AGENDA

1> Computer Vision Basics
2> Understand Images / Video data (Theory)
3> Understand Images / Video data (Practical)
4> Filters and its use cases (T/P)
5> CNN

1> CV Basics

Enables machine or computer to have a vision system close to human.

Classification

P 90%　　Dog 95%　　Person 70% Dog 30%

Detection (Regression, Classification)

NOB

(x, y)

P → 90%　　D → 94%

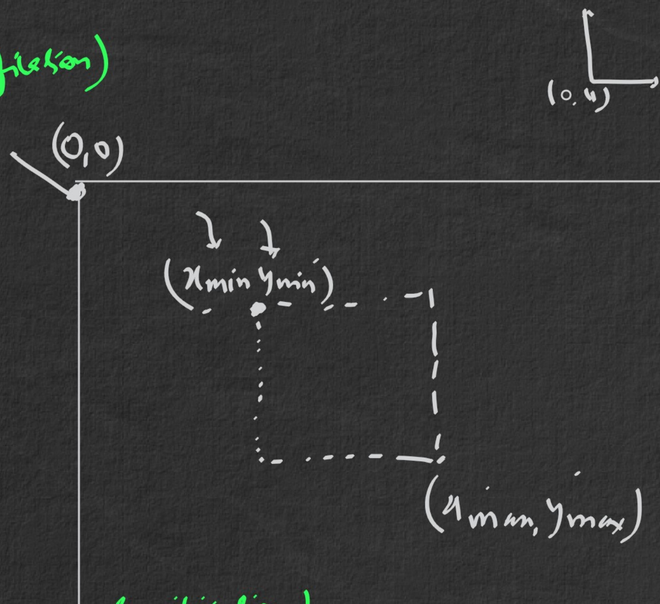
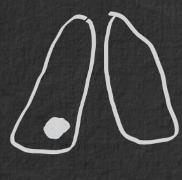


Image Segmentation (Pixel wise classification)

P　　D

healthcare domain



model

covid +ve → img classification

image segmentation

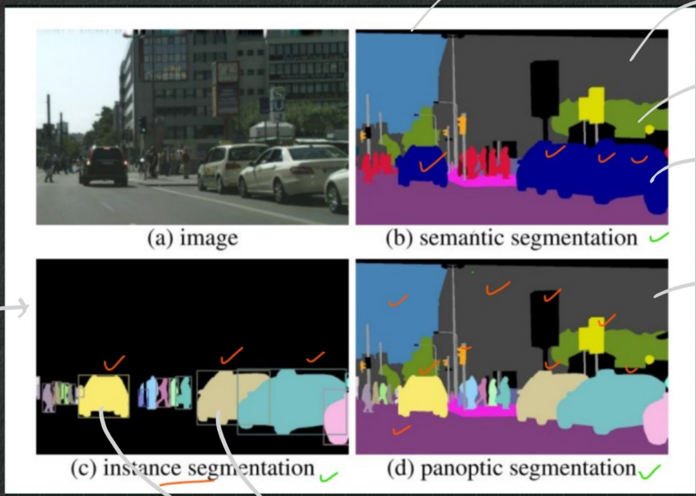
Sky

Building

Tree

Car → Blue

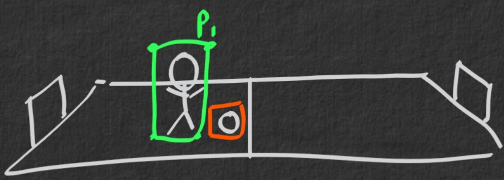
Building



car1 car2

semantic + instance = panoptic segmentat^n

Object Tracking

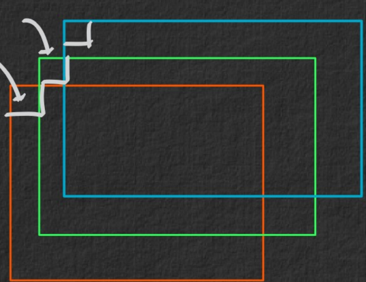


2> Image Dataset.

Primary Colours : RGB

(0,255)

IMG →



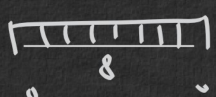
R G B
(0, 150, 205) →

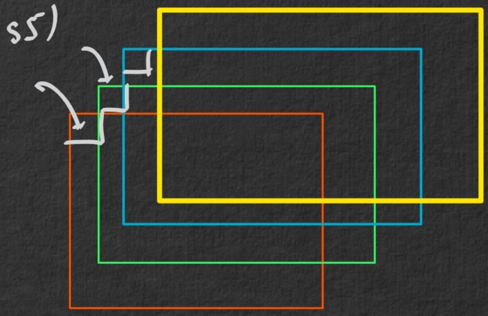
256

8 bit Images

$0 \to 2^8 - 1$

$0 \to 255$





alpha → transparency.

8-bit images $\quad 0 \rightarrow \underbrace{255}$ → $\underbrace{00000000}_{0}$ $\quad \underbrace{11111111}_{2^8 - 1}$

$2^8 - 1 \rightarrow 256$

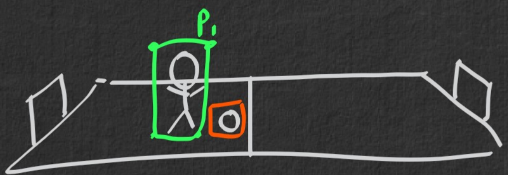
frame1 frame2 frame...

↓

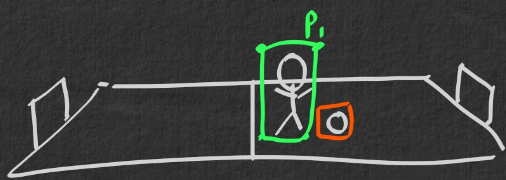
Video

P1 P2 P3

t=0 t=1 → time

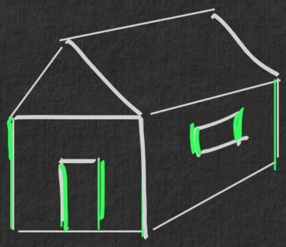


frame 1

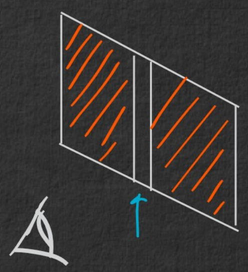


frame 2

Filters .

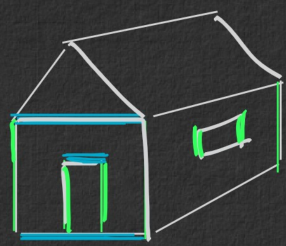


Window

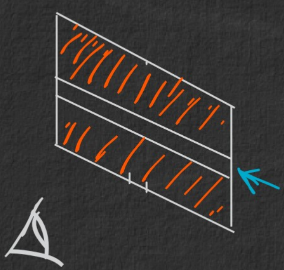


eye

$$\begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}$$ sobel filter



Window

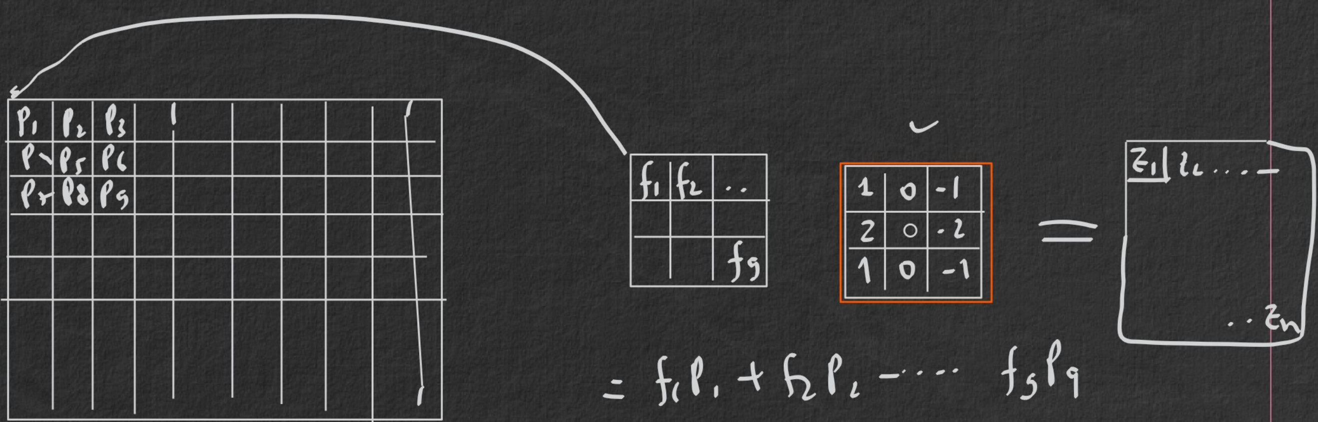


eye

$$\begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}^T$$ sobel filter

$$\| $$

$$\begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$



$$= f_1 p_1 + f_2 p_2 \cdots\cdots f_9 p_9$$

$$z_1 = \sum_{i=1}^{9} f_i p_i$$

Img

pixel length of image

New Image $\dfrac{p - f}{1} + 1$ (f → len of filter)

stride → 1

z_2

image = $(6,6)$ filter $(3,3)$ Stride, step = 1

new image len $= \dfrac{6-3}{1} + 1 = 4$

width $= \dfrac{6-3}{1} + 1 = 4$

$(4,4)$

STEP 1

$$\text{pic} = 6\left\{\underbrace{\begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 7 & 8 & 9 & 10 & 11 & 12 \\ 13 & 14 & 15 & 16 & 17 & 18 \\ 19 & 20 & 21 & 22 & 23 & 24 \\ 25 & 26 & 27 & 28 & 29 & 30 \\ 31 & 32 & 33 & 34 & 35 & 36 \end{bmatrix}}_{6}\right. \qquad 3\left\{\underbrace{\begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}}_{3}\right. = \overbrace{\begin{bmatrix} z_1 & z_2 & z_3 & z_4 \\ & & & \\ & & & \\ & & & z_{16} \end{bmatrix}}^{4}\Bigg\}4$$

$$\frac{6-3}{1} + 1 = 4$$

pic$[0:3,\ 0:3]$ 0, 1, 2

$$= \begin{bmatrix} 1 & 2 & 3 \\ 7 & 8 & 9 \\ 13 & 14 & 15 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}$$

$z_1 =$

$$\text{sum}\begin{bmatrix} 1 & 0 & -3 \\ 14 & 0 & -18 \\ 13 & 0 & -15 \end{bmatrix} = \text{sum}\left[28,\ 0,\ -36\right]$$

$$z_1 = -8$$

$z_1 = 1\times1 + 2\times0 + 3\times(-1)\ \ldots\ldots\ 15\times(-1)$

$= -8$

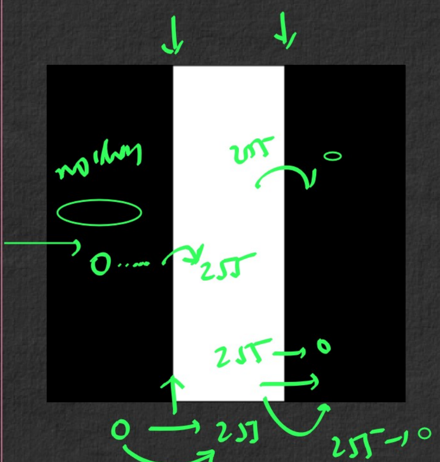
1	2	3	4	5	6
7	8	9	10	11	12
13	14	15	16	17	18
19	20	21	22	23	24
25	26	27	28	29	30
31	32	33	34	35	36

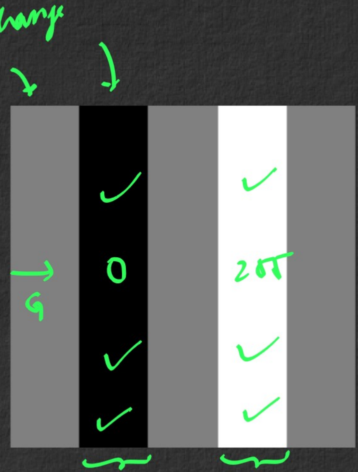
pic, 6 × 6

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

3 × 3

$$\begin{matrix} 2 & 3 & 4 \\ 8 & 9 & 10 \\ 14 & 15 & 16 \end{matrix} \times \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} = \longrightarrow z_2$$





$$\frac{d5}{dx} = 0$$

$$\frac{d}{dx} \neq 0$$

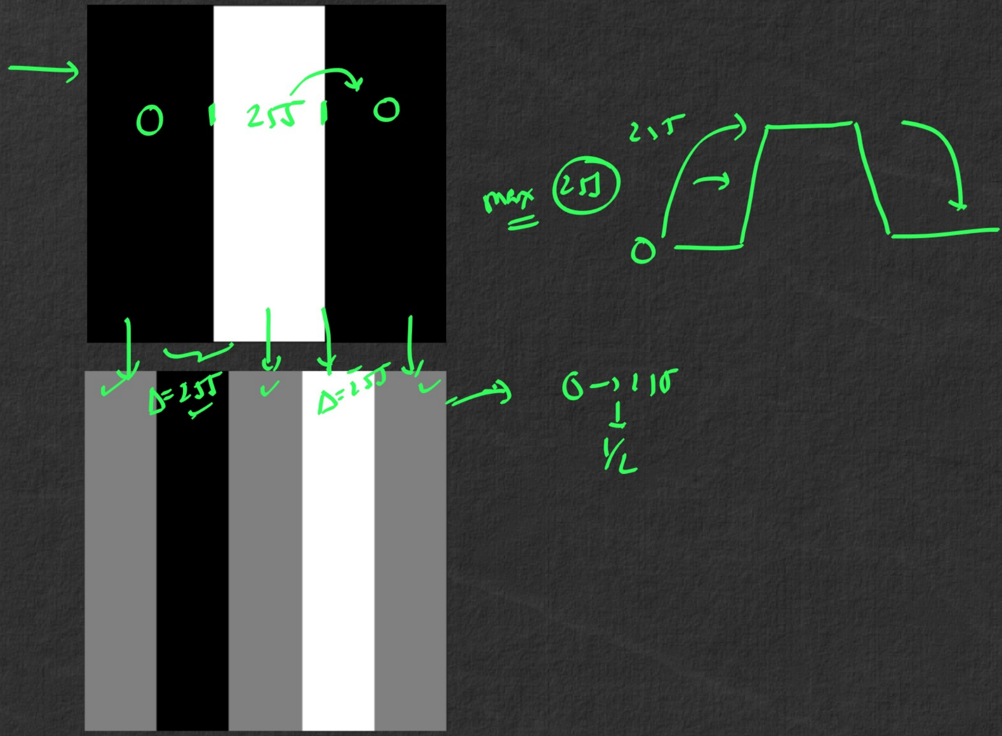
Edge → change in intensity of pixel values

255 ✓

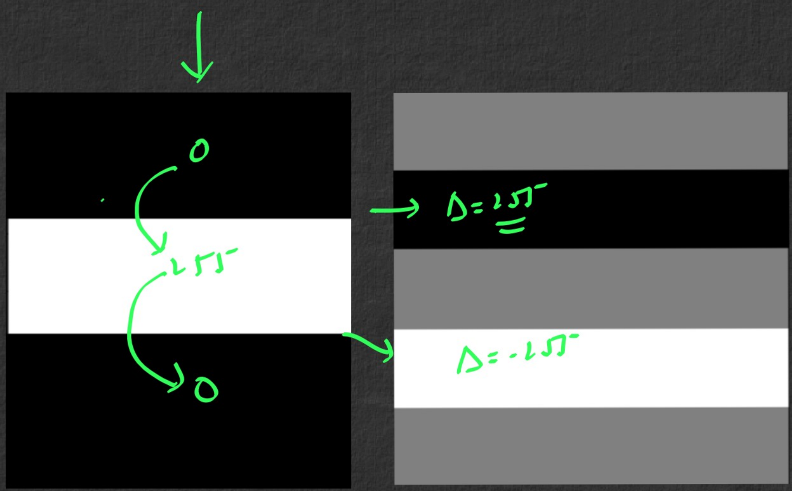
$y = mx$

$y = 2x \rightarrow 2$

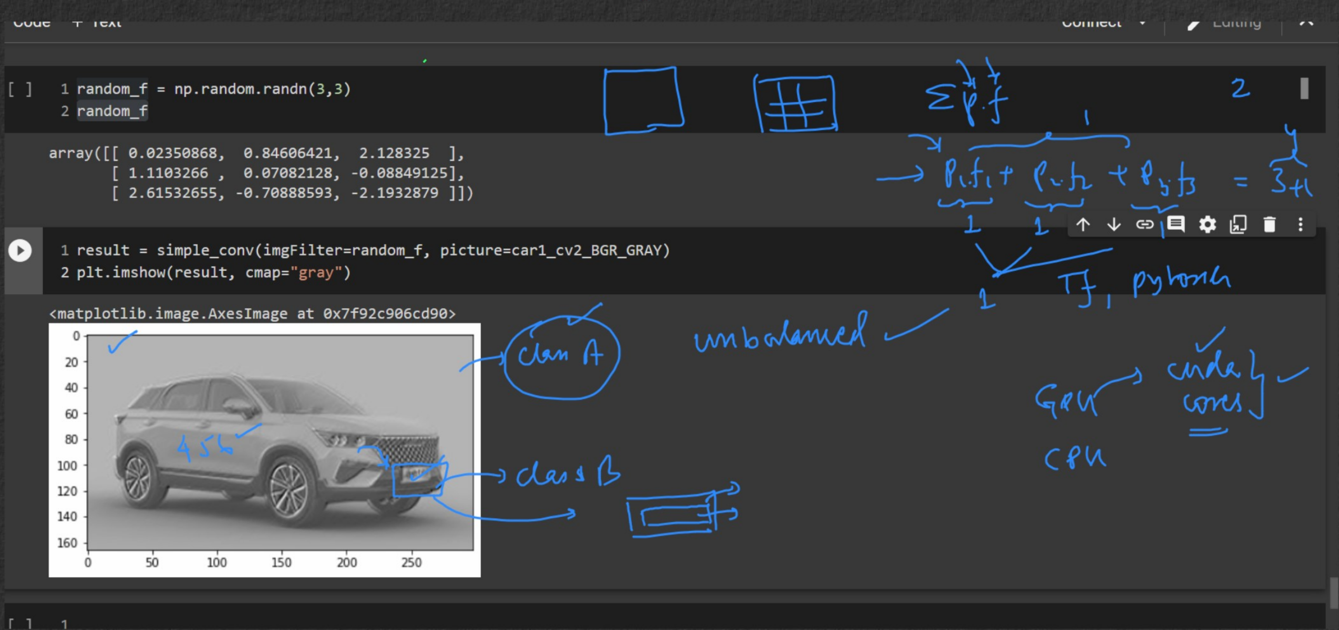
$y = 2 \longrightarrow 0$



0
255
0
max = 255
255
0
Δ=255
Δ=255



0
255
0
Δ=255
Δ=-255



```
[ ] 1 random_f = np.random.randn(3,3)
    2 random_f

array([[ 0.02350868,  0.84606421,  2.128325  ],
       [ 1.1103266 ,  0.07082128, -0.08849125],
       [ 2.61532655, -0.70888593, -2.1932879 ]])

    1 result = simple_conv(imgFilter=random_f, picture=car1_cv2_BGR_GRAY)
    2 plt.imshow(result, cmap="gray")

<matplotlib.image.AxesImage at 0x7f92c906cd90>
```
Σ p.f
$p_1 f_1 + p_2 f_2 + p_3 f_3 = 3+1$
TF, pytorch
unbalanced
class A
class B
GPU
CPU
cuda cores