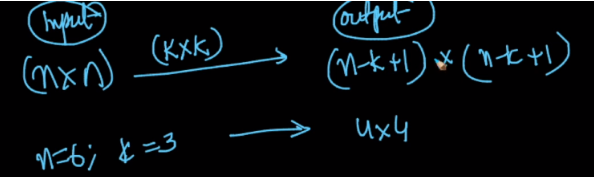
Till now after performing convolution on n\*n image we are getting output image of (n-k+1) \* (n-k+1)

Where n => dim in i/p image

K => kernel dim

Example for 6\*6 image and kernel with 3\*3 dim, output will be of 6-3+1 ie 4\*4



So how to get 6\*6 output image instead of 4\*4 o/p image? Ans is Padding

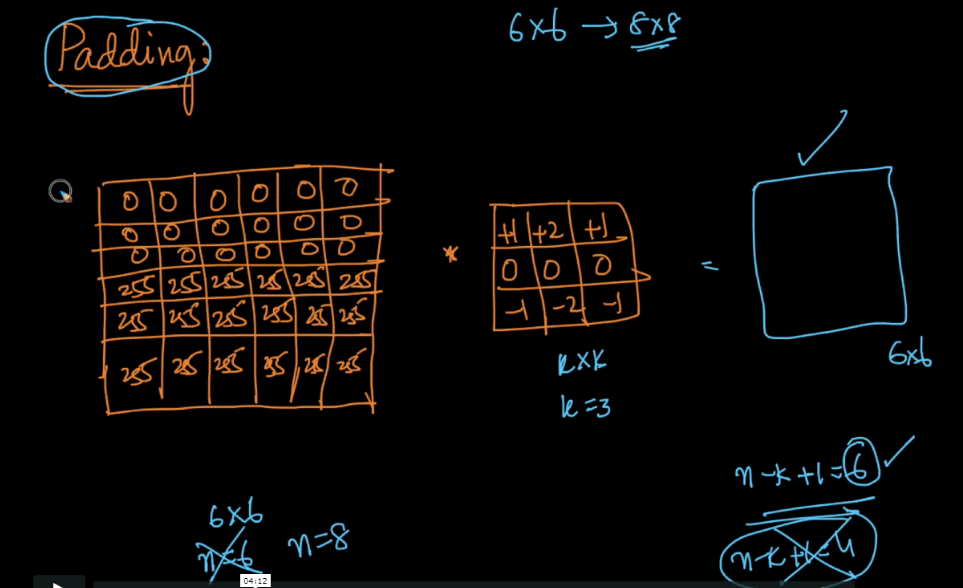
**Padding:**

let’s say i/p image is of 6\*6 which generates o/p of 4\*4, so we need to get an o/p of 6\*6, but how?

First find which no. of i/p dim will get o/p of 6\*6, so ans is 8\*8, so now we’ll make our 6\*6 i/p image as 8\*8 image by adding 1-1 row in top and bottom, 1-1 column in left and right, and we’ll put 0 in newly added rows and columns.

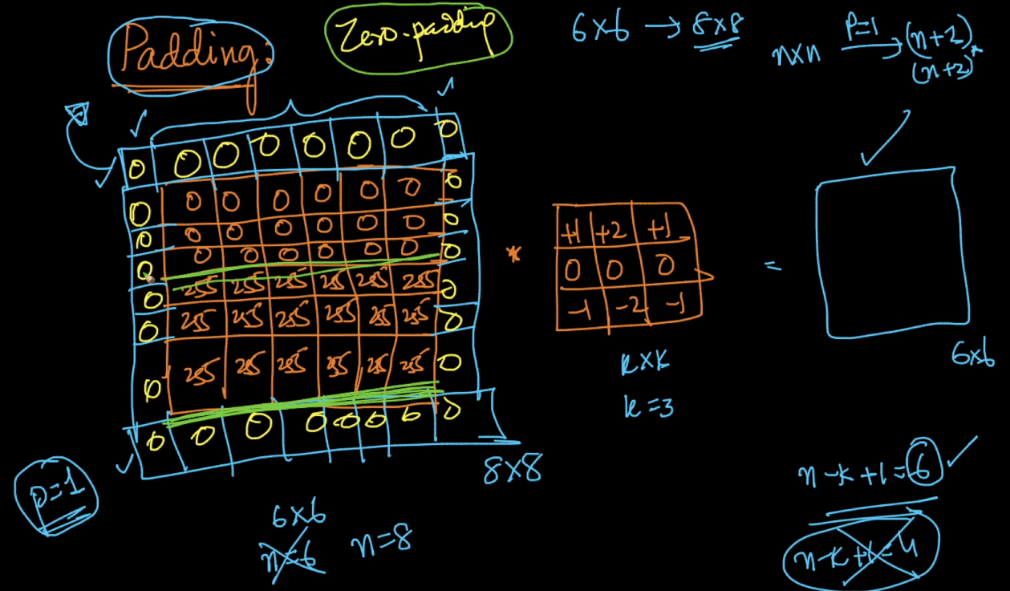
And we’ll perform the convolution on that image and we’ll get output image of 6\*6.

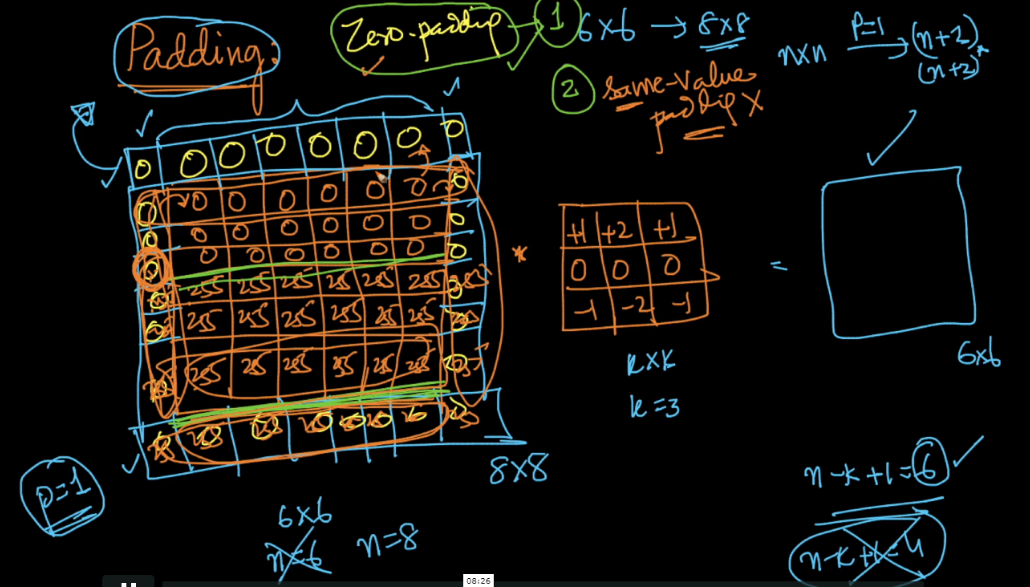
This way of putting 0 on padding columns and rows is called zero padding. And it’s used most of the times



There is another padding called same value padding which will pad the value at cell, to it’s closest cell value, but there is a problem in that which is see at 4th row in below figure and 1st column where the value at right will be 0 and value at below will be 255, so which value to choose, therefore it’s least used.

This padding of adding 1-1 row and column is padding with p=1, because we are adding 1-1 row and column in top, bottom, left and right.

