**OpenCV using Python**

**Basic Terminal commands**

**1. cd (change directory) :** The **cd** command will allow you to change directories. When you open a terminal you will be in your home directory. To move around the file system you will use cd.

**2. man (manual):** **man** is an interface to online reference manuals. man page provides description about the commands. This is a super command that lets you learn about all other commands.

3. **ls (listing)** : The **ls** command shows you ('list') the files in your current directory. Used with certain options, it shows size of files, permissions etc. Use ***man*** *ls* to see different options in ***ls*** .

**Opening editor from Terminal**

Use one of the method:

**Method 1 : nano** editor

* Goto Terminal ---(ctrl + alt + t )
* Type: *nano* ***filename****.py*
* **Press 'Enter' to start typing.**
* **Write given code.**
* **Press Ctrl+O to save the file.**
* **Press Ctrl+X to exit from the editor.**

**Run the file using below command**

**python filename.py**

**Method 2 : gedit** editor

* Goto Terminal ---(ctrl + alt + t )
* Type: *gedit* ***filename****.py*
* **Press 'Enter' to start typing.**
* **Write given code.**
* **Press Ctrl+s to save the file.**

**Run the file using below command**

**python filename.py**

**1 . Code to read an IMAGE**

import cv2

import numpy as np

img=cv2.imread('img1.jpg') **#read the file and store it as matrix in img**

cv2.imshow('frame',img) **#display the image in a window named frame**

cv2.waitKey(0) **#wait for any key to be pressed**

cv2.destroyAllWindows()  **#close all the windows created**

**2. Code to read IMAGE properties**

*// import the modules*

*// read an img1.jpg image and save it as matrix in img*

px=img[100,100] **#read the row & column wise and store in px**

print px **#get the RGB value of px**

img[100,100]=[255,255,255] **#assigning RGB value to row and column**

print img[100,100]

print img.shape  **#returns a tuple of number of rows, columns and channels**

print img.size **#returns the number of pixels**

print img.dtype **#returns the data type of image**

**3.Code to Capture an Image from Web camera**

*// import the modules*

cap=cv2.VideoCapture(0)  **# initialise the camera**

while(1): **#goes inside an infinite loop**

ret,img=cap.read() **#read from camera and store it in a variable img , ret=1 if image is captured else ret = 0**

cv2.imshow('frame',img)  **#show the read image on the window frame**

cv2.waitKey(0)  **#wait for a key to be pressed**

break **#exit from loop**

cap.release() **#release the camera**

cv2.destroyAllWindows() **#destroy all the windows**

**4.Code to Capture an Image from Web camera and save it in the folder**

// import modules

cap=cv2.VideoCapture(0)  **# initialise the camera**

while(1):  **#goes inside an infinite loop**

ret,img=cap.read()  **#read from camera and store it in a**

cv2.imshow('frame',img)  **#show the read image on the window**

cv2.imwrite('save.jpg',img)  **#write the image with a .jpg filename**

cv2.waitKey(0)  **#wait for a key to be pressed**

break **#exit from loop**

cap.release()  **#release the camera**

cv2.destroyAllWindows()

**5. Click a selfie using webcam with a key- press and save it**

## ***Hint : for getting inputs from keyboard***

k = cv2.waitKey(5) & 0xFF **# waits for a key to be pressed and stores in k** if k==ord('c'): **# if key k is pressed**

*//write your code to capture image*

elif k == ord(‘q’):  **#if key q is pressed**

*//write code to exit and release camera*

***Extended Task :***

**Save 20 selfies with different names Starting from DSC0000 till DSC0020**

**6. Draw a line, circle,rectangle on an image**

*// import modules*

img=cv2.imread('img4.jpg')

**#To draw a line, you need to pass starting and ending coordinates of line.**

**# Draw a diagonal blue line with thickness of 5 px**

cv2.line(img,(0,0),(511,511),(255,0,0),5)

**#To draw a rectangle, you need top-left corner and bottom-right corner of rectangle. This time we will draw a green rectngle at the top-right corner of image.**

cv2.rectangle(img,(384,0),(510,128),(0,255,0),3)

**#To draw a circle, you need its center coordinates and radius.**

cv2.circle(img,(447,63), 63, (0,0,255), -1)

**#Font type (Check cv2.putText() docs for supported fonts)**

font = cv2.FONT\_HERSHEY\_SIMPLEX

cv2.putText(img,'Heee heee!',(10,400), font, 2, (200,255,155), 12, cv2.CV\_AA)

cv2.imshow('image',img)

cv2.waitKey(0)

cv2.destroyAllWindows()

***Extended Task :***

Open an image named img3.jpg and write a code to put an interesting caption on it .The caption has to come inside a box and has to be underlined.

