

Digital Signal Processing Lab

Demo 22 - Exercise 3 (Video processing)

Saad Zubairi
shz2020

November 17, 2025

Solution

For this, I used the color demo D7 - `color_operations/find_blue_in_image.py` as the base. The entire flow (load the sample fruit image, convert to HSV, build a mask, display/save the results) was untouched of course, with the following changes added:

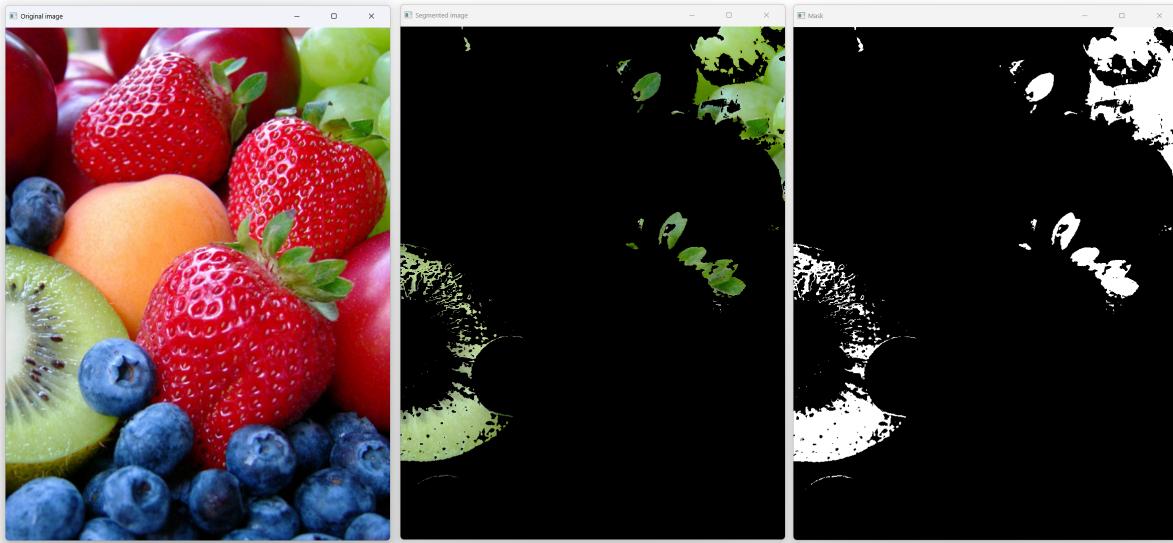
- Replaced the BGR reference color with pure green and reused the same hue-based bandpass idea.
- Clamped the hue limits inside the valid HSV range and renamed the saved images to `fruit_mask_green.jpg` / `fruit_green.jpg`.

The HSV mask isolates green pixels, and `cv2.bitwise_and` paints only those pixels in the detected image.

```
1 img = cv2.imread('fruit.jpg', 1)
2 green = np.uint8([[[0, 255, 0]]])
3 green_hsv = cv2.cvtColor(green, cv2.COLOR_BGR2HSV)
4 h = green_hsv[0,0,0]
5 lower = np.array([max(h-20, 0), 50, 50])
6 upper = np.array([min(h+20, 179), 255, 255])
7 mask = cv2.inRange(img_hsv, lower, upper)
8 output = cv2.bitwise_and(img, img, mask = mask)
```

Snippet 1: Key changes for green detection

Screenshots



Addendum: Full implementation

```
1 import cv2
2 import numpy as np
3
4 img = cv2.imread('fruit.jpg', 1)
5 # Convert to different color space
6 img_hsv = cv2.cvtColor(img, cv2.COLOR_BGR2HSV)
7
8 print(type(img_hsv))
9 print(img_hsv.shape)
10 print(img_hsv.dtype)
11
12 blue = np.uint8([[0, 255, 0]])      # 3D array describing green in BGR
13 blue_hsv = cv2.cvtColor(blue, cv2.COLOR_BGR2HSV)
14 h = blue_hsv[0,0,0]
15 print('Blue in HSV color space:', blue_hsv)
16 print('Hue = ', h)    # see that h = 120
17
18 lower = np.array([max(h-20, 0), 50, 50])
19 upper = np.array([min(h+20, 179), 255, 255])
20 print('lower = ', lower)
21 print('upper = ', upper)
22
23 # Determine binary mask
24 blue_mask = cv2.inRange(img_hsv, lower, upper)
25
26 # Apply mask to color image
27 output = cv2.bitwise_and(img, img, mask = blue_mask)
28
29 # Show images:
30 cv2.imshow('Original image', img)
31 cv2.imshow('Mask', blue_mask)
32 cv2.imshow('Segmented image', output)
33
34 print('Switch to images. Then press any key to stop')
35
36 cv2.waitKey(0)
37 cv2.destroyAllWindows()
38
39 # Write the image to a file
40 cv2.imwrite('fruit_mask_green.jpg', blue_mask)
41 cv2.imwrite('fruit_green.jpg', output)
42
43
44 # Reference
45 # http://docs.opencv.org/3.2.0/df/d9d/tutorial_py_colorspaces.html
```

Snippet 2: Full implementation