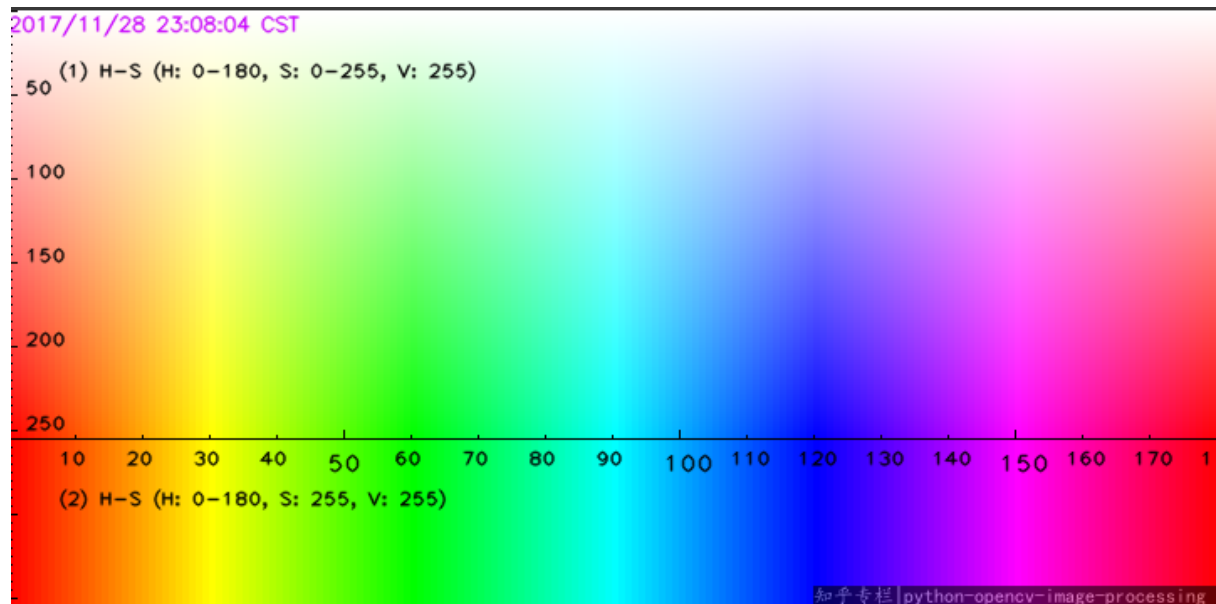


Report Lab1

Part 1 - The red cup

Just to remember, here is the color spectrum:



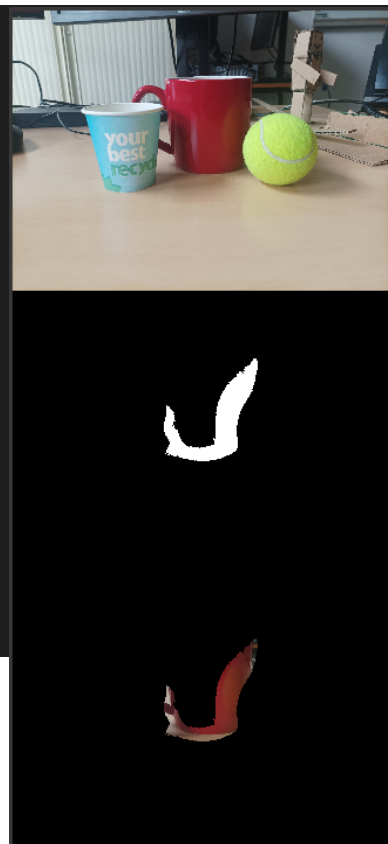
If we only take H: 0 - 10, here is the result we get:

```
# define color segmentation intervals
lower = np.array([0,50,50])
upper = np.array([10,255,255])

# Threshold the HSV image to get only red colors
mask = cv.inRange(hsv, lower, upper)
mask.shape

# Bitwise-AND mask and original image
res = cv.bitwise_and(img_r,img_r, mask=mask)

cv2_imshow(img_r)
cv2_imshow(mask)
cv2_imshow(res)
k = cv.waitKey(0)
```



We are therefore missing a good part of the red from the cup, which corresponds to H: 170 - 180

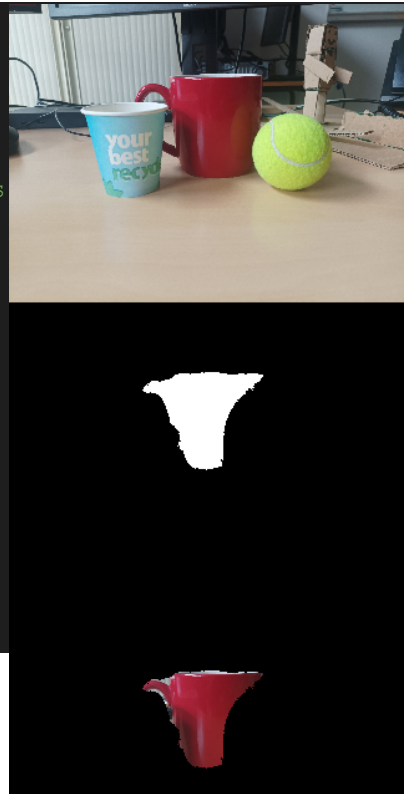
If we only take H: 170 - 180, here is the result we get:

```
# define color segmentation intervals
lower = np.array([170,50,50])
upper = np.array([180,255,255])

# Threshold the HSV image to get only red colors
mask = cv.inRange(hsv, lower, upper)
mask.shape

# Bitwise-AND mask and original image
res = cv.bitwise_and(img_r,img_r, mask=mask)

cv2_imshow(img_r)
cv2_imshow(mask)
cv2_imshow(res)
k = cv.waitKey(0)
```



We observe that this result complements the previous one well. If we group the two together, it works well for images 3.jpg and 2.jpg but for image 1.jpg, we have noise, we can reduce the first interval to 8, here is what we obtain:

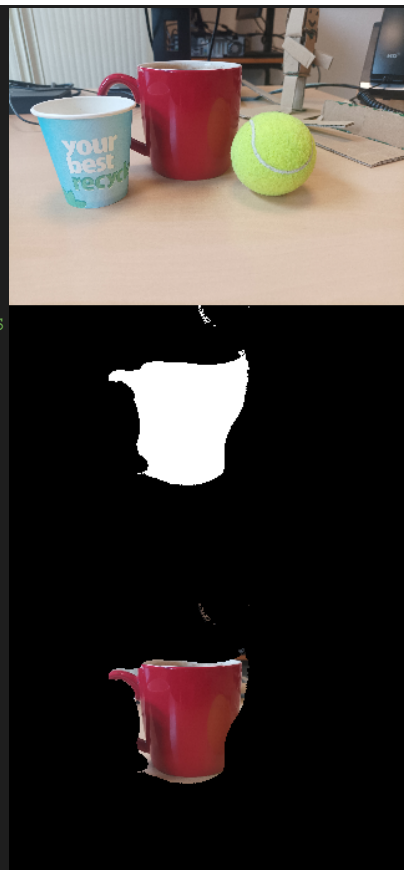
```
# define color segmentation intervals
lower = np.array([0,50,50])
upper = np.array([8,255,255])

lower2 = np.array([170,50,50])
upper2 = np.array([180,255,255])

# Threshold the HSV image to get only red colors
mask = cv.inRange(hsv, lower, upper)
mask2 = cv.inRange(hsv, lower2, upper2)
mask = mask + mask2
mask.shape

# Bitwise-AND mask and original image
res = cv.bitwise_and(img_r,img_r, mask=mask)

cv2_imshow(img_r)
cv2_imshow(mask)
cv2_imshow(res)
k = cv.waitKey(0)
```



Part 2 - The blue cup

We notice that the cup is a light blue so there is no need to take the whole spectrum. On the other hand, it contains green.

Here is the color transformation equations used in the course;

$$V \leftarrow \max(R, G, B)$$

$$S \leftarrow \begin{cases} \frac{V - \min(R, G, B)}{V} & \text{if } V \neq 0 \\ 0 & \text{otherwise} \end{cases}$$

$$H \leftarrow \begin{cases} 60(G - B)/(V - \min(R, G, B)) & \text{if } V = R \\ 120 + 60(B - R)/(V - \min(R, G, B)) & \text{if } V = G \\ 240 + 60(R - G)/(V - \min(R, G, B)) & \text{if } V = B \\ 0 & \text{if } R = G = B \end{cases}$$

If $H < 0$ then $H \leftarrow H + 360$. On output $0 \leq V \leq 1, 0 \leq S \leq 1, 0 \leq H \leq 360$.

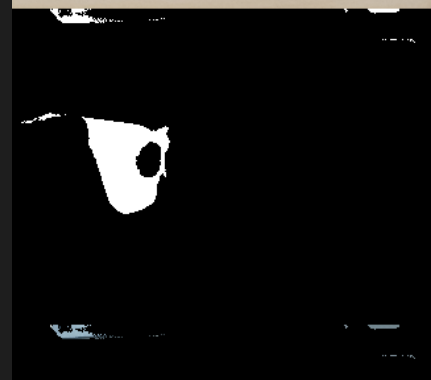
Here are the settings I kept on image 3.jpg:

```
# define color segmentation intervals
lower = np.array([50,38,130])
upper = np.array([100,255,255])

# Threshold the HSV image to get only red color
mask = cv.inRange(hsv, lower, upper)
mask.shape

# Bitwise-AND mask and original image
res = cv.bitwise_and(img_r, img_r, mask=mask)

cv2_imshow(img_r)
cv2_imshow(mask)
cv2_imshow(res)
k = cv.waitKey(0)
```



Here is the final result and only the cup is detected.



During tests on image 1.jpg, we observe that the detected image is smaller but the algorithm still works:

