

BÀI TẬP SEGMENTATION COLOR SPACES BASED

Trương Chí Diễm – 19520464

CS338.M22.KHCL

Nội dung

1. Segmentation cá hề 1
2. Thực hành nhận diện Parking slot line 6

1. Segmentation cá hề

>>> Đọc ảnh và convert màu sang RGB (mặc định của OpenCV là BGR)

```
nemo = cv2.cvtColor(cv2.imread('Clown_fish.jpg'), cv2.COLOR_BGR2RGB)
plt.imshow(nemo)
plt.show()
```



>>> Chuyển hình ảnh thành hệ màu HSV

```
hsv_nemo = cv2.cvtColor(nemo, cv2.COLOR_RGB2HSV)
```

>>> Chọn khoảng màu của màu cam

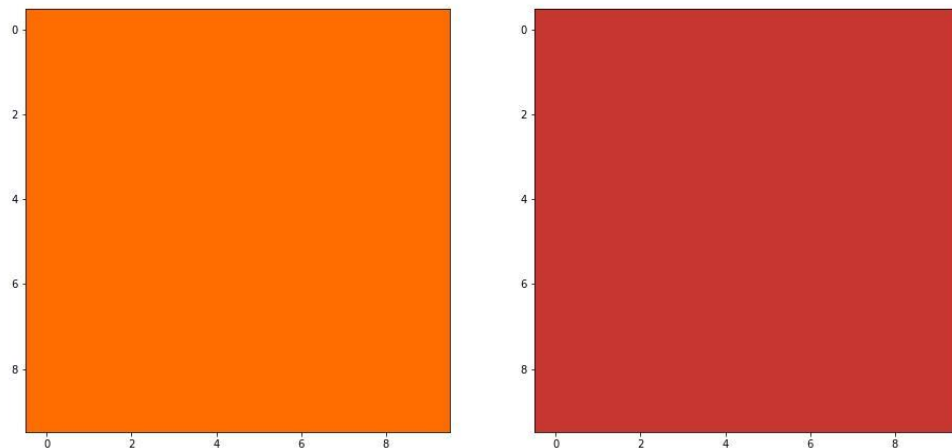
```
light_orange = (1, 190, 200)
dark_orange = (18, 255, 255)
```

>>> Tạo 2 ma trận với giá trị của 2 khoảng màu

```

lo_square = np.full((10, 10, 3), light_orange, dtype=np.uint8) / 255.0
do_square = np.full((10, 10, 3), dark_orange, dtype=np.uint8) / 255.0
plt.figure(figsize=(16, 8))
plt.subplot(1, 2, 1)
plt.imshow(hsv_to_rgb(do_square))
plt.subplot(1, 2, 2)
plt.imshow(hsv_to_rgb(lo_square))
plt.show()

```

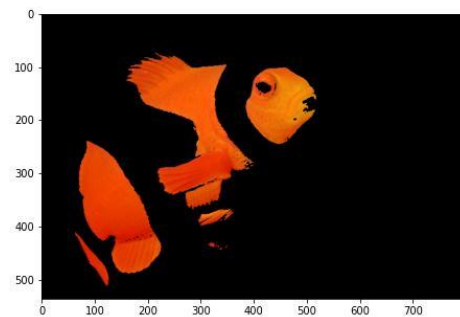
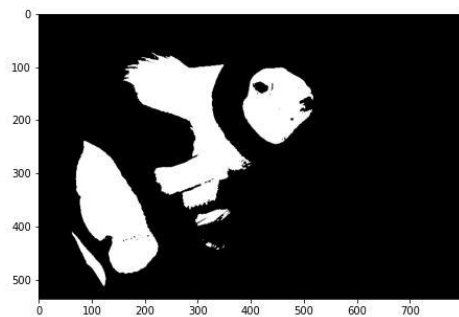


>>> Lấy mask của ảnh là những pixel có giá trị HSV nằm trong khoảng màu

```
mask = cv2.inRange(hsv_nemo, light_orange, dark_orange)
```

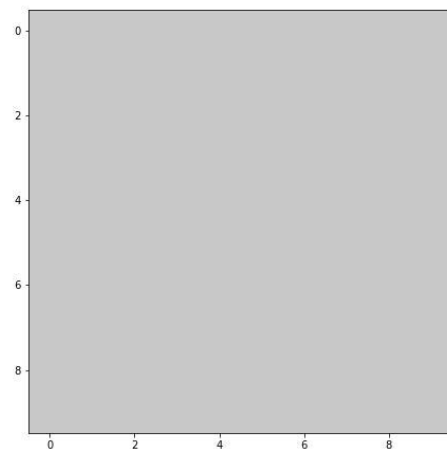
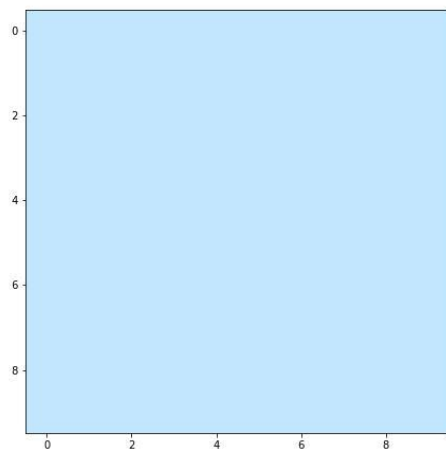
>>> Áp dụng mask lên ảnh gốc để lấy được những phần có màu cam

```
result = cv2.bitwise_and(nemo, nemo, mask=mask)
```

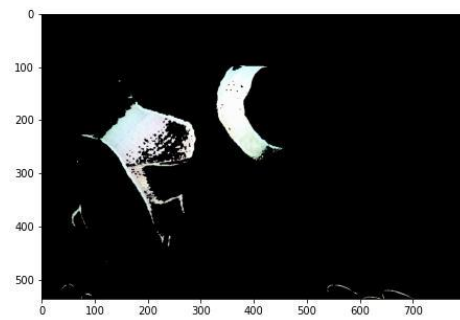
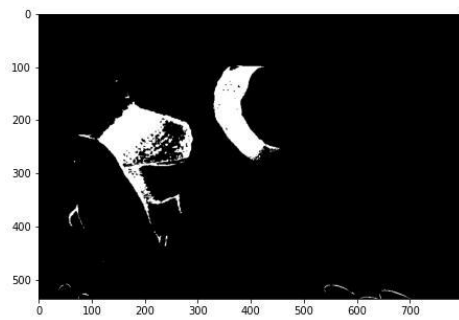


>>> Cá hề có 2 màu, cần lấy thêm màu trắng bằng cách tương tự

```
light_white = (0, 0, 200)
dark_white = (145, 60, 255)
lo_square = np.full((10, 10, 3), light_white, dtype=np.uint8) / 255.0
do_square = np.full((10, 10, 3), dark_white, dtype=np.uint8) / 255.0
```

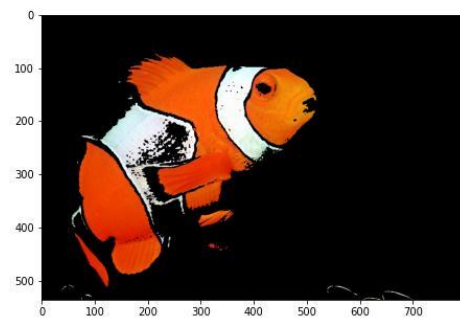
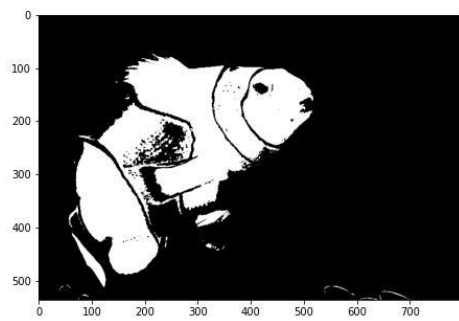


```
mask_white = cv2.inRange(hsv_nemo, light_white, dark_white)
result_white = cv2.bitwise_and(nemo, nemo, mask=mask_white)
```



>>> Cộng 2 phần lại để được kết quả cuối cùng

```
final_mask = mask + mask_white  
final_result = cv2.bitwise_and(nemo, nemo, mask=final_mask)
```

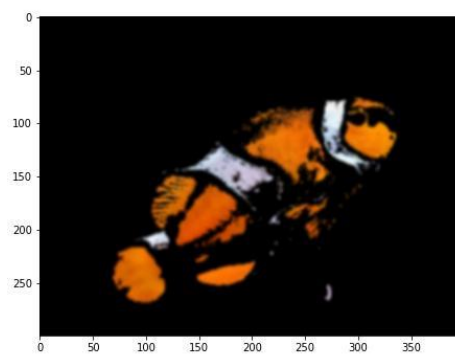
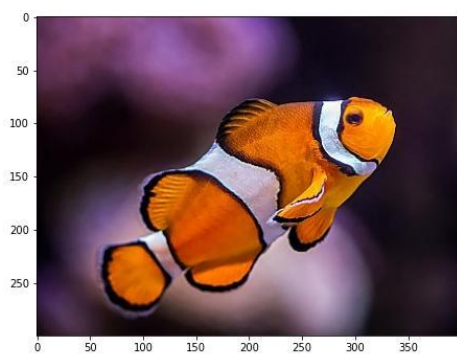
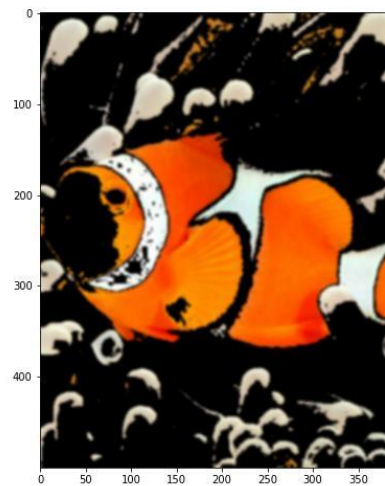
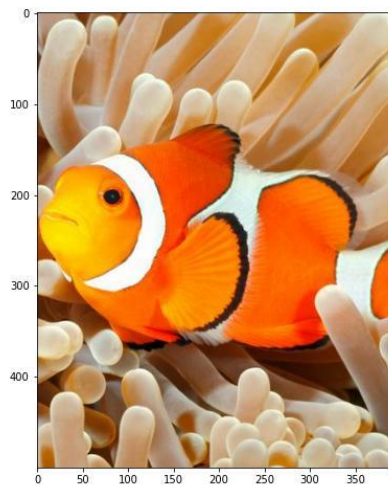


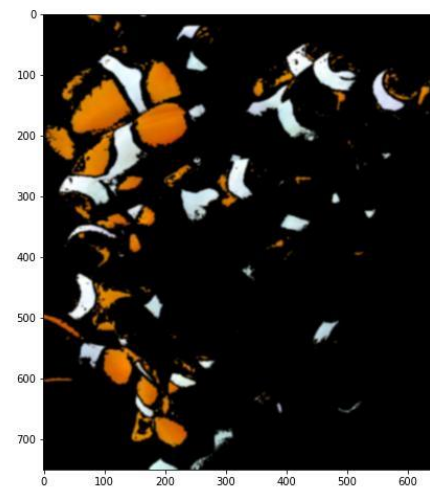
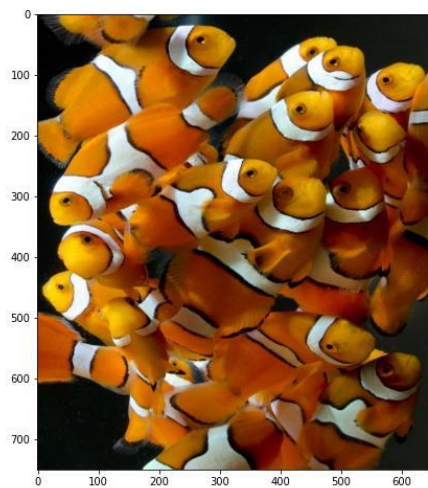
>>> Áp dụng Gaussian Blur để thu được ảnh mịn hơn

```
blur = cv2.GaussianBlur(final_result, (7, 7), 0)
```

>>> Gộp tất cả lại để được 1 hàm

```
def segment_fish(image):
    hsv_image = cv2.cvtColor(image, cv2.COLOR_RGB2HSV)
    light_orange = (1, 190, 200)
    dark_orange = (18, 255, 255)
    mask = cv2.inRange(hsv_image, light_orange, dark_orange)
    light_white = (0, 0, 200)
    dark_white = (145, 60, 255)
    mask_white = cv2.inRange(hsv_image, light_white, dark_white)
    final_mask = mask + mask_white
    result = cv2.bitwise_and(image, image, mask=final_mask)
    blur = cv2.GaussianBlur(result, (7, 7), 0)
    return blur
```

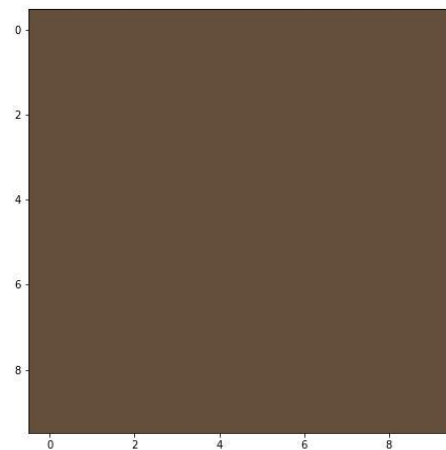
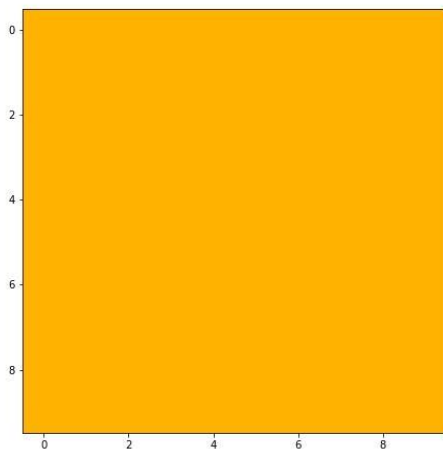




2. Thực hành nhận diện Parking slot line

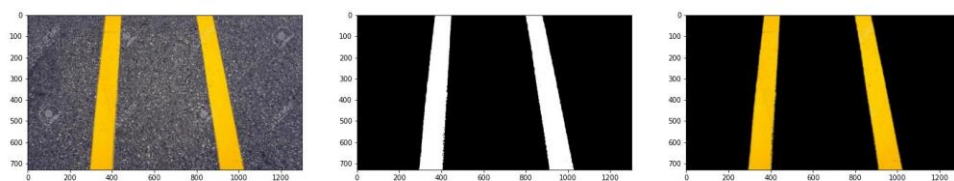
>>> Chọn Threshold cho màu vàng của Parrking slot line

```
light_yellow = (20 , 100, 100)
dark_yellow = (30, 255, 255)
```

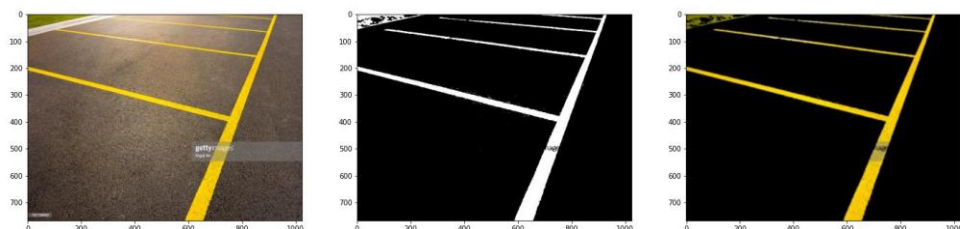


>>> Xây dựng hàm segmentation, vì line chỉ có 1 màu nên chỉ cần segment màu vàng là đủ.

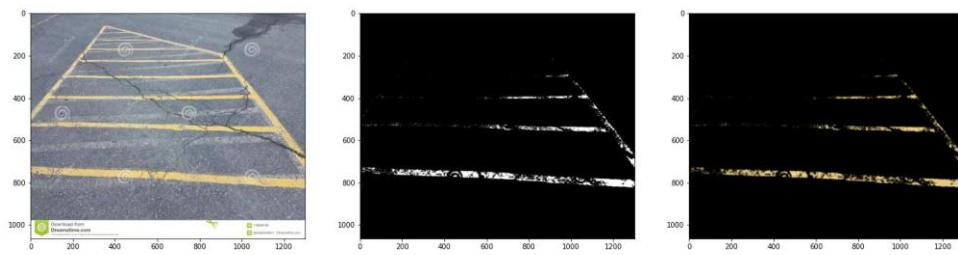
```
def segment_parking_slot(image):
    hsv_image = cv2.cvtColor(image, cv2.COLOR_RGB2HSV)
    light_yellow = (20, 100, 100)
    dark_yellow = (30, 255, 255)
    mask = cv2.inRange(hsv_image, light_yellow, dark_yellow)
    result = cv2.bitwise_and(image, image, mask=mask)
    blur = cv2.GaussianBlur(result, (7, 7), 0)
    return mask, blur
```



Nhận xét: Với ảnh phân tách rõ ràng giữa line và đường, màu sắc không mờ hay bị nhiễu thì thuật toán hoạt động tốt.



Nhận xét: Với ảnh có những objects có màu tương đồng thì có thể dùng phương pháp nhận diện object (phát hiện cạnh) trước khi segmentation.



Nhận xét: Với object mờ, màu nhạt nhạt, nhiều nhiễu cần chọn một threshold khác hoặc phát triển thuật toán có thể tự động điều chỉnh threshold dựa trên điều kiện sáng, tổng quan màu sắc (adaptive), cùng các phương pháp lọc nhiễu để đạt kết quả tốt hơn.