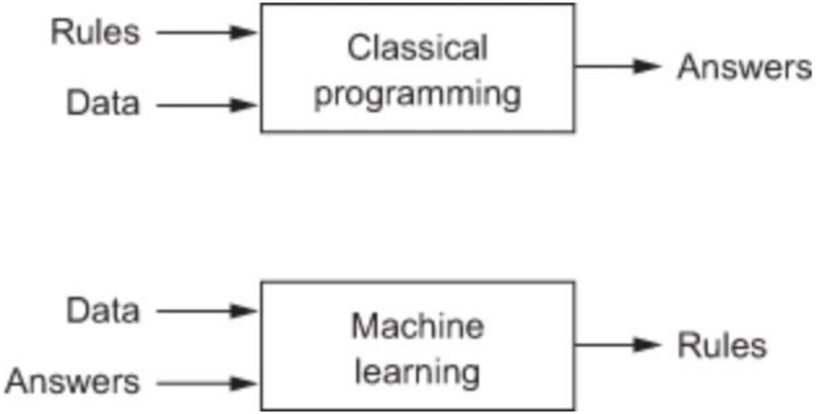
Not Car

Output

What is Deep Learning?

Deep Learning is a subset of Machine Learning that uses mathematical functions to map the input to the output.

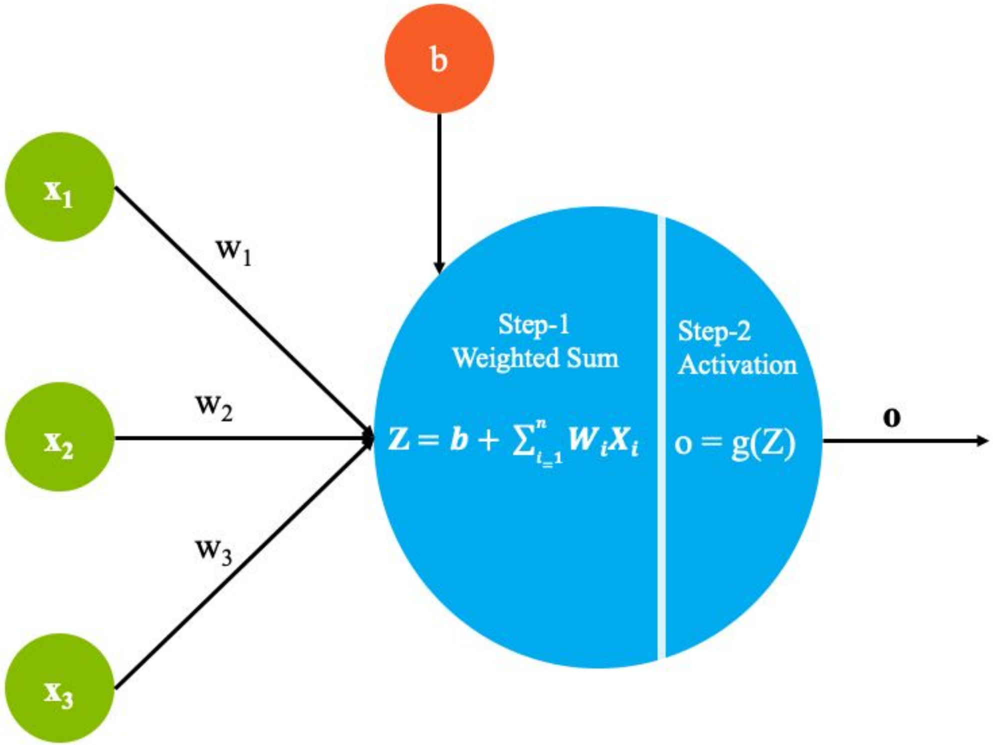
These functions can extract non-redundant information or patterns from the data, which enables them to form a relationship between the input and the output.



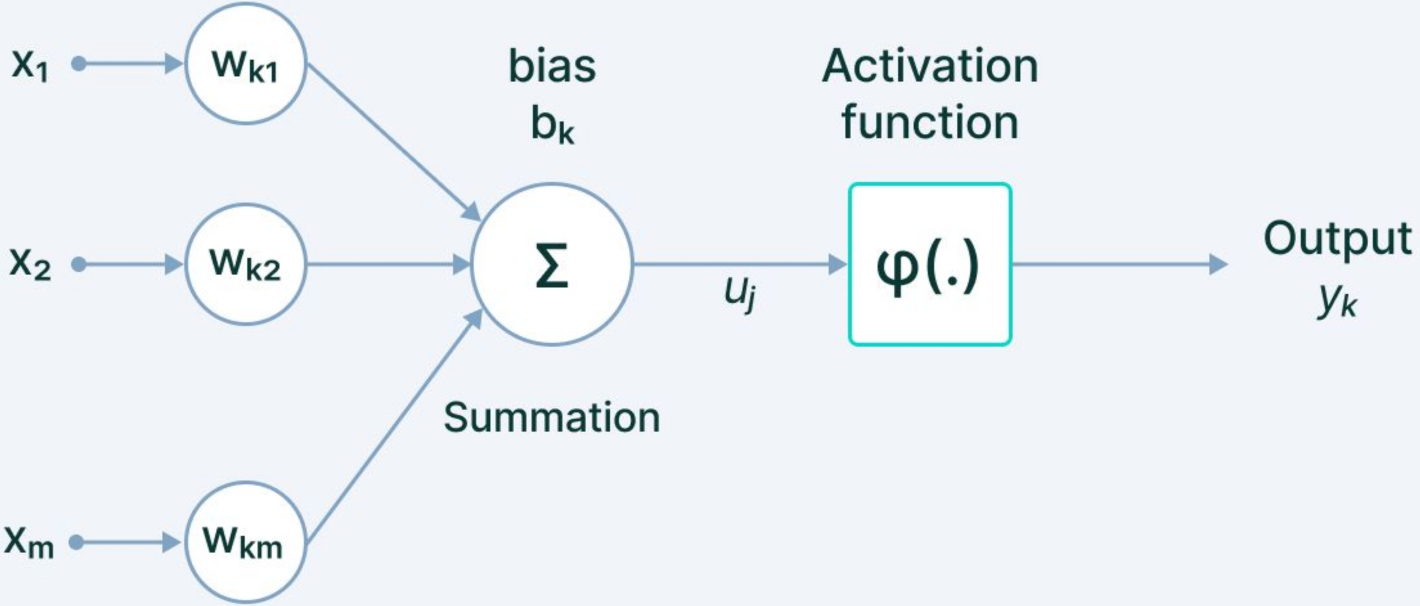
Artificial Neural Network (ANN)

This section provides an overview of the architecture behind deep learning, artificial neural networks (ANN), and discusses some of the key terminology.

As shown in the following figure, each perceptron is made up of the following parts:

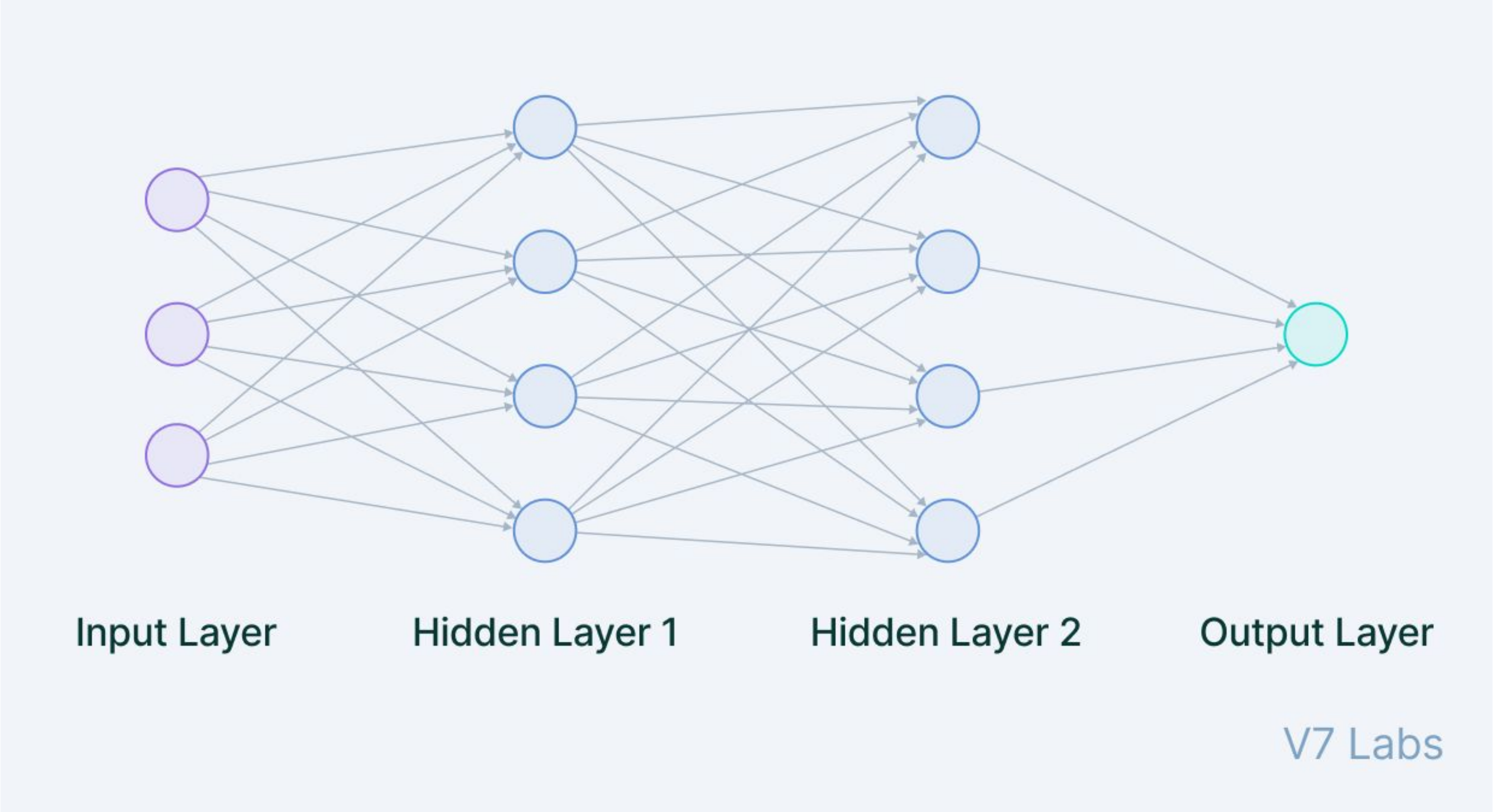


Neuron



V7 Labs

How does Deep Learning work?