**Resize image command:**

**dst\_name = cv2.resize(src\_name , (600,400))**

**7. ROI (region of image)**

*// import modules*

img=cv2.imread('img5.jpg')

orig=cv2.imread('img5.jpg')

part = img[50:130, 360:420] **#img[y1:y2, x1:x2]**

rb=cv2.flip(part,1)

img[50:130, 70:130] = rb

cv2.imshow('original',orig)

cv2.imshow('Roi',img)

cv2.imwrite('edit1.jpg',img)

cv2.waitKey(0)

cv2.destroyAllWindows()

**Extended Task :**

Open the image named ‘messi.jpg’ and Crop the ball from the image and place it somewhere

**8.Color detection**

*// import modules*

cap = cv2.VideoCapture(1)  
while(1):  
  **#capture a frame**  
 ret, frame = cap.read()

**#convert bgr to hsv**  
 hsv = cv2.cvtColor(frame, cv2.COLOR\_BGR2HSV)  
 ***# define range of desired color in HSV***  
 lower\_orange = np.array([0,144,215])

upper\_orange = np.array([34,255,255])

**#threshold the hsv image to get desired color**  
 mask = cv2.inRange(hsv, lower\_orange, upper\_orange)

***# Bitwise-AND mask and original image***

res = cv2.bitwise\_and(frame,frame, mask= mask)  
 ***# morphological operations***  
kernel = np.ones((5,5),np.uint8)

erosion = cv2.erode(mask,kernel,iterations = 1)  
 dilation = cv2.dilate(mask,kernel,iterations = 1)

opening = cv2.morphologyEx(mask, cv2.MORPH\_OPEN, kernel)  
 closing = cv2.morphologyEx(mask, cv2.MORPH\_CLOSE, kernel)

cv2.imshow('Erosion',erosion)  
 cv2.imshow(‘dilation',dilation)  
 cv2.imshow('Original',frame)  
 cv2.imshow('Mask',mask)  
 cv2.imshow('Opening',opening)  
 cv2.imshow('Closing',closing)  
  
 k = cv2.waitKey(5) & 0xFF  
 if k == 27:  
 break  
  
cv2.destroyAllWindows()  
cap.release()

**9.Simple Code for tracking a coloured object :**

(Here we are tracking an Orange TT ball)

import cv2

import numpy as np

import cv2.cv as cv

import time

import serial

cap = cv2.VideoCapture(0)

while(1):

***#capture a frame***

ret, frame = cap.read()

***#convert bgr to hsv***

hsv = cv2.cvtColor(frame,cv2.COLOR\_BGR2HSV)

***# define range of desired color in HSV***

lower\_orange = np.array([0,144,215])

upper\_orange = np.array([34,255,255])

***#threshold the hsv image to get desired color***

mask = cv2.inRange(hsv, lower\_orange, upper\_orange)

***# Bitwise-AND mask and original image***

res = cv2.bitwise\_and(frame,frame, mask= mask)

***# morphological operations***

kernel = np.ones((5,5),np.uint8)

erosion = cv2.erode(mask,kernel,iterations = 1)

dilation = cv2.dilate(mask,kernel,iterations = 1)

opening = cv2.morphologyEx(mask, cv2.MORPH\_OPEN, kernel)

closing = cv2.morphologyEx(mask, cv2.MORPH\_CLOSE, kernel)

***# Detect circles using HoughCircles***

circles = cv2.HoughCircles(closing,cv.CV\_HOUGH\_GRADIENT,2,120,

param1=100,param2=55,minRadius=10,maxRadius=0)

***#Draw Circles***

if circles is not None:

for i in circles[0,:]:

***# If the ball is far, draw it in green***

cv2.circle(frame,(int(round(i[0])),int(round(i[1]))),int(round(i[2])),(255,0,0),4)

cv2.circle(frame,(int(round(i[0])),int(round(i[1]))),1,(0,255,0),4)

cv2.imshow('tracking',frame)

cv2.imshow('Original',frame)

cv2.imshow('Mask',mask)

cv2.imshow('Result',res)

k = cv2.waitKey(5) & 0xFF

if k == 27:

break

cv2.destroyAllWindows()

cap.release()

