1. Image Classification:
   It is the process of assigning a single label to the whole/entire image based on what is present in the image.
   i/p: An image of ‘mxn’ matrix of pixel values
   o/p: A class label out of the multiple class label (this class label will be selected probabilistically using Soft Max activation function)

   Object Detection:
   It is the process of locating all the instances of the image which belongs to a specific category of objects and localizing the categories with the help of rectangular bounding boxes.
   i/p: An image of ‘mxn’ matrix of pixel values
   o/p: Bounding box indicating the ‘class’ and the ‘confidence score’

   Semantic Segmentation:
   It is the process in which classifying each and every pixel in an image into some predefined categories.
   i/p: An image of ‘mxn’ matrix of pixel values
   o/p: A pixel wise labelling of image in which each and every pixel is assigned to a class label segmented with different colors

   Instance Segmentation:
   Instance segmentation is actually Semantic Segmentation + Object Detection
   It is the process in which detecting and delineating each and every individual object in the image and categorizing between different instances which belongs to the same class.
   i/p: An image of ‘mxn’ matrix of pixel values
   o/p: A set of object masks which indicates the detected class along with localization with a bounding box

   Similarities and Differences:

   Similarities:
   - All the four types of these computer vision tasks involves handling with visual images so they mostly rely on Covolutional Neural Networks.
   - Another similarity is all of these four types perform feature extraction either for classifying into class categories or localization.

   Differences:
   - Out of all these tasks, instance segmentation is computationally expensive since it has to combine both Semantic Segmentation as well as Object detection to clearly distinguish between two different objects of a same class.
   - Image Classification’s output is just a single label, object detection - bounding box with class and confidence score, semantic segmentation - pixel wise class labels with segmentation in colours, Instance Segmentation - Pixel wise class label segmentation mask in colors and bounding boxes.
   - In classification task we’ll handle with the entire image but in segmentation tasks we have to operate in a pixel level.

ii) Real World Use cases:

1. Image Classification: Medical Image Diagnosis - Pneumothorax Disease Detection
2. Object Detection: Autonomous Vehicles - Detecting road lanes, obstacles and traffic signals
3. Instance Segmentation: Medical Imaging - Brain Tumor Segmentation
4. Semantic Segmentation: Satellite Imagery Analysis - Mapping land area, water bodies and urban community living areas.

2) i/p $3 \times 224 \times 224$

a) i) i/p: $3 \times 224 \times 224 = 150528$ (when this convolution layer is converted to 1D flattened layer no. of input features)

Hidden layer: It will have half the neurons as i/p layer

so, $\frac{150528}{2} = 75264$

1st FC layer: weights: 150528×75264

bias: 75264

Total = weight + bias

$= 150528 \times 75264 + 75264$

2nd FC layer: o/p: 100

weights = 75264×100

biases = 100

Total = weights + biases o/p → 100 classes

$= (75264 \times 100) + 100$

ii) Total Model parameters,

= 1st FC layer + 2nd FC layer

$= (150528 \times 75264) + 75264 + (75264 \times 100) + 100$

b) i)

filter size $= 7 \times 7$ [∵ weights $= 7 \times 7 \times 3 = 147$]

i/p channel = 3

o/p channels = 3

Total parameters = (weights × i/p channel) + o/p channel

$= (147 \times 3) + 3$

$= 441 + 3$

Total parameters = 444

ii)

i/p : 224×224 $(m \times n)$

stride : 1×1 (s_h, s_w)

filter : 7×7 (f_h, f_w)

1st conv. layer

$$o/p = \left(\frac{m - f_h}{s_h} + 1\right) \times \left(\frac{n - f_w}{s_w} + 1\right)$$

$$= \left(\frac{224 - 7}{1} + 1\right) \times \left(\frac{224 - 7}{1} + 1\right)$$

o/p = 218×218

2nd conv layer

o/p $= (218 - 7) + 1 \times (218 - 7) + 1$

$= 212 \times 212$

⋮

As per the Arithmetic progression

WKT $a + (n-1)d = 8$ [∵ since we need to reduce the i/p to $8 \times 8 \times 8$]

$a = 218$

we are reducing '6' for every next layer

so $d = -6$

$218 + (n-1)(-6) = 8$

$218 - 6n + 6 = 8$

$216 = 6n$

$\boxed{n = 36}$

∴ Total 36 convolution layers are needed to reduce to $8 \times 8 \times 8$

iii) Total parameters for FC layer after flattening with o/p (100)

= weights + biases

$= (3 \times 8 \times 8) \times 100 + 100$

$= 19200 + 100$

$= 19300$

iv) Total no. of parameters for the entire architecture

$= \underbrace{(444 \times 36)}_{\text{CONV}} + \underbrace{19300}_{\text{FC}}$

$= 15984 + 19300$

$= 35284$

∴ The no. of parameters in FC layers are more as it will be more complex and computationally expensive compared to that of convolution layers.