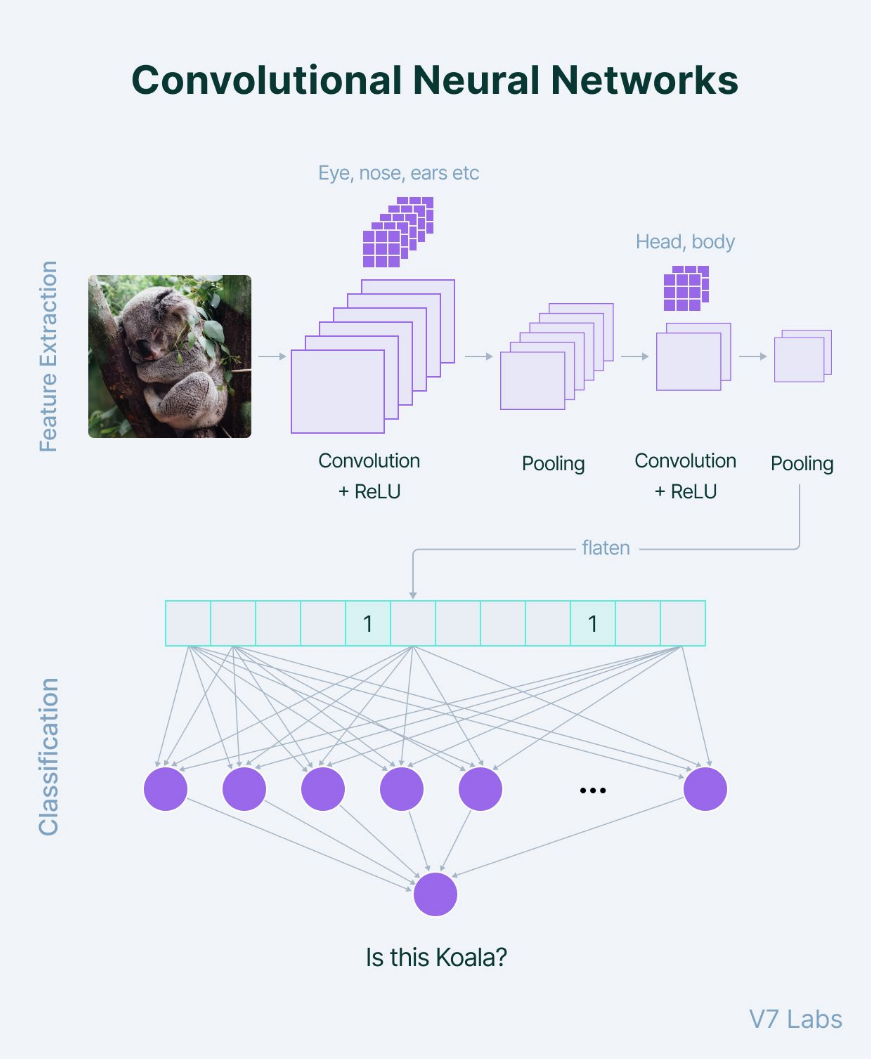


The Convolutional Neural Networks or CNNs

The Convolutional Neural Networks or CNNs are primarily used for tasks related to computer vision or image processing.

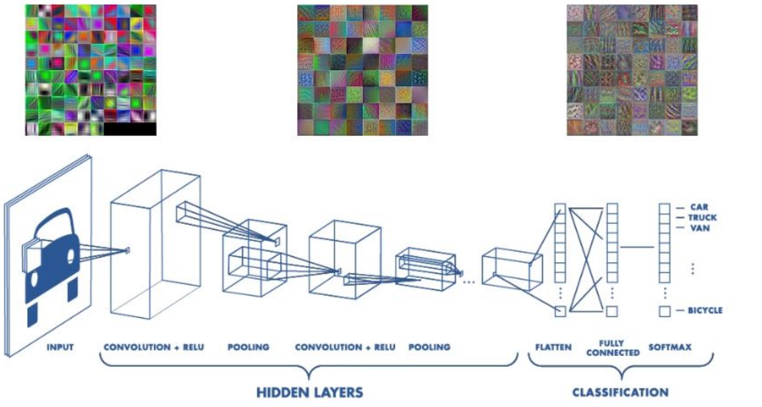
CNNs are extremely good in modeling spatial data such as 2D or 3D images and videos.

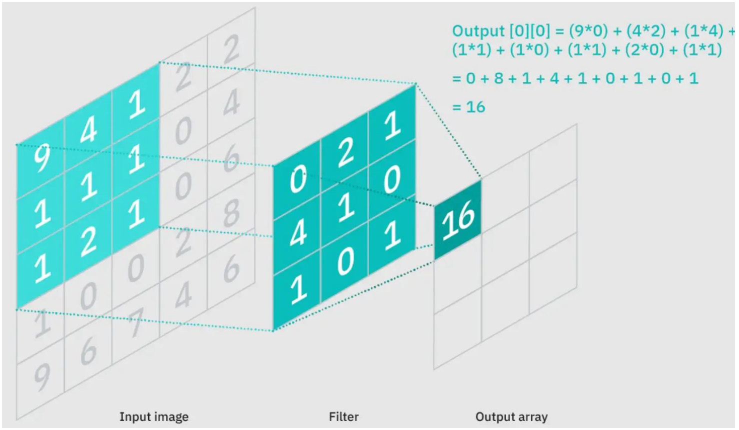
They can extract features and patterns within an image, enabling tasks such as image classification or object detection.



Convolutional Neural Networks

Convolutional Neural Network

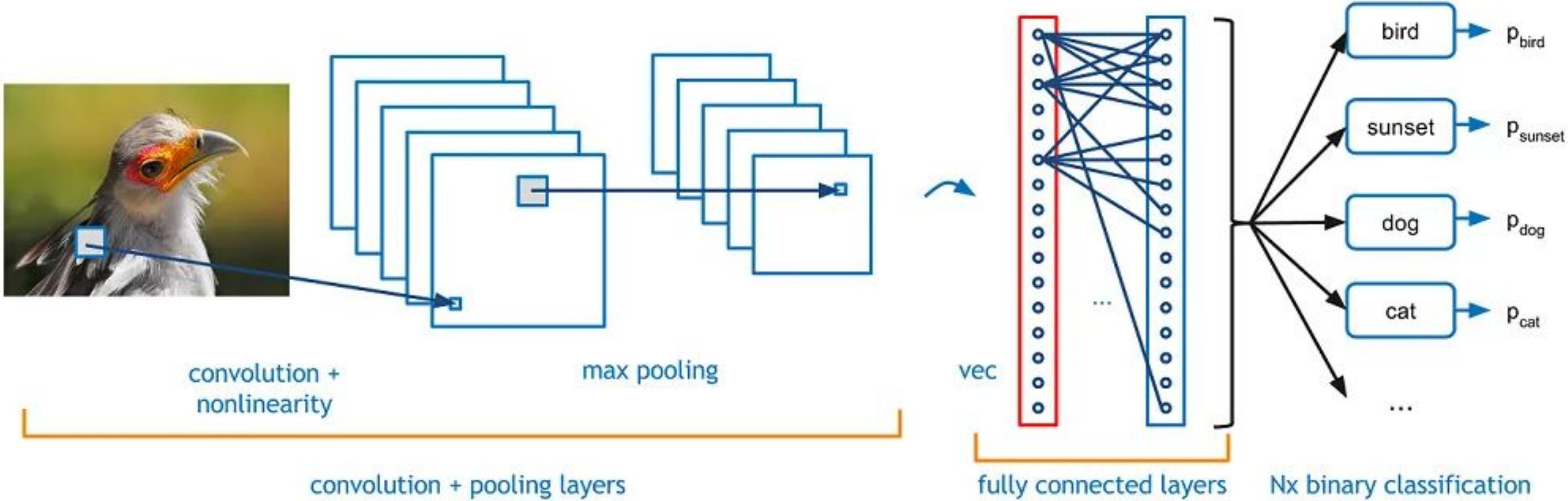




Output [0][0] = (9*0) + (4*2) + (1*4) + (1*1) + (1*0) + (1*1) + (2*0) + (1*1)

= 0 + 8 + 1 + 4 + 1 + 0 + 1 + 0 + 1

= 16



convolution + nonlinearity

max pooling

vec

bird

sunset

dog

cat

...

p_{bird}

p_{sunset}

p_{dog}

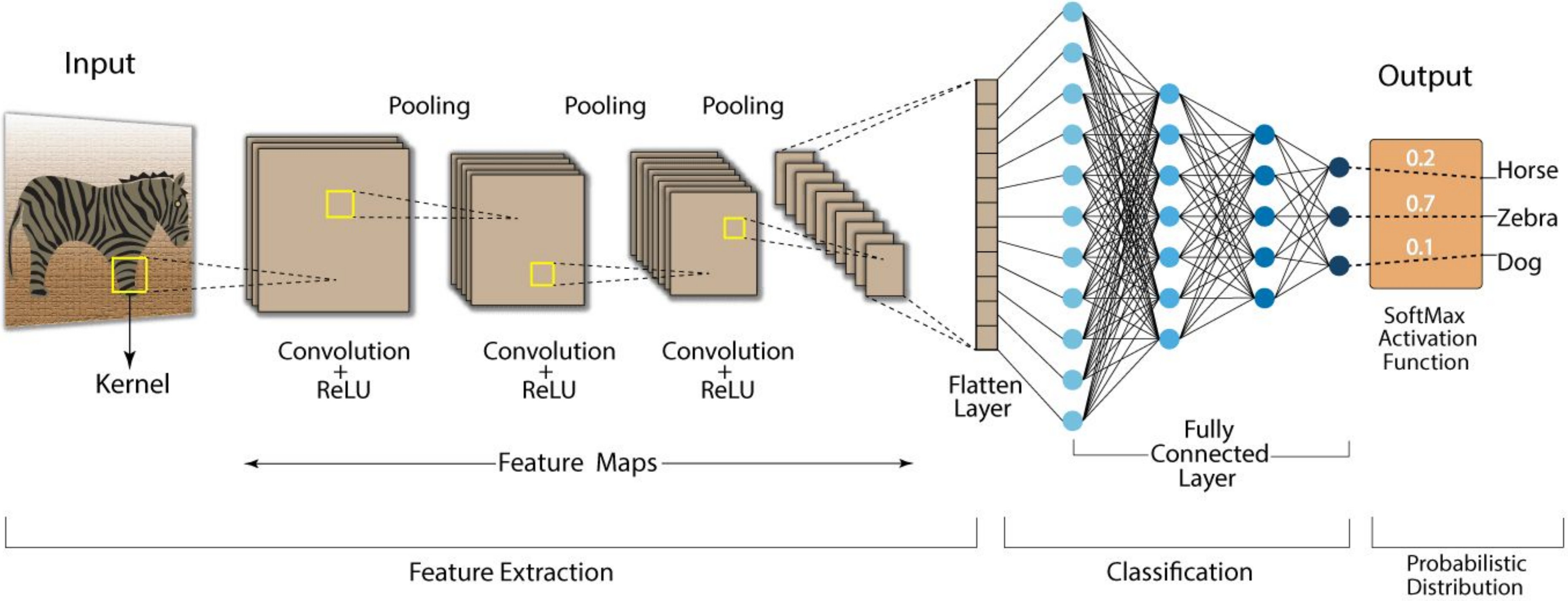
p_{cat}

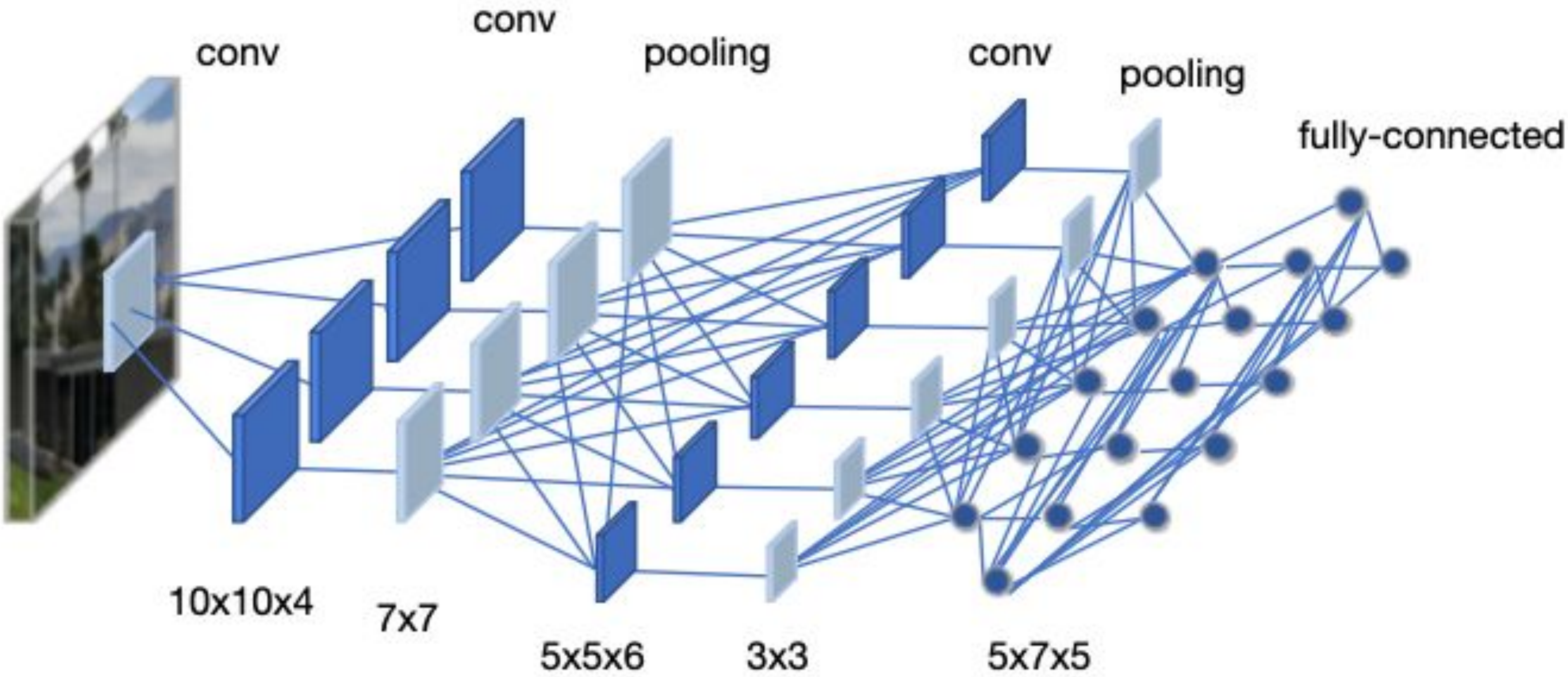
convolution + pooling layers

fully connected layers

Nx binary classification

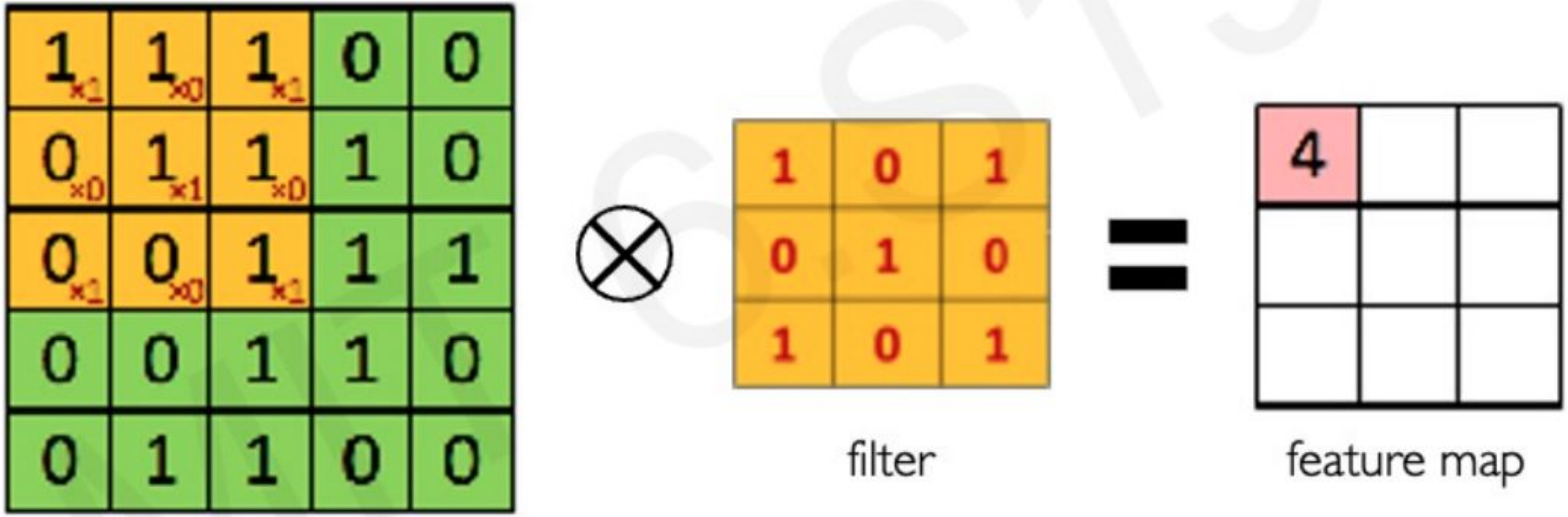
Convolution Neural Network (CNN)




conv
conv
pooling
conv
pooling
fully-connected
10x10x4
7x7
5x5x6
3x3
5x7x5

The Convolution Operation

We slide the 3x3 filter over the input image, element-wise multiply, and add the outputs:



$1_{\times 1}$	$1_{\times 0}$	$1_{\times 1}$	0	0
$0_{\times 0}$	$1_{\times 1}$	$1_{\times 0}$	1	0
$0_{\times 1}$	$0_{\times 0}$	$1_{\times 1}$	1	1
0	0	1	1	0
0	1	1	0	0

