Image Processing through Open CV

① Reading an image :)

for this we need to import two libraries

```
import cv2
  cv2.imread('parameter-1', paramiter2)
```

Parameter_1 : image to be read
(allows us to provide name of image to be read, we can also specify path of image if it is not in same working directory)

Parameter_2 :) can have any of three values
1 :- Loads a colour image neglecting any transparency of image
0 : Loads image in grayscale mode
-1 : Loads image as such including alpha channel

[Alpha channel are masks through which we can display images. The alpha channel is on 8-bit channel, which means it has 256 levels of gray from 0 (black) to 255 (white)]

#1) Display an Image ~~#~~

cv2.imshow('window_name', display_image)
cv2.waitkey(0) (Time)
↳ (This function is used for waiting some millisecond for any keyboard event)

cv2.destroyAllWindows()
↳ (used to destroy all the windows we have created)

2) Accessing image pixels and planes OpenCV :)

Visualizing an image as an array it is easy to access its pixels by simply array indexing, for example if you have to access the pixel at position [100,100]

eg: pixel = image[100,100]
print(pixel)

↳ we then obtain the BGR values at position [100,100] and we can see that intensity range in b/w 0→255

In openCV, the Red, Green and blue planes are arranged in order
[Blue; Green; Red] i.e. BGR
rather than RGB

for accessing only blue plane
eg: blue = image[:, :, 0]
print(blue)

To get the shape of image we use # eg,
image_name.shape → (512, 512, 3)
R C No. of Plane

To get the size of image we use #
image_name.size

Keras :) is a python based deep learning framework which is actually the high level api of tensorflow.

- opensource, high level neural-network library
- written in python, so easy to debug
- Runs on top of tensorflow, Theano, CNTK
- Runs smoothly on CPU & GPU, if used in D.L.

Tensorflow :)

- opensource and free S/W library for ~~windows~~ dataflow

- Symbolic math library
- Used for ML apps: like neural networks
- Used for research and production at google:

Numpy stands for numerical python

It is a library which consists of multidimensional array objects and collection of routines for processing those arrays.

- We can perform mathematical and logical operation on arraye.
- Performs operation related to Linear algebra

① Import an image using PIL i.e. (Python Imaging Library)

```
from PIL import Image
    import numpy as np

  img = Image.open("file path.ext")
```

(used for create the image object)

To convert it to numpy array, we simply used inbuilt method of numpy

i.e. (II) `img_array = np.array(img)`

img-array.shape
(256,256,3) — (No. of channels in image)
↳ (Height & width of image)

for on black-white image, there is only one channel [Hp: only either two classes that is 0 & 1]

for grayscale image - only one channel [0-255]

Data Visulisation :)

```
import matplotlib.pyplot as plt

X = np.array([1,2,3,4,5])
Y = X**2

plt.plot(X,Y)  # plots curve or lines
plt.scatter(X,Y)  # plots points only
plt.scatter(X,Y, color = 'red')
                   # can also define color
plt.show() # show the things before what we
              write above the code.
```