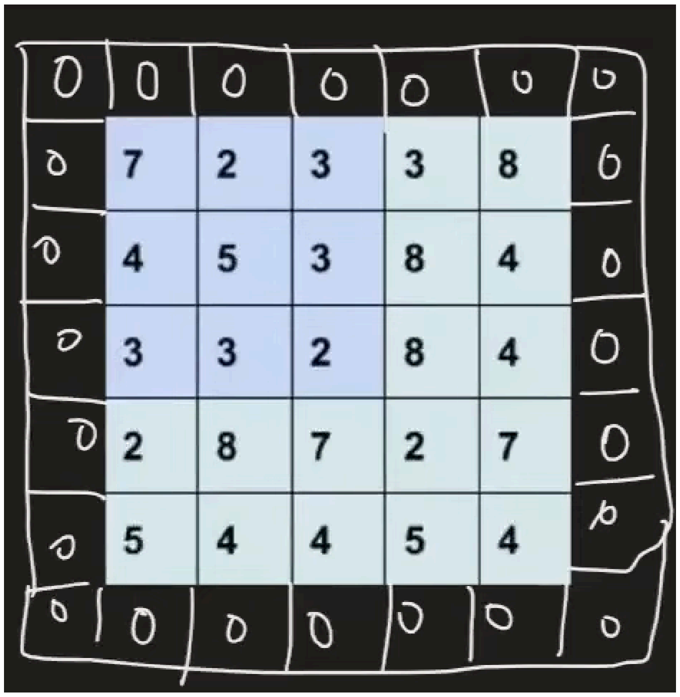
Padding & Strides in CNN

- Critical hyperparameters in CNN

- **Padding** = Adding extra pixels (usually zeros) around the input image to **control the size** of the output.
- **Stride** = Number of pixels the filter **moves (or "jumps")** each time it slides over the input.

🔵 A. Padding: Adding Extra Pixels Around Image



0	0	0	0	0	0	0
0	7	2	3	3	8	6
0	4	5	3	8	4	0
0	3	3	2	8	4	0
0	2	8	7	2	7	0
0	5	4	4	5	4	p
0	0	0	0	0	0	0

❓ Why Padding?

When a filter moves over the image, it only works where the filter **fully fits** inside the image. So:

- Output size **shrinks**.
- Edge pixels get **less attention**.

Padding helps:

- **Preserve size** of input.
- **Capture edge features** better.

🧱 How Padding Works?

Let's say you have:

Input Image (5×5):

`padding="valid"` **(Default)**

$$\text{Output Size} = \left\lfloor \frac{\text{Input Size} - \text{Kernel Size}}{\text{Stride}} \right\rfloor + 1$$

```
[1 2 3 4 5
 6 7 8 9 10
 11 12 13 14 15
 16 17 18 19 20
 21 22 23 24 25]
```

Apply padding = 1, it becomes:

`padding="same"`: 👇

```
[0 0 0 0 0 0 0
 0 1 2 3 4 5 0
 0 6 7 8 9 10 0
 0 11 12 13 14 15 0
 0 16 17 18 19 20 0
 0 21 22 23 24 25 0
 0 0 0 0 0 0 0]
```

- This keeps the **output size same** after convolution.

$$\text{Output Size} = \left\lfloor \frac{\text{Input Size}}{\text{Stride}} \right\rfloor$$

Padding in Keras:

```
model = Sequential()

model.add(Conv2D(32,kernel_size=(3,3),padding='same', activation='rel
u', input_shape=(28,28,1)))
model.add(Conv2D(32,kernel_size=(3,3),padding='same', activation='rel
u'))
model.add(Conv2D(32,kernel_size=(3,3),padding='same', activation='rel
u'))

model.add(Flatten())

model.add(Dense(128,activation='relu'))
model.add(Dense(10,activation='softmax'))
```

🔴 B. Stride: How Much the Filter Moves Each Step

- By default, a filter moves **1 pixel at a time** (stride = 1).

❓ Why Stride?

- When we only want high level features
- We want to make the output smaller.
- Lower computing power
- We want to skip some positions for **speed** or **feature reduction**.

🧮 How Stride Works?

Example:

- Input: 6×6
- Filter: 3×3
- Stride = 1 → output = 4×4
- Stride = 2 → output = 2×2

Stride in Keras:

```
model = Sequential()

model.add(Conv2D(32,kernel_size=(3,3),padding='same',strides=(2,2), activa
tion='relu', input_shape=(28,28,1)))
model.add(Conv2D(32,kernel_size=(3,3),padding='same',strides=(2,2), activa
tion='relu'))
model.add(Conv2D(32,kernel_size=(3,3),padding='same',strides=(2,2), activa
tion='relu'))

model.add(Flatten())

model.add(Dense(128,activation='relu'))
model.add(Dense(10,activation='softmax'))
```

`strides=(2,2)` → **Horizontal movement = 2 pixels, vertical movement=2 pixels**

🧠 Trivia & Insights

- Large padding + stride = keeps model shallow but efficient.
- **Strided convolutions** can be used **instead of pooling**.
- Padding is like giving the image **"invisible margins"** so filters can slide fully even at the edge.
- Stride is like telling the filter: **"jump ahead this many pixels"**.