$\otimes$

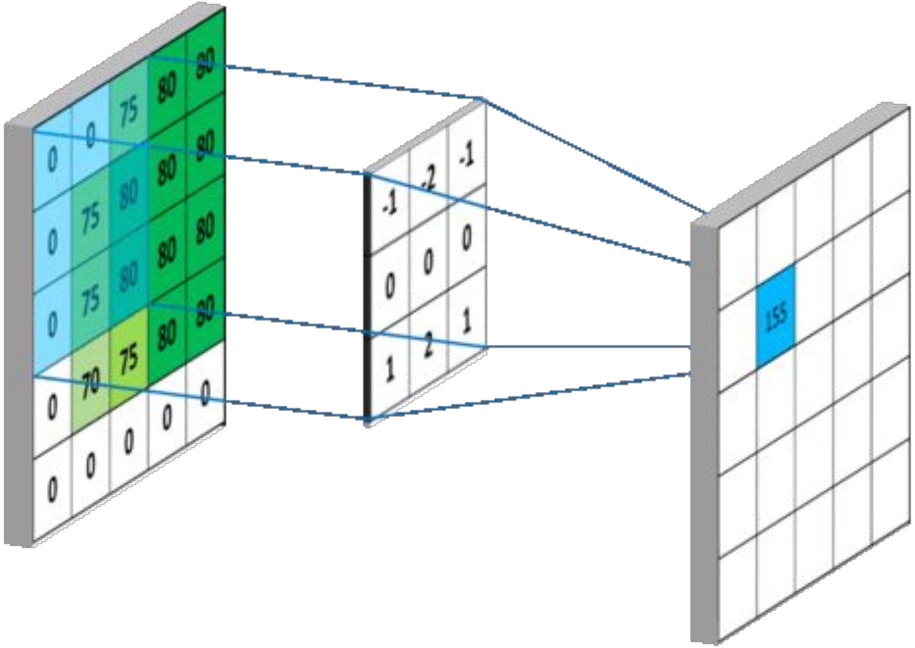
1	0	1
0	1	0
1	0	1

filter

=

4		

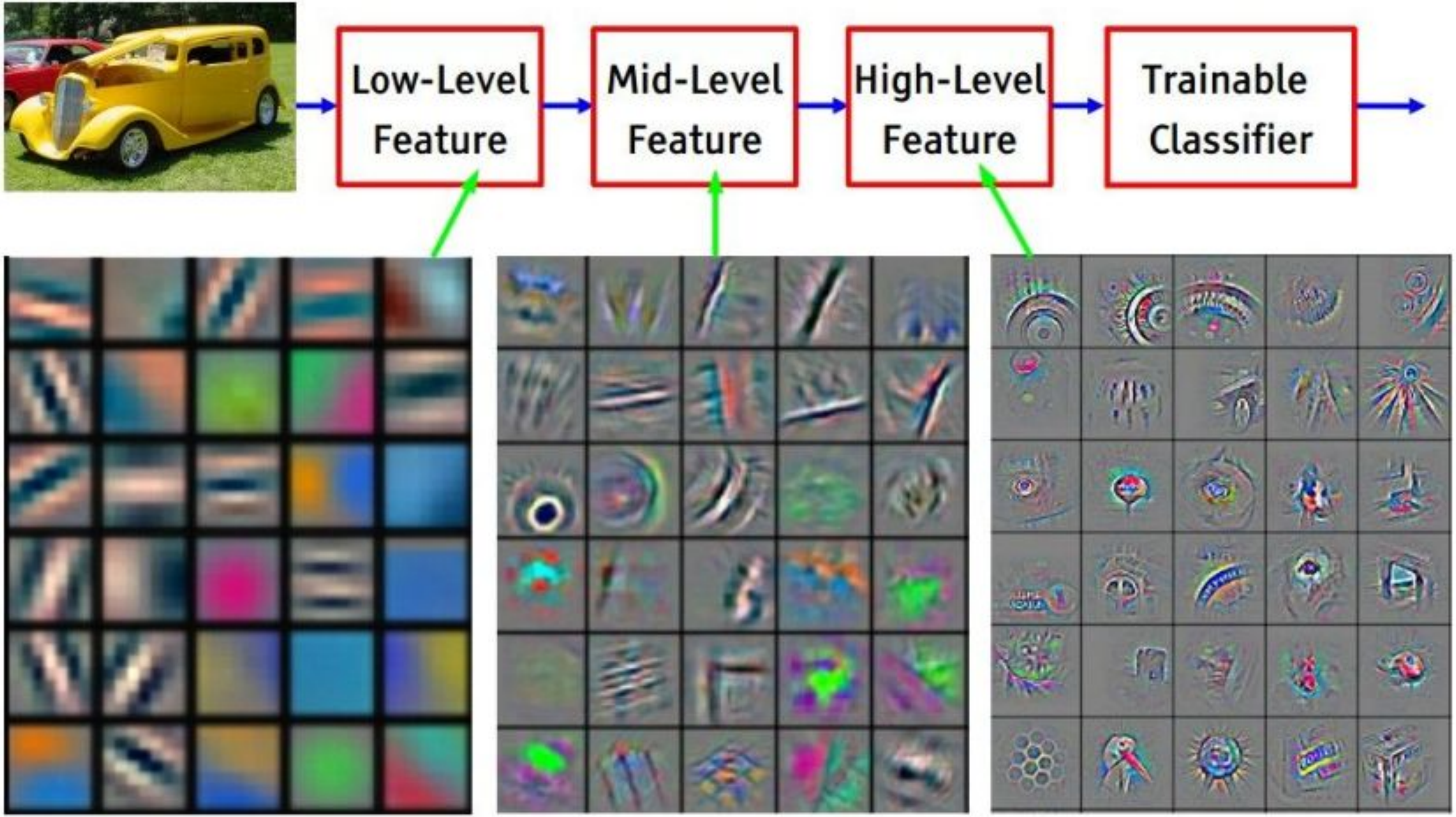
feature map



0	0	75	80	80
0	75	80	80	80
0	75	80	80	80
0	70	75	80	80
0	0	0	0	0

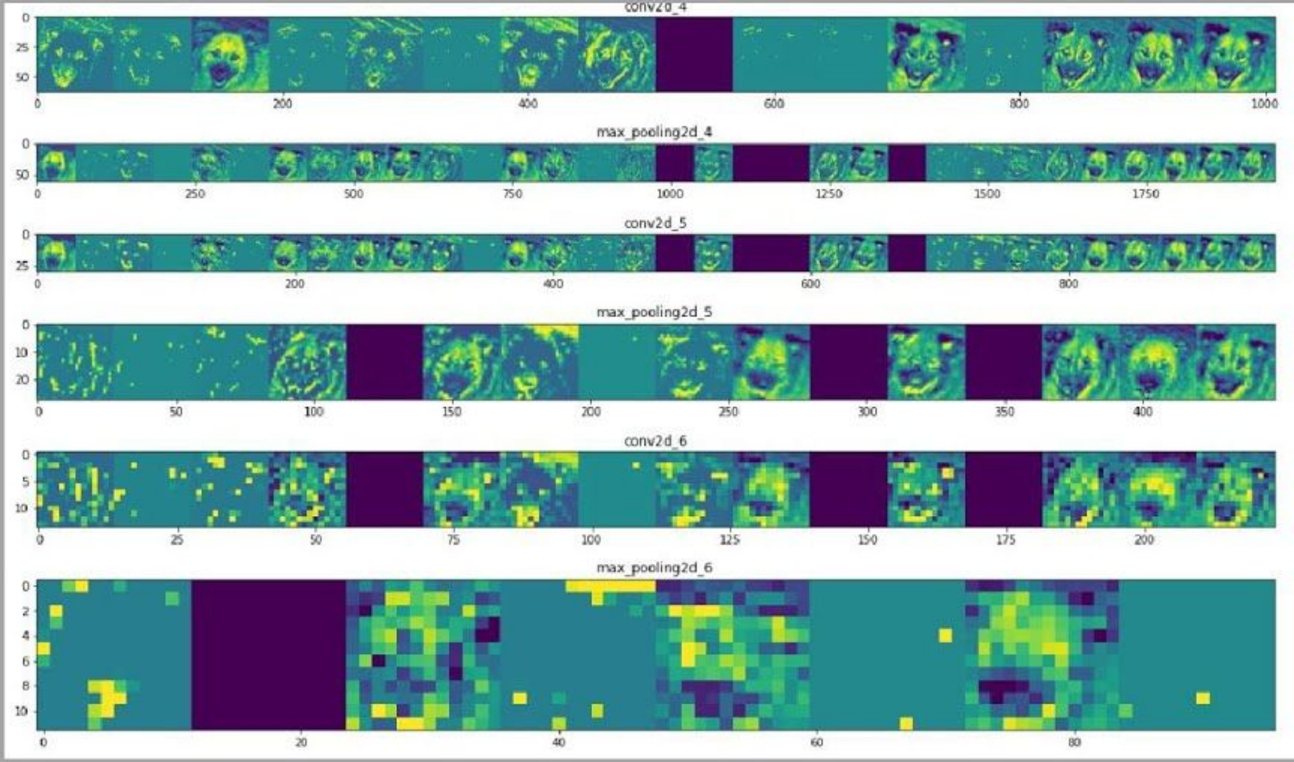
-1	-2	-1
0	0	0
1	2	1

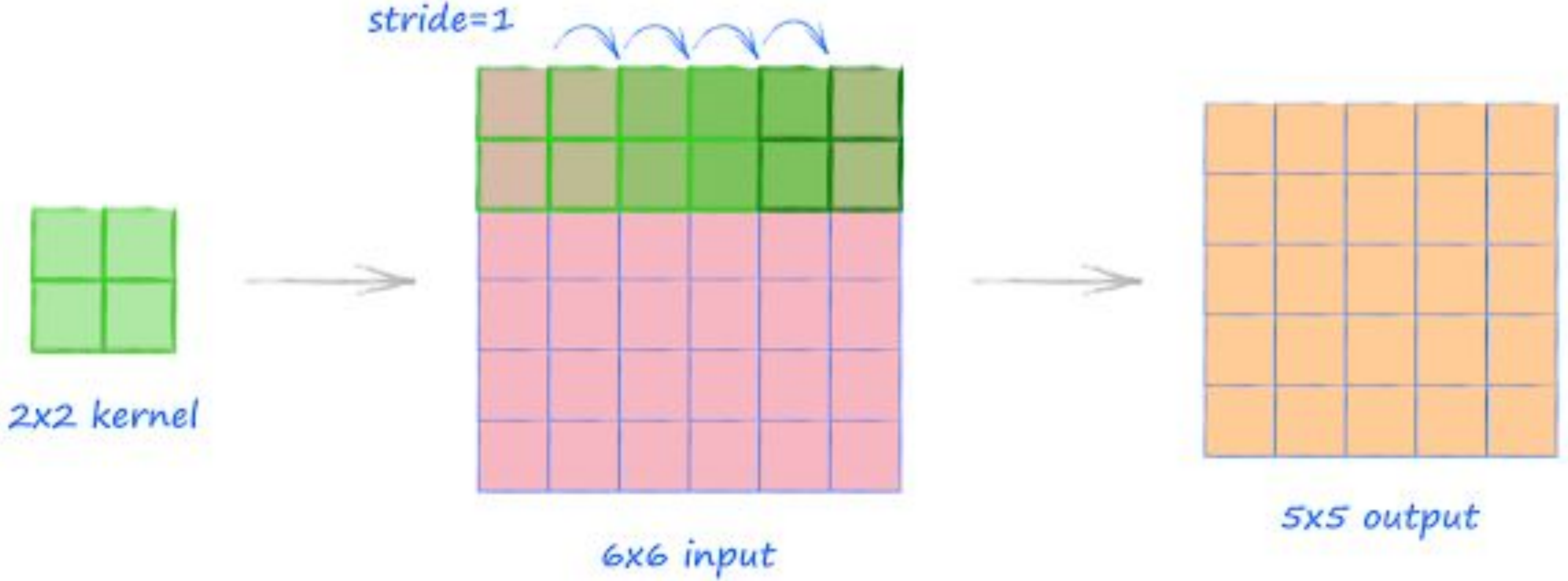
	155			

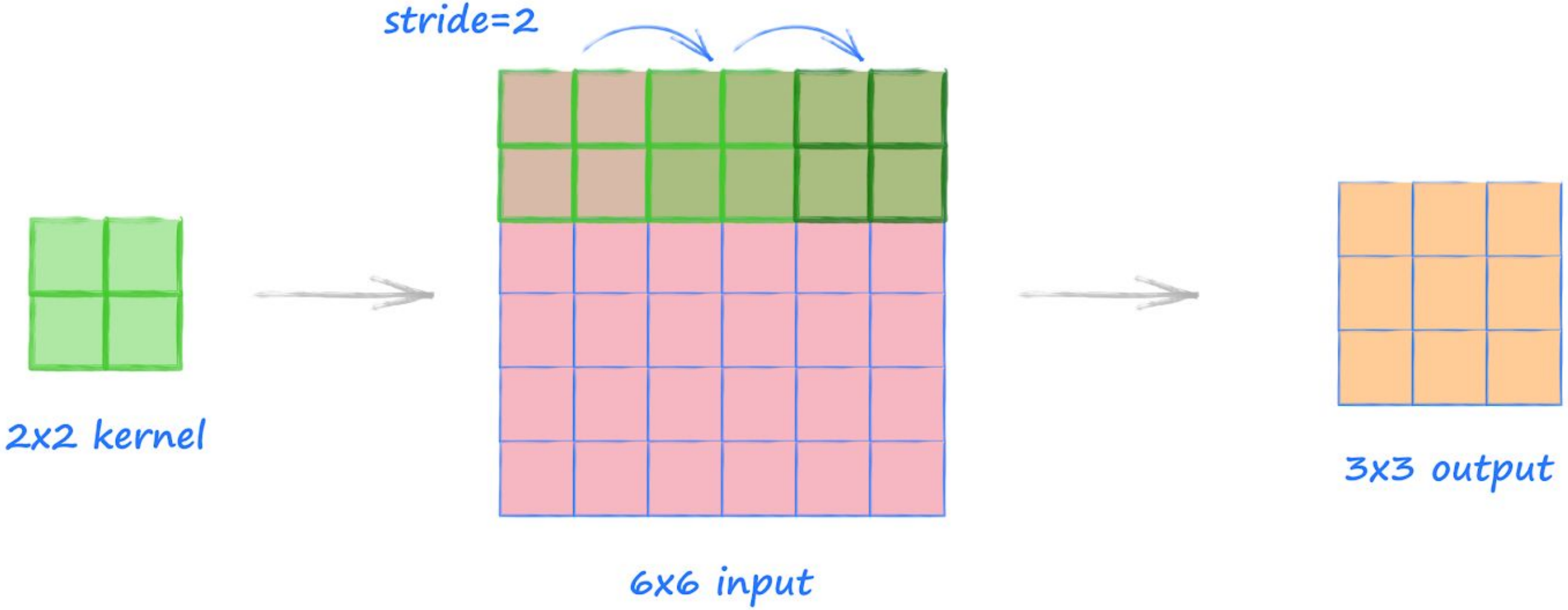

Low-Level Feature
Mid-Level Feature
High-Level Feature
Trainable Classifier

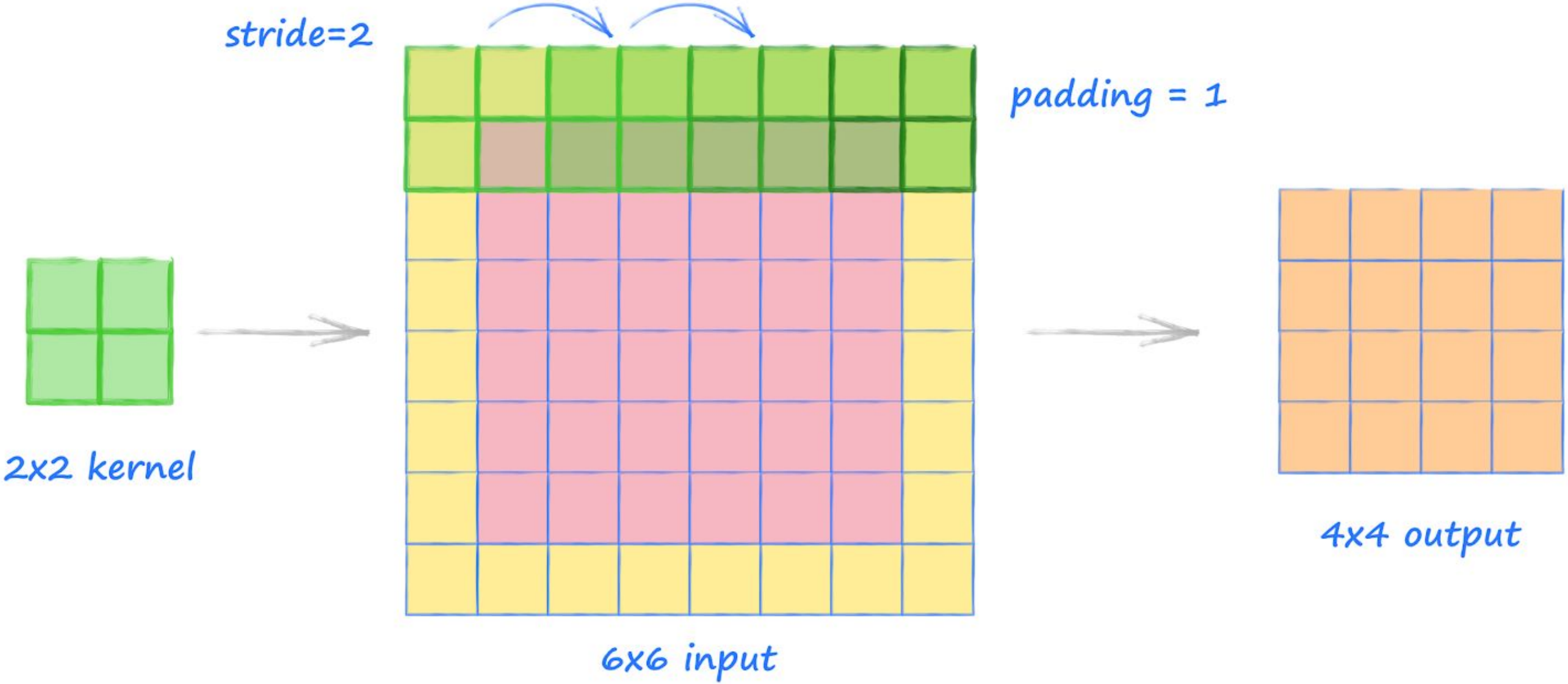
Visualize Feature Maps of CNN Layers to Understand

