Homework 1.2

June 23, 2023

1 Bài tập 1.2

1.1 Đề bài

Dùng Thresholding để tách phần màu da, trích xuất ra cánh tay từ hình ảnh sau:

1.2 Đáp án gợi ý

1. Gọi thư viện

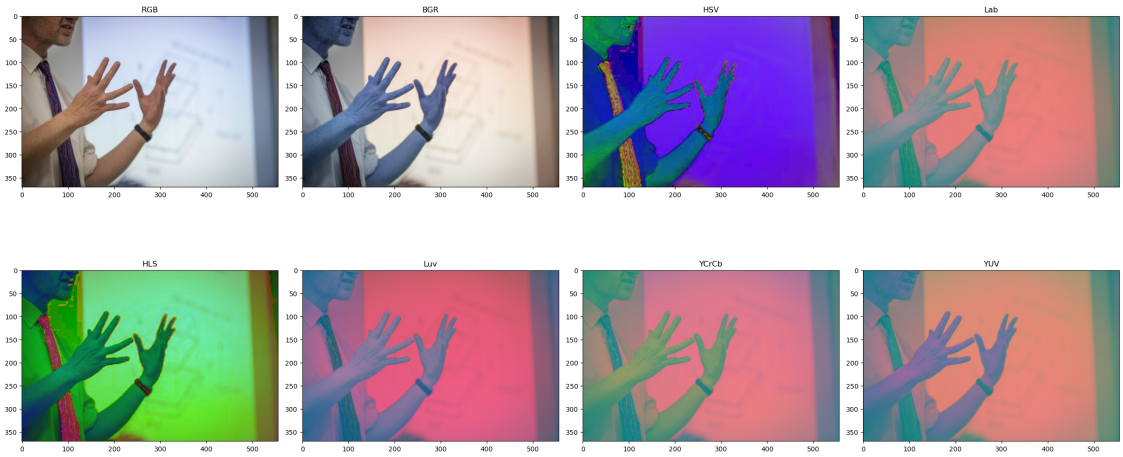
[72]:
```
import numpy as np
import matplotlib.pyplot as plt
import cv2
```

2. Đọc ảnh. Ảnh đọc vào đang ở không gian màu RGB

[73]:
```
img = cv2.imread('./images/Img_2.jpg')
```

3. Phân tích
   Theo như Homework 1, các không gian màu khác sẽ cho thấy sự phân biệt rõ ràng hơn về các màu. Ta thử vẽ ở các không gian màu khác nhau xem như thế nào

[74]:
```
img_rbg = cv2.cvtColor(img,cv2.COLOR_BGR2RGB)
img_hsv = cv2.cvtColor(img,cv2.COLOR_BGR2HSV)
img_lab = cv2.cvtColor(img_rbg, cv2.COLOR_RGB2Lab)
img_hls = cv2.cvtColor(img, cv2.COLOR_BGR2HLS)
img_luv = cv2.cvtColor(img, cv2.COLOR_BGR2Luv)
img_ycrb = cv2.cvtColor(img, cv2.COLOR_BGR2YCrCb)
img_yuv = cv2.cvtColor(img, cv2.COLOR_BGR2YUV)

img_list = [img_rbg, img, img_hsv, img_lab, img_hls, img_luv, img_ycrb, img_yuv]
titles = ["RGB", "BGR", "HSV", "Lab", "HLS", "Luv", "YCrCb", "YUV"]

so_hang = 2
so_cot  = 4

fig, ax = plt.subplots(so_hang, so_cot, figsize = (24,12))
for i in range(so_cot*so_hang):
    ax[i // so_cot, i % so_cot].imshow(img_list[i])
    ax[i // so_cot, i % so_cot].set_title(titles[i])
plt.tight_layout()
```

