# CNN (Convolutional Neural Network)

1. Internal working (Mathematics)
2. Implementation

MNIST Data (Images of numbers (0-9)) — 28 × 28

→ Multiclass classification

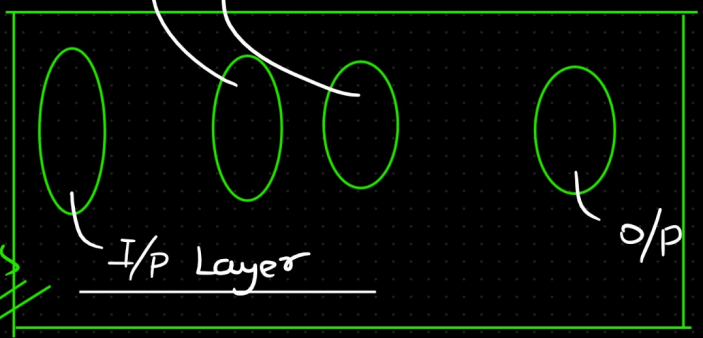
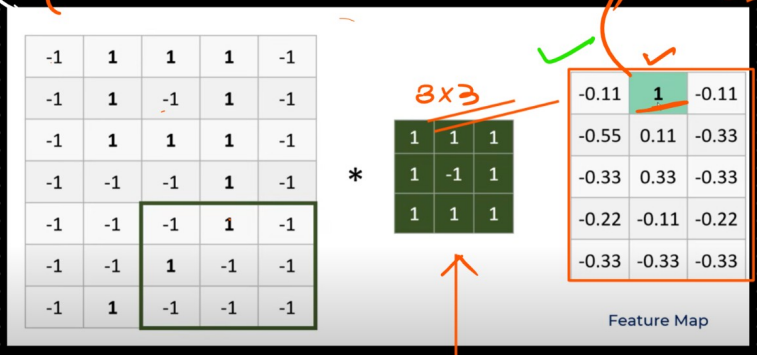


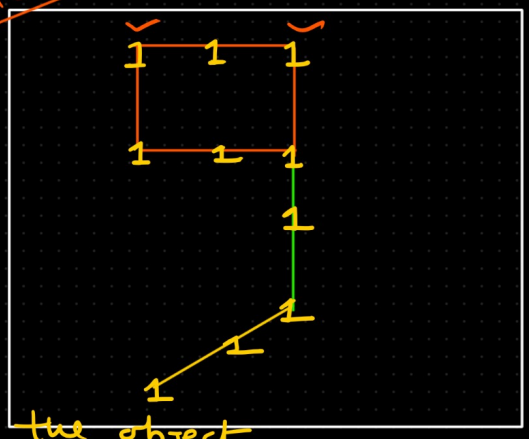

Image → Pixel (Matrices)

Input Image




Feature detection

| | | |
|---|---|---|
| -0.11 | 1 | -0.11 |
| -0.55 | 0.11 | -0.33 |
| -0.33 | 0.33 | -0.33 |
| -0.22 | -0.11 | -0.22 |
| -0.33 | -0.33 | -0.33 |



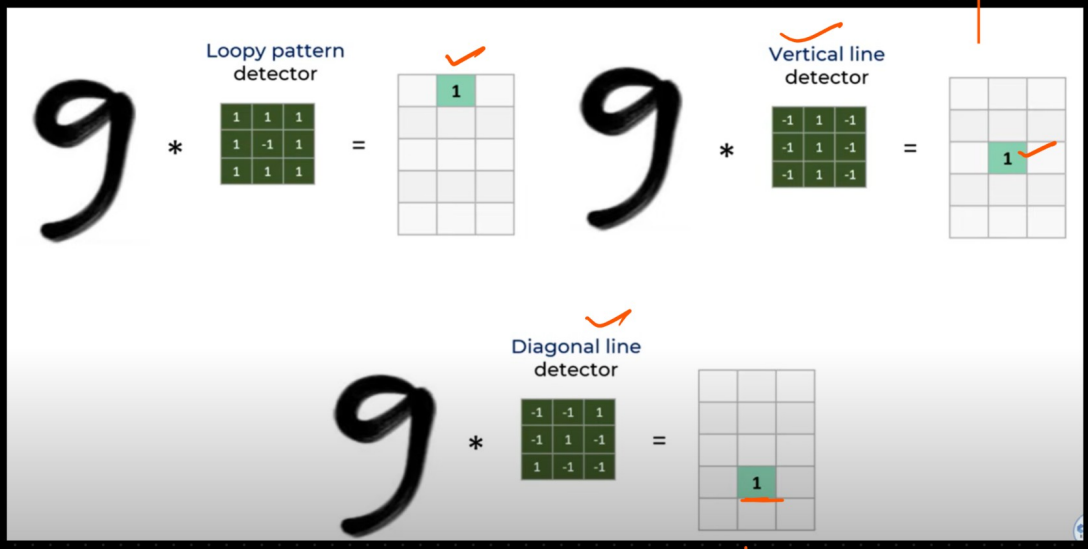
→ detect the object

Filters / Kernel → detect & extract two features

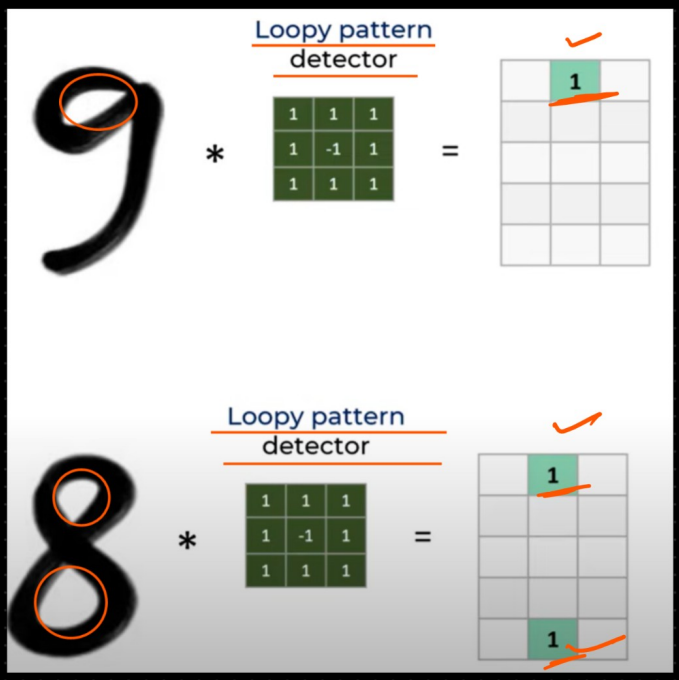
$-1*1 + 1*1 + 1*1 + -1*1 + 1*-1 + -1*1 + -1*1 + 1*1 + 1*1 \Rightarrow -1 \rightarrow -1/9 = -0.11$

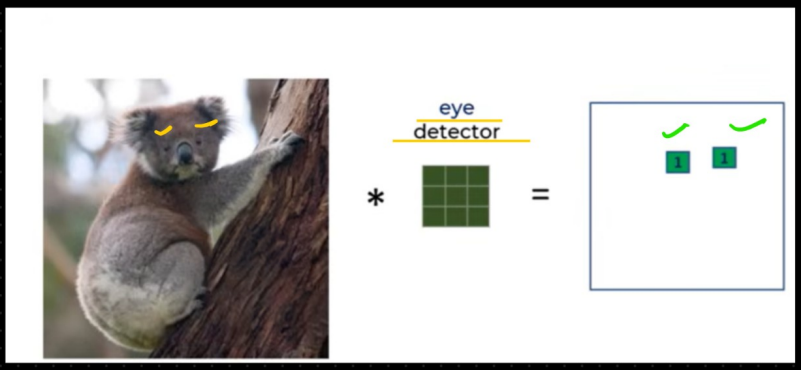
Project → size of all images in a given dataset is same or not

- Same
- different
  - Standard size
    - fix one size




feature map


Loopy pattern
detector
1 1 1
1 -1 1
1 1 1
*
=
1
Loopy pattern
detector
1 1 1
1 -1 1
1 1 1
*
=
1
1


eye
detector
*
=
1
1

Input Image = 5x5

Kernel size = 3x3

Stride = 1

Paddig = 0

Interview question

Output Image ⇒ (3x3)

$$\frac{\text{Input Size} - \text{Kernel size} + 2 * \text{Paddig}}{\text{Stride}} + 1$$

$$\frac{5 - 3 + 2*0}{1} + 1$$

$$2 + 1 = 3$$

Input Image → 10x10

Kernel → 3x3

Stride → 2

$$\text{Output} \longrightarrow \left\lfloor \frac{10 - 3 + 0}{2} \right\rfloor + 1$$