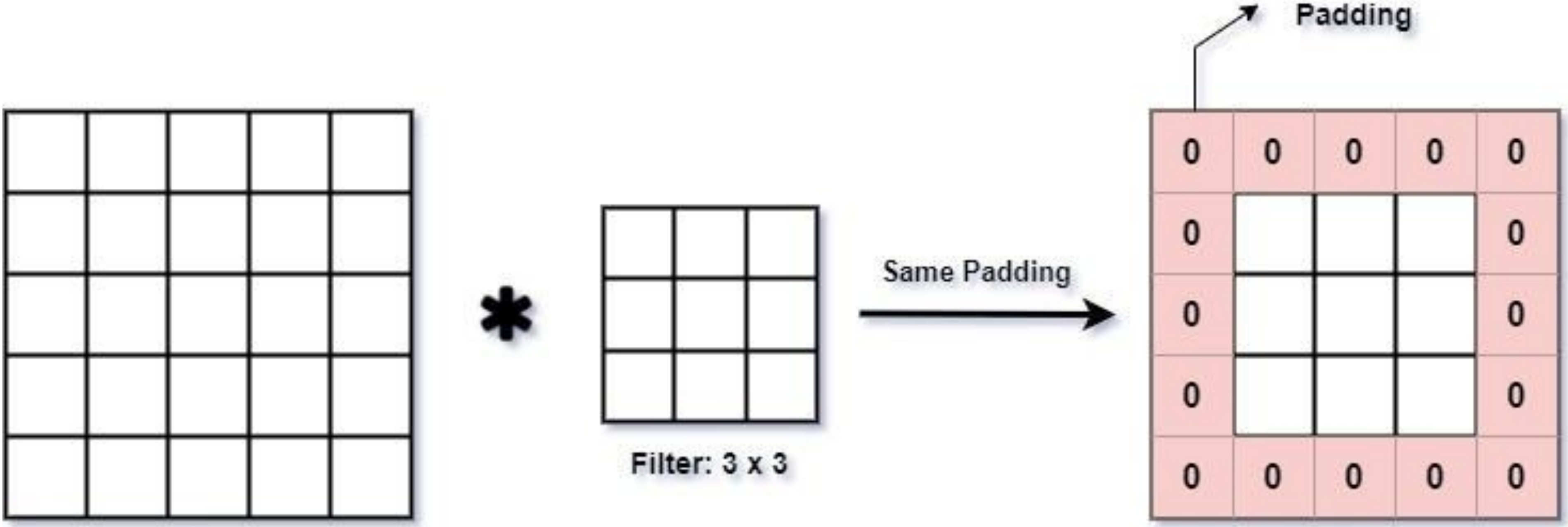
What Layers are Learning




stride=1
2x2 kernel
6x6 input
5x5 output


stride=2
2x2 kernel
6x6 input
3x3 output


stride=2
padding = 1
2x2 kernel
6x6 input
4x4 output



Original Image: 5 x 5

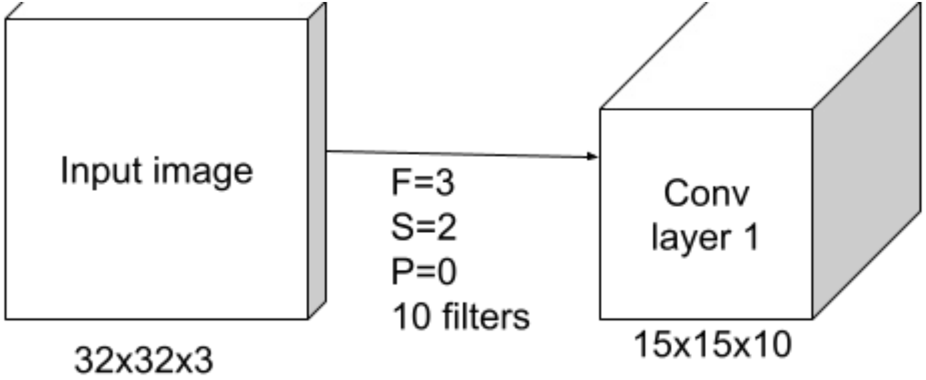
Filter: 3 x 3

Feature Map: 5 x 5

Calculating Output Sizes

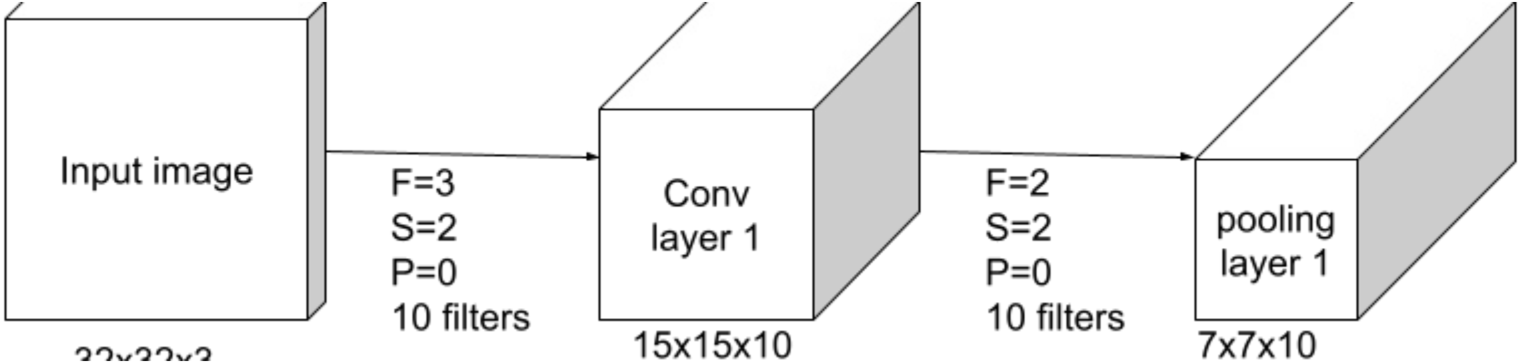
$$\text{output size} = \left\lfloor \frac{(\text{input size}) + 2*\text{padding} - (\text{kernel size} - 1) - 1}{\text{stride}} + 1 \right\rfloor$$

Calculate Output Size of Convolutional and Pooling layers



$$\text{Floor}\left(\frac{(W-F+2P)}{S}+1\right)$$

Calculate Output Size of Convolutional and Pooling layers



Input image

32x32x3

F=3
S=2
P=0
10 filters

Conv layer 1

15x15x10

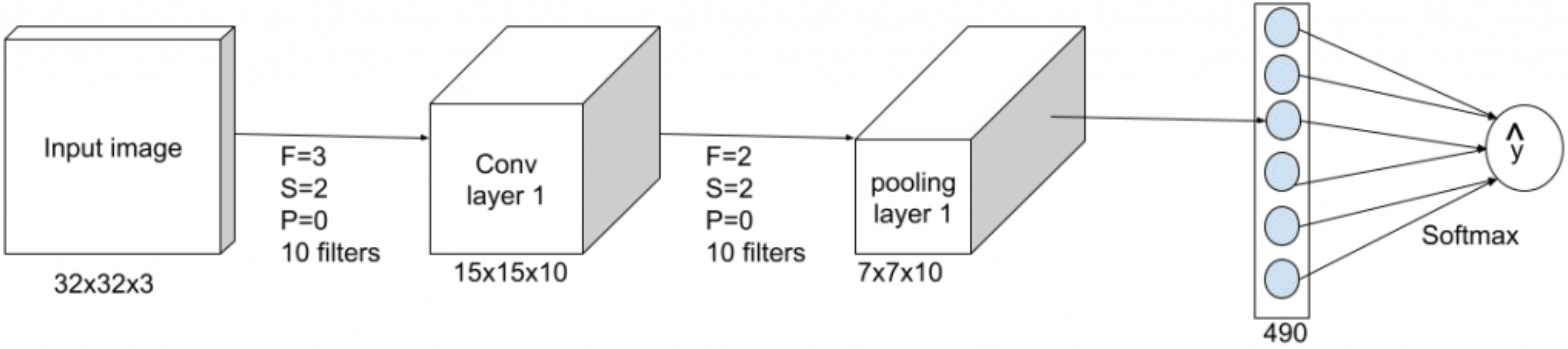
F=2
S=2
P=0
10 filters

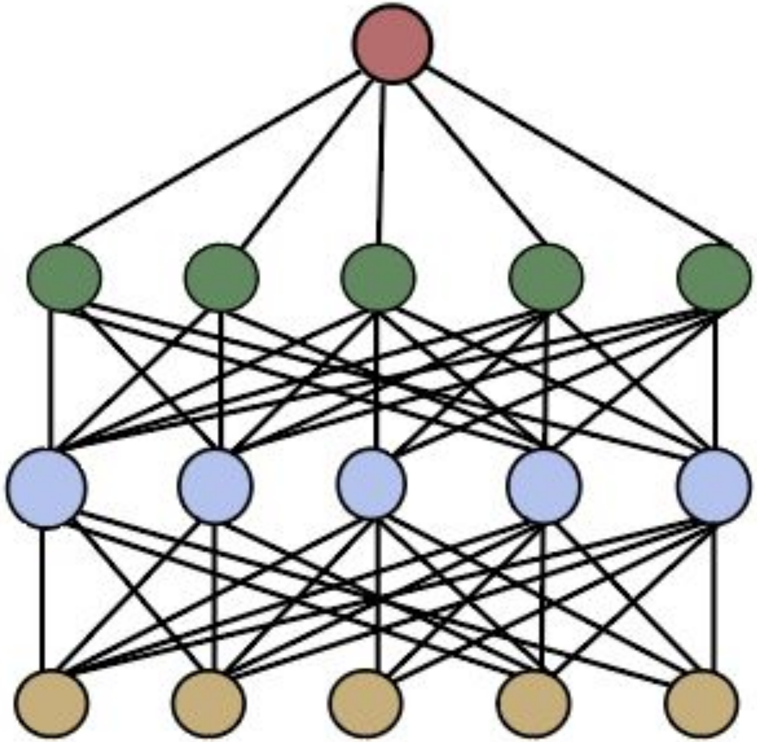
pooling layer 1

7x7x10

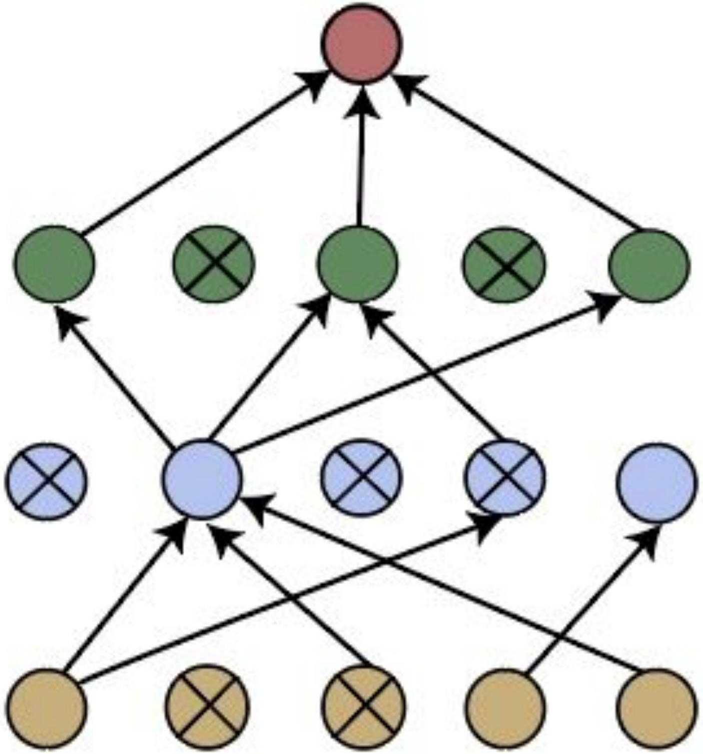
$$\mathrm{Floor}\left(\frac{(W-F)}{S} + 1 \right)$$

