Convolutional Neural Network
(CNN)

Topics to be covered

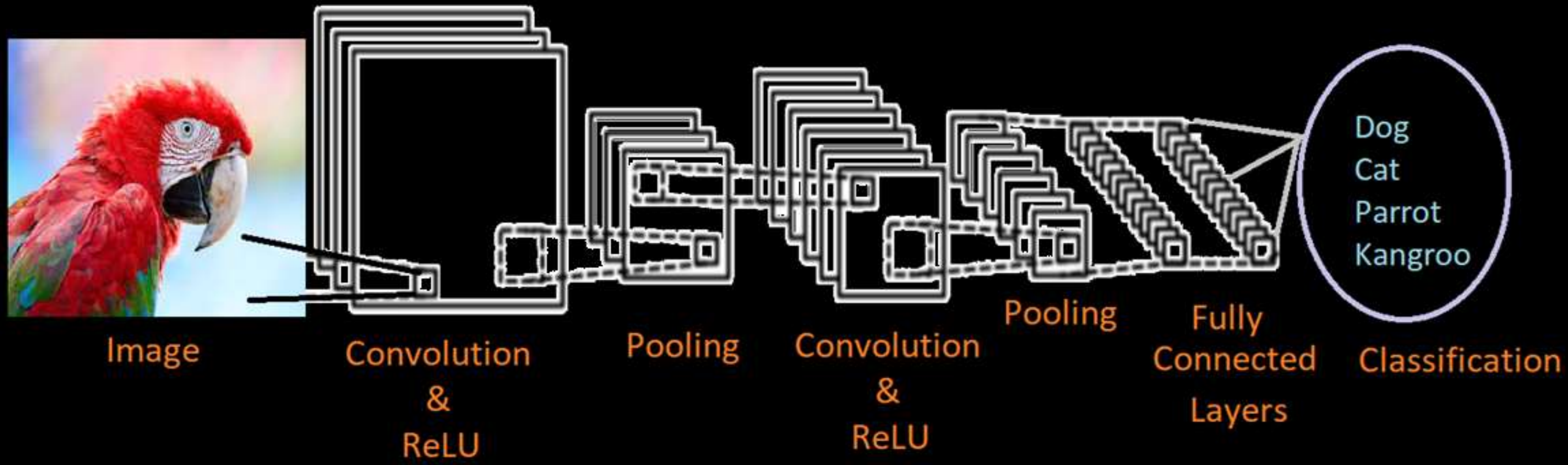
- CNN Architecture and Main Operations.
- 2D Convolution
- Shapes of Feature Maps
- Pooling
- Transfer Learning
- Four Projects

CNN Architecture
and
Operations

CNN Main Operations

- Convolution
- Non Linearity (ReLU)
- Pooling OR Subsampling
- Classification

Typical CNN Architecture



CNN Main Operations

- Convolution
- Non Linearity (ReLU)
- Pooling OR Subsampling
- Classification

2D Convolution

Steps in 2D Convolution

- Flipping
- Shifting / Sliding
- Multiplication
- Addition

6	3	7	9	1
7	0	3	1	3
8	1	5	3	6
3	9	4	3	5
7	8	2	3	0

Image

1	3	1
2	-1	2
3	-2	-1

Kernel

Flipping the Kernel

1	3	1
2	-1	2
3	-2	-1

Kernel

3	-2	-1
2	-1	2
1	3	1

Horizontally Flipped

-1	-2	3
2	-1	2
1	3	1

Flipped Kernel

6	3	7	9	1
7	0	3	1	3
8	1	5	3	6
3	9	4	3	5
7	8	2	3	0

Image

-1	-2	3
2	-1	2
1	3	1

Flipped Kernel

Multiplication and Addition

6	3	7	9	1
7	0	3	1	3
8	1	5	3	6
3	9	4	3	5
7	8	2	3	0

Image

-1	-2	3
2	-1	2
1	3	1

	45			

Convolution

(6 x -1) + (3 x -2) + (7 x 3)
(7 x 2) + (0 x -1) + (3 x 2)
(8 x 1) + (1 x 3) + (5 x 1)

