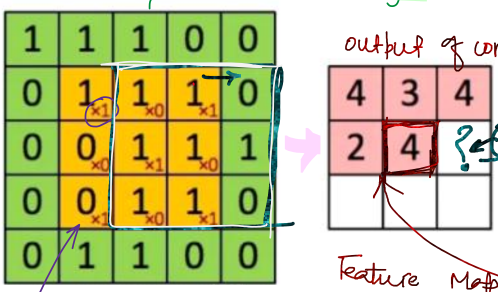
CNN Model Architecture Contd.

15 November 2025 10:14

How does convolution work?

Raw input image



output of convolution operation → Feature Map.

↓

(2D-array)

$(1\times1 + \cancel{1\times0} + \cancel{0\times1}) + (\cancel{1\times0} + 1\times1 + \cancel{1\times0}) + (1\times1 + \cancel{1\times0} + \cancel{0\times1}) = 1+1+1 = 3$

3

Feature Map

Filter aka Kernel

1	0	1
0	1	0
1	0	1

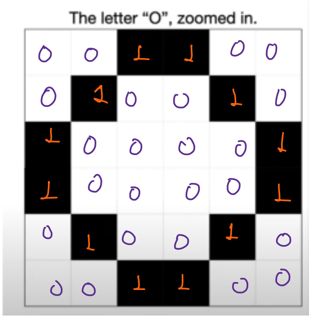
Filtering Calculation

$(1\times1 + \cancel{1\times0}^{0} + 1\times1) + (\cancel{0\times0}^{0} + 1\times1 + \cancel{1\times0}^{0}) + (\cancel{0\times1}^{0} + \cancel{1\times0}^{0} + 1\times1)$

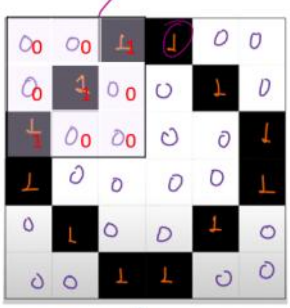
$= (1+0+1) + (0+1+0) + (0+0+1)$

$= 2+1+1 = 4$

Tic-Tac-Toe



Filtering / Convolution operation



0	0	1
0	1	0
1	0	0

=

Dot product operation

$(0\times0 + 0\times0 + 1\times1) + (0\times0 + 1\times1 + 0\times0) + (1\times1 + 0\times0 + 0\times0) = 3$

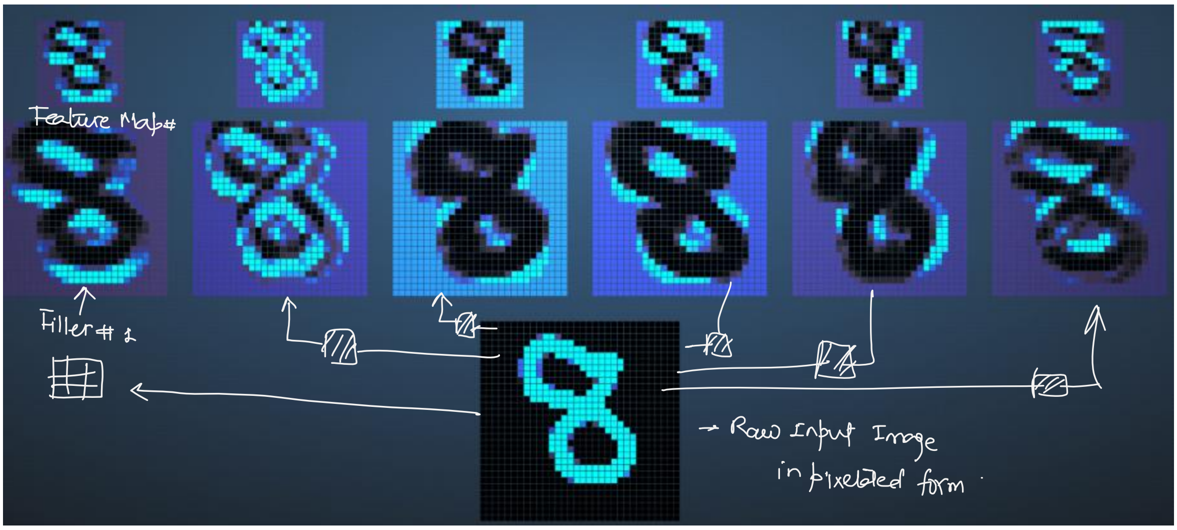
↓

$\boxed{\text{Bias: } -2}$

$3-2=1$

Feature Map.

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



CONVOLUTION / Filtering operation

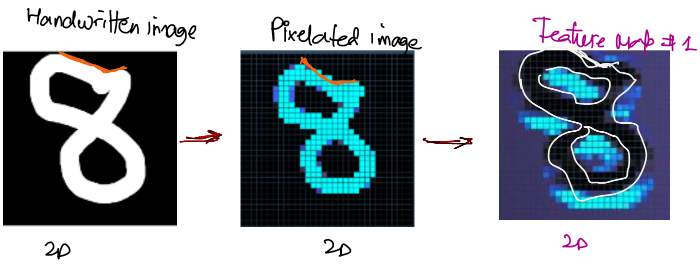
In simple terms, convolution is like sliding a small filter aka kernel over an input image → so basically a mathematical operation [dot product → sumproduct of cell values overlapped with filter values]

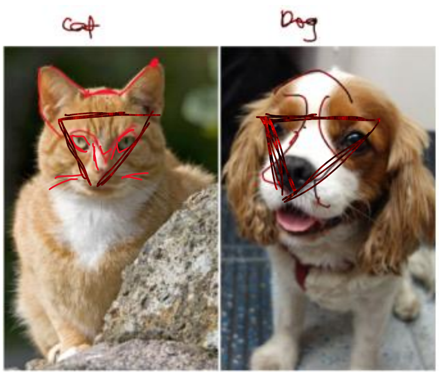
↓

[to extract specific features such as edges, fine lines, strokes, corners, curves etc.]

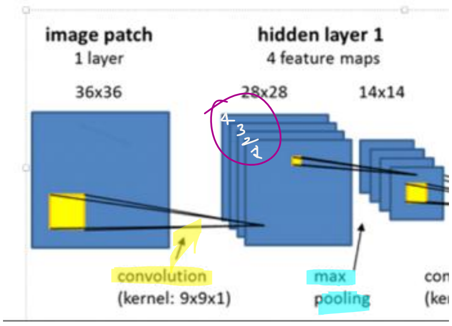
↓

to form shapes → patterns





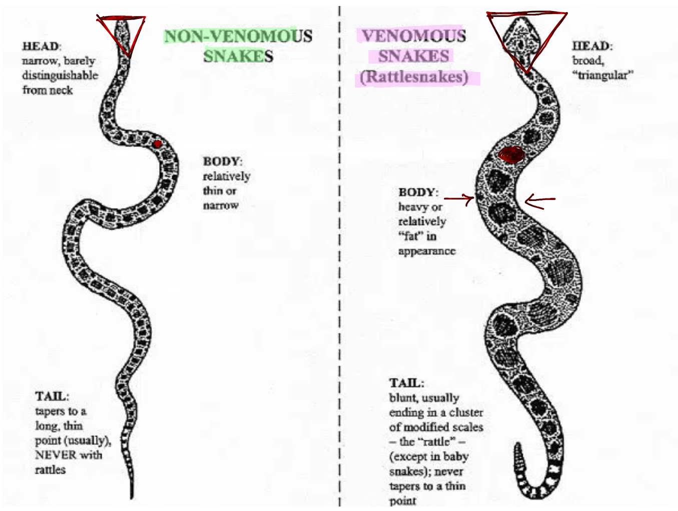
→ convolution



Note: convolutional layer applies $\overbrace{\text{a set of filters or kernels}}^{\text{4 filters}}$ to an input image and filters slide over the image and generate $\underbrace{\text{a set of feature maps}}_{\text{4 feature maps}}$ as the output.