Neural Networks & Deep Learning

3.

i/p

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

$m \times n \Rightarrow 4 \times 4$

filter

1	-1	1
0	1	0
1	-1	1

$f \times f \Rightarrow 3 \times 3$

here the stride is 1×1

padding is 'NO PADDING'

The output size will be $= (m-f+1) \times (n-f+1)$

$= (4-3+1) \times (4-3+1)$

$= (2 \times 2)$

Convolution Operation :

1)

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

i/p 4×4

$\circledast$ Convolution operation

1	-1	1
0	1	0
1	-1	1

filter (3×3)

$(2 \times 1) + (0 \times -1) + (-1 \times 1) + (0 \times 0) + (1 \times 1) + (0 \times 0) + (-2 \times 1)$
$+ (0 \times -1) + (1 \times 1)$

$\Rightarrow 2 - 1 + 1 - 2 + 1 = 1$

o/p

1	

2×2

2)

stride $= 1 \times 1$

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

4×4

$\circledast$ Convolution Operation

1	-1	1
0	1	0
1	-1	1

3×3

$(0 \times 1) + (-1 \times -1) + (-1 \times 1) + (1 \times 0) + (0 \times 1) + (2 \times 0)$
$+ (0 \times 1) + (1 \times -1) + (1 \times 1)$

$\Rightarrow 1 - 1 - 1 + 1 = 0$

o/p :

1	0

2×2

3)

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

4×4

$\circledast$ Convolution Operation

1	-1	1
0	1	0
1	-1	1

3×3

$\Rightarrow (0 \times 1) + (1 \times -1) + (0 \times 1) + (-2 \times 0) + (0 \times 1) + (1 \times 0)$
$+ (1 \times 1) + (1 \times -1) + (0 \times 1)$

$\Rightarrow -1 + 1 - 1 = -1$

o/p :

1	0
-1	

2×2

4)

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

4×4

$\circledast$ Convolution Operation

1	-1	1
0	1	0
1	-1	1

3×3

$(1 \times 1) + (0 \times -1) + (2 \times 1) + (0 \times 0) + (1 \times 1) + (1 \times 0)$
$+ (1 \times 1) + (0 \times -1) + (-1 \times 1)$

$\Rightarrow 1 + 2 + 1 + 1 - 1 = 4$

o/p :

1	0
-1	4

2×2

3.b) i/p

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

(4×4)
$(m \times n)$

filter

1	-1	1
0	1	0
1	-1	1

$(f_h \times f_w)$ $\Rightarrow 3 \times 3$

Transposed convolution operation is used as a oversampling approach which will increase the size of the i/p image.

Steps: ① Each and every pixel in the input image should be multiplied with all the pixels in the filter and the output is filled in corresponding positions in the o/p image.

② If overlapping occurs at any location we have to sum up the pixels at the overlapped region.

③ The formula for finding out the size of the output image is

here the stride is 1×1 $(s_h \times s_w)$
padding is '0'

o/p size :

$$[(m-1)s_h + f_h - 2p] \times [(n-1)s_w + f_w - 2p]$$

$$[(4-1)1 + 3 - 2(0)] \times [(4-1)1 + 3 - 2(0)]$$

$$[3+3] \times [3+3]$$

o/p size $\Rightarrow (6, 6)$

Let us initialize the initial matrix as $[0]_{6 \times 6}$ [zero matrix]

$\Rightarrow$ i/p @ $(0, 0) \Rightarrow 2$

$2 \times$

1	-1	1
0	1	0
1	-1	1

$\Rightarrow$

2	-2	2
0	2	0
2	-2	2

o/p :

2	-2	2	0	0	0
0	2	0	0	0	0
2	-2	2	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0