= -6 -6 +21 + 14 + 0 + 6 + 8 + 3 + 5

= 45

6	3	7	9	1
7	0	3	1	3
8	1	5	3	6
3	9	4	3	5
7	8	2	3	0

Image

-1	-2	3
2	-1	2
1	3	1

Multiplication and Addition

0	0	0			
0	6	3	7	9	1
0	7	0	3	1	3
	8	1	5	3	6
	3	9	4	3	5
	7	8	2	3	0

Image

-1	-2	3
2	-1	2
1	3	1

21				
	45			

Convolution

(6 x -1) + (3 x 2) + (7 x 3)
+ (0 x 1)

= -6 + 6 + 21 + 0
= 21

Convolution Results

21	33	27	16	27
15	45	28	9	9
0	61	24	41	11
31	43	35	33	0
10	1	12	6	0

Convolution

Shape of Output Feature Map After Convolution

Assume we have K filters or kernels, each of size $m \times m$, then the shape of output feature map can be calculated by using the following equations

$$W_{out} = \frac{W_{in} - m + 2 * Padding}{Stride} + 1$$

$$H_{out} = \frac{H_{in} - m + 2 * Padding}{Stride} + 1$$

$$D_{out} = K$$

Concept of Stride

6	3	7	9	1
7	0	3	1	3
8	1	5	3	6
3	9	4	3	5
7	8	2	3	0

Image

1	3	1
2	-1	2
3	-2	-1

Kernel

6	3	7	9	1
7	0	3	1	3
8	1	5	3	6
3	9	4	3	5
7	8	2	3	0

6	3	7	9	1
7	0	3	1	3
8	1	5	3	6
3	9	4	3	5
7	8	2	3	0

6	3	7	9	1
7	0	3	1	3
8	1	5	3	6
3	9	4	3	5
7	8	2	3	0

Concept of Stride

6	3	7	9	1
7	0	3	1	3
8	1	5	3	6
3	9	4	3	5
7	8	2	3	0

Image

1	3	1
2	-1	2
3	-2	-1

Kernel

6	3	7	9	1
7	0	3	1	3
8	1	5	3	6
3	9	4	3	5
7	8	2	3	0

6	3	7	9	1
7	0	3	1	3
8	1	5	3	6
3	9	4	3	5
7	8	2	3	0

$\text{Stride} = 2$

Zero Padding

0	0	0	0	0	0	0
0	6	3	7	9	1	0
0	7	0	3	1	3	0
0	8	1	5	3	6	0
0	3	9	4	3	5	0
0	7	8	2	3	0	0
0	0	0	0	0	0	0

Zero Padding $= 1$

Shape of Output Feature Map After Convolution

Assume we have K filters or kernels, each of size $m \times m$, then the shape of output feature map can be calculated by using the following equations