Fully Connected layer





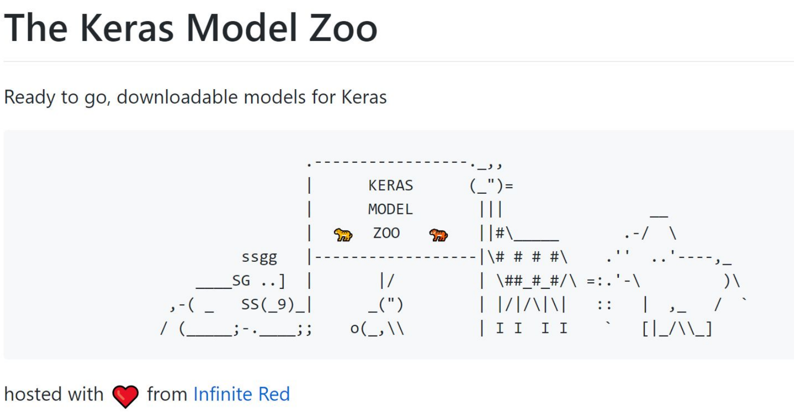
Standard Neural Network



After Applying Dropout

TensorFlow 2 Classification **Model Zoo**

https://github.com/tensorflow/models/blob/master/research/object_detection/g3doc/tf2_classification_zoo.md

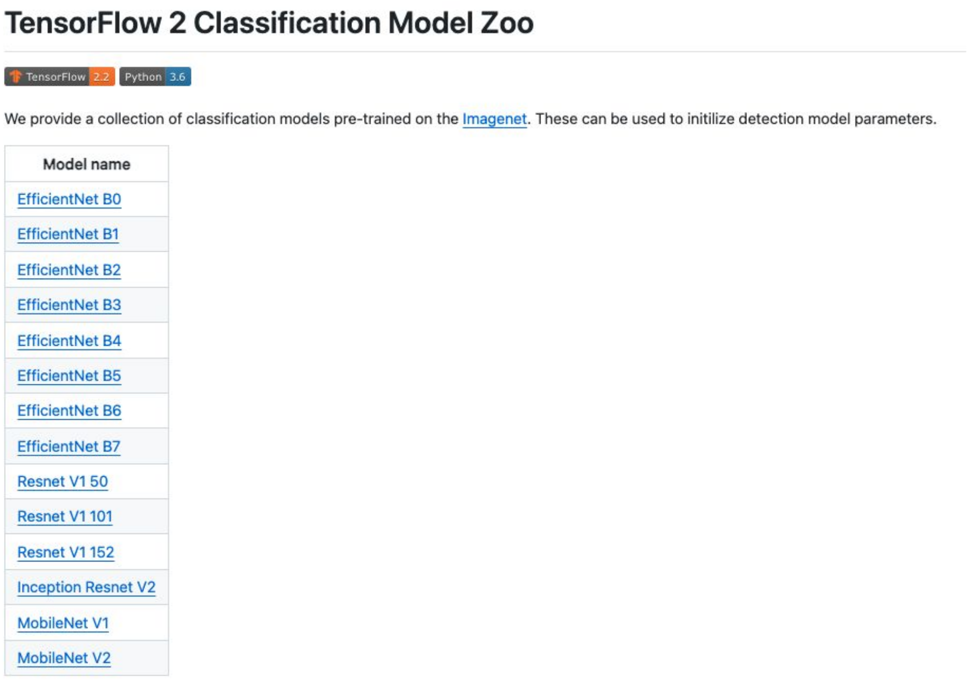


The Keras Model Zoo

Ready to go, downloadable models for Keras

```
                .-----------------._,,
                |      KERAS      (_")=
                |      MODEL       |||                __
                |  🐆  ZOO   🐅    ||#\_____      .-/  \
         ssgg   |------------------|\# # # #\    .''  ..'----,_
    ____SG ..]  |       |/         | \##_#_#/\ =:.'-\         )\
 ,-( _   SS(_9)_|      _(")        | |/|/\|\|   ::   |  ,_   /  `
/ (_____;-.____;;    o(_,\\         | I I  I I    `   [|_/\\_]
```

hosted with ❤️ from Infinite Red



TensorFlow 2 Classification Model Zoo

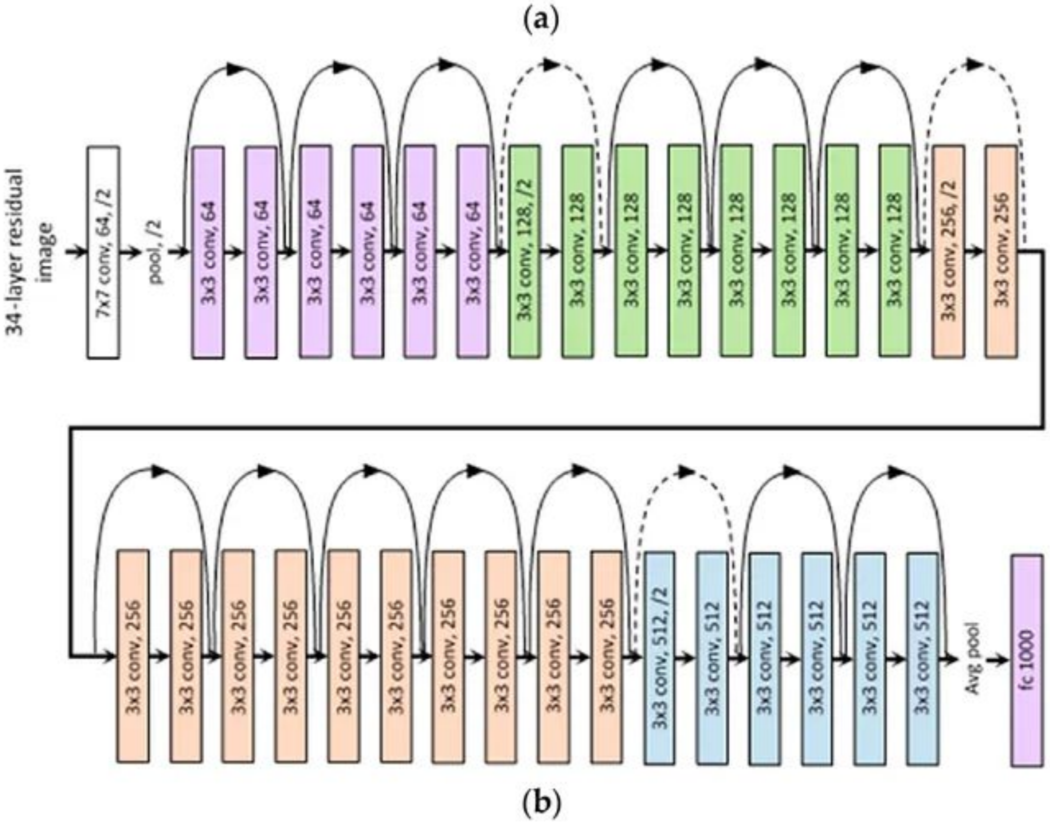
TensorFlow 2.2 Python 3.6

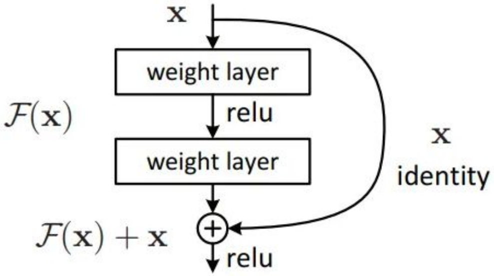
We provide a collection of classification models pre-trained on the Imagenet. These can be used to initilize detection model parameters.

Model name
EfficientNet B0
EfficientNet B1
EfficientNet B2
EfficientNet B3
EfficientNet B4
EfficientNet B5
EfficientNet B6
EfficientNet B7
Resnet V1 50
Resnet V1 101
Resnet V1 152
Inception Resnet V2
MobileNet V1
MobileNet V2

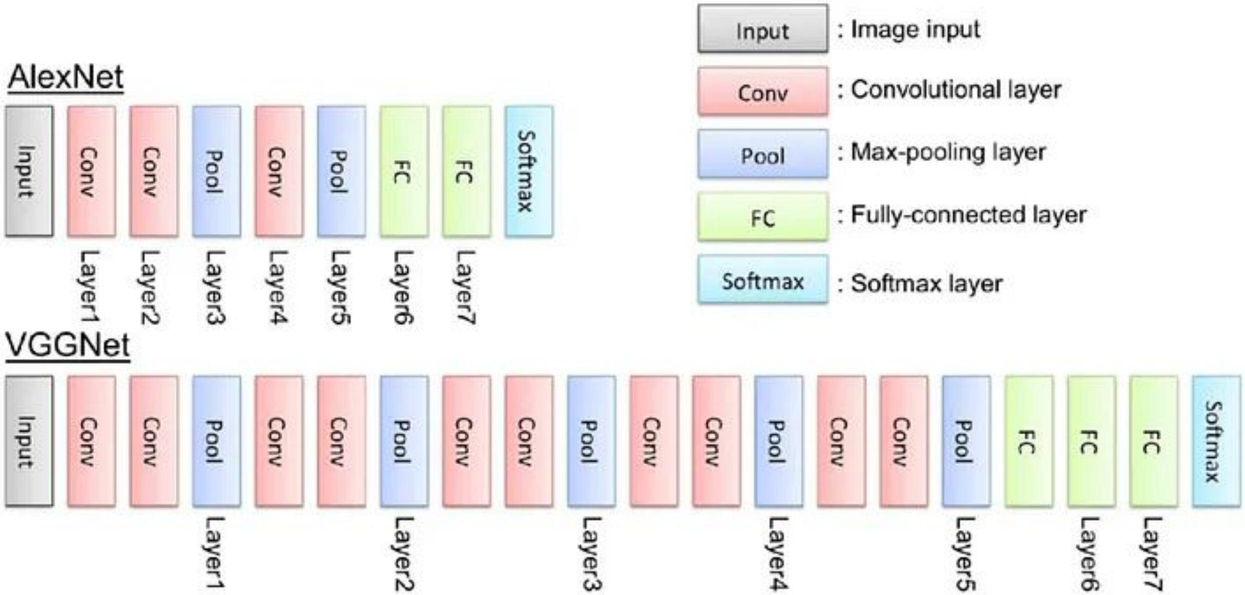
https://medium.com/@siddheshb008/resnet-architecture-explained-47309ea9283d

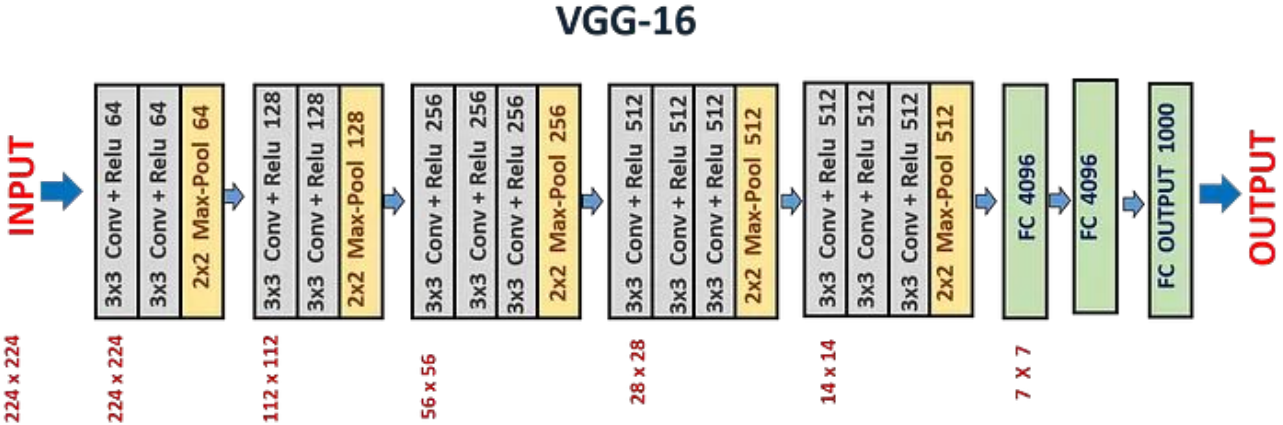
Resnet Architecture Explained

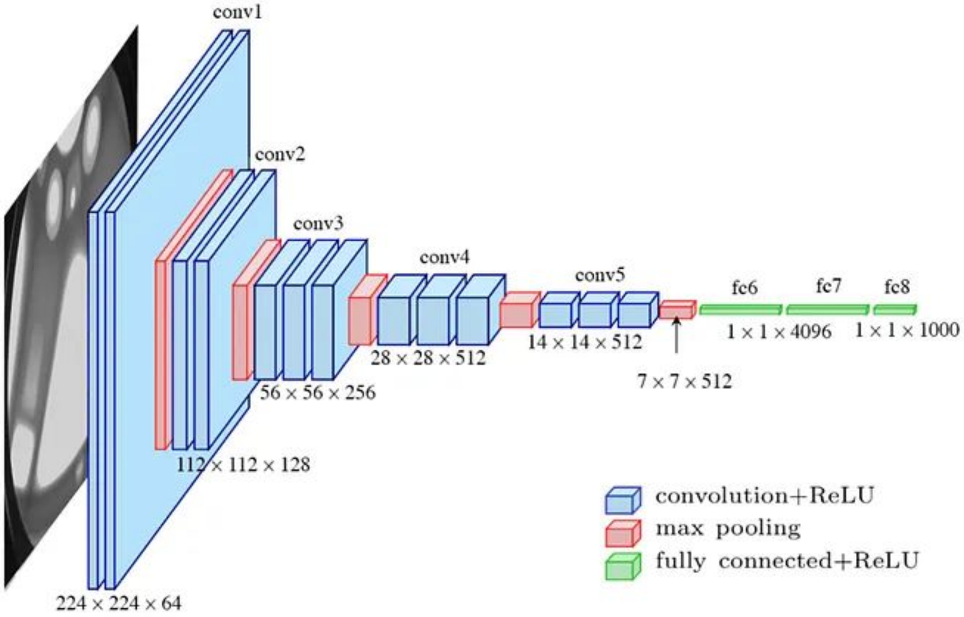


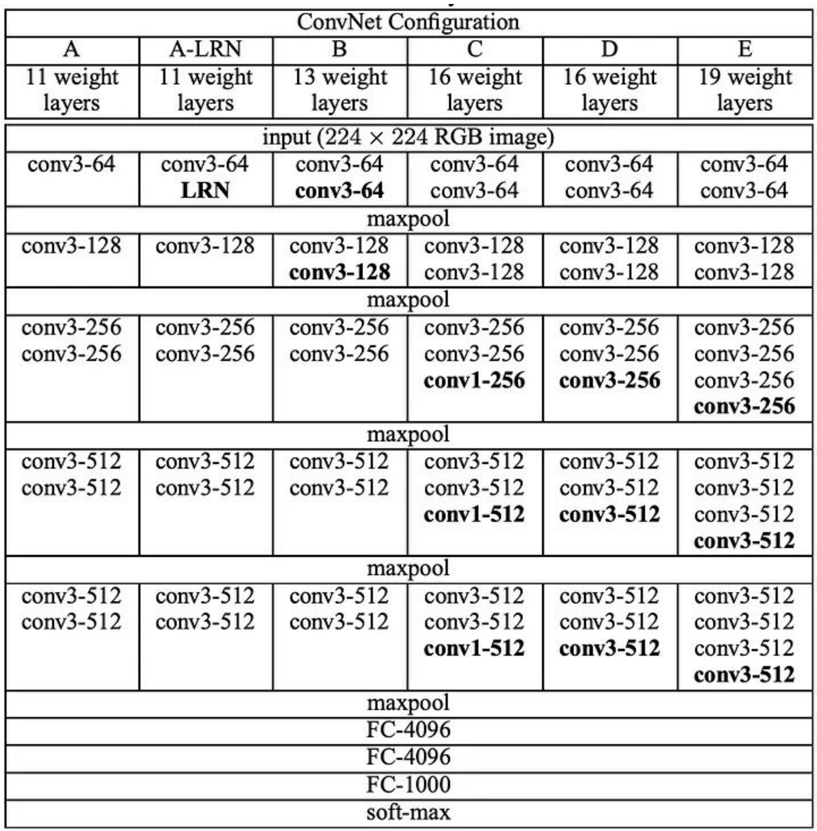


Skip connection or Shortcut


AlexNet
Input
Conv
Conv
Pool
Conv
Pool
FC
FC
Softmax
Layer1
Layer2
Layer3
Layer4
Layer5
Layer6
Layer7
Input : Image input
Conv : Convolutional layer
Pool : Max-pooling layer
FC : Fully-connected layer
Softmax : Softmax layer
VGGNet
Input
Conv
Conv
Pool
Conv
Conv
Pool
Conv
Conv
Pool
Conv
Conv
Pool
Conv
Conv
Pool
FC
FC
FC
Softmax
Layer1
Layer2
Layer3
Layer4
Layer5
Layer6
Layer7


VGG-16
INPUT
3x3 Conv + Relu 64
3x3 Conv + Relu 64
2x2 Max-Pool 64
3x3 Conv + Relu 128
3x3 Conv + Relu 128
2x2 Max-Pool 128
3x3 Conv + Relu 256
3x3 Conv + Relu 256
3x3 Conv + Relu 256
2x2 Max-Pool 256
3x3 Conv + Relu 512
3x3 Conv + Relu 512
3x3 Conv + Relu 512
2x2 Max-Pool 512
3x3 Conv + Relu 512
3x3 Conv + Relu 512
3x3 Conv + Relu 512
2x2 Max-Pool 512
FC 4096
FC 4096
FC OUTPUT 1000
OUTPUT
224 x 224
224 x 224
112 x 112
56 x 56
28 x 28
14 x 14
7 X 7

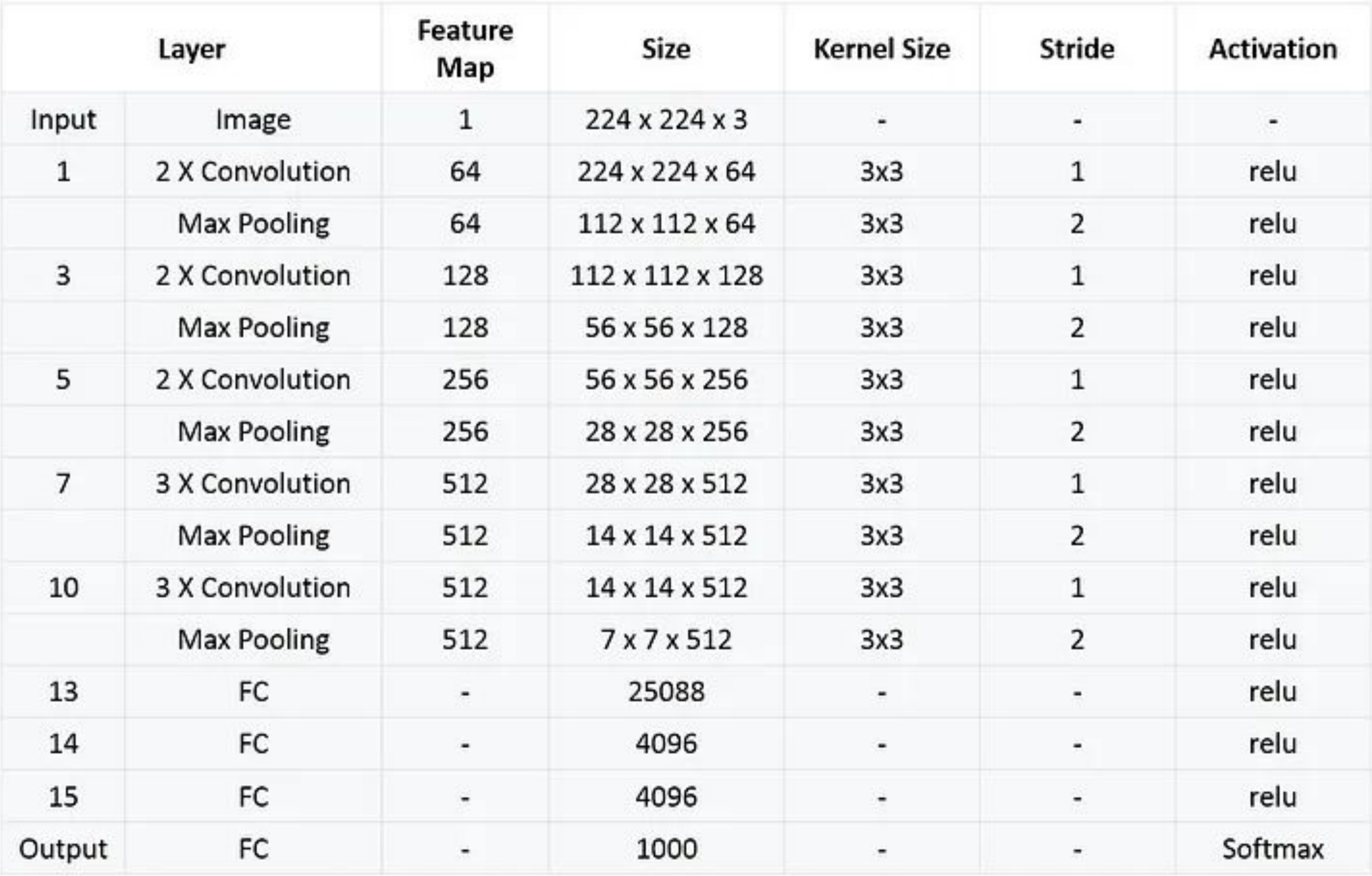




ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 **LRN**	conv3-64 **conv3-64**	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 **conv3-128**	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 **conv1-256**	conv3-256 conv3-256 **conv3-256**	conv3-256 conv3-256 conv3-256 **conv3-256**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

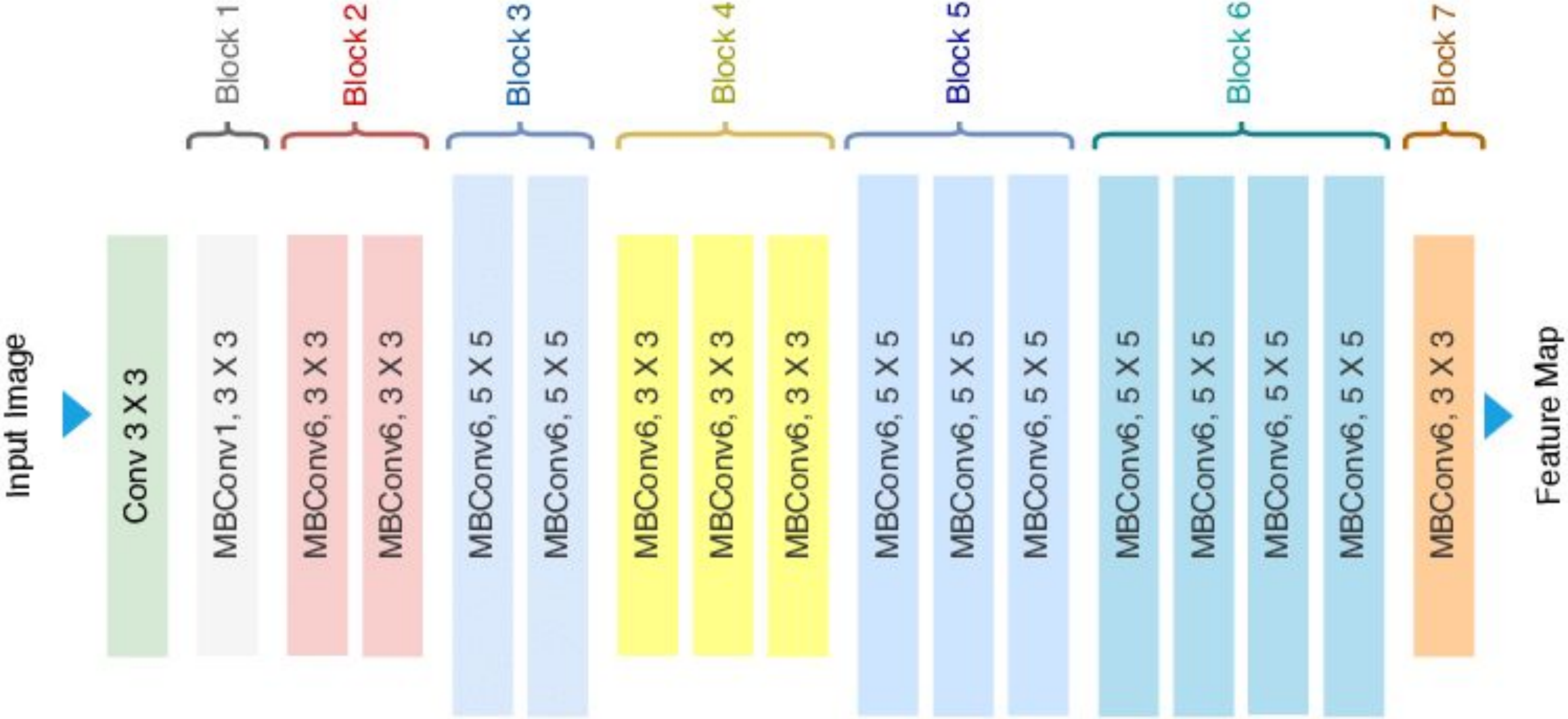
https://medium.com/@siddheshb008/vgg-net-architecture-explained-71179310050f

