

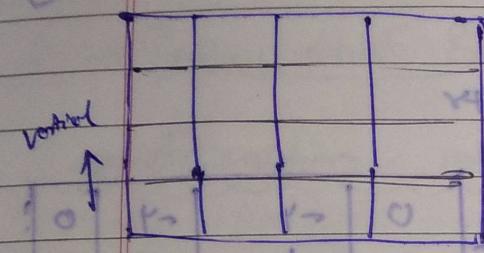
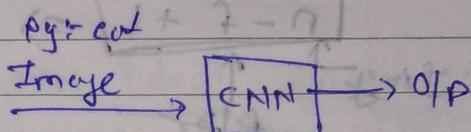
CNN

- whenever your I/P are in the form of Images, videos frames at that time you should prefer using CNN

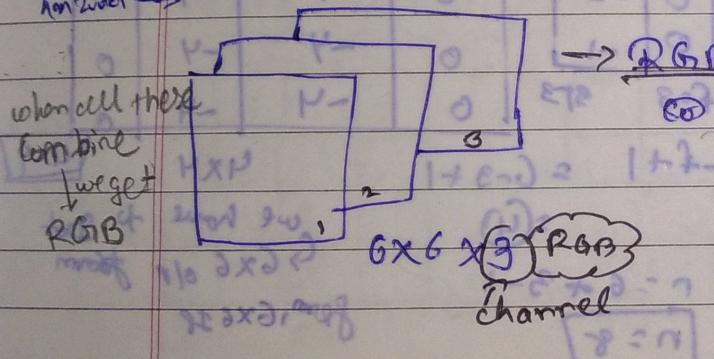
App Object detection, face classification, object classification and object recognition, etc..

* What is Convolution

Basic things about images



- Gray Scale (Black & white)
- Values ranges Between 0 to 255
 - Black
 - white
- 4x4 in pixels



$\rightarrow \text{RGB} \leftarrow \text{color}$ (Pixel values between 0 & 255)

min, max = 0 to 1

We can also normalize the pixels (or) reduce the pixel using ratio, more scalar it will reduce between 0 to 1.

$$\begin{array}{c}
 \text{Vertical Edge Filter} \\
 \hline
 \begin{array}{|c|c|c|c|c|c|c|} \hline
 0 & 0 & 0 & 1 & 1 & 1 & * \\ \hline
 0 & 0 & 0 & 1 & 1 & 1 & \\ \hline
 0 & 0 & 0 & 1 & 1 & 1 & \\ \hline
 0 & 0 & 0 & 1 & 1 & 1 & \\ \hline
 0 & 0 & 0 & 1 & 1 & 1 & \\ \hline
 \end{array} \quad
 \begin{array}{|c|c|c|c|} \hline
 1 & 0 & -1 & \\ \hline
 2 & 0 & -2 & = \\ \hline
 1 & 0 & -1 & \\ \hline
 \end{array} \quad
 \begin{array}{|c|c|c|c|} \hline
 0 & -4 & -4 & 0 \\ \hline
 0 & -4 & -4 & 0 \\ \hline
 0 & -4 & -4 & 0 \\ \hline
 0 & -4 & -4 & 0 \\ \hline
 \end{array} \\
 \text{Rowden Value} \\
 \text{filter} \quad 3 \times 3 \\
 \hline
 \begin{array}{|c|c|c|c|c|c|} \hline
 0 & 0 & 0 & 1 & 1 & 1 \\ \hline
 0 & 0 & 0 & 1 & 1 & 1 \\ \hline
 0 & 0 & 0 & 1 & 1 & 1 \\ \hline
 0 & 0 & 0 & 1 & 1 & 1 \\ \hline
 0 & 0 & 0 & 1 & 1 & 1 \\ \hline
 \end{array} \quad
 \begin{array}{|c|c|c|c|} \hline
 1 & 0 & -1 & \\ \hline
 2 & 0 & -2 & = \\ \hline
 1 & 0 & -1 & \\ \hline
 \end{array} \quad
 \begin{array}{|c|c|c|c|} \hline
 0 & -4 & -4 & 0 \\ \hline
 0 & -4 & -4 & 0 \\ \hline
 0 & -4 & -4 & 0 \\ \hline
 0 & -4 & -4 & 0 \\ \hline
 \end{array} \\
 \hline
 \begin{array}{|c|c|c|c|c|c|} \hline
 0 & 0 & 0 & 1 & 1 & 1 \\ \hline
 0 & 0 & 0 & 1 & 1 & 1 \\ \hline
 0 & 0 & 0 & 1 & 1 & 1 \\ \hline
 0 & 0 & 0 & 1 & 1 & 1 \\ \hline
 0 & 0 & 0 & 1 & 1 & 1 \\ \hline
 \end{array} \quad
 \begin{array}{|c|c|c|c|} \hline
 1 & 0 & -1 & \\ \hline
 2 & 0 & -2 & = \\ \hline
 1 & 0 & -1 & \\ \hline
 \end{array} \quad
 \begin{array}{|c|c|c|c|} \hline
 0 & -4 & -4 & 0 \\ \hline
 0 & -4 & -4 & 0 \\ \hline
 0 & -4 & -4 & 0 \\ \hline
 0 & -4 & -4 & 0 \\ \hline
 \end{array} \\
 \hline
 \end{array}$$

Size of image (6,6)

$$n + 2p - f + 1 \quad | \quad p \rightarrow \text{padding}$$

* filter (or) Kernel are used to find Edges

PAGE No.

DATE

DATE / /

horizontal Edge filter

1	2	1
0	0	0
-1	-2	-1

$$\text{Image size} = 16 \times 6$$

Note:- As a firefly so off we get
4x4 image size, But we are looking some imputed
information from image

$$\lceil n - f + 1 \rceil \geq 6 - 3 + 1 = 4$$

No. of rings ↑
(Polar) ↑ filter

To overcome the problem we used ~~perdition~~

* Padding In CNN

Note:- How we can get output $G \times G$ (so we use padding)

Random values				Input			
0	-4	-4	0	0	-4	-4	0
0	0	0	0	0	-4	-4	0
0	0	0	1	1	1	1	0
0	0	0	1	1	1	1	0
0	0	0	1	1	1	1	0
0	0	0	1	1	1	1	0
0	0	0	1	1	1	1	0

for getting O/P GX6 we will be adding Padding

$P = \text{padding}$ During Backpropagation filter will update
 $S = \text{stride}$

PAGE No.	
DATE	/ /

- Two type of values can be basically filled
 - (i) Zero padding (where will insert zero everywhere)
 - (ii) Try to find out which nearest value and you try to put the same value, Neighor you will try to put

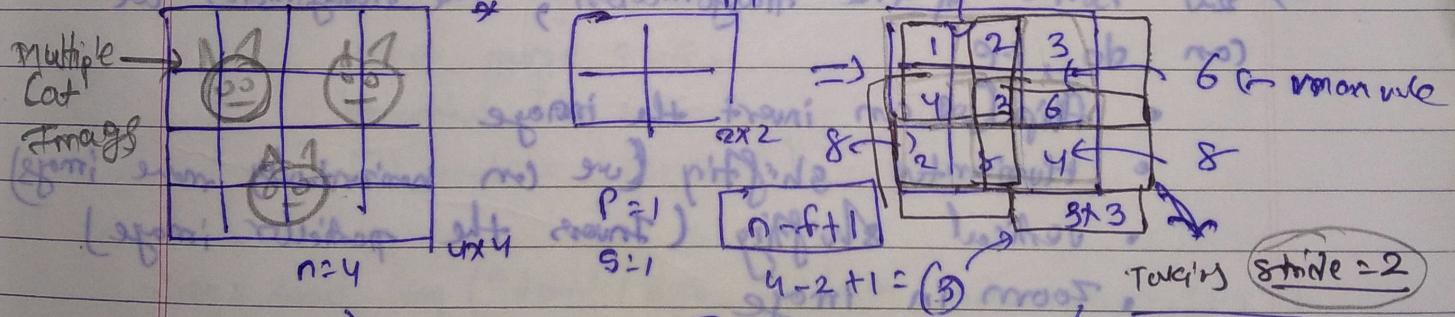
Note: Most commonly used is 0 padding

Note: In padding we will be getting more Edges in our I/P
 Padding will give you some image size in O/P

- Padding adds Extra rows and columns

* Operations of CNN (CNN vs ANN)

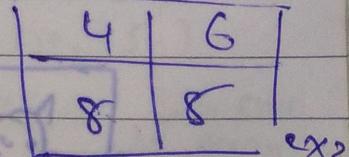
* Max Pooling



Note Imp

→ More pooling basically means at a specific location, see I am taking the higher intensity values and I am placing it in my output. Hence will be properly detected, because we are taking the high intensity high pixel value.

Convolution operation



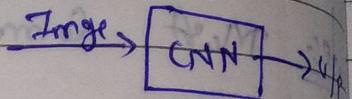
Max Pooling

Types of Pooling

→ Max Pooling / Avg pooling

↳ Here we will take lower intensity value, lower value.

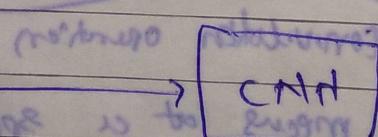
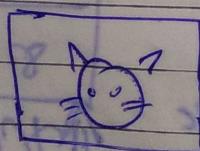
Data Augmentation CNN



→ Dog & Cat image
 { x_i, y_i }
 if my x image is dog then my o/p is
 if my x image is cat then my o/p is

- With the help of Data Augmentation we can transform this particular image into different kinds of images, the o/p will be not only.
- But some of the transformation, the transformation we can do, i.e.

- flip (Can invert the image)
- Horizontal shifting (Can horizontally move image)
- Vertical shifting (Inverts the particular image)
- zoom in image



→ O/P

- why Data Augmentation
- Suppose we have very less Data set e.g. - 1000 images of cats and dogs
- Suppose I take the cat image and do Data augmentation, I will get many more images, all different kinds of images.

→ we will be adding some INVARIANCE into the images, some different styles, whatever CNN model we are creating it should be Robust.

Imp → ~~→~~ Data Augmentation → helps to ~~→~~ many images with the same off, Due to this our Data set increases ^{create more}

* Code Data Augmentation

it allows all the Data Augmentation properties.

from keras.preprocessing.image import ImageDataGenerator, array_to_img, image_to_array, load_img,

we can load the image and apply various operations

```
datagen = ImageDataGenerator(  
    rotation_range=40,  
    width_shift_range=0.2,  
    height_shift_range=0.2,  
    shear_range=0.2,  
    zoom_range=0.2,  
    horizontal_flip=True,  
    fill_mode="nearest")
```

img = load_img("dataset/signs_dataset/cut-orange.jpg")

x = img_to_array(img) # this is np array (3, 150, 150)

x = x.reshape(1,) + x. shape) # this is np array with shape (1, 150, 150) ✓ 4 Dim array

#flow command generate batch of images and save the results to the previous dir
i = 0

for batch in datagen.flow(x, batch_size=1,

save_to_dir="Previous", save_prefix="ad",
save_format=

i += 1

if i > 20: # after loading 20 images to dir (otherwise)
break
then stop.