$$W_{out} = \frac{W_{in} - m + 2 * Padding}{Stride} + 1$$

$$H_{out} = \frac{H_{in} - m + 2 * Padding}{Stride} + 1$$

$$D_{out} = K$$

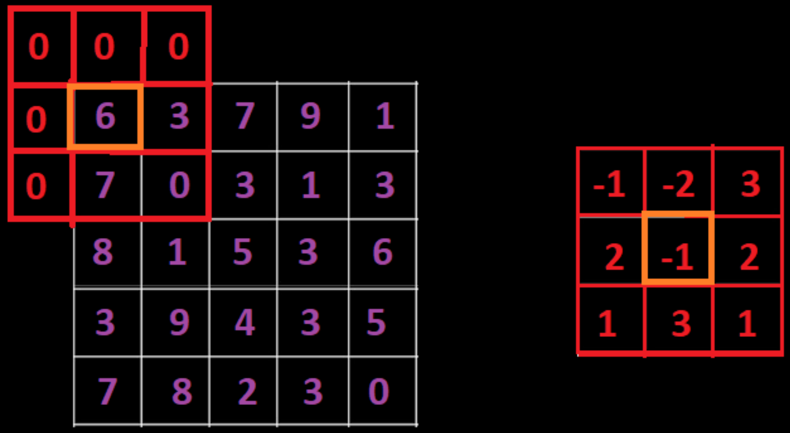
Zero Padding

6	3	7	9	1
7	0	3	1	3
8	1	5	3	6
3	9	4	3	5
7	8	2	3	0

Image

-1	-2	3
2	-1	2
1	3	1

Filter



Image

Shape of Output Feature Map

$$W_{out} = \frac{5 - 3 + 2 * 1}{1} + 1$$

$$\boxed{W_{out} = 5}$$

$$H_{out} = \frac{5 - 3 + 2 * 1}{1} + 1$$

$$\boxed{H_{out} = 5}$$

$$\boxed{D_{out} = 1}$$

Output Feature Map

6	3	7	9	1
7	0	3	1	3
8	1	5	3	6
3	9	4	3	5
7	8	2	3	0

Image

*

-1	-2	3
2	-1	2
1	3	1

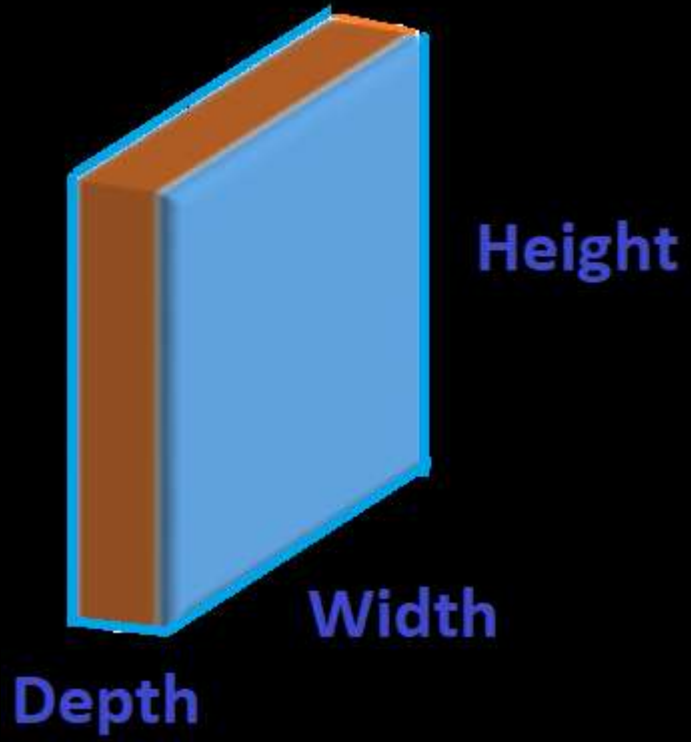
Filter

=

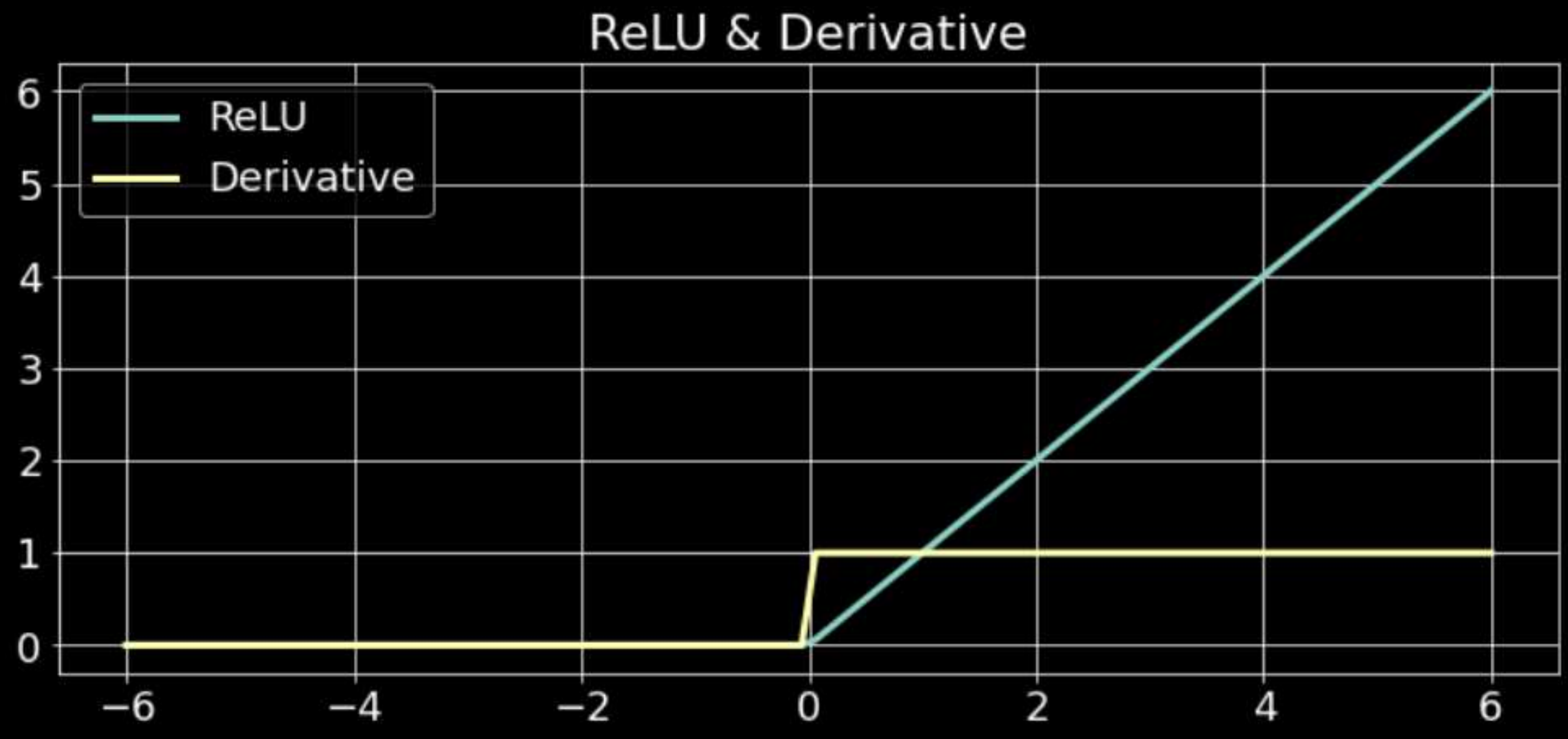
21	33	27	16	27
15	45	28	9	9
0	61	24	41	11
31	43	35	33	0
10	1	12	6	0

Feature Map

Volume of Feature Map



Rectified Linear Unit (ReLU) Function



$$ReLU(x) = max(0, x)$$

Rectified Linear Unit (ReLU) Function

0.77	-0.11	0.11	0.33	0.55	-0.11	0.33
-0.11	1.00	-0.11	0.33	-0.11	0.11	-0.11
0.11	-0.11	1.00	-0.33	0.11	-0.11	0.55
0.33	0.33	-0.33	0.55	-0.33	0.33	0.33
0.55	-0.11	0.11	-0.33	1.00	-0.11	0.11
-0.11	0.11	-0.11	0.33	-0.11	1.00	-0.11
0.33	-0.11	0.55	0.33	0.11	-0.11	0.77

0.77	0	0.11	0.33	0.55	0	0.33
0	1.00	0	0.33	0	0.11	0
0.11	0	1.00	0	0.11	0	0.55
0.33	0.33	0	0.55	0	0.33	0.33
0.55	0	0.11	0	1.00	0	0.11
0	0.11	0	0.33	0	1.00	0
0.33	0	0.55	0.33	0.11	0	0.77

