## (#) Reading, Manipulating and Saving Grayscale Image using Matplotlib.

We are going to talk about the real grayscale, real grayscale images, how to read those images, how to create the images, how to manipulate those images, how to create the images as matrices, and how to write the modified images back to the computer. First, we will be using matplotlib then further openCV.

Escape+Y → code.

Date : Escape+M → Markdown

# Processing Grayscale Images.

im = plt.imread("path").

type(im) → numpy.ndarray

im.shape → dimensions in (rows, columns) i.e. (height, width)

(949, 728)

im.dtype → dtype('uint8') → 8 bit unsigned integer

display

plt.imshow(im, cmap = "gray")

↳ colormap.

im → array([[85, 85, 85, ..., 62, 62, 62],
[85, 85, 85, ..., 64, 65, 64],
...
[29, 37, 37, ..., 36, 43, 44],
[33, 40, 37, ..., 35, 43, 42]]
, dtype = uint8)

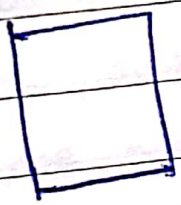
img is a grid of numbers.

im[23, 300] → colorcode at that position

row (0 to 948) column (0 to 727)

→ column increase

row increase ↓



↳ Making an img copy. (coz original copy (im) is read only)

im2 = im.copy()

Date :

im2[23:100, 40:100] = 255

↓ Start from row number 24 coz 0.

till row no. 100 (exclusively)

↳ This means we have picked a patch and assigned pure white color in that patch.

And not only that, we can ~~a~~ assign different colors, different intensity values to different patches or different individual pixels, pure white, pure black, or any other color.

im2[300:400, 40:100] = ~~255~~ 100 → grayish color.

plt.imshow(im2, cmap='gray')

* saving img in the computer

plt.imsave(r'path/filename.jpg', im2, cmap='gray')

So, this is how we treat a grayscale image just like a two dimensional array of numbers.

Quiz

Can you replace eyes with white color rectangular patches in python

#Solution.

```
im = plt.imread('albert.jpg').
im3 = im.copy()

im3 = [330:430, 300:400] = 255 # left eye
im3 = [330:430, 440:540] = 255 # right_eye

plt.show()  plt.imshow(im3, cmap="grey")
```