**10 . Code for finding HSV values using trackbars :**

import cv2

import numpy as np

***#function to apply HSV value***

def getthresholdedimg(hsv):

threshImg

=cv2.inRange(hsv,np.array((cv2.getTrackbarPos('Hue\_Low','Trackbars'),cv2.getTrackbarPos('Saturation\_Low','Trackbars'),cv2.getTrackbarPos('Value\_Low','Trackbars'))),np.array((cv2.getTrackbarPos('Hue\_High','Trackbars'),cv2.getTrackbarPos('Saturation\_High','Trackbars'),cv2.getTrackbarPos('Value\_High','Trackbars'))))

return threshImg

***#function to get present trackbar value***

def getTrackValue(value):

return value

c = cv2.VideoCapture(0)

width,height = c.get(3),c.get(4)

print "frame width and height : ", width, height

***#Create trackbars***

cv2.namedWindow('Output')

cv2.namedWindow('Trackbars', cv2.WINDOW\_NORMAL)

cv2.createTrackbar('Hue\_Low','Trackbars',0,255, getTrackValue)

cv2.createTrackbar('Saturation\_Low','Trackbars',0,255, getTrackValue)

cv2.createTrackbar('Value\_Low','Trackbars',0,255, getTrackValue)

cv2.createTrackbar('Hue\_High','Trackbars',0,255, getTrackValue)

cv2.createTrackbar('Saturation\_High','Trackbars',0,255, getTrackValue)

cv2.createTrackbar('Value\_High','Trackbars',0,255, getTrackValue)

cv2.createTrackbar('Caliberate','Trackbars',0,1, getTrackValue)

while(1):

\_,f = c.read()

f = cv2.flip(f,1)

***# Convert BGR to HSV***

hsv = cv2.cvtColor(f,cv2.COLOR\_BGR2HSV)

thrImg = getthresholdedimg(hsv)

***#Morphological operations***

erode = cv2.erode(thrImg,None,iterations = 3)

dilate = cv2.dilate(erode,None,iterations = 10)

***#finding contours***

contours,hierarchy = cv2.findContours(dilate,cv2.RETR\_LIST,cv2.CHAIN\_APPROX\_SIMPLE)

for cnt in contours:

x,y,w,h = cv2.boundingRect(cnt)

cx,cy = x+w/2, y+h/2

cv2.rectangle(f,(x,y),(x+w,y+h),[0,0,255],2)

if(cv2.getTrackbarPos('Caliberate','Trackbars') == 1):

cv2.imshow('Output',thrImg)

else:

cv2.imshow('Output',f)

if cv2.waitKey(10) & 0xFF == ord('q'):

break

cv2.destroyAllWindows()

cap.release()

**11. To find the Mouse Coordinates :**

import numpy as np

import cv2

#function to print mouse coordinates

def mousePosition(event,x,y,flags,param):

if event == cv2.EVENT\_MOUSEMOVE:

print x,y

img = cv2.imread('hawk.jpg')

cv2.imshow('image',img)

cv2.setMouseCallback('image',mousePosition)

cv2.waitKey(0)

cv2.destroyAllWindows

**Python – Arduino Communication**

**4a. Python Program**

import numpy as np

import cv2 as cv

import serial

import time

#Initializing the device the for communication

ser = serial.Serial('/dev/ttyUSB0', 9600)

while(1):

#for example if you want to send tha data a, b , c

a=100

b=200

c=300

output = "X{0:d}Y{1:d}Z{2:d}".format(a,b,c)

ser.write(output)

**4b. Arduino program**

int x,y,z;

void setup()

{

Serial.begin(9600);

}

void loop()

{

if (Serial.available() > 0)

{

if (Serial.read() == 'X')

{

x = Serial.parseInt();

if (Serial.read() == 'Y')

{

y = Serial.parseInt();

if (Serial.read() == 'Z')

{

z = Serial.parseInt();

}

}

}

// Put your Code Here

while (Serial.available() > 0)

{

Serial.read();

}

}

}

The above programs are just examples. You may need to send information of different data types.

**Extended Task :**

Write an open cv program to detect wether an object of a given color is to the right or left of a reference point and correspondingly blink One LED for Right and two LEDs for Left. Pass information from python to arduino to achieve this.