

Image Processing workshop 2017

DAY-1

#C++ tutorial:

- >Header file <iostream> should be included
- >After header file use- using namespace std; (in order to avoid write std before every line)
- >To take input- cin>>a (a is the variable)
- >To take output- cout<<a<<endl (endl to change a line)

#ubuntu commands:

- >to open a c++ file: subl file_name.cpp
- >To compile a c++ file: g++ file_name.cpp
- >To check present location: pwd
- >To create a new directory at present location: mkdir xyz
- >To open that : cd xyz

#IMAGE:

- >A 2-D array of pixels.
- >Different color scales-**RGB**(red green blue),**HSV**(hue saturation value)

#RGB:

- >Any color can be made through these 3 basic primary colors-red,green,blue.
- >Every color can be further classified into 256 different intensities.
- >0 has lowest intensity and 255 has highest.
- >e.g- yellow is (255,255,0)(R,G,B).

#image processing:

- >Header files to be included except #include<iostream>
 1. include "opencv2/highgui/highgui.hpp"
 2. include "opencv2/imgproc/imgproc.hpp"
 3. include "opencv2/core/core.hpp"
- >using namespace cv;
- >inside main function.
- >to define a set of pixels we use data type Mat.
- >to define image a using Mat:
 - Mat a(400,300,CV_8UC3,Scalar(0,255,78))
 - Mat a(400,300,CV_8UC1,Scalar(135)) //for grey scale image
 - >this will construct a 400X300 array of pixels.
 - >8 bit,3 channel(BGR)
 - >U-unsigned

>Scalar(blue value,green value,red value)

->to print an image

Use imshow("image",a);

waitKey(0); (K is capital)

In the end of program;

->to assign a value to a particular pixel we use

img.at<Vector3b>(i,j)[0]=.....

>(0 for blue,1 for green,2 for red)

>img name of variable.

>(i,j)-ith row,jth col

img.at<uchar>(i,j)=.....

>for grey scale image

->to work on images

> Mat a=imread("....address...",0 or 1) //0 for grey scale and vice versa

> no.of rows and cols in given image a- a.rows,a.cols

->to create a copy

>Mat b=a.clone()

To convert a colored image to grey scale:

->using weighted mean formula:

>img.at<uchar>(i,j)=...0.114xB + 0.587xG + 0.299xB

To convert an image in RGB to different scale:

cvtColor(source,destination,flag)

>source-original image variable

>destination- final image variable

>flag- CV_BGR2HSV

CV_BGR2GRAY