Họ và tên: Trần Trung Tín

MSSV: 19522351

Lớp: CS338.M21.KHCL

**COLOR SPACE TRANSFORMATION**

1. **Methods**
2. **Changing the Color-space**

There are more than 150 color-spaces conversion methods available in OpenCv. But we will look into only two, which are most widely used ones: BGR ↔ Gray and BGR ↔ HSV.

For color conversion, we use the function cv.cvtColor(input\_image, flag) where flag determines the type of conversion. But here we might need just 1-2 flags. But if you want to use more others flags this is what you shuld type in the terminal to get more flags

import cv2 as cv

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

[print](https://docs.opencv.org/4.x/df/d57/namespacecv_1_1dnn.html#a43417dcaeb3c1e2a09b9d948e234c366)( flags )

1. **Object tracking**

We know how to convert a BGR image to HSV, we can use this to extract a colored object  In HSV, it is easier to represent a color than in BGR color-space. In our application, we will try to extract a yellow and white colored object. So here is the method:

* Take each picture from our folder
* Convert them from BGR2RGB then convert them again from RGB2HSV (explaining for this I think that cv2 when they read a image they just automatic change their color-space into BGR so you will need to change it to normal)
* We threshold the HSV image for a range of that specific color
* Now extract the blue object alone, we can do whatever we want on that image.

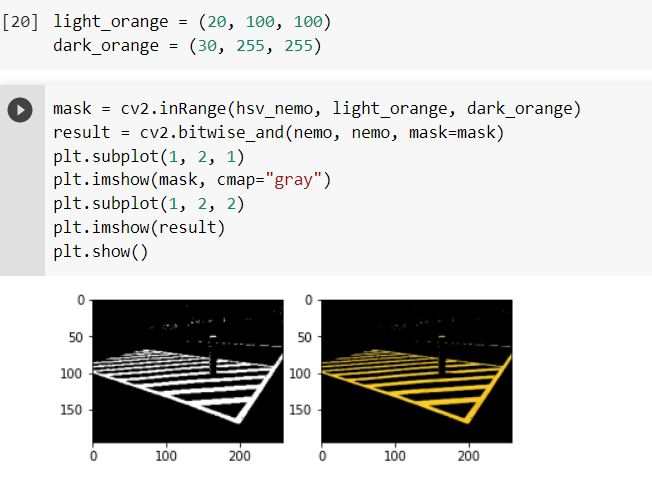
1. **Result**



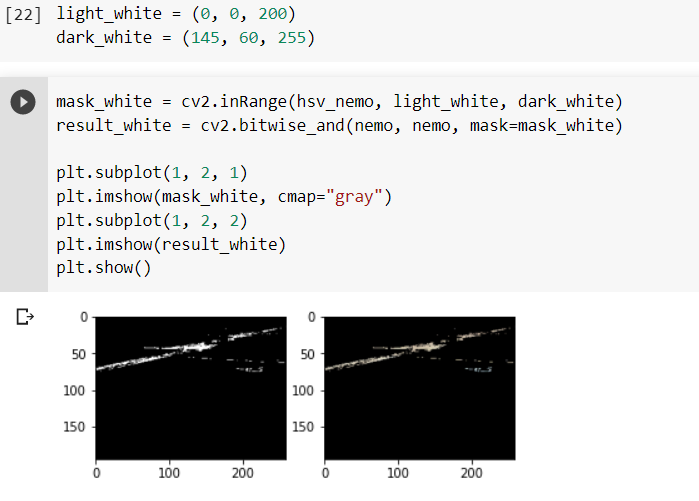
* We read the sample images first



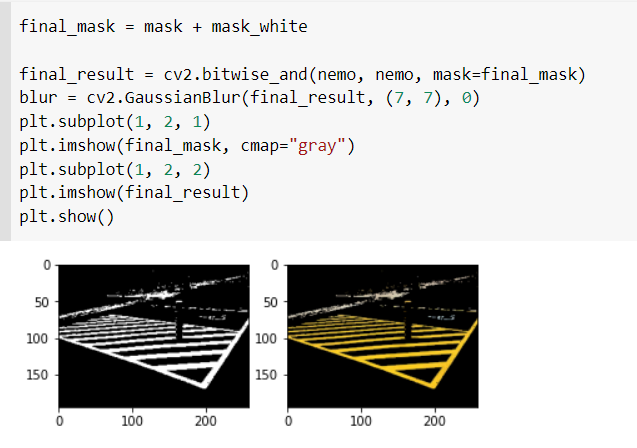
* We convert it to RGB Color-spaces and then convert one more time to HSV Color-Space



* We then set 1 threshold for the first color is orange (set the upper bound and lower bound for this color ) . Then we extract the orange line alone using **inRange ()** and **bitwise\_and ()** functions.

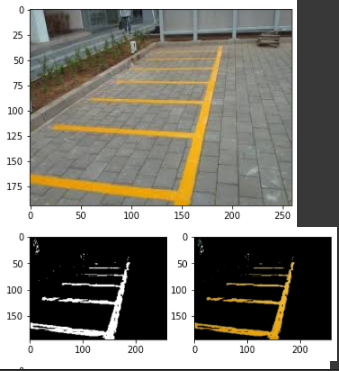


* We then do the same with the white object. Finally combine them all together

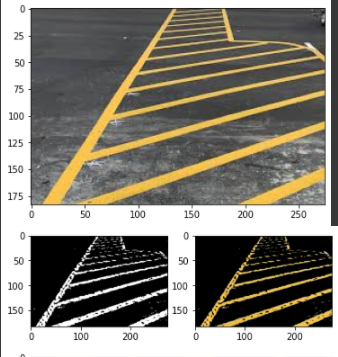


* Do the same for all images :

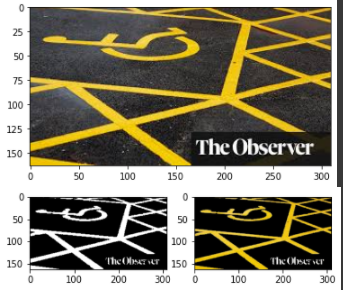
**1.**



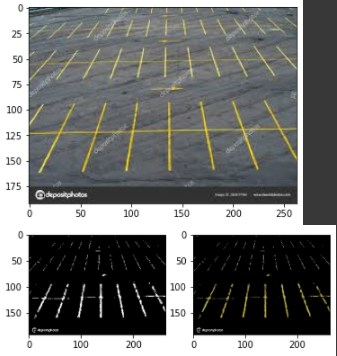
**2.**



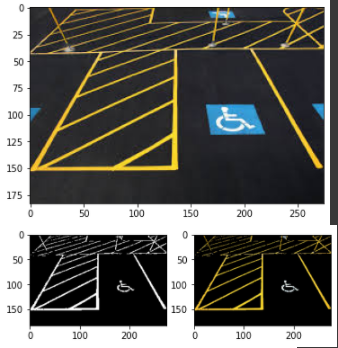
**3.**



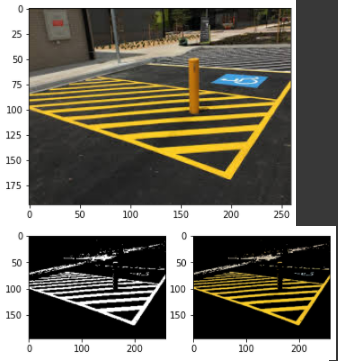
**4.**



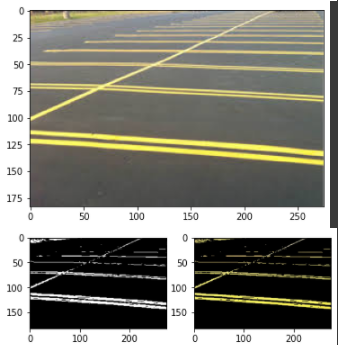
**5.**



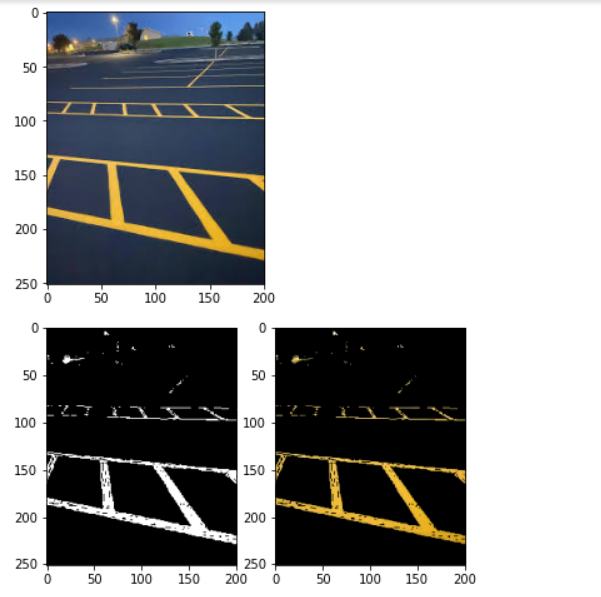
**6.**



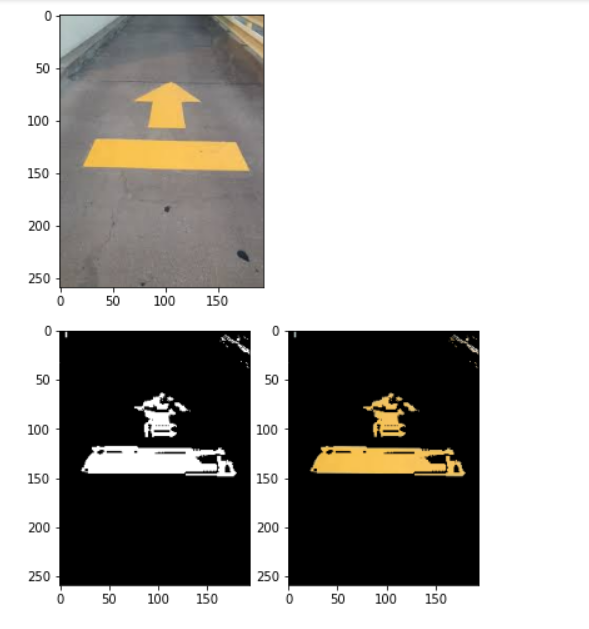
**7.**



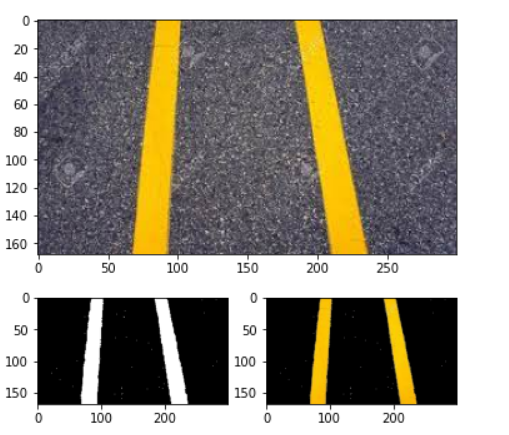
**8.**



**9.**



**10.**



1. **Advantages and Disadvantages**
2. **Advantages**

* Really fast, easy to understand, easy to implement.
* Simple than Deep Learn not so complicated like other mothods.
* High efficiency in detect some simple color for a lót of images.
* Because it easier and less complicated so it consume less resources and that’s good for your computer.

1. **Disadvantages**

* Have to change the threshold every single time we detect other pictures.
* Not so good with images that contain many different colors.

**Source code:**

<https://colab.research.google.com/drive/1d_2VCwl_5TvP_pImIi07ke_f1XcChgzN?usp=sharing>