CNN leverages convolution to:

- preserve the spatial relationship in the image
- efficiently learn patterns from the local regions of the image
- reduce the computational load compared to ANNs.



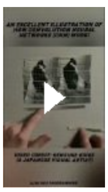
Intuitive Idea behind CNN

↳ convolution still with downsampling.

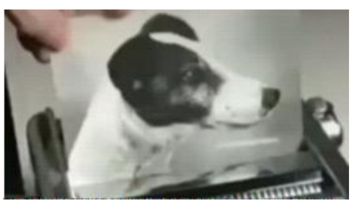
Original Image

[An excellent illustration of how CNN work! #artificialintelligence #deeplearning](https://www.youtube.com/shorts/N6NBT-n9mmo)

https://www.youtube.com/shorts/N6NBT-n9mmo

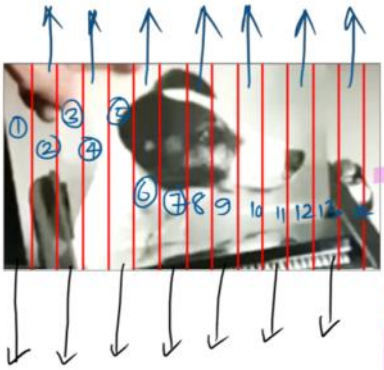


Original Image



2. cut the original dogs image vertically into equal strips

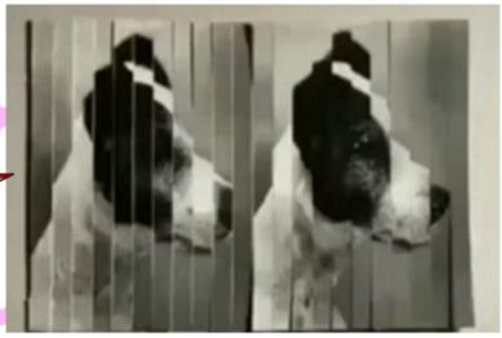
1 dog



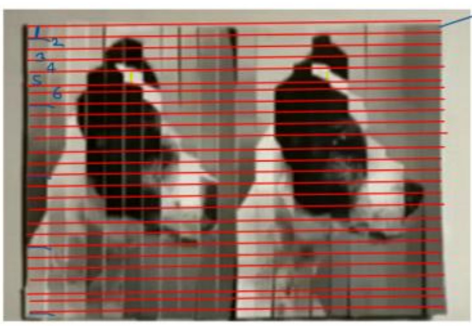
1 3 5 7 9 11 13

2 4 6 8 10 12 14

2 dogs



③ cut the image on the right horizontally into equal strips



(2) (2)

1 2

3 4

5 6

7 8

9 10

11 12

13 14

4 dogs

$\frac{1}{4}$th $\frac{1}{4}$th $\frac{1}{4}$th $\frac{1}{4}$th

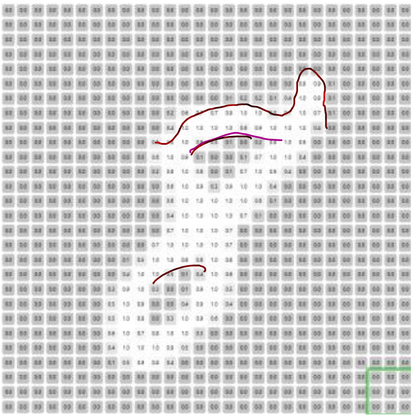
4 dogs

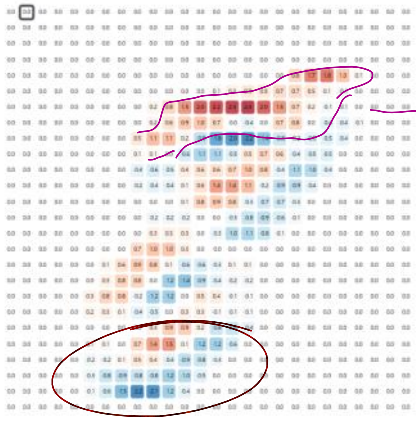
categorized / classified as a

DOG

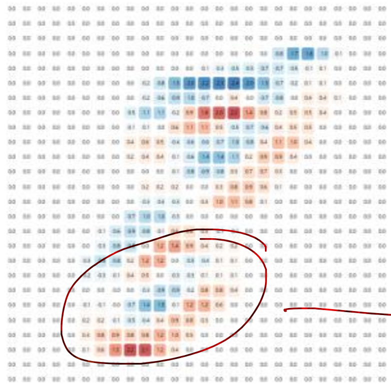
https://deeplizard.com/resource/pavq7noze2

[Mathematical Interpretation of filtering operation]

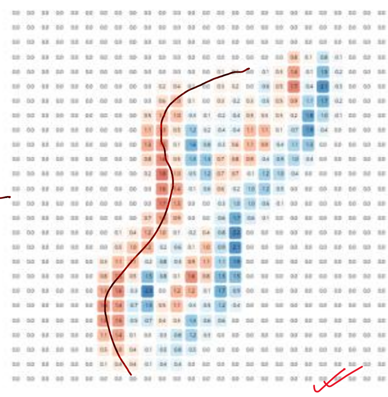




→ Top Edge getting captur



→ Bottom ~~Edge~~ getting captured ,,



left edge getting captured ←

'Master Oogway'

→

→ Past.

Future

Present

+ (Phone has to be in silent) → [Push calls] →

~~Flat → BLR~~

~~2 crores~~

~~3yrs~~

~~DS world.~~

~~(26 unmarried)~~

Future

[exceptions ↓ allow → calls → filtering important]

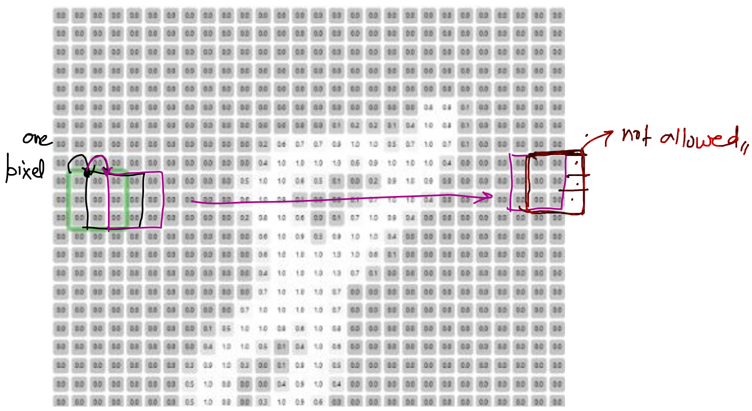
*** Pro-tip

***Pro-tip

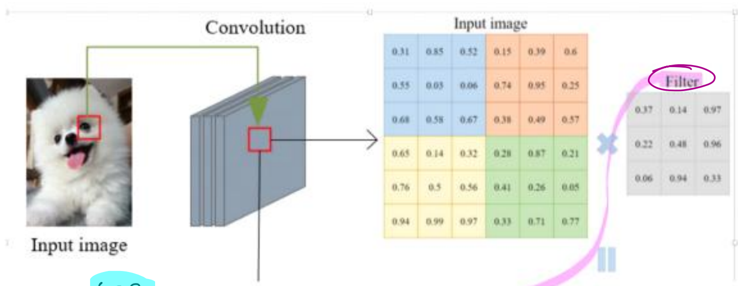
STRIDE & PADDING

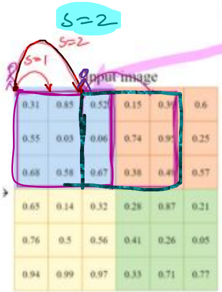
Stride: Stride is a hyperparameter in CNN that determines the no. of pixels by which the filter (aka kernel) moves across the input image during the convolution operation.

Default setting S = 1 → Filter moves one pixel at a time



Stride # S = 2 Filter jumps by 2 pixels at a time → reducing the overlap and hence the filter map size





Why stride matters?

- Small (S = 1) → High resolution output with more details captured at the granular level → computation would be slower.
- Large (S = 2 or more) 2, 3 or 4 → small resolution output → feature maps' size would be smaller → Better downsampling ↓

It is faster in computation

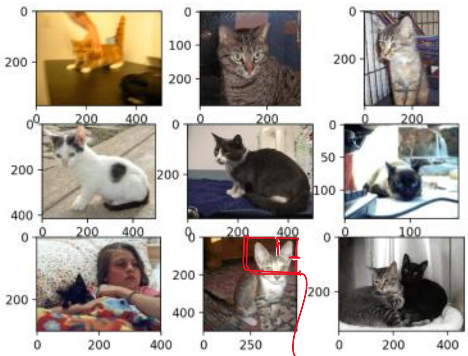
— may loose fine details of the input image

* For any small input images (upto 100 x 100 pixels) → stride # 1 is the most recommended.

* In some cases or large input images (upto 224 x 224 pixels) → stride # 2 or more can also be used but less frequently and generally in deeper layers to reduce the feature size for computational efficiency while not losing on essential features of the image.

Padding

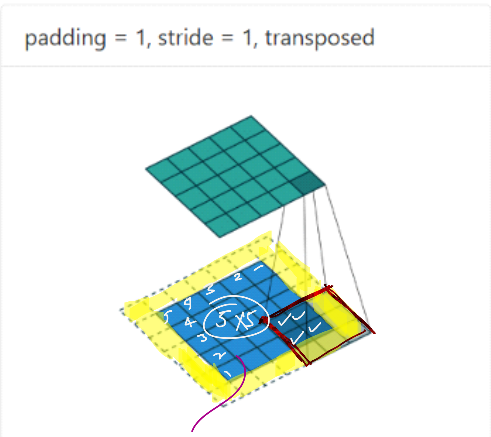
— margin / some room → to the images



In such cases, it is evident that padding is needed.

Padding is a technique used in CNNs to adjust the spatial dimensions of the input image

- it involves adding extra row(s) and/or column(s) pixels usually with zero around the border of the input image



Raw input size: 5x5 pixels

Before padding

p=0 and s=1

size of the feature map??

Formula for feature map dimensions

Height of feature map

$$H_{out} = \left\lfloor \frac{H_{in} - K + 2P}{S} \right\rfloor + 1$$

where $\lfloor \cdot \rfloor \rightarrow$ Greatest Integer Function (Floor)

$\lfloor 2.3 \rfloor \rightarrow 2$

$\lfloor 3.6 \rfloor \rightarrow 3$

Width of the feature map

$$W_{out} = \left\lfloor \frac{W_{in} - K + 2P}{S} \right\rfloor + 1$$

where

- **Input dimensions**

  H_{in} : Height of the input image

  W_{in} : Width —"— —"— —

- **Filter parameters**

  K or f : size of the filter (assuming square matrix of order K)

  S : Stride of the filter

  P : Padding for the filter

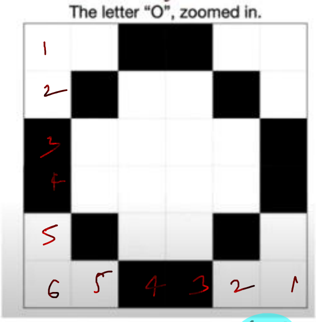
- **No. of filters**

  F : no. of filter applied → which determines the depth of the output

Stride = 1 // Padding = 0 // Filter size = 3

Input Image

The letter "O", zoomed in.



6x6

$\odot$

→ Filter ←

0	0	1
0	1	0
1	0	0

3x3

Feature Map

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

4x4

$H_{in} = 6$

$W_{in} = 6$

$K = 3$ or $f = 3$

$$H_{out} = \left\lfloor \frac{H_{in} - K + 2P}{S} \right\rfloor + 1$$

$$= \left\lfloor \frac{6 - 3 + 2 \times 0}{1} \right\rfloor + 1 = 4$$

$$W_{out} = \left\lfloor \frac{W_{in} - K + 2P}{S} \right\rfloor + 1 = 4$$

Feature Map dimensions : 4x4

MNIST Use-case

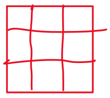
28x28

28px

8 (8)

28px

x

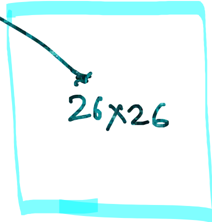


3x3

K=3

or

f=3

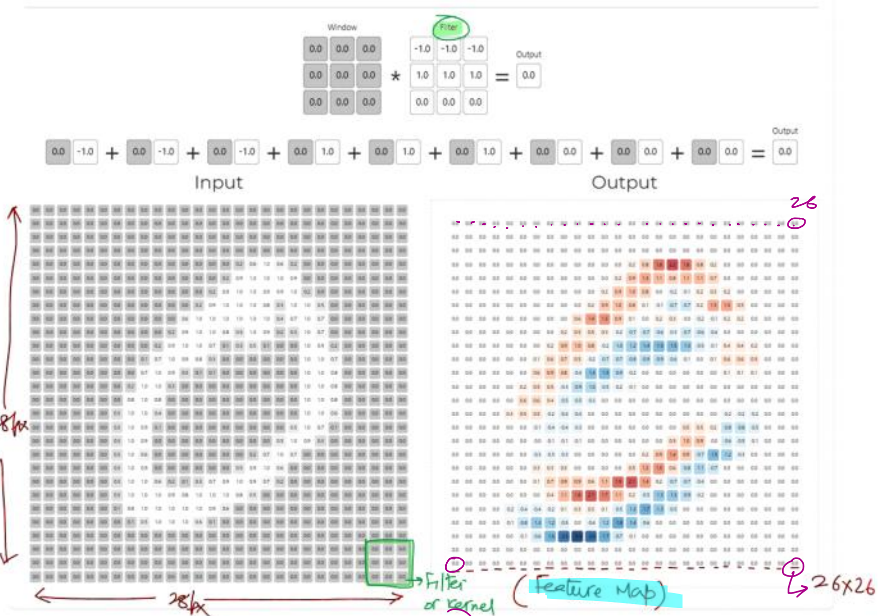


26x26

Assume s=1 and p=0,

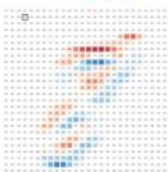
$$H_{out} = \left\lfloor \frac{28-3+2\times 0}{1} \right\rfloor + 1 = 26$$

$$W_{out} = 26$$



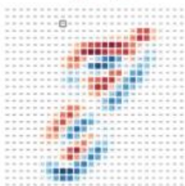
https://deeplizard.com/resource/pavq7noze2

Filter #1



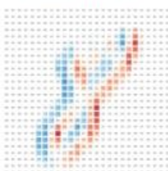
26x26

Filter #2



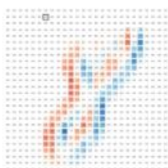
26x26

Filter #3



26x26

Filter #4



26x26

Why is padding so important?

TASK #4