Padding & Strides in CNN

In CNN, padding & strides are 2 important hyperparameters that affect the

Behavior of the convolution operation.

Understanding their intuition is crucial for grasping how CNNs process data and how these parameters influence the output size of each layer.

Why is Padding Required ?

$$\begin{bmatrix} 7 & 2 & 3 & 3 & 8 \\ 4 & 5 & 3 & 8 & 4 \\ 3 & 3 & 2 & 8 & 4 \\ 2 & 8 & 7 & 2 & 7 \\ 5 & 4 & 4 & 5 & 4 \end{bmatrix} * \begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix} = \begin{bmatrix} 6 & -9 & -8 \\ -3 & -2 & \\ & & \end{bmatrix}$$

Problem:

Filter ko Image pr Jbh Convolve krrhe then humme jo Filter Map millraha vo Image se chota hai

Border ya Edge Vale Pixel Kamm Convolution ke Part hai but jo Center vale hai vo bhaut saare Convolution ke part hai as Filter (Kernel) zyaada move horaha unkepass se

So Jo Feature Map Banega beech vale Pixel ka zyaada information millega,

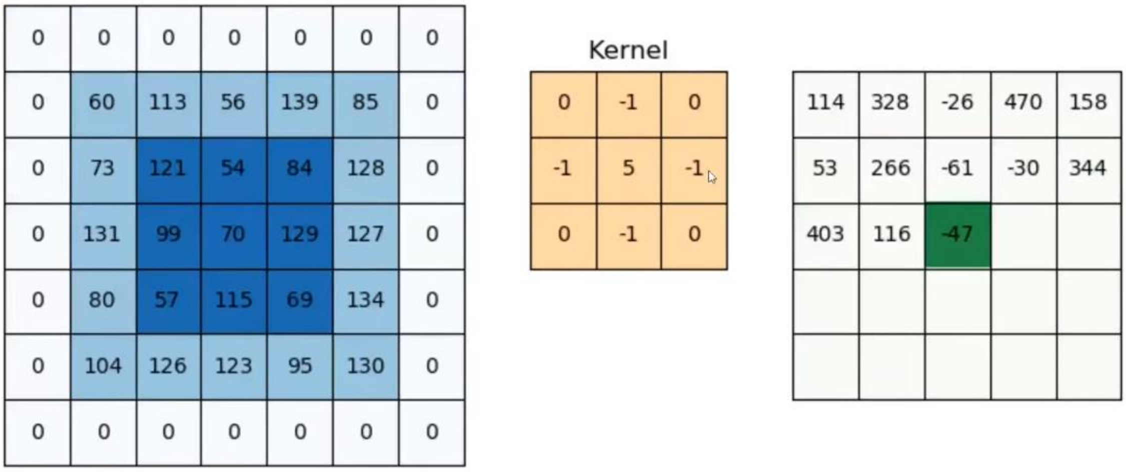
So agar information Side mai border side information zyaada relvent hoga so vo humme nahi millega so vahi issue aayega humme

Padding:

In Padding we change the Image Size

We change the image Top, Bottom , Right Left we add one row & column

Zero-Padding



As we can see:

Input Image == Output Image

Keras mai 2 Padding option hota hai

1.Valid -> iska mtlb padding nahi hoga

2.Same -> keras will automatically findout ki kitna padding dena hai

---

Padding refers to adding additional layers of pixel around the input image before applying convolution operation.

The purpose of padding is primarily to control spatial dimensions of the output volume.

Padding helps to preserve spatial dimensions of input volume so that output volume hqas same spatial dimensions.

Padding is commonly expressed in terms of number of pixels added to each dimension.

If the padding is ‘p’ , it means ‘p’ pixels are added to each side of input volume.

Intuition:

Padding allows convolution operation to Capture More information from the input volumes Edges & Corners.

It helps in reducing the problem of Shrinking or Reduced Output Volumes , which can when applying successive convolutional layers without padding, specially in Deep Networks.

Padding Ensures that spatial or occupied dimensions are not significantly reduced after each convolution operation, which can be crucial for preserving information throughout network.

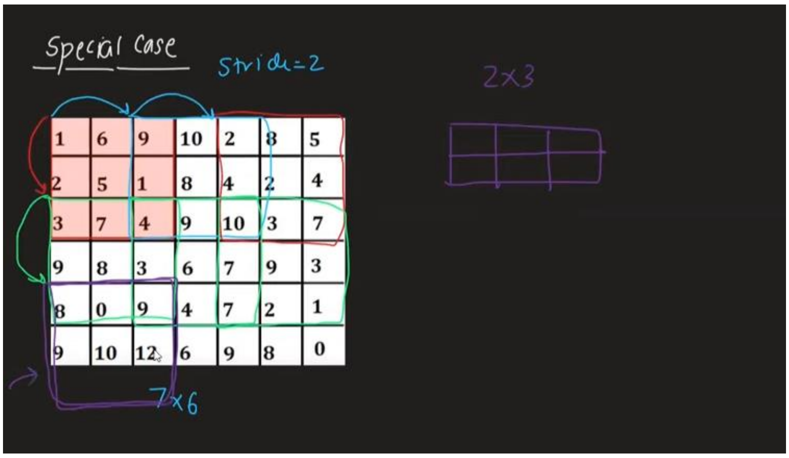
Strides

Stride refers to number of pixels by which convolutional filter is moved across input volume.

In a typical convolution operation, filter moves 1 Pixel at a time, But strides allow it to move more than 1 Pixel at Each Step.

Strides Determine how much the filter shifts or Slides over the input volume.

Strides can be different along different dimensions that is horizontal and vertical can be different.



Intuition

Larger stride value lead to a reduction in spatial dimension or occupied dimensions of output volume.

Using larger strides effectively reduces resolution of output volume

Strides can be used to control computational complexity of network by reducing number of computations required, as larger strides reduce number of operations performed in each layer.

Strides aajke time itna bhi use nahi hote

Why Strides?

1.) When you only want High Level Features not Low Level Features
2.) Computing Power (Not that important in today's time)

Stride aaj ke time utne important nahi hai jitne pehle theah ,

But ya kuch specific problems mai use hote hai.

Padding in Convolutional Neural Networks (CNNs) solves several problems related to the dimensions of feature maps and the preservation of spatial information throughout the network. Here are some key problems that padding helps to address:

1. **Preventing Information Loss at Edges**:
    - Without padding, as the filter slides across the input image, the spatial dimensions of the feature maps decrease with each convolution operation.
    - This reduction in dimensions can lead to information loss, especially at the edges of the input image, where the filter might not cover the entire receptive field.
    - Padding ensures that the convolution operation captures information from the entire input image, including the edges, by adding additional pixels around the input.
2. **Preserving Spatial Dimensions**:

- Padding allows the spatial dimensions of the input and output feature maps to remain the same.
- This preservation of spatial dimensions can be crucial, especially in deep CNN architectures where multiple convolutional layers are stacked.
- Preserving spatial dimensions helps maintain the network's ability to capture spatial hierarchies of features across multiple layers.

3. **Enabling Symmetry in Convolutional Operations**:
    - Padding ensures that the convolution operation is symmetric around the central pixels of the input.
    - This symmetry can be important for certain tasks, such as image processing, where preserving spatial relationships is essential.
4. **Facilitating Design Flexibility**:
    - Padding provides flexibility in designing CNN architectures by allowing researchers to control the spatial dimensions of feature maps.
    - It enables the use of smaller convolutional filters while maintaining the spatial dimensions of the input and output feature maps.
    - With padding, designers can choose from various architectures and hyperparameters to optimize performance for specific tasks.

Overall, padding in CNNs addresses issues related to information loss at the edges of input images, preserves spatial dimensions throughout the network, enables symmetry in convolutional operations, and enhances design flexibility. These benefits contribute to improved performance and effectiveness of CNNs in various computer vision tasks, such as image classification, object detection, and segmentation

Demo:

<u>ANN mai jaise Dense hota hai vaise</u>

<u>Convolution keliye Conv2D hota hai</u>

```
[2] import tensorflow
    from tensorflow import keras
    from keras.layers import Dense,Conv2D,Flatten
    from keras import Sequential
    from keras.datasets import mnist

[4] (X_train, y_train), (X_test, y_test) = mnist.load_data()

model = Sequential()

model.add(Conv2D(32,kernel_size=(3,3),padding='valid', activation='relu', input_shape=(28,28,1)))
model.add(Conv2D(32,kernel_size=(3,3),padding='valid', activation='relu'))
model.add(Conv2D(32,kernel_size=(3,3),padding='valid', activation='relu'))

model.add(Flatten())

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

Padding – valid means no padding

Intuition: