**To build convnet, first we perform convolution for each several things, in this chapter we’ll see for edge detection**

**Horizontal edge:**

(Note color def: 0 -> black, 1-> white, in below images 0, 1 are replced with each other so consider them by interchanging them)

Let’s we are given a 6 \* 6 image in which first half is black and second half is white. And we need to detect the edge in it(line where white and black are meeting)

Here we’ll do convolution of image with kernel/filter/mask for horizontal edge detection

For horizontal edge detection kernel is sobel edge detection it’s a 3\*3(it can be changed) which is shown in below.

Now we just take a first 3\*3 window in input image and then do component multiplication and then addition.

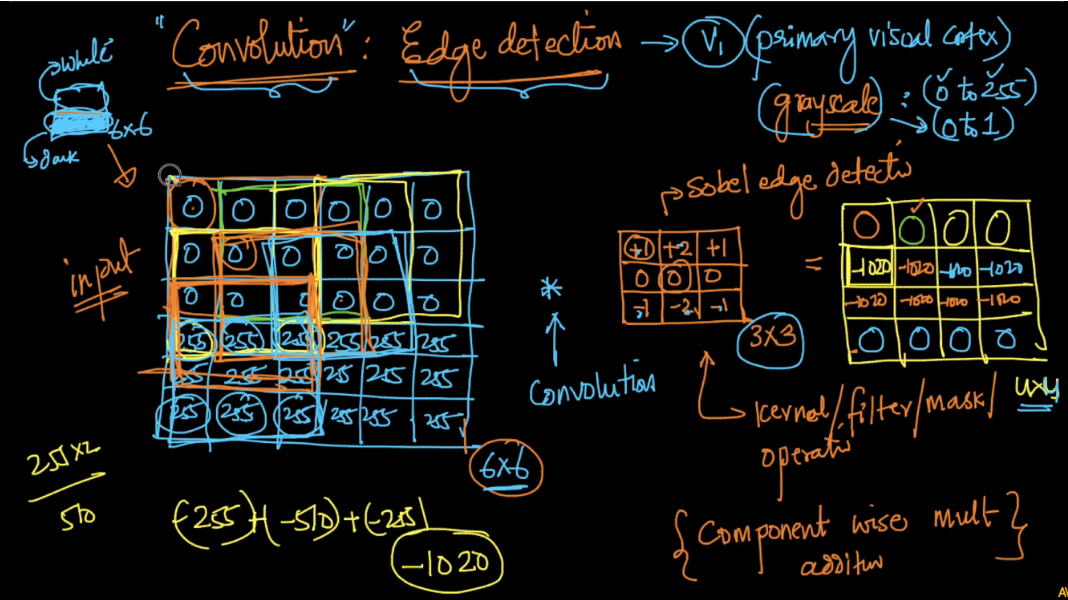
Example for fifth window, which starts from 2nd row of i/p image and 1st column of i/p image:

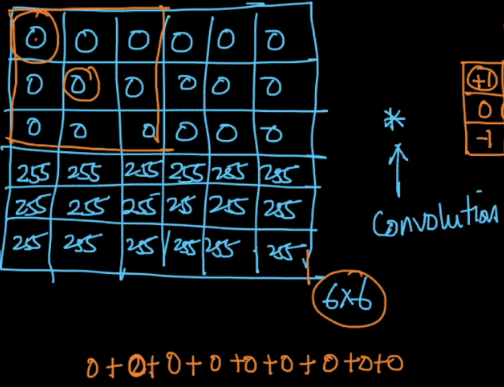
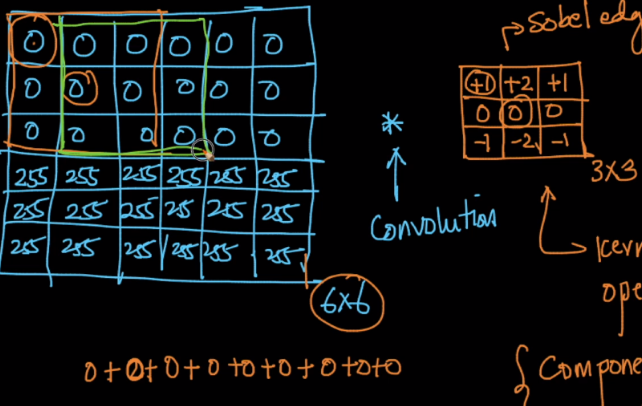
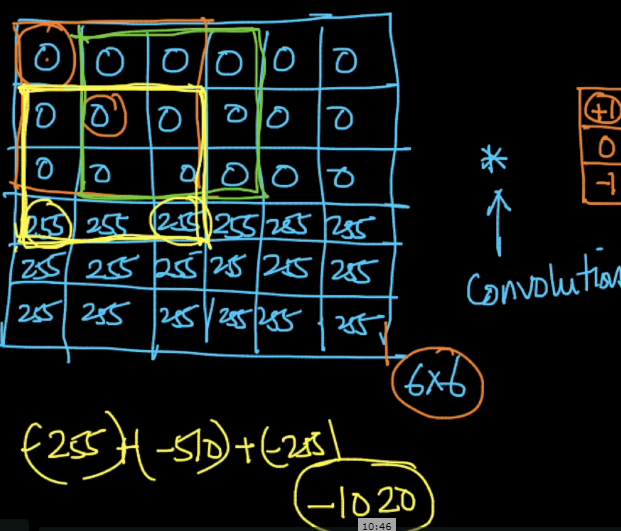
0\*1 + 0\*2 + 0\*1 + 0\*0 + 0\*0 + 0\*0 + 255 \* (-1) + 255 \* (-2) + 255 \* (-1) = -1020

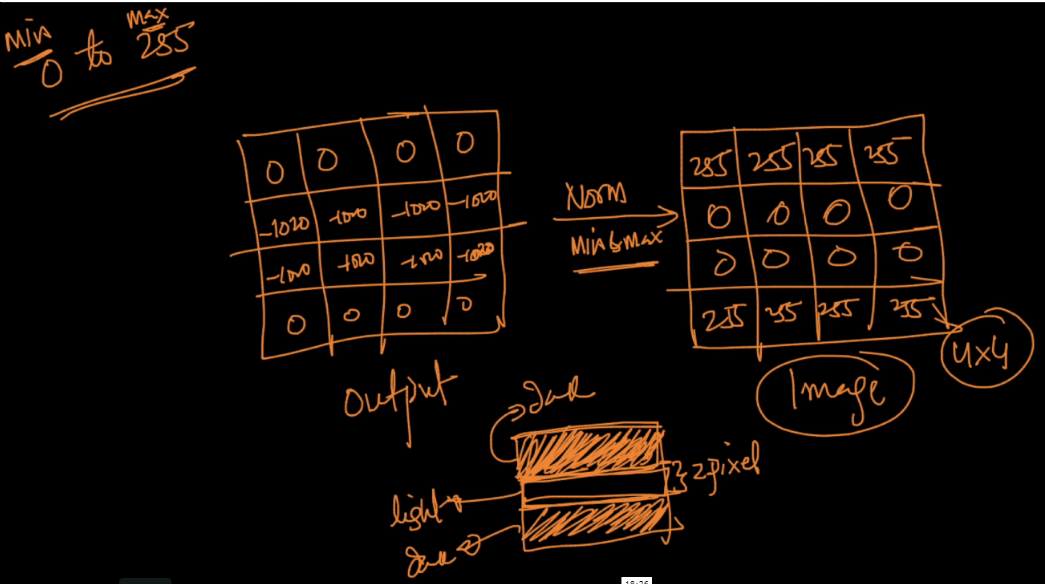
So we’ll make this entry in 2nd row and 1st column of o/p matrix.