Pooling

Pooling With Filter $= 2 \times 2$ and stride $= 2$

21	8	8	12
12	19	9	7
8	10	4	3
18	12	9	10

Feature Map

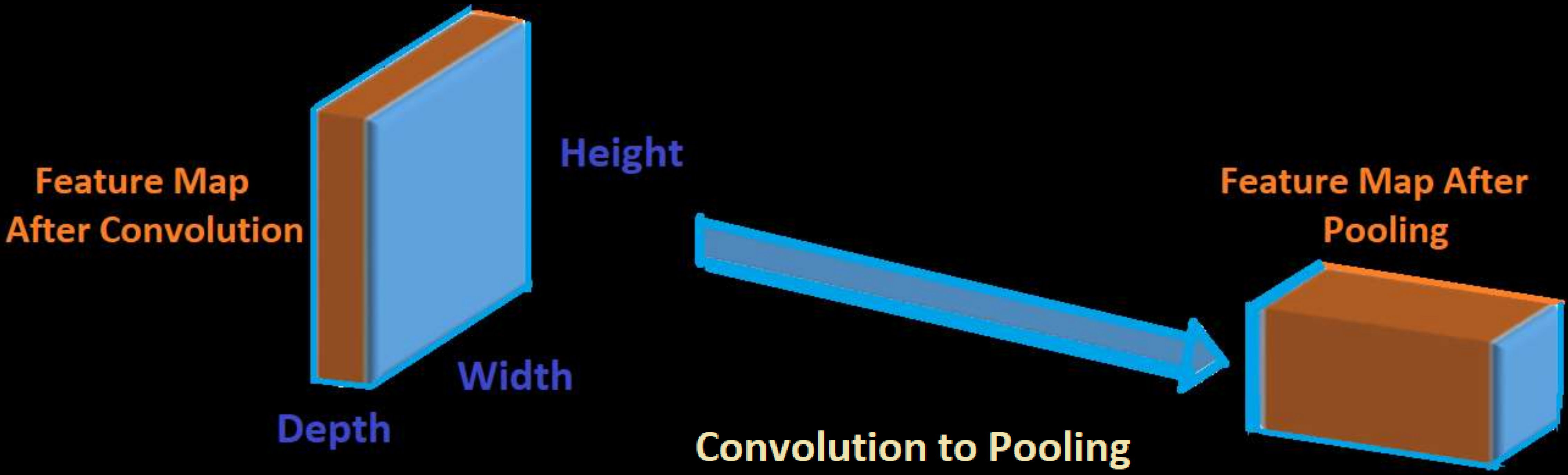
15	9
12	7

Average Pooling

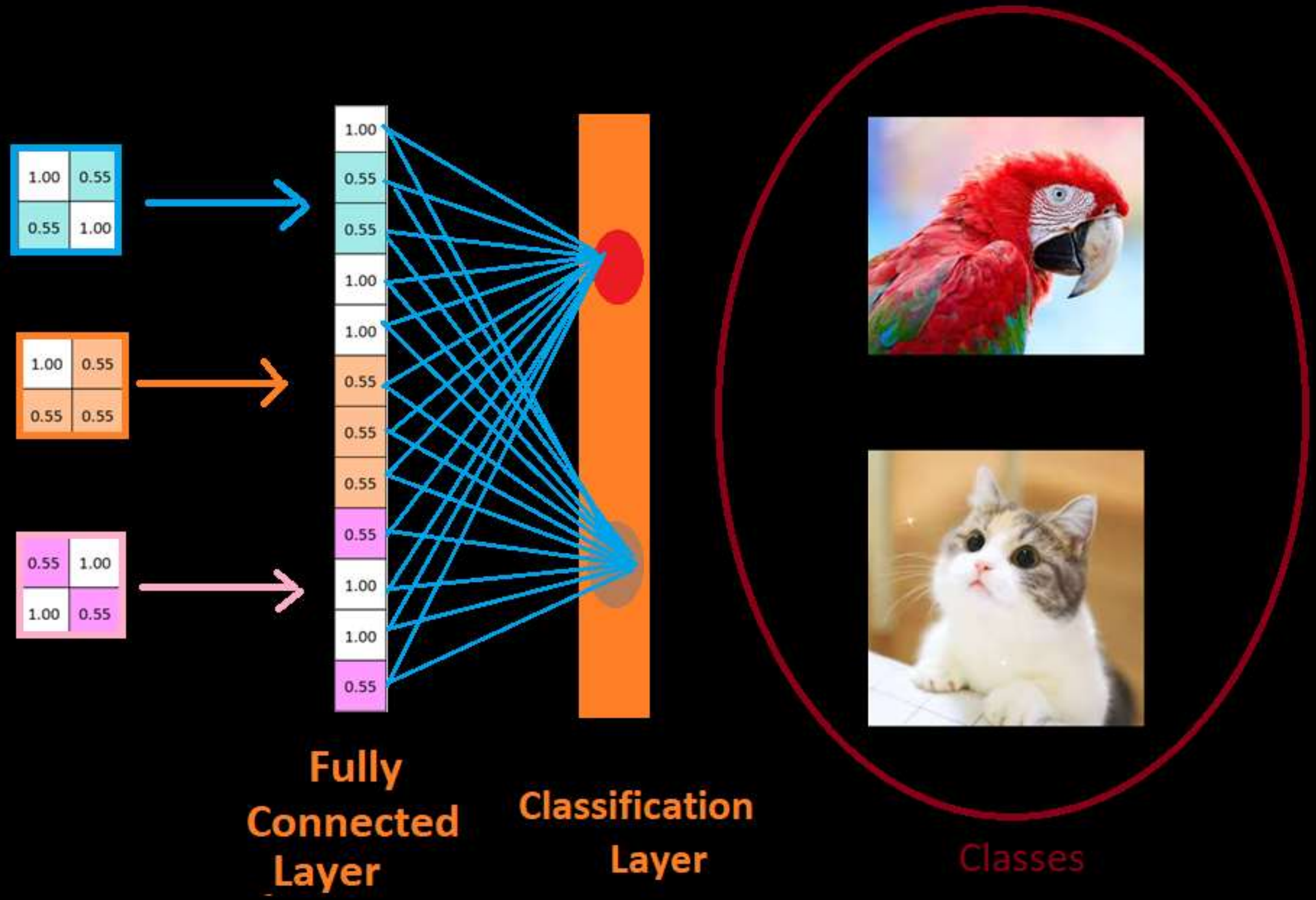
21	12
18	10

Maximum Pooling

Convolution to Pooling



Pooling to Classification


1.00 0.55
0.55 1.00
1.00 0.55
0.55 0.55
0.55 1.00
1.00 0.55
1.00
0.55
0.55
1.00
1.00
0.55
0.55
0.55
0.55
1.00
1.00
0.55
Fully Connected Layer
Classification Layer
Classes

Transfer Learning

Transfer Learning

Transfer learning involves fine-tuning a model trained on one set of data and then applying it to another collection of data and a different task.

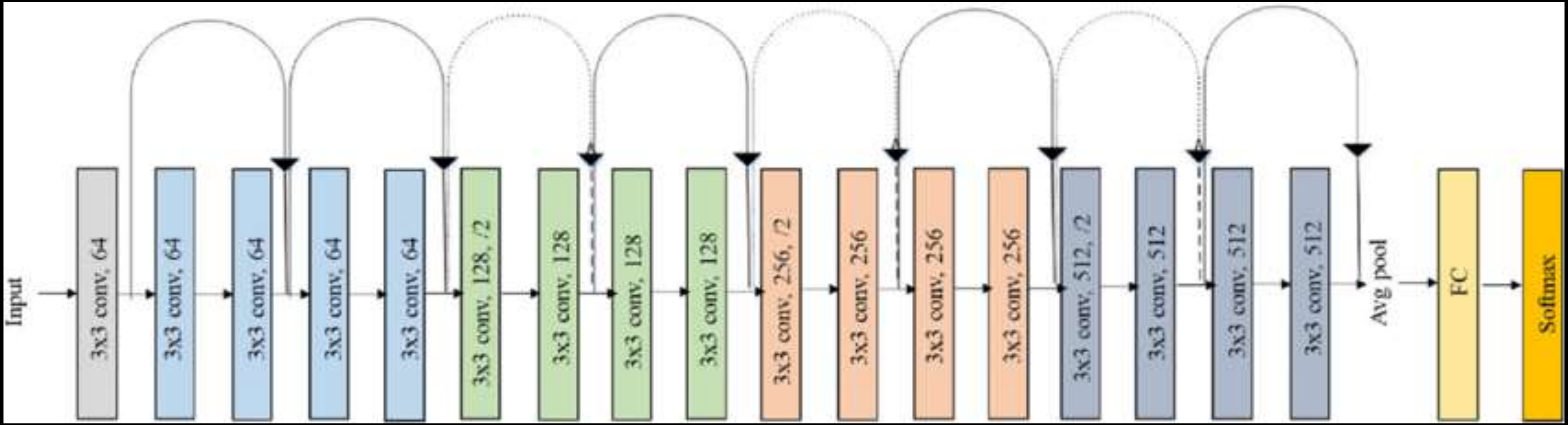
CNN Models for Transfer Learning

- AlexNet
- Inception
- MobileNet
- ResNet-18
- VGG-16

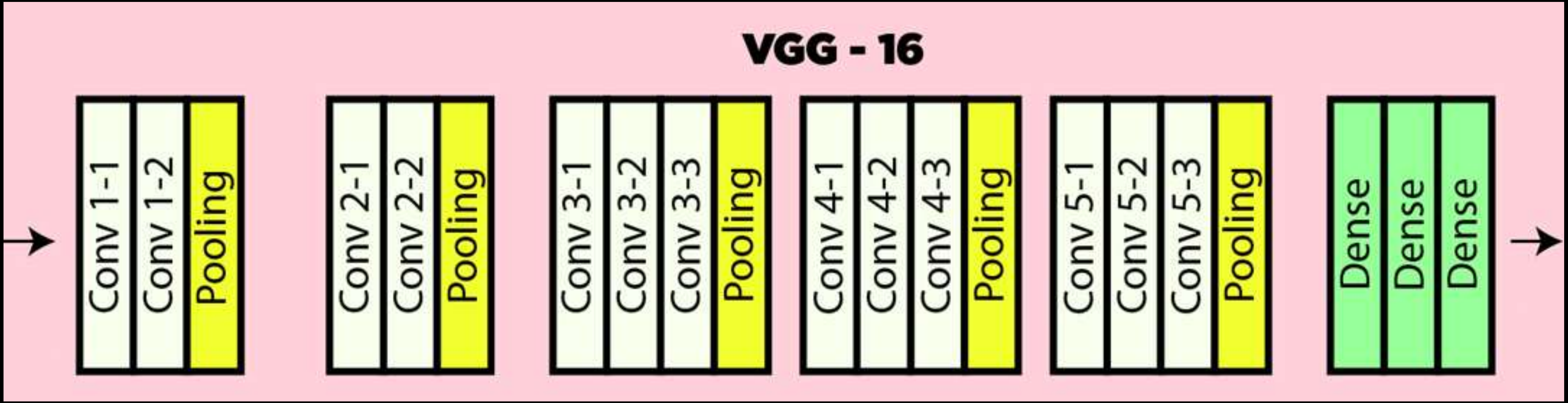
ResNet-18 and VGG-16

- Trained on ImageNet Dataset.
- The dataset has 1000000 samples.
- The dataset has 1000 classes.

ResNet-18 Architecture



VGG-16 Architecture



Advantages of Transfer Learning

- Less Training Time
- Large datasets are not required
- Improved Performance (in most cases).

Dataset for Transfer Learning

500 Images of each class