پاسخ تمرین نهم **مهدیه نادری: ۹۸۵۲۲۰۷۶**

سوال چهارم)

الف)منبع: https://pyimagesearch.com/2016/02/01/opencv-center-of-contour/

https://pyimagesearch.com/2016/02/08/opencv-shape-detection/

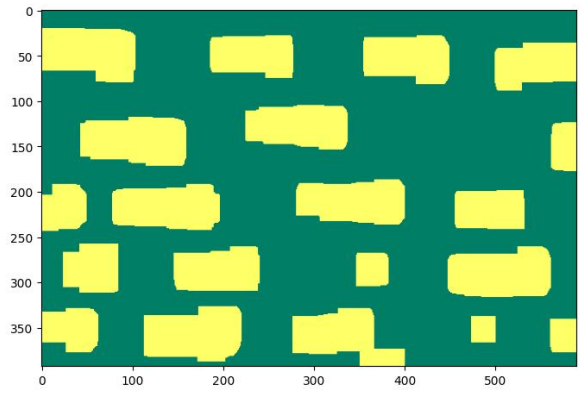
برای حل این سوال من ابتدا تصویر را با کرنل مناسب فیلتر کردم. تا تصویر به اندازه ی خوبی هموار شود. سپس آن را به سطح خاکستری بردم و بعد با استفاده از آستانه گذاری وفقی تصویر را باینری کردم. سپس برای هر عملگر مورفولوژی یک کرنل تعریف کردم و از آن ها برای انجام عملیات ها پشت سر هم استفاده کردم. سعی کردم کرنل های مختلف را امتحان کنم تا در نهایت نتیجه ی مناسبی حاصل شود. بعد از اعمال هر عملگر روی تصویر نتیجه اش را به عنوان ورودی عملگر بعدی دادم. سپس تصویر را رسم کردم تا مطمئن شوم جواب میتواند درست باشد.

در مرحله ی بعد با استفاده از لینک های بالا و توابع آماده ی opencv و imutils ماشین ها را پیدا کردم.

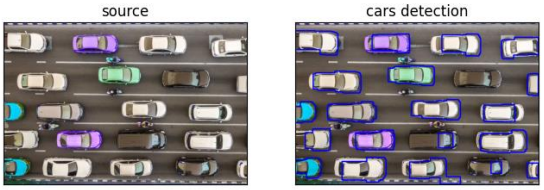
در آخرین محیط ماشین های پیدا شده را در تصویر اصلی رسم کردم و خروجی لازم تولید شد.

این کد ۲۰ ماشین در تصویر پیدا کرد.

```
result = image.copy()
cars_num = None

#Write your code here

blurred = cv2.GaussianBlur(image, (5, 5), 0)

blurred = cv2.filter2D(blurred, -1, np.array([1]))

gray_scale_image = cv2.cvtColor(blurred, cv2.COLOR_BGR2GRAY)

thresh = cv2.adaptiveThreshold(gray_scale_image, 255, cv2.ADAPTIVE_THRESH_MEAN_C, cv2.THRESH_BINARY, 225, 1)

OpenSE = np.ones((17, 15))

CloseSE = np.ones((20, 30))

ErodeSE = np.ones((11, 11))

DilateSE = np.ones((13, 13))

opening = cv2.morphologyEx(thresh, cv2.MORPH_OPEN, OpenSE)

closing = cv2.morphologyEx(opening, cv2.MORPH_CLOSE, CloseSE)

erodtion = cv2.morphologyEx(closing, cv2.MORPH_CLOSE, ErodeSE)

dilation = cv2.dilate(erodtion, DilateSE)

plt.imshow(dilation, cmap='summer')
plt.show()

contours = cv2.findContours(dilation, cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE)
contours = imutils.grab_contours(contours)
cars_num = len(contours)


print(f'Number of car is {cars_num}.')

for i in range(cars_num):
    contour = contours[i]
    bound = (0, 0, 255)
    cv2.drawContours(result, [contour.astype("int")], -1, bound, 2)

return result,cars_num
```




```
Number of car is 20.
```



نتیجه ی اجرای کد بالا:

پاسخ تمرین نهم **مهدیه نادری: ۹۸۵۲۲۰۷۶**

ب)

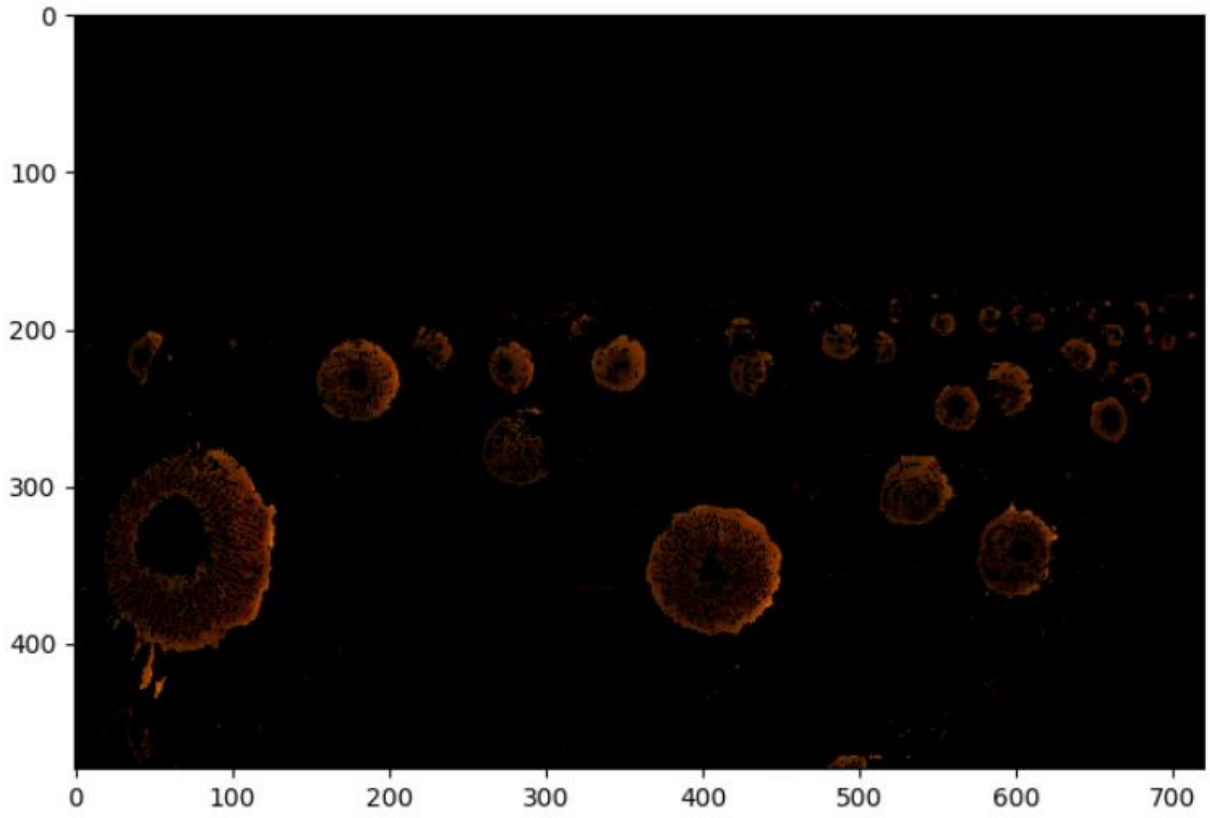
منبع:

https://www.tutorialspoint.com/how-to-find-the-hsv-values-of-a-color-using-opencv-python