$\Rightarrow$ i/p @ $(0, 1) \Rightarrow 0$

$0 \times$

1	-1	1
0	1	0
1	-1	1

$\Rightarrow$

0	0	0
0	0	0
0	0	0

NO CHANGE

$\Rightarrow$ i/p @ $(0, 2) = -1$

$-1 \times$

1	-1	1
0	1	0
1	-1	1

$\Rightarrow$

-1	1	-1
0	-1	0
-1	1	-1

There is an overlap, so we have to sum up

-1	1	-1
0	-1	0
-1	1	-1

+

2	0	0
0	0	0
2	0	0

=

1	1	-1
0	-1	0
1	1	-1

o/p:

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

$\Rightarrow$ i/p @ $(0, 3) \Rightarrow -1$

$-1 \times$

1	-1	1
0	1	0
1	-1	1

$\Rightarrow$

-1	1	-1
0	-1	0
-1	1	-1

Overlap, so we have to sum up

-1	1	-1
0	-1	0
-1	1	-1

+

1	-1	0
-1	0	0
1	-1	0

$\Rightarrow$

0	0	-1
-1	-1	0
0	0	-1

o/p:

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

i/p @ (1,0) ⇒ 0

0 × filter = $[0]_{3\times3}$

no change

i/p @ (1,1) ⇒ 1

1 × filter =

1	-1	1
0	1	0
1	-1	1

overlapping

1	-1	1
0	1	0
1	-1	1

+

2	0	-1
-2	1	0
0	0	0

⇒

3	-1	0
-2	2	0
1	-1	1

o/p:

2	-2	1	0	0	-1
0	3	-1	0	-1	0
2	-2	2	0	0	-1
0	1	-1	1	0	0
0	0	0	0	0	0
0	0	0	0	0	0

i/p @ (1,2) ⇒ 0

0 × filter = $[0]_{3\times3}$

No change

i/p @ (1,3) ⇒ 2

2 ×

1	-1	1
0	1	0
1	-1	1

=

2	-2	2
0	2	0
2	-2	2

overlapping so sum up

2	-2	2
0	2	0
2	-2	2

+

0	-1	0
0	0	-1
1	0	0

⇒

2	-3	2
0	2	-1
3	-2	2

o/p:

2	-2	1	0	0	-1
0	3	-1	2	-3	2
2	-2	2	0	2	-1
0	1	-1	3	-2	2
0	0	0	0	0	0
0	0	0	0	0	0

i/p @ (2,0) ⇒ -2

-2 ×

1	-1	1
0	1	0
1	-1	1

⇒

-2	2	-2
0	-2	0
-2	2	-2

overlapping so sum up

-2	2	-2
0	-2	0
-2	2	-2

+

2	-2	2
0	1	-1
0	0	0

⇒

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

o/p:

2	-2	1	0	0	-1
0	3	-1	2	-3	2
0	0	0	0	2	-1
0	-1	-1	3	-2	2
-2	2	-2	0	0	0
0	0	0	0	0	0

i/p @ (2,1) ⇒ 0

0 × filter = $[0]_{3\times3}$

No change

i/p @ (2,2) = 1

1 × filter =

1	-1	1
0	1	0
1	-1	1

overlapping so sum up

1	-1	1
0	1	0
1	-1	1

+

0	0	2
-1	3	-2
-2	0	0

⇒

1	-1	3
-1	4	-2
-1	-1	1

o/p:

2	-2	1	0	0	-1
0	3	-1	2	-3	2
0	0	1	-1	3	-1
0	-1	-1	4	-2	2
-2	2	-1	-1	1	0
0	0	0	0	0	0

i/p @ (2,3) → 1

1 × filter =

1	-1	1
0	1	0
1	-1	1

overlapping so sum up

1	-1	1
0	1	0
1	-1	1

+

-1	2	-1
4	-2	2
-1	1	0

⇒

0	2	0
4	-1	2
0	0	1

o/p:

2	-2	1	0	0	-1
0	3	-1	2	-3	2
0	0	1	0	2	0
0	-1	-1	4	-1	2
-2	2	-1	0	0	1
0	0	0	0	0	0

i/p @ (3,0) ⇒ 1

1 × filter =

1	-1	1
0	1	0
1	-1	1

overlapping so sum up

1	-1	1
0	1	0
1	-1	1

+

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

⇒

1	-2	0
-2	3	-1
1	-1	1

o/p:

2	-2	1	0	0	-1
0	3	-1	2	-3	2
0	0	1	0	2	0
1	-2	0	4	-1	2
-2	3	-1	0	0	1
1	-1	1	0	0	0

i/p @ (3,1) ⇒ 1

1 × filter =

1	-1	1
0	1	0
1	-1	1

overlapping so sum up

1	-1	1
0	1	0
1	-1	1

+

-2	0	4
3	-1	0
-1	1	0

⇒

-1	-1	5
3	0	0
0	0	1

O/p:

2	-2	1	0	0	-1
0	3	-1	2	-3	2
0	0	1	0	2	0
1	-1	-1	5	-1	2
-2	3	0	0	0	1
1	0	0	1	0	0

i/p (3,2) ⇒ 0

0 × filter = $[0]_{3\times3}$

no change

i/p (3,3) ⇒ -1

-1 ×

1	-1	1
0	1	0
1	-1	1

=

-1	1	-1
0	-1	0
-1	1	-1

overlap, so we have to sum up

-1	1	-1
0	-1	0
-1	1	-1

+

5	-1	2
0	0	1
1	0	0

⇒

4	0	1
0	-1	1
0	1	-1

O/p:

2	-2	1	0	0	-1
0	3	-1	2	-3	2
0	0	1	0	2	0
1	-1	-1	4	0	1
-2	3	0	0	-1	1
1	0	0	0	1	-1

∴ find O/p image after transposed convolution operation

4)

i/p : 3 x 224 x 224

latent representation: 256 x 16 x 16

final o/p : 3 x 224 x 224

filter : 3 x 3

Channels : [3, 16, 32, 64, 128, 256, 256, 256 ...]

256 channels until reaching the desired latent representation of 256 x 16 x 16

No. of parameters per filter are

for layer 1: [3 → 16]

$(3 \times 3 \times 3 + 1) \times 16 = 448$

o/p size $= (224 - 3) + 1 = 222$

for layer 2: [16 → 32]

$(3 \times 3 \times 16 + 1) \times 32 = 4640$ o/p size = 220

for layer 3: [32 → 64]

$(3 \times 3 \times 32 + 1) \times 64 = 18496$ o/p size = 218

for layer 4: [64 → 128]

$(3 \times 3 \times 64 + 1) \times 128 = 73856$ o/p size = 216

for layer 5: [128 → 256]

$[3 \times 3 \times 128 + 1] \times 256 = 295168$ o/p size = 214

for layer 6: [256 → 256]

$[3 \times 3 \times 256 + 1] \times 256 = 590080$ o/p size = 21

o/p size

⋮

arithmetic progression

$a_n = a_1 + (n-1)d$

$a_n = 16$ (last)

$a_1 = 224$ (first)

$n = \frac{\text{first} - \text{last}}{d} = \frac{224 - 16}{2} = 104$

number of parameters in downsampling $= 448 + 4640 + 18496 + 73856 + 295168 + 590080 + (104 - 5)(590080)$

$= 58810528$

Number of parameters in upsampling with filter 9x9

here 16 → 224 i.e. $(224 - 9) + 1 = 216$

so here size increase by '8' in each step.

so as per arithmetic progression

$n = \frac{224 - 16}{8} = \frac{208}{8}$ (208 → 26)

$\boxed{n = 26}$

for the layers upto 5th layer:

the parameters $= (9 \times 9 \times 256 + 1)(256)$

$= 5308672$

for the layer 5

$(9 \times 9 \times 256 + 1)\, 128 = 2654336$

for the layer 4

$(9 \times 9 \times 64 + 1)\, 64 = 663616$

for the layer 3

$(9 \times 9 \times 32 + 1)\, 32 = 165920$

for the layer 2

$(9 \times 9 \times 16 + 1)\, 16 = 41488$

for the layer 1

$(9 \times 9 \times 16 + 1)\, 3 = 3891$

no of parameters in upsampling $= 2654336 + 663616 + 165920 + 41488 + 3891 + (26 - 1)(5308672)$

$= 115011363$

Total number of parameters = parameters in downsampling + parameters in upsampling

$= 58810528 + 115011363$

$= 173821891$