VGG-16 Summary

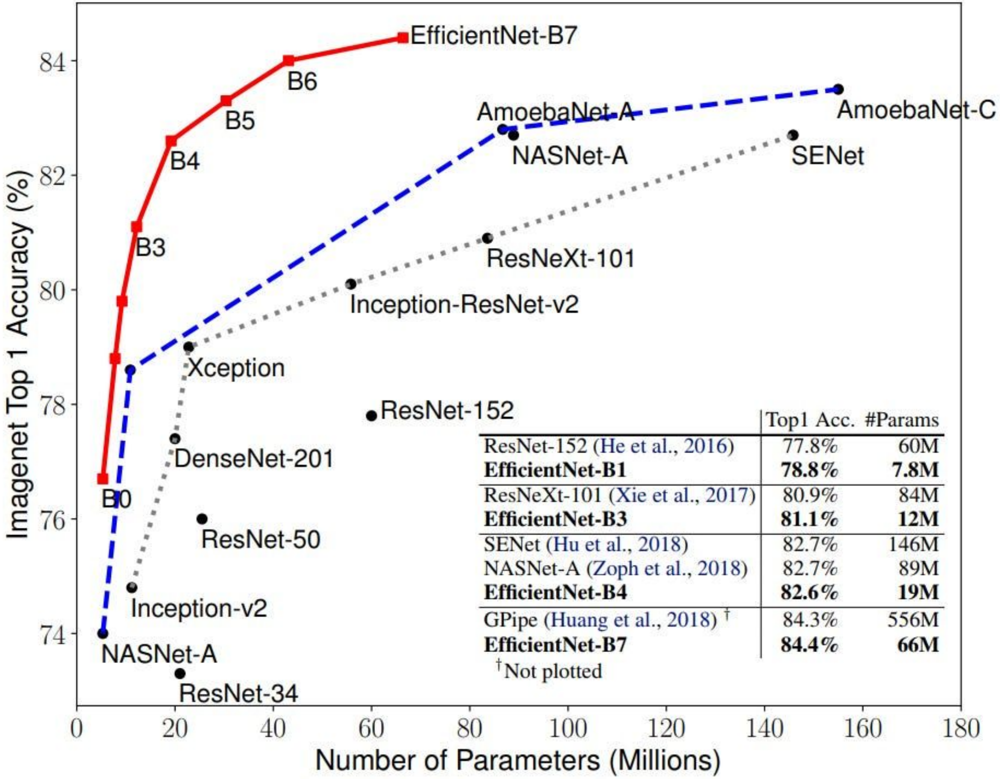


Layer		Feature Map	Size	Kernel Size	Stride	Activation
Input	Image	1	224 x 224 x 3	-	-	-
1	2 X Convolution	64	224 x 224 x 64	3x3	1	relu
	Max Pooling	64	112 x 112 x 64	3x3	2	relu
3	2 X Convolution	128	112 x 112 x 128	3x3	1	relu
	Max Pooling	128	56 x 56 x 128	3x3	2	relu
5	2 X Convolution	256	56 x 56 x 256	3x3	1	relu
	Max Pooling	256	28 x 28 x 256	3x3	2	relu
7	3 X Convolution	512	28 x 28 x 512	3x3	1	relu
	Max Pooling	512	14 x 14 x 512	3x3	2	relu
10	3 X Convolution	512	14 x 14 x 512	3x3	1	relu
	Max Pooling	512	7 x 7 x 512	3x3	2	relu
13	FC	-	25088	-	-	relu
14	FC	-	4096	-	-	relu
15	FC	-	4096	-	-	relu
Output	FC	-	1000	-	-	Softmax

EfficientNet



EfficientNet Performance



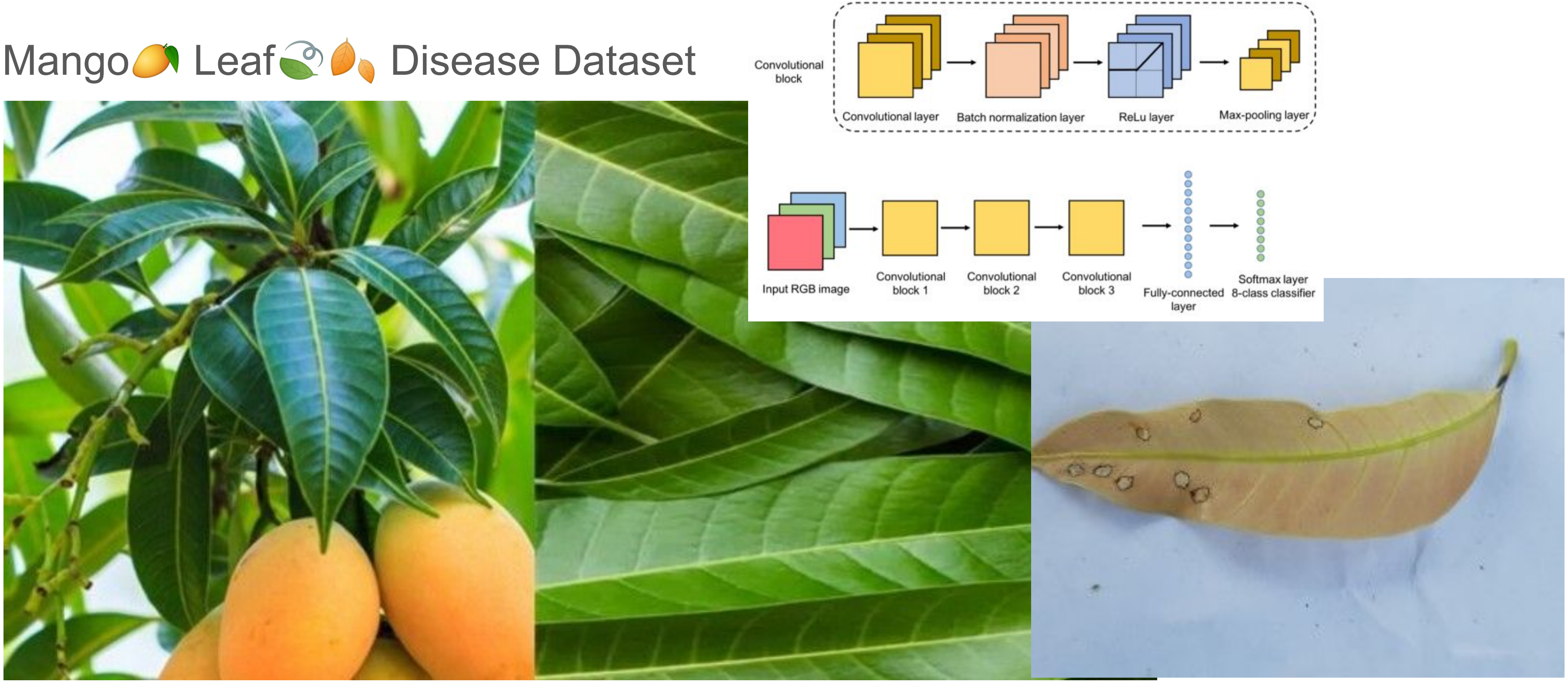
	Top1 Acc.	#Params
ResNet-152 (He et al., 2016)	77.8%	60M
EfficientNet-B1	**78.8%**	**7.8M**
ResNeXt-101 (Xie et al., 2017)	80.9%	84M
EfficientNet-B3	**81.1%**	**12M**
SENet (Hu et al., 2018)	82.7%	146M
NASNet-A (Zoph et al., 2018)	82.7%	89M
EfficientNet-B4	**82.6%**	**19M**
GPipe (Huang et al., 2018) †	84.3%	556M
EfficientNet-B7	**84.4%**	**66M**

†Not plotted

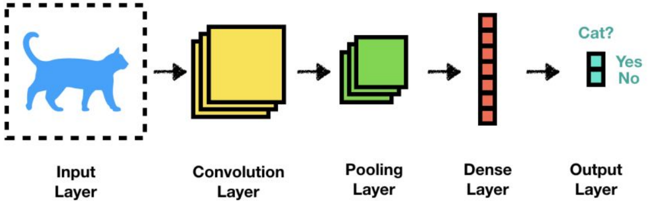
Deep Learning Exercise:

TensorFlow

Mango🥭 Leaf🍃🍂 Disease Dataset

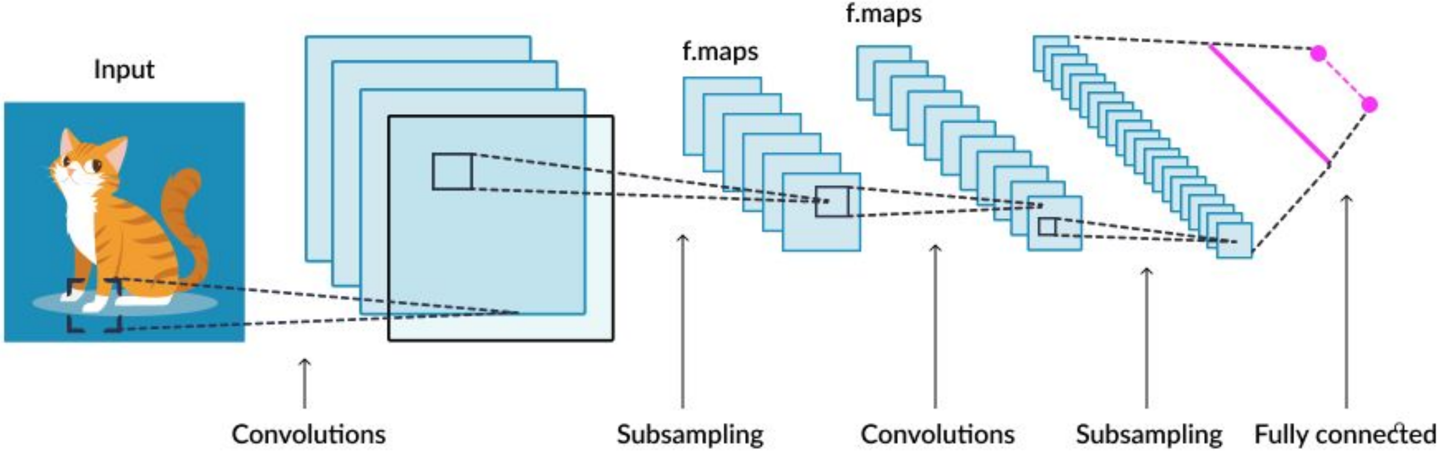


Week 10: Assignment



Build **Your Own Image** Classification in TensorFlow

This assignment will help you understand the fundamentals of image classification, TensorFlow, and neural network architectures.


Input
f.maps
f.maps
Convolutions
Subsampling
Convolutions
Subsampling
Fully connected