Color - Recognition

Indian Institute of Technology, Delhi

Hauz Khas, New Delhi, 110016

Project Report

for

ELL 205: Signals and Systems

Semester 2nd, 2021-22

Slot no. 2

Team Members: -

PRAHLAD PALLAV, 2019PH10647

ALOK YADAV, 2019PH10614

Submitted to: -

Prof. Abhishek Dixit

Assistant Professor

Department of Electrical Engineering

IIT Delhi

ELL 205 Project – Color Recognition

Color Recognition

Color recognition is the process of detection of any color as per our need.

By recognizing different color, we can do different sorts of things and it has many real-life applications.

Libraries

Libraries in programming languages are collections of prewritten code that users can use to optimize tasks.

In this project, we have use two libraries – OpenCV and NumPy.

OpenCV -> Open-Source Computer Vision Library

OpenCV is a huge open-source library for computer vision, machine learning, and image processing. OpenCV supports a wide variety of programming languages like Python, C++, Java, etc. It can process images and videos to identify objects, faces, or even the handwriting of a human.

Features of OpenCV

- ➔ Open Source
- ➔ Optimized
- ➔ Cross-Platform

NumPy

NumPy is a general-purpose array-processing package. It provides a high-performance multidimensional array object, and tools for working with these arrays.

Why NumPy?

- ➔ Very similar to python list and it works more efficient and faster way.
- ➔ We can quick built advanced functionalities such as Tensorflow, Pandas.

Different Color Format

RGB

When it comes to digital image processing and OpenCV, the RGB color model is by far the most often employed. There are three channels in a color picture. Each color has its own channel. The primary colors of this model are red, green, and blue. Each color is created by varying the proportions of these three primary colors. The darker the value, the greater the intensity of the color.

RGB

Red: 0 – 255

Green: 0 – 255

Blue: 0 – 255

HSV

Three channels comprise the picture. Three channels exist: Hue, Saturation, and Value. This color model does not make direct use of basic colors. It makes use of color in the manner that people see it. When a cone is used to symbolize the HSV color.

In RGB, we cannot separate color information from luminance. HSV or Hue Saturation Value is used to separate image luminance from color information.

- Hue, saturation, and value are the main color properties that allow us to distinguish between different colors.
- Hues are the three primary colors (red, blue, and yellow) and the three secondary colors (orange, green, and violet) that appear in the color wheel or color circle.
- Color saturation is the purity and intensity of a color as displayed in an image.
- Color value refers to the relative lightness or darkness of a color.

Color Filter, Color Space

 # HSV

Hue: 0 – 180

Saturation: 0 – 255

Value: 0 – 255

 # RGB

Red: 0 – 255

Green: 0 – 255

Blue: 0 – 255

Function Used:

VideoCapture() -> Record real time video from our webcam.

read() -> It is a function used to read image in which we pass file as an argument.

cv2.cvtColor(frame, cv2.COLOR_BGR2HSV) -> convert the frame in BGR to HSV.

numpy.array() -> returns an ndarray. The ndarray is an array object which satisfies the specified requirements.

cv2.inRange() -> returns an array consisting of elements equal to 0 if the elements of the source array lie between the elements of the two arrays representing the upper bounds and the lower bounds.

cv2.bitwise_and() -> read two images using imread() function and then merge the given two images using bitwise_and operator and then display the resulting image as the output on the screen.

Imshow() -> It is a function used to show image as output in which we passes two argument, the first one the name of the frame, and second one is the name of the matrix (name of the variable in which imread() Is called).

release() -> the camera port got released, no longer in work.

Waitkey() -> It is a function that is used to hold the image on the screen until any key is pressed.

Destroyallwindows() -> It with hold all the window until any key is pressed.

Steps followed:

Step 1

Import cv2 and numpy.

Step 2

VideoCapture() -> Record real time video from our webcam.

Step 3

Reads the input and stored in frame.

Step 4

cv2.cvtColor(frame, cv2.COLOR_BGR2HSV) -> convert the frame in BGR to HSV.

Step 5

Define the upper and lower bound to create mask.

Step 6

Use bitwise AND operator between frame and mask.

Step 7

imshow() -> Return the output.

Step 8

waitKey() -> An interrupt is required to break out the function.

Step 9

release() -> the camera port got released, no longer in work.

Step 10

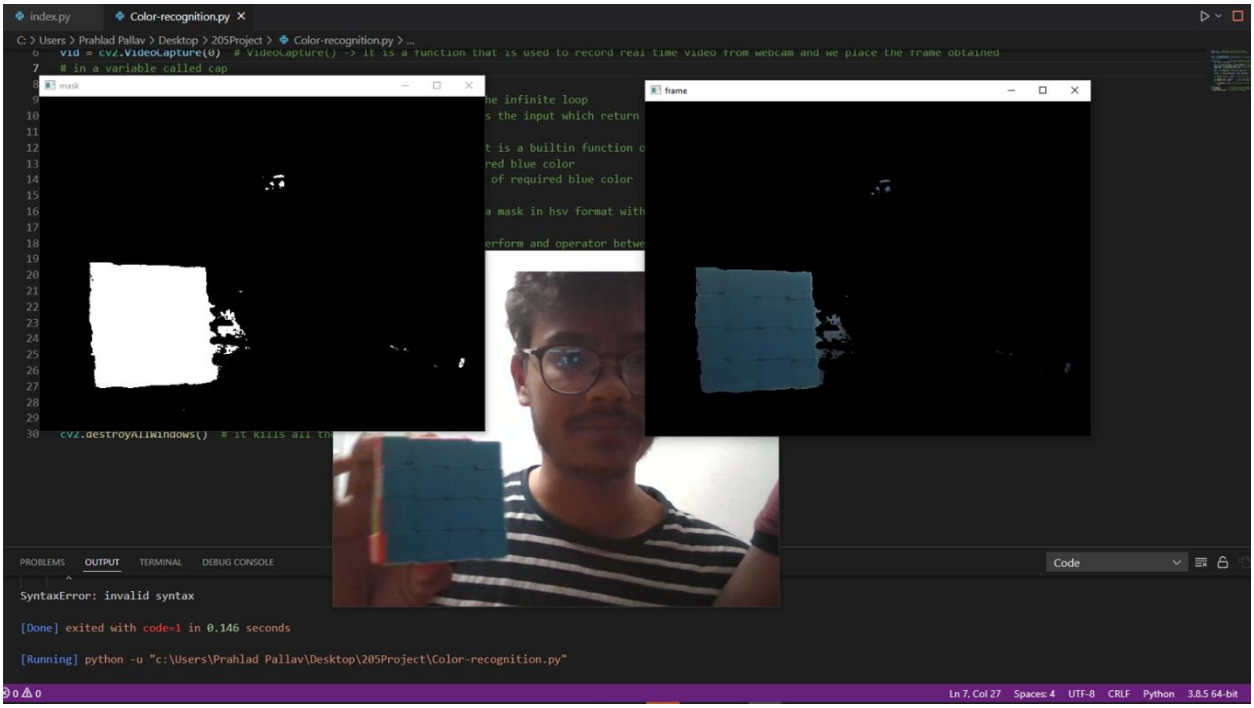
destroyAllWindows()-> it kills all the window.

Code:

C: > Users > Prahlad Pallav > Desktop > 205Project > Color-recognition.py > ...

```python

import cv2    # importing cv2 library from openCv
import numpy as np # importing np library from numpy


vid = cv2.VideoCapture(0)  # VideoCapture() -> It is a function that is used to record real time video from webcam and we place the frame obta
# in a variable called cap

while True:               # For the implemention of function for the infinite loop
    ret, frame = vid.read() # read() -> It is a function that reads the input which return tuples as output.

    hsv = cv2.cvtColor(frame, cv2.COLOR_BGR2HSV) # cvtColor() -> It is a builtin function of openCv which converts input into different format
    lower_blue = np.array([90,50,50]) # It is lower bound of required blue color
    upper_blue = np.array([130, 255, 255]) # It is the upper bound of required blue color

    mask = cv2.inRange(hsv, lower_blue, upper_blue)  # we created a mask in hsv format with defined color.

    result = cv2.bitwise_and(frame, frame, mask=mask)  # it will perform and operator between frame and mask.

    cv2.imshow("Original", frame)  # return the original frame

    cv2.imshow('frame', result)     # return the frame after implementing bitwise operator
    cv2.imshow('mask', mask)       # return the inRange frame.

    if cv2.waitKey(1) == 13:  # An interrupt is required to break out the function -> 13 -> ASCII CODE for Enter
        break

vid.release()            # the camera port got released, no longer in work.
cv2.waitKey(0)           # it helps to retrain the output until an interrupt is performed.
cv2.destroyAllWindows()  # it kills all the window
```

Result:



Some Real-world Applications

- Face Recognition.
- Biometric Detection.
- Medical Diagnosis.
- In self-driving car, to detect the traffic signals.