Open Cv

Jk31.m96@

# Image Processing

## Goal:

-Various operation like changing value from RGB 🡪Gray,

-cv2.cvt.Color(),cv2.cvt.Range()

## Changing Color Space:

🡪For color conversion, we use the function cv2.cvtColor(input\_image, flag) where flag determines the type of conversion.

For BGR \rightarrow Gray conversion we use the flags cv2.COLOR\_BGR2GRAY. Similarly for BGR \rightarrow HSV, we use the flag cv2.COLOR\_BGR2HSV.

>>> import cv2

>>> flags = [i for i in dir(cv2) if i.startswith('COLOR\_')]

>>> print flags

### How to find HSV values to track?

It is very simple and you can use the same function, cv2.cvtColor(). Instead of passing an image, you just pass the BGR values you want. For example, to find the HSV value of Green, try following commands in Python terminal:

>>> green = np.uint8([[[0,255,0 ]]])

>>> hsv\_green = cv2.cvtColor(green,cv2.COLOR\_BGR2HSV)

>>> print hsv\_green

[[[ 60 255 255]]]

# To load an Image:

🡪**cv2.imread()**, **cv2.imshow()** , **cv2.imwrite()**

**🡪**Use the function **cv2.imread()** to read an image. The image should be in the working directory or a full path of image should be given.

First argument is the path of the image

Second argument is a flag which specifies the way image should be read.

* cv2.IMREAD\_COLOR : Loads a color image. Any transparency of image will be neglected. It is the default flag. 1
* cv2.IMREAD\_GRAYSCALE : Loads image in grayscale mode 0
* cv2.IMREAD\_UNCHANGED : Loads image as such including alpha channel -1
* Instead of these three flags, you can simply pass integers 1, 0 or -1 respectively.

import numpy as np

import cv2

# Load an color image in grayscale

img = cv2.imread('messi5.jpg’0)

cv2.imshow('image',img)

cv2.waitKey(0)

cv2.destroyAllWindows()

To display image we cv2.imshow() which take two argument one is the name given to the window and other is the img we read earlier in the variable img

**cv2.waitKey()** is a keyboard binding function. Its argument is the time in milliseconds. The function waits for specified milliseconds for any keyboard event. If you press any key in that time, the program continues. If **0** is passed, it waits indefinitely for a key stroke.

**cv2.destroyAllWindows()** simply destroys all the windows we created. If you want to destroy any specific window, use the function **cv2.destroyWindow()** where you pass the exact window name as the argument.

# Video Processing

* import numpy as np
* import cv2
* cap = cv2.VideoCapture(0)#0 represents the webcam if u have a multiple cameras us 1,2..
* while(True):
* # Capture frame-by-frame
* ret, frame = cap.read()
* # Our operations on the frame come here
* gray = cv2.cvtColor(frame, cv2.COLOR\_BGR2GRAY)
* # Display the resulting frame
* cv2.imshow('frame',gray)
* if cv2.waitKey(1) & 0xFF == ord('q'):
* break
* # When everything done, release the capture
* cap.release()
* cv2.destroyAllWindows()