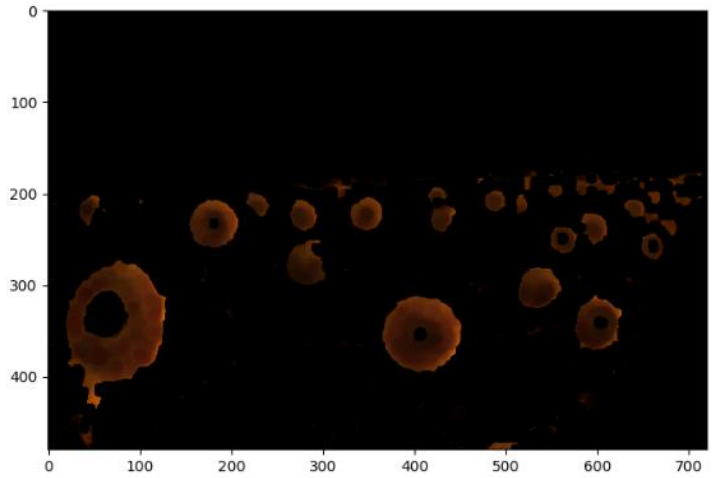
برای حل این سوال ابتدا باید تصویر را به حالت RGB ببریم. سپس باید hsv را برای تصویر بدست آمده را محاسبه کنیم. برای هر دو این ها از تابع cvtcolor استفاده میکنیم.

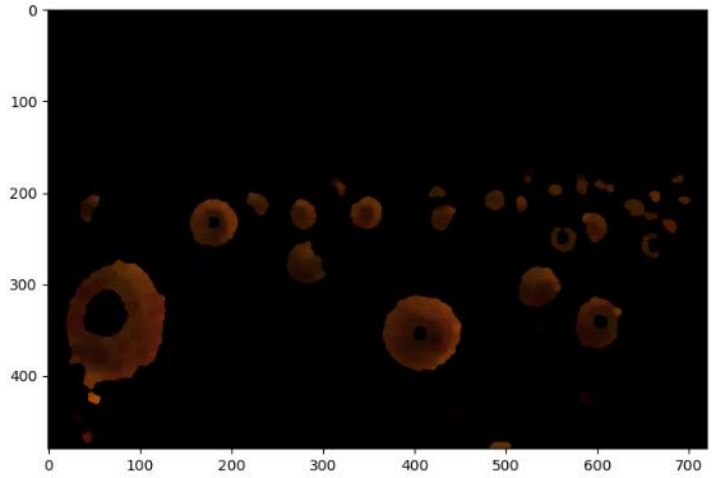
بعد از آن ما نیاز به یک mask داریم که با استفاده از آن گل های آفتابگردان را پیاده سازی کنیم. من ابتدا رنگ زرد را برای اینکار در نظر گرفته بودم اما به دلیل همپوشانی گلبرگ ها معیار مناسبی نبود. برای همین مرکز گل ها که رنگ قهوه ای دارد را برای این کار انتخاب کردم. برای اینکه این دایره ها را استخراج کنیم باید محدوده ی تقریبی رنگ قهوه ای را جست و جو کنیم و با چند بار امتحان کردن میتوان این را تقریب زد. مثلا من مقدایر کمینه را [0, 220, 0] و مقادیر بیشینه را [13, 255, 255] در نظر گرفتم که هر درایه ی آرایه ها مربوط به رنگ های آبی و قرمز و سبز است. بزرگترین بازه مربوط به رنگ قرمز است که مینیمم آن ۰ و ماکسیمم آن را ۲۵۵ در نظر گرفتم که بیشترین تاثیر را در رنگ قهوه ای دارد.

نتیجه ی کار شکل زیر شد:



برای اینکه تصویر هموار شود و نقاط قهوه ای یک ناحیه ی دایره ای را تشکیل بدهند از عملگر های باز و بسته با کرنل های مناسب استفاده کردم و حاصل آن شکل زیر شد. به ترتیب بالایی برای بسته و پایین برای باز است.





در نهایت مانند سوال قبل با استفاده از تابع findContours این دایره ها را پیدا کردیم و تعدادشان ۳۹ شد که با کمی تقریب قابل قبول است.

```
Number of flowers is 39.
```