By doing this we’ll get a output matrix which shown in image below.

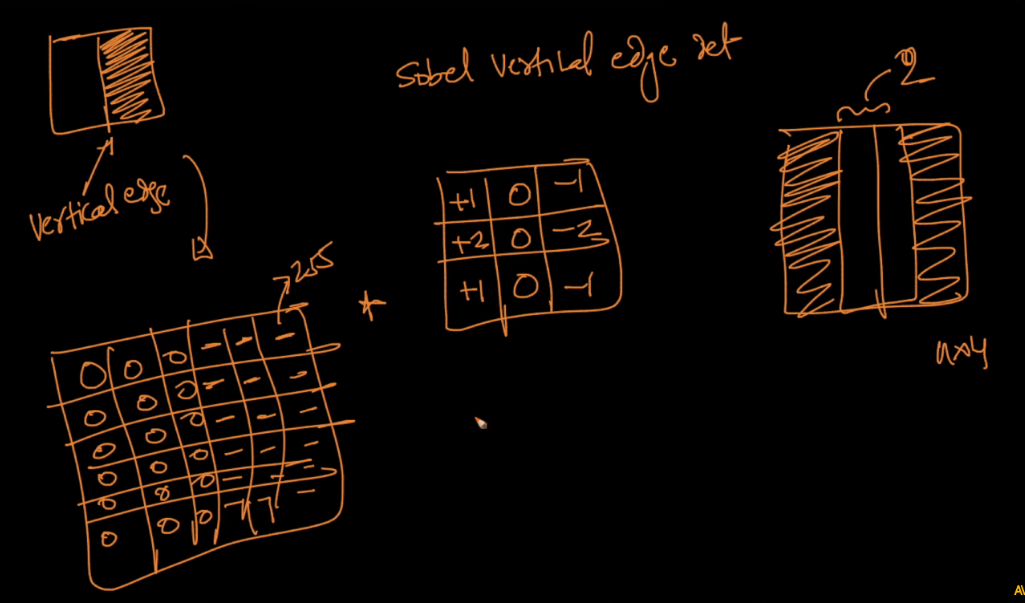
Then we perform normalization(0-255) where 0 will be changed to 255 and -1020 will be changed to 0, and by doing this we’ll see that we get a image(output matrizx) with a white horizontal line which reflects the edge.

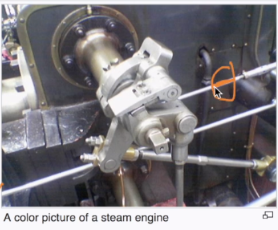
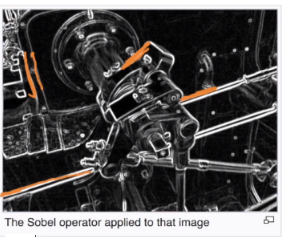


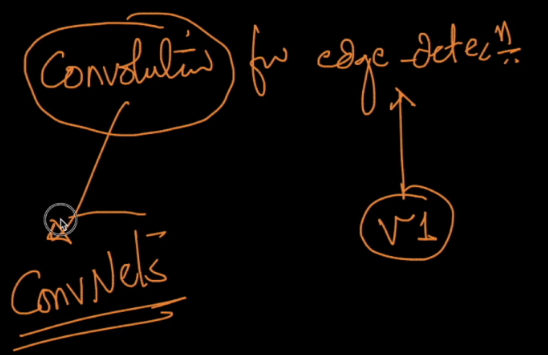
  



**Vertical Edge:** For vertical edge we just replace the kernel above with sobel vertical edge detector which is given below, and rest operation will be same.





Visual Explanation of CNN: <http://setosa.io/ev/image-kernels/>

<https://stackoverflow.com/questions/42883547/intuitive-understanding-of-1d-2d-and-3d-convolutions-in-convolutional-neural-n>