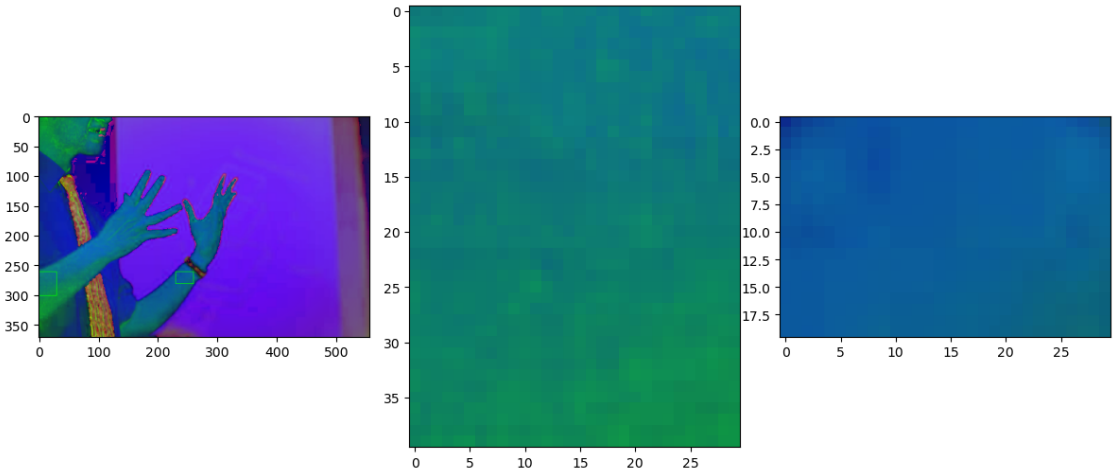


Ta thấy không gian HSV có phân biệt màu sắc và các bộ phận khá rõ. Lưu ý là có thể dùng các không gian màu khác nhau hoặc kết hợp các không gian màu.

Để chọn ngưỡng (Threshold), ta xét ảnh màu HSV và 2 vị trí của tay

[75]:
```
fig, ax = plt.subplots(1 , 3, figsize = (12,6))
a = img_hsv[260:300,0:30]
b = img_hsv[260:280,230:260]
```

```
#
marked_img = img_hsv.copy()
marked_img = cv2.rectangle(marked_img, (0, 260), (30,300),(0,255,0))
marked_img = cv2.rectangle(marked_img, (230, 260), (260,280),(0,255,0))
for i in range(3):
    ax[0].imshow(marked_img)
    ax[1].imshow(a)
    ax[2].imshow(b)
plt.tight_layout()
plt.show()
print(a)
print(b)
```



```
[[[ 14 119 120]
  [ 14 123 120]
  [ 14 119 120]
  …
  [ 13 122 132]
  [ 13 123 131]
  [ 13 125 129]]

 [[ 14 125 120]
  [ 14 123 122]
  [ 14 123 122]
  …
  [ 13 131 123]
  [ 13 126 128]
  [ 13 124 130]]

 [[ 14 137 115]
```