$$\longrightarrow 4 \times 4$$

Padding → 1

↳ Input & output shape exactly same

Input → 5 x 5

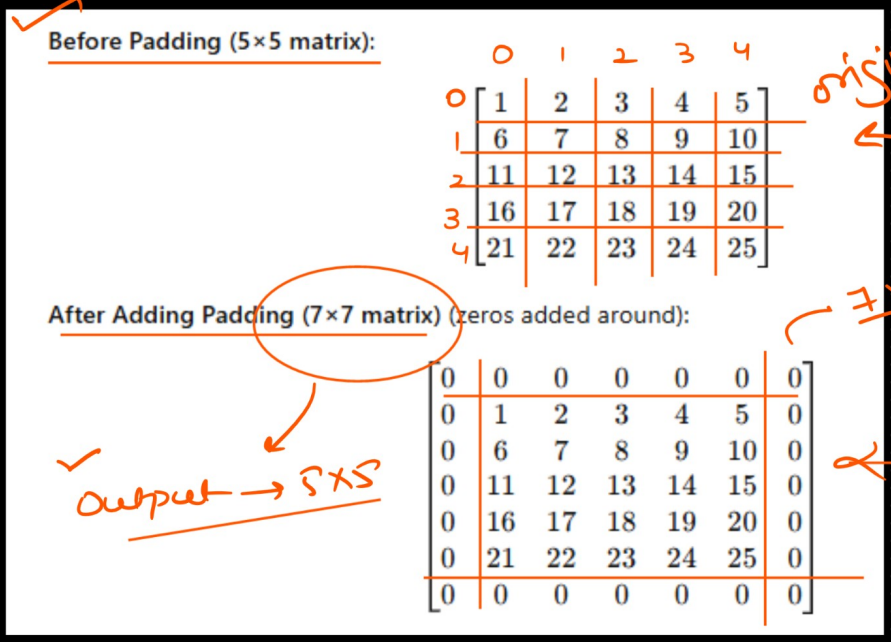
Kernel → 3 x 3

Stride → 1

Output Image → $\frac{5 - 3 + 2 * 1}{1} + 1$

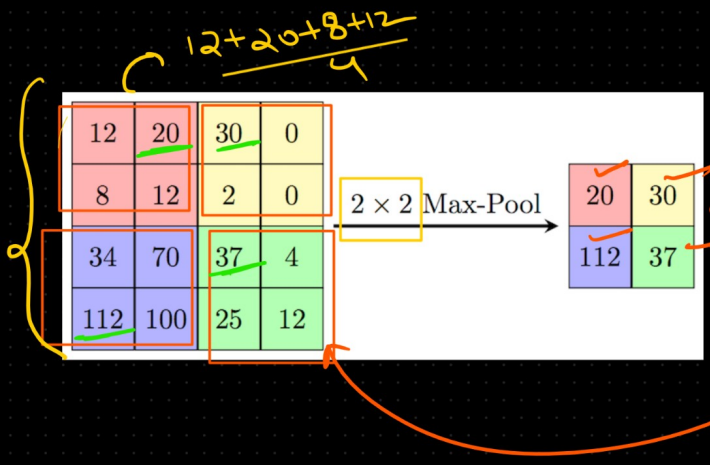
$4 + 1 = 5$

Output → 5 x 5




→ No Padding → Input image > { Output image

→ 1 Padding → Input image = Output image



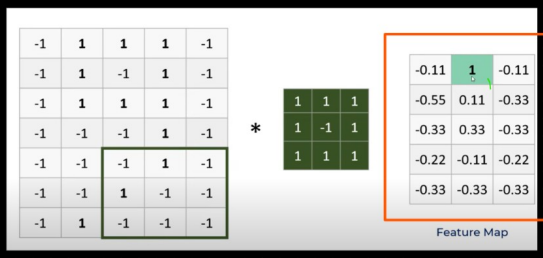

output (1. Reduce Computational cost)

(2. Essential features)

Neural Network

Step 2

I/p




flatten()

-0.11
1
-0.11
-0.55
0.11
-0.33
-0.33

Hidden layers 128, relu

output

10

Softmax

9 → 0.90

target = 9

updating weights & biases

loss function

Categorical_crossentropy

Step1

CNN

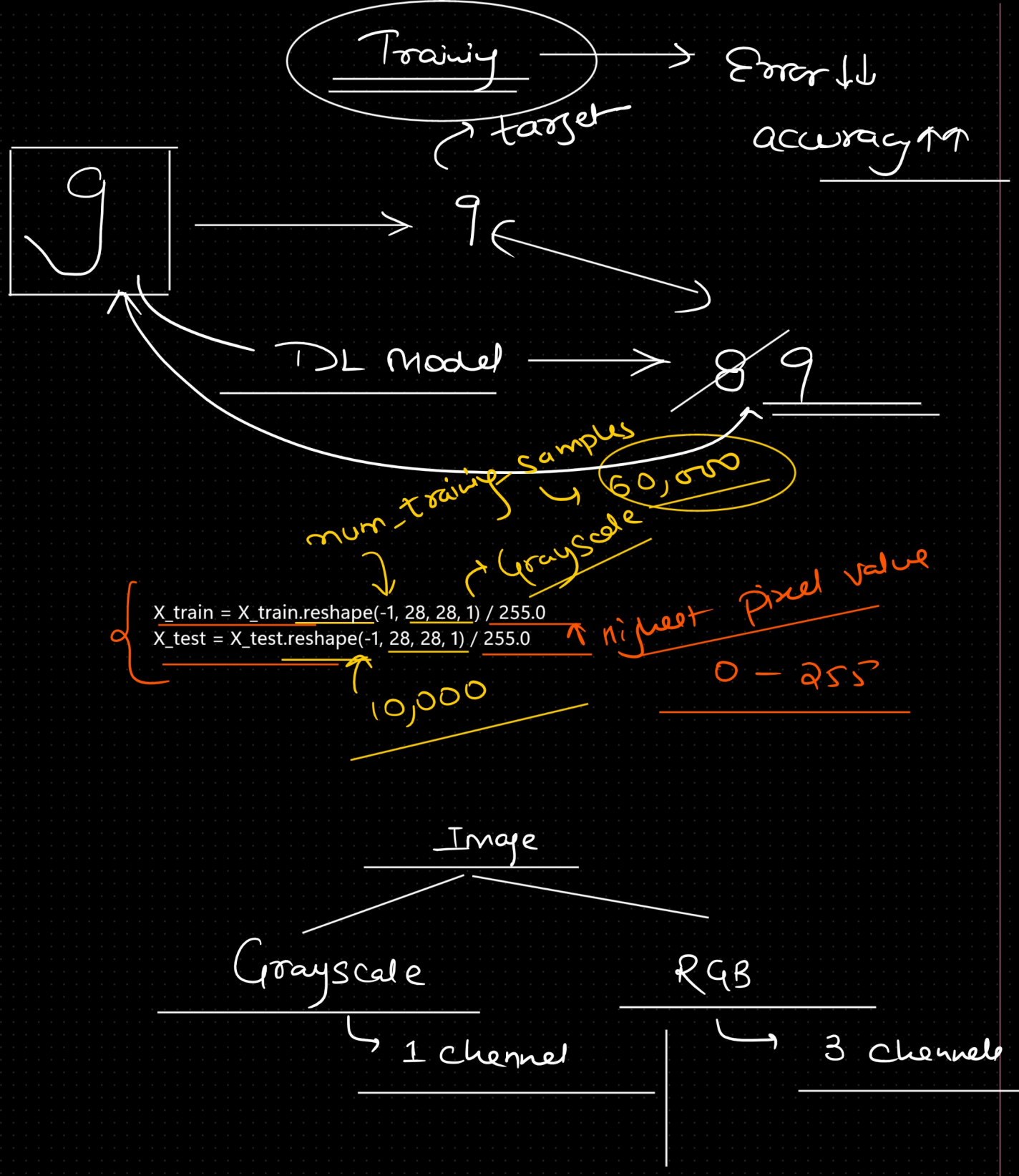
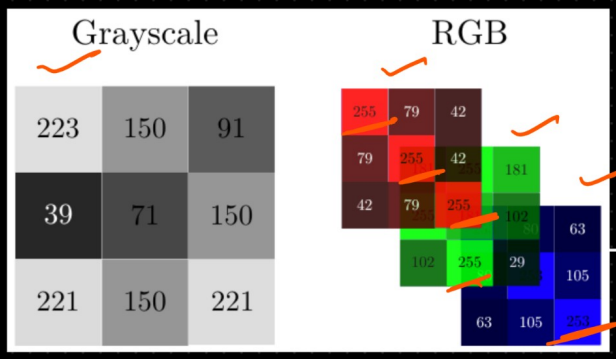

Training
Error ↓↓
accuracy ↑↑
target
9
9
DL Model
8 9
num_training_samples
60,000
Grayscale
X_train = X_train.reshape(-1, 28, 28, 1) / 255.0
X_test = X_test.reshape(-1, 28, 28, 1) / 255.0
10,000
highest Pixel value
0 - 255


Image
Grayscale
1 channel
RGB
3 channels




original data

Data Preprocessing

RGB

Grayscale

most common practice