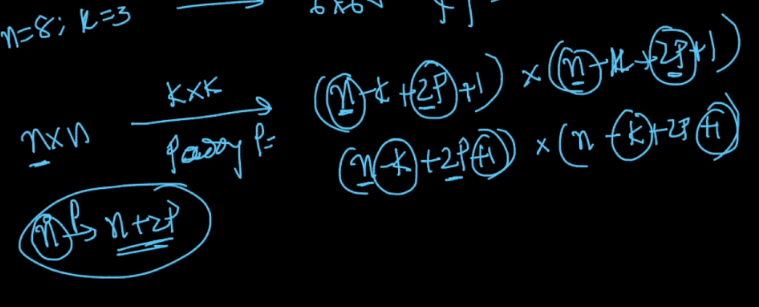




Now dim of output image when we use padding is (n - k + 2p + 1) \* (n – k + 2p + 1)

Let’s take an ex: i/p image=> 6\*6, kernel => 3\*3 and padding => 1

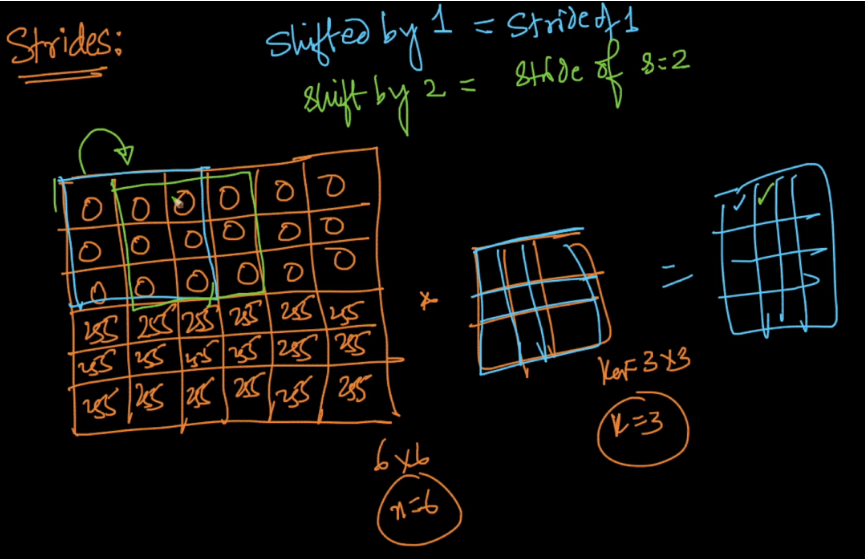
Then o/p matrix dim: 6 – 3 + 2\*1 + 1 = 6

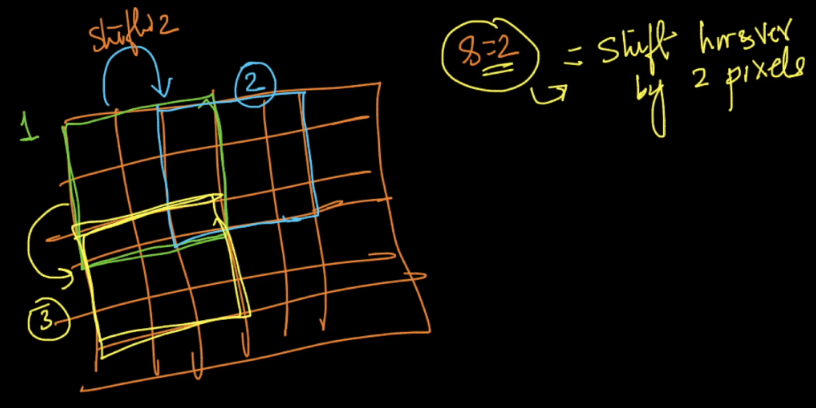


**Strides:**

Till now when we are doing convolution we are shifting window which multiplies with kernel by 1 step in horizontal and vertical, and this is called 1 stride.

We can shift by more than one step in both horizontal and vertical direction.

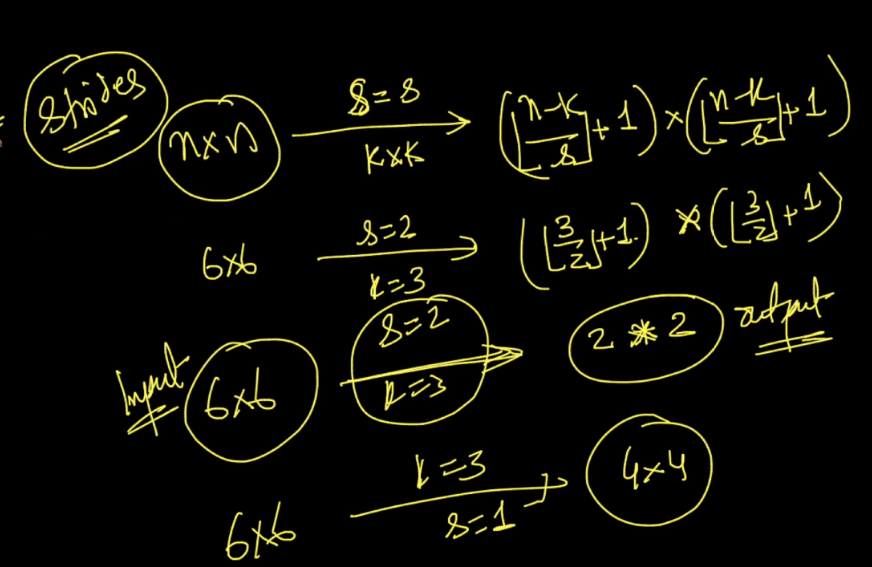




Dim of output matrix we get after stride will be floor ( (n-k)/s + 1 ) \* ( (n-k)/s + 1 )

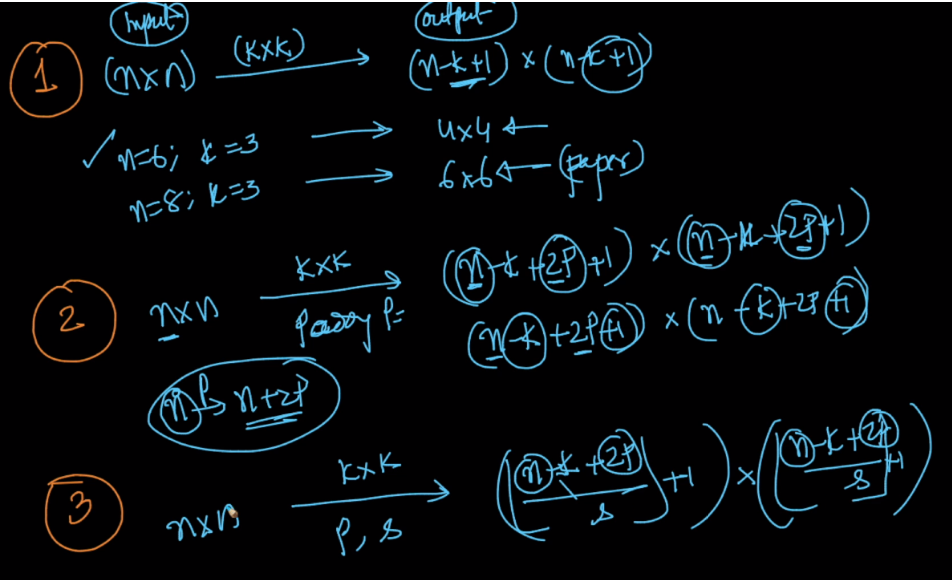
Let’s take an ex: i/p image=> 6\*6, kernel => 3\*3 and stride => 2

Ouput image dim: (6-3) / 2 + 1 = 3/2 + 1 = 1 + 1 = 2



Summary of formulas we have for o/p matrix

1. Without any padding and stride
2. With padding
3. With padding and stride



Nice Read: <https://www.quora.com/What-are-the-roles-of-stride-and-padding-in-a-convolutional-neural-network>