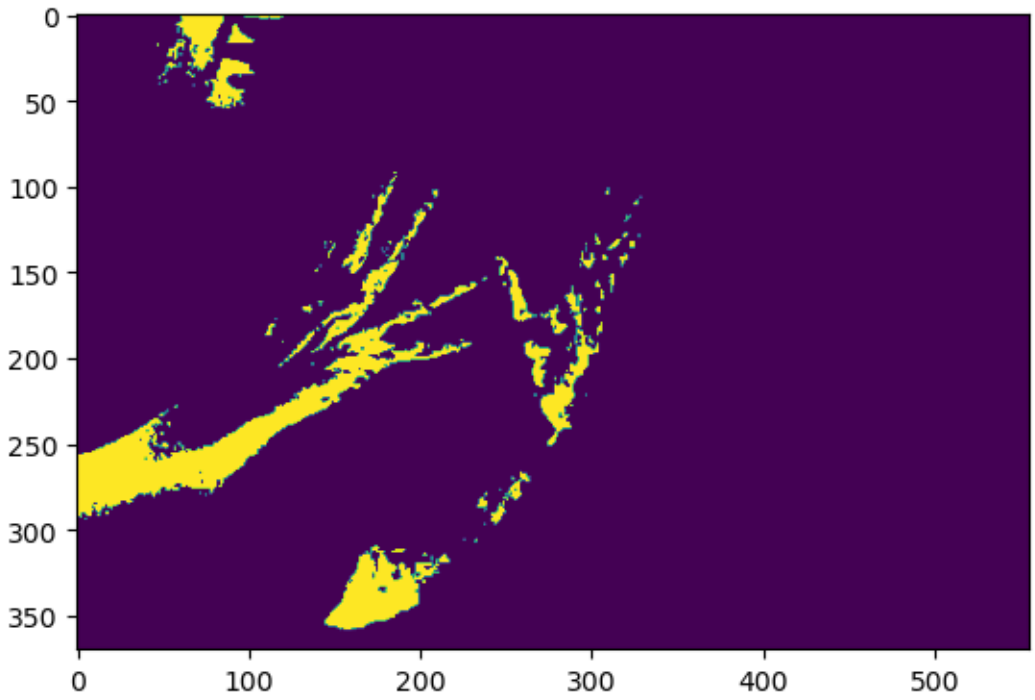
```
  [ 14 133 119]
  [ 14 131 121]
  …
  [ 13 126 128]
  [ 13 122 132]
  [ 13 120 134]]

 …

 [[ 13 132  93]
  [ 13 126  97]
  [ 13 133  92]
  …
  [ 14 138  74]
  [ 14 146  70]
  [ 14 144  71]]

 [[ 12 126  93]
  [ 12 127  92]
  [ 12 135  87]
  …
  [ 14 142  72]
  [ 14 148  69]
  [ 14 146  70]]

 [[ 12 129  91]
  [ 12 135  87]
  [ 12 140  84]
  …
  [ 14 138  74]
  [ 14 146  70]
  [ 14 146  70]]]
[[[ 12  42 140]
  [ 13  58 144]
  [ 12  67 149]
  …
  [ 13  81 158]
  [ 13  82 150]
  [ 15  84 130]]

 [[ 13  59 142]
  [ 12  68 147]
  [ 12  77 152]
  …
  [ 13  91 160]
  [ 13  87 158]
  [ 14  85 147]]
```

```
[[ 12  67 148]
 [ 11  78 153]
 [ 12  85 156]
 …
 [ 13 100 161]
 [ 13  98 159]
 [ 13  90 153]]

…

[[ 11  89 158]
 [ 11  89 158]
 [ 11  89 157]
 …
 [ 13 101 124]
 [ 11 100 125]
 [ 11  97 124]]

[[ 11  89 158]
 [ 11  89 157]
 [ 11  89 158]
 …
 [ 14 103 121]
 [ 13 103 121]
 [ 11  98 122]]

[[ 11  89 158]
 [ 11  89 157]
 [ 11  88 159]
 …
 [ 14 105 119]
 [ 13 105 119]
 [ 11 100 120]]]
```

3. Dựa vào khoảng màu trong 2 hình a và b, ta chọn các ngưỡng cho từng màu:
   $0 < \mathrm{H} < 15$
   $100 < \mathrm{S} < 150$
   $100 < \mathrm{V} < 150$
4. Test với ngưỡng vừa chọn

[76]:
```
img_mask = cv2.inRange(img_hsv,(0,100,100),(15,150,150))
plt.imshow(img_mask)
```

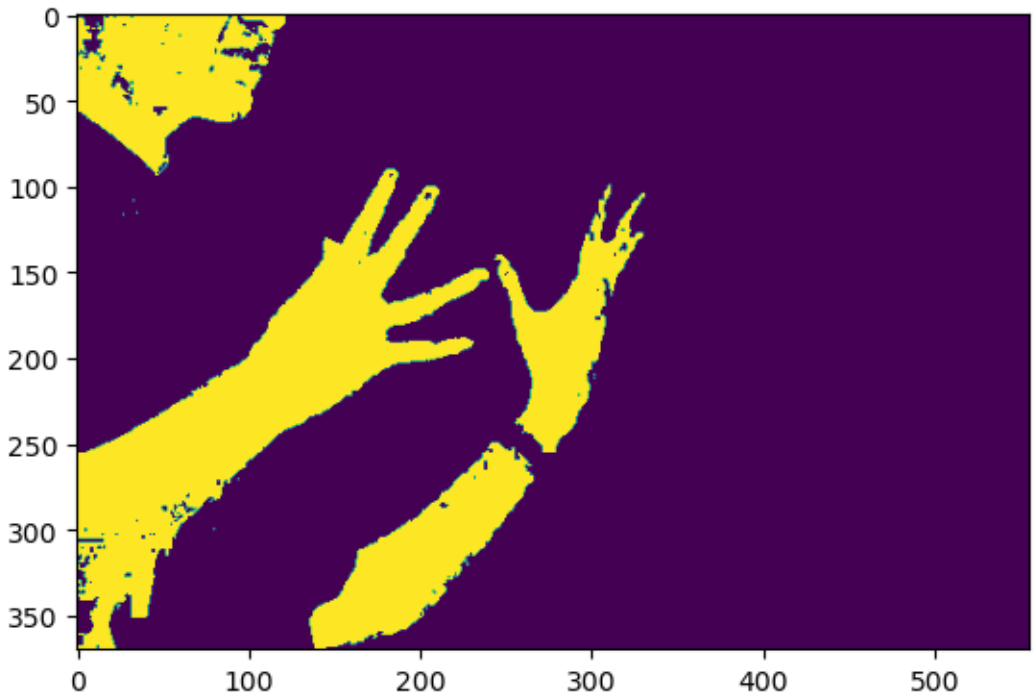
[76]:
```
<matplotlib.image.AxesImage at 0x19a7a22d190>
```



5. Nhận xét và sửa ngưỡng
   Ta cần tăng ngưỡng lên để lấy được nhiều giá trị hơn (lấy cho trọn vẹn bàn tay)
   H: Hue - Tông màu
   S: Saturation - Độ đồng nhất
   V: Giá trị

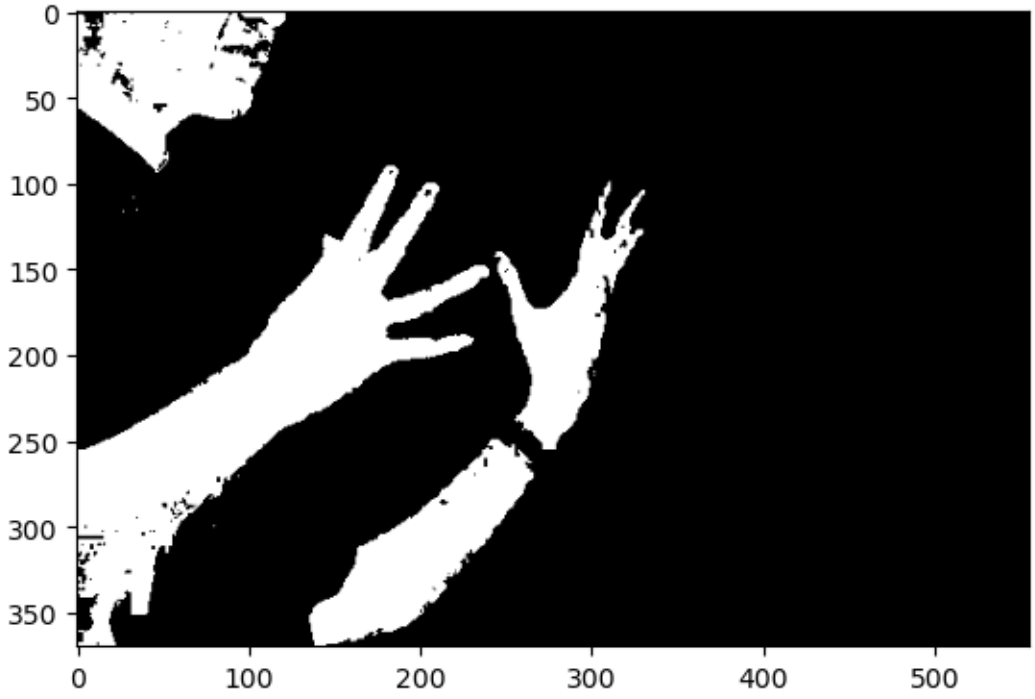
[77]:
```
img_mask = cv2.inRange(img_hsv,(0,80,50),(15,150,255))
plt.imshow(img_mask)
```

[77]:
```
<matplotlib.image.AxesImage at 0x19a7a33c850>
```



[78]:
```
plt.imshow(img_mask, cmap='gray')
```

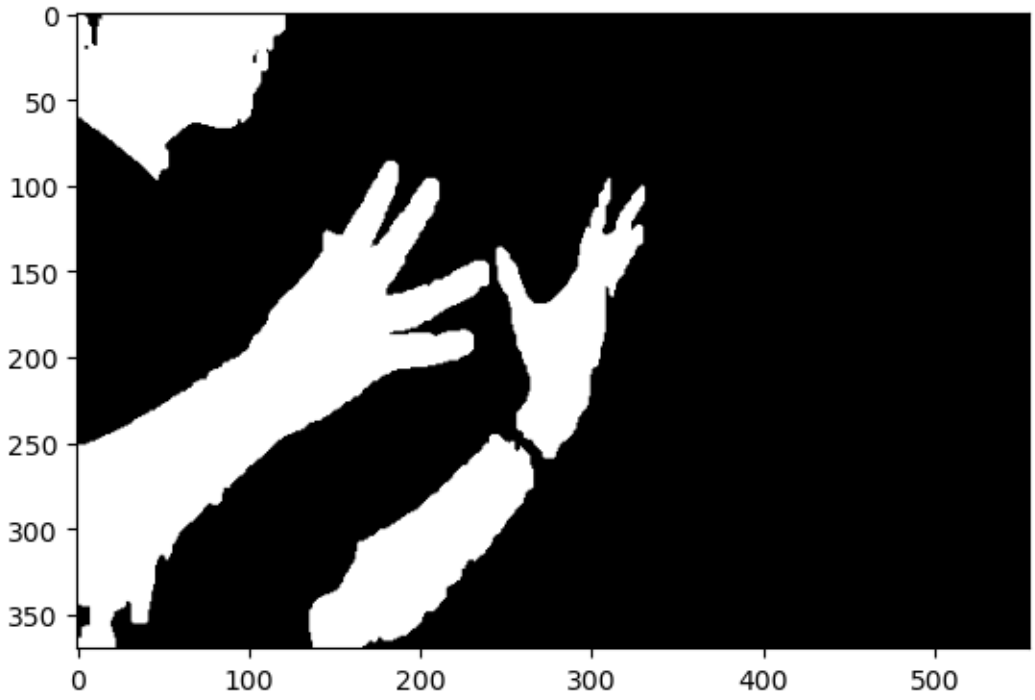
[78]: <matplotlib.image.AxesImage at 0x19a78606790>



6. Dùng các thuật toán Lọc và Morphology để hiệu chỉnh tới khi ưng ý nhất

[79]:
```
kernel = np.ones(9, dtype = np.uint8)
img_mask = cv2.medianBlur(img_mask,3,3)
img_mask = cv2.dilate(img_mask,kernel, 70)
#img_mask = cv2.erode(img_mask,kernel, 5)
plt.imshow(img_mask, cmap='gray')
```

[79]:
```
<matplotlib.image.AxesImage at 0x19a7a1bc700>
```



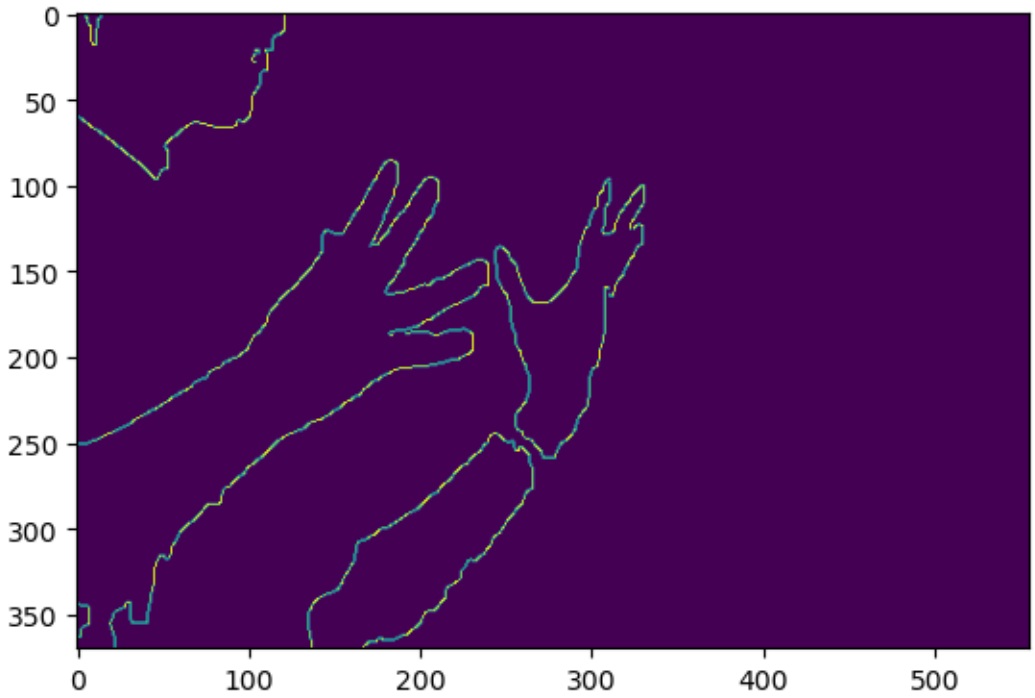
7. (Optional) Thể hiện trên ảnh ban đầu

- Lấy edges
- Lấy contours
- Vẽ các contours đủ lớn

[80]:
```
#img_mask = cv2.GaussianBlur(img_mask,(5,5),0)
edges = cv2.Canny(img_mask,127,254, apertureSize=3)
contours, hierarchy = cv2.findContours(edges,cv2.RETR_EXTERNAL, cv2.
↪CHAIN_APPROX_NONE)
```

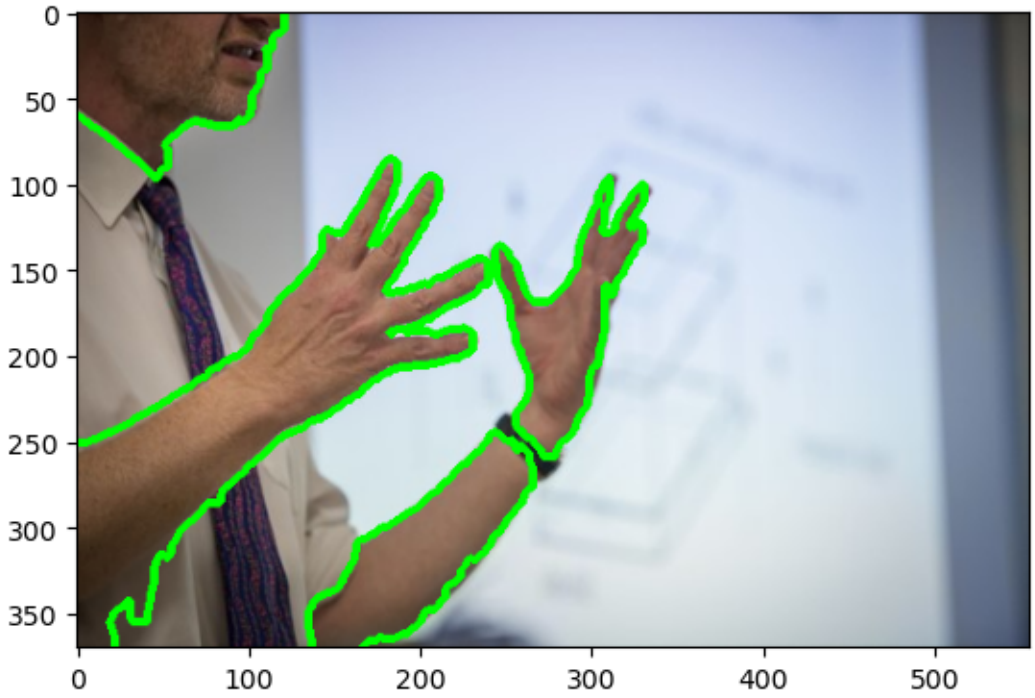
[81]:
```
plt.imshow(edges)
```

[81]: <matplotlib.image.AxesImage at 0x19a7a1e6f40>



[85]:
```
for contour in contours:
    if (cv2.contourArea(contour) > 5):
        cv2.drawContours(img, contour, -1, (0, 255, 0), 3)

plt.imshow(cv2.cvtColor(img,cv2.COLOR_BGR2RGB))
```

[85]: <matplotlib.image.AxesImage at 0x19a02dcaca0>


0
50
100
150
200
250
300
350
0
100
200
300
400
500

[ ]: