
Assignment 2

Subsection 1:

The assigned working image for me was im_3.jpg.

```
# im_3.jpg
# round objects
# set.jpg & geister.jpg
# SIFT

import cv2
import numpy as np
import matplotlib.pyplot as plt

#Subsection 1-----
-----

img = cv2.imread(r"C:\Users\Vladuts\Desktop\IPIVA\TEMA2\poze 2021\im_3.jpg",
1)
plt.figure(1)
plt.imshow(img[:, :, ::-1])
plt.title("Original image - im_3.jpg")
plt.show()
#img2 = cv2.medianBlur(img, 1)

imgHSV = cv2.cvtColor(img, cv2.COLOR_BGR2HSV)
H, S, V = cv2.split(imgHSV)
#clahe = cv2.createCLAHE(clipLimit=2, tileGridSize=(4, 4))
#V = clahe.apply(V)

plt.figure(2)
plt.imshow(V, cmap='gray')
plt.title("V channel")
plt.show()

tresh=185
maxValue = 1
retval, dst= cv2.threshold(V, thresh, maxValue, cv2.THRESH_BINARY)

plt.figure(3)
plt.imshow(dst, cmap='gray')
plt.title("V-THRESH_BINARY- segmented_3.jpg")
plt.show()

cv2.imwrite('segmented 3.jpg', dst)
```

Assignment 2

Tudorache Vlad Adrian 442C

```
kernel = cv2.getStructuringElement(cv2.MORPH_DILATE, ksize=(7,7))
out1 = cv2.dilate(dst, kernel, iterations=2)

plt.figure(4)
plt.imshow(out1, cmap='gray')
plt.title("V after dilate")
plt.show()

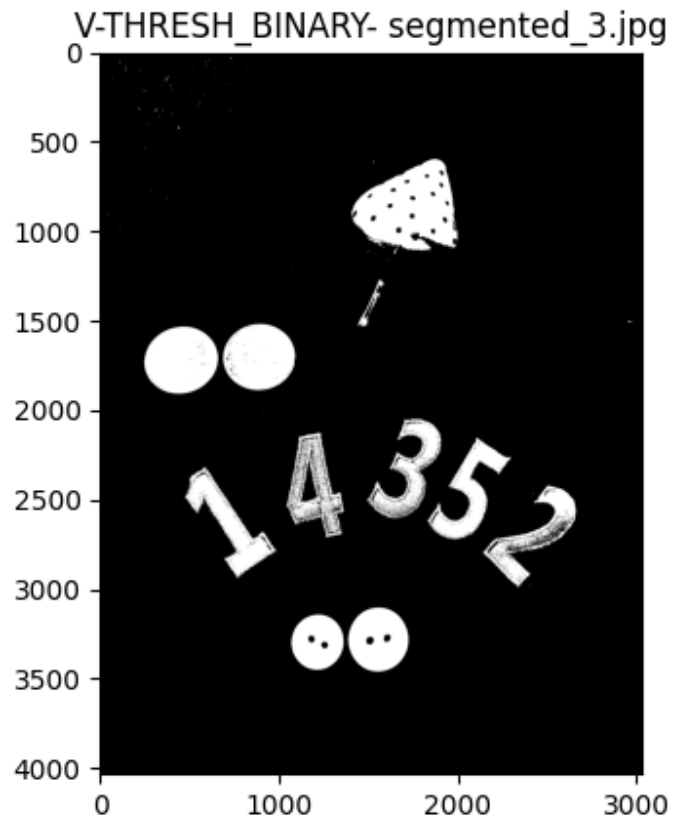
kernel = cv2.getStructuringElement(cv2.MORPH_ERODE, ksize=(3,3))
out2 = cv2.erode(out1, kernel, iterations=5)

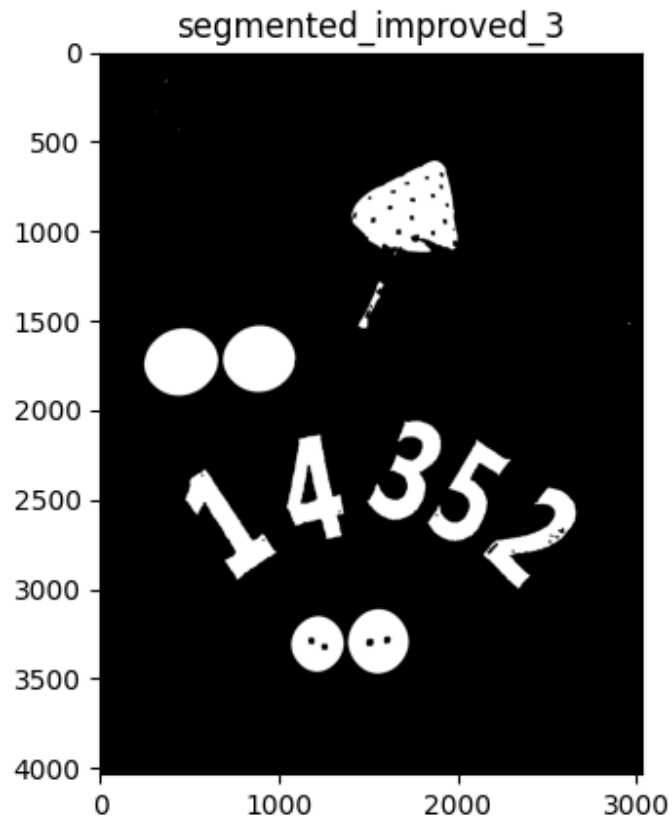
plt.figure(5)
plt.imshow(out1, cmap='gray')
plt.title("V after erode")
plt.show()

out3 = cv2.morphologyEx(out2, cv2.MORPH_CLOSE, (5,5), iterations=10)

plt.figure(6)
plt.imshow(out3, cmap='gray')
plt.title("segmented_improved_3")
plt.show()
cv2.imwrite('segmented_improved_3.jpg', out3)
```

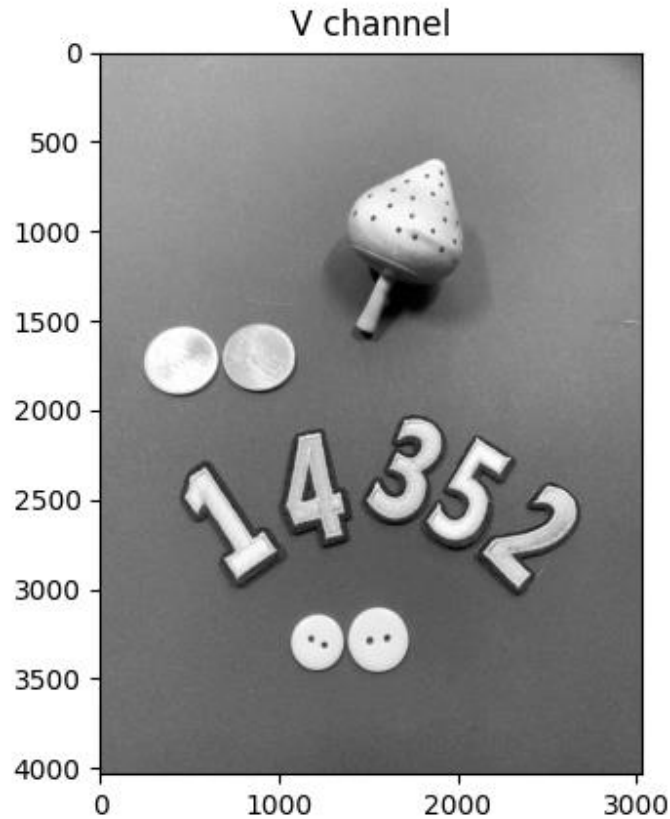






As it can be seen in the code, I chose the HSV color space. I chose this color space because of the multiple color objects, and if I chose the RGB color space some objects could not be identified in some channels from the background (which is gray).

From the HSV color space I chose the V channel because I got the best distinction between the background and the objects, and most of the reflections disappeared.



Subsection 2:

I have to extract the round objects.

```
#Subsection 2-----  
-----  
num_labels, imLabels = cv2.connectedComponents(out3)  
plt.figure(6)  
plt.imshow(imLabels, cmap='gray')  
plt.title("image with blobs labeled- blobs_3.jpg")  
plt.show()  
print(num_labels)  
  
cv2.imwrite('blobs_3.jpg', imLabels)  
blobs = cv2.imread('blobs_3.jpg', 1)  
blobs1 = cv2.cvtColor(blobs, cv2.COLOR_BGR2GRAY)  
blobs2 = cv2.medianBlur(blobs1, 21)  
circles = cv2.HoughCircles(blobs2, cv2.HOUGH_GRADIENT, 1, 100, param1=60,  
param2=30, minRadius=121, maxRadius=250)  
circles = np.uint16(np.around(circles))  
print(circles)  
  
for circ in circles[0,:]:
```

Assignment 2

Tudorache Vlad Adrian 442C

```
cv2.circle(blobs1, (circ[0], circ[1]), circ[2], (255, 0, 0), 5)

plt.figure(7)
plt.subplot(121);plt.imshow(blobs1, cmap='gray')
plt.subplot(122);plt.imshow(blobs2, cmap='gray')
plt.show()

# Display the labels
nComponents = imLabels.max()
displayRows = np.ceil(nComponents/3.0)
plt.figure(figsize=[20,12])
for i in range(nComponents+1):
    plt.subplot(displayRows,3, i+1)
    plt.imshow(imLabels==i)
    if i == 0:
        plt.title("Background, Component ID : {}".format(i))
    else:
        plt.title("Component ID : {}".format(i))
plt.show()

#components: 21, 22, 28, 29

nb_components, output, stats, centroids =
cv2.connectedComponentsWithStats(out3, connectivity=8)

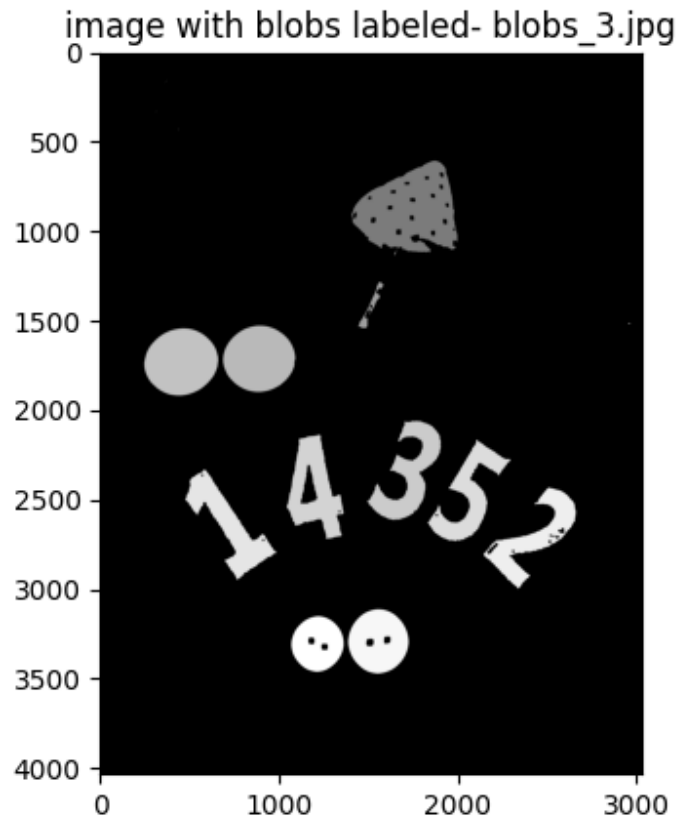
imFinal = (imLabels == 21) + (imLabels == 22) + (imLabels == 21) + (imLabels
== 28) + (imLabels == 29)
plt.figure(10)
plt.imshow(imFinal, cmap='gray')
plt.title('Final Image')
plt.show()

imFinal = cv2.cvtColor(np.float32(imFinal), cv2.COLOR_GRAY2BGR);
indices = np.where(imFinal == 1)
imFinal[indices[0], indices[1], :] = [255, 0, 0]

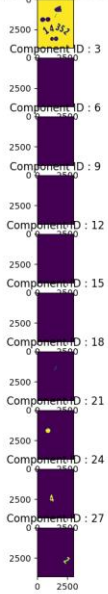
font = cv2.FONT_HERSHEY_SIMPLEX
cv2.putText(imFinal, 'Counted objects:4', (2000,150), font, 3, (255, 255,
255), 2, cv2.LINE_AA)

plt.figure(11)
plt.imshow(imFinal[:, :, ::-1])
plt.show()

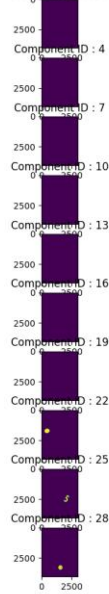
cv2.imwrite('valid_blobs_3.jpg', imFinal)
```



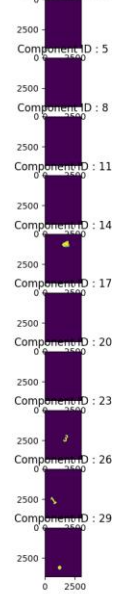
Background, Component ID : 0

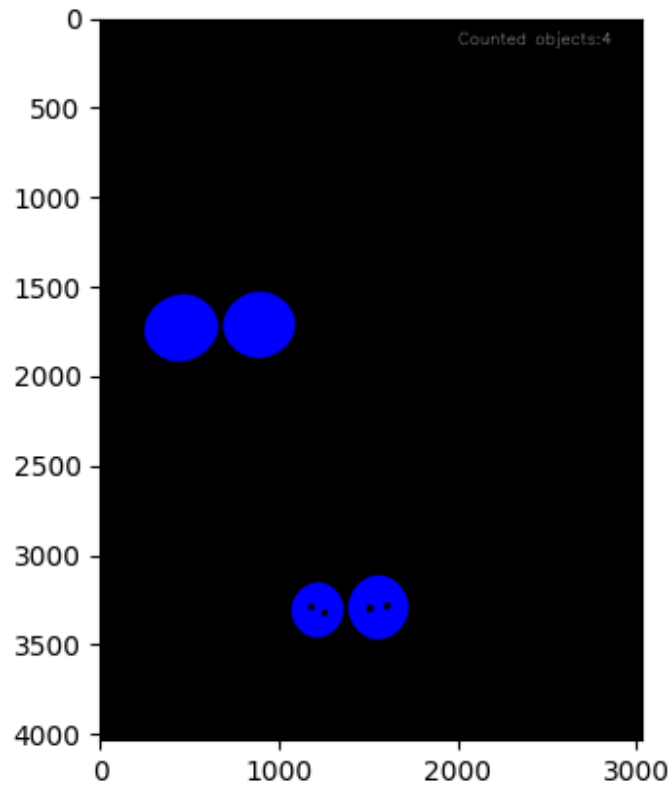


Component ID : 1



Component ID : 2





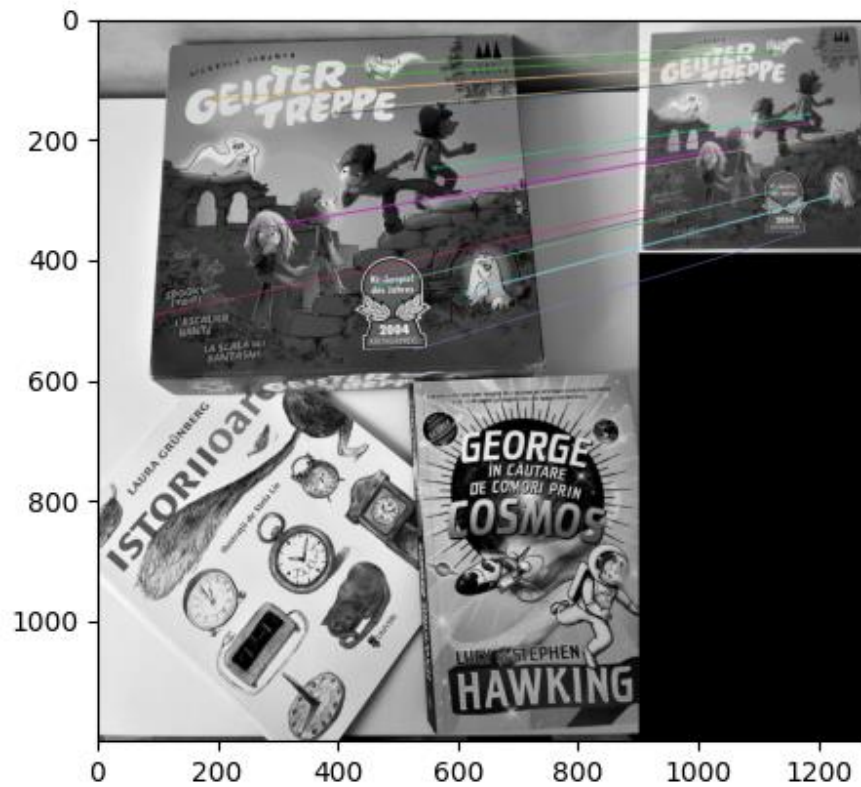
Subsection 3:

Using SIFT I had to match the set.jpg & geister.jpg images.

```
#Subsection 3-----  
-----  
  
img1 = cv2.imread(r'C:\Users\Vladuts\Desktop\IPIVA\TEMA2\set.jpg',0)  
img2 = cv2.imread(r'C:\Users\Vladuts\Desktop\IPIVA\TEMA2\geister.jpg',0)  
  
sift = cv2.SIFT_create()  
  
kp1, des1 = sift.detectAndCompute(img1,None)  
kp2, des2 = sift.detectAndCompute(img2,None)  
  
bf = cv2.BFMatcher(cv2.NORM_L2, crossCheck=True)  
matches = bf.match(des1,des2)  
matches = sorted(matches, key = lambda x:x.distance)  
  
img3 = cv2.drawMatches(img1,kp1,img2,kp2,matches[:20],None, flags=2)  
plt.imshow(img3)
```


Assignment 2
Tudorache Vlad Adrian 442C

```
plt.show()  
cv2.imwrite('matched.jpg', img3)
```



As we can see the algorithm performed correctly, all the matches are spot on.