

영상처리

5장 과제 #1



학과/분반	컴퓨터공학과 / 02	학번	20210262
이름	김우혁	제출일	2025. 05. 26 (월)
주제	동전 검출하기		

■ 목차

- 실습 목표
- 해결 과정 및 주요 코드 설명
- 실행 결과
- 느낀점

■ 실습 목표

불균일한 조명에서 촬영한 영상에서 임계화 연산을 사용하여 동전 영역만을 분리하여 연결 요소를 생성한다.

■ 해결 과정 및 주요 코드 설명

1. 컬러 영상을 그레이 스케일 영상으로 변환하고 가우시안 평활화를 수행한다.

```
gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
gray = cv2.GaussianBlur(gray, (3, 3), 0)
```

2. 적응적 임계화 연산을 수행한다.

```
binary = cv2.adaptiveThreshold(gray, 255,
                               cv2.ADAPTIVE_THRESH_GAUSSIAN_C, cv2.THRESH_BINARY_INV, 11, 2)
```

3. 모폴로지 연산을 수행하여 단순화된 도형을 생성한다.

```
kernel = cv2.getStructuringElement(cv2.MORPH_ELLIPSE, (3, 3))
img = cv2.morphologyEx(img, cv2.MORPH_CLOSE, kernel)
```

4. 연결 요소를 생성한다.

```
def findContours(img):

    version = int(cv2.__version__.split(".")[0])
    if version == 2 or version == 4:
        (contours, hierarchy) \
            = cv2.findContours(img, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)
    elif version == 3:
        (_, contours, hierarchy) \
            = cv2.findContours(img, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)
    print("Total number of contours: ", len(contours))

    return contours

#####

contours = findContours(binary)

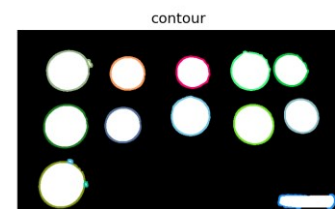
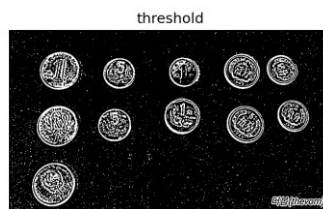
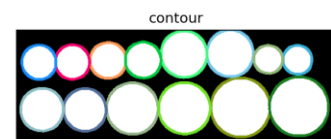
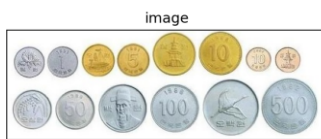
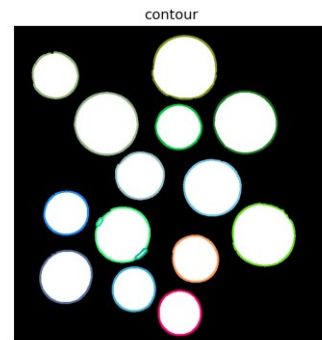
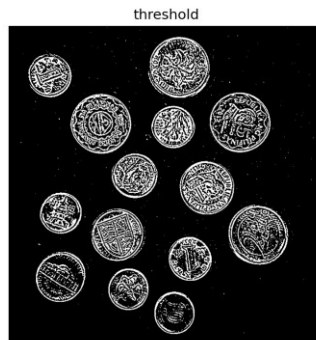
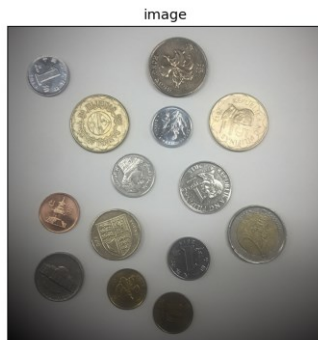
new_img = np.zeros_like(image, dtype="uint8")
cntrarray = sorted(contours, key = cv2.contourArea, reverse = True)
box_img = deepcopy(new_img) # 필요하다면 반복 전에 복제
seed(9001)
for cntr in cntrarray:
    if cv2.contourArea(cntr) > 5:
        r = randint(0, 256)
        g = randint(0, 256)
        b = randint(0, 256)
        cv2.drawContours(new_img, [cntr], 0, (b, g, r), 10)
        hull = cv2.convexHull(cntr)
        cv2.drawContours(new_img, [hull], 0, (255, 255, 255), -1)
        print("면적: ", cv2.contourArea(cntr))
```

5. 결과 영상을 화면에 표시한다.

```
image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
plt.subplot(1, 3, 1), plt.imshow(image)
plt.title('image'), plt.xticks([]), plt.yticks([])
plt.subplot(1, 3, 2), plt.imshow(binary, cmap='gray')
plt.title('threshold'), plt.xticks([]), plt.yticks([])
plt.subplot(1, 3, 3), plt.imshow(new_img)
plt.title('contour'), plt.xticks([]), plt.yticks([])
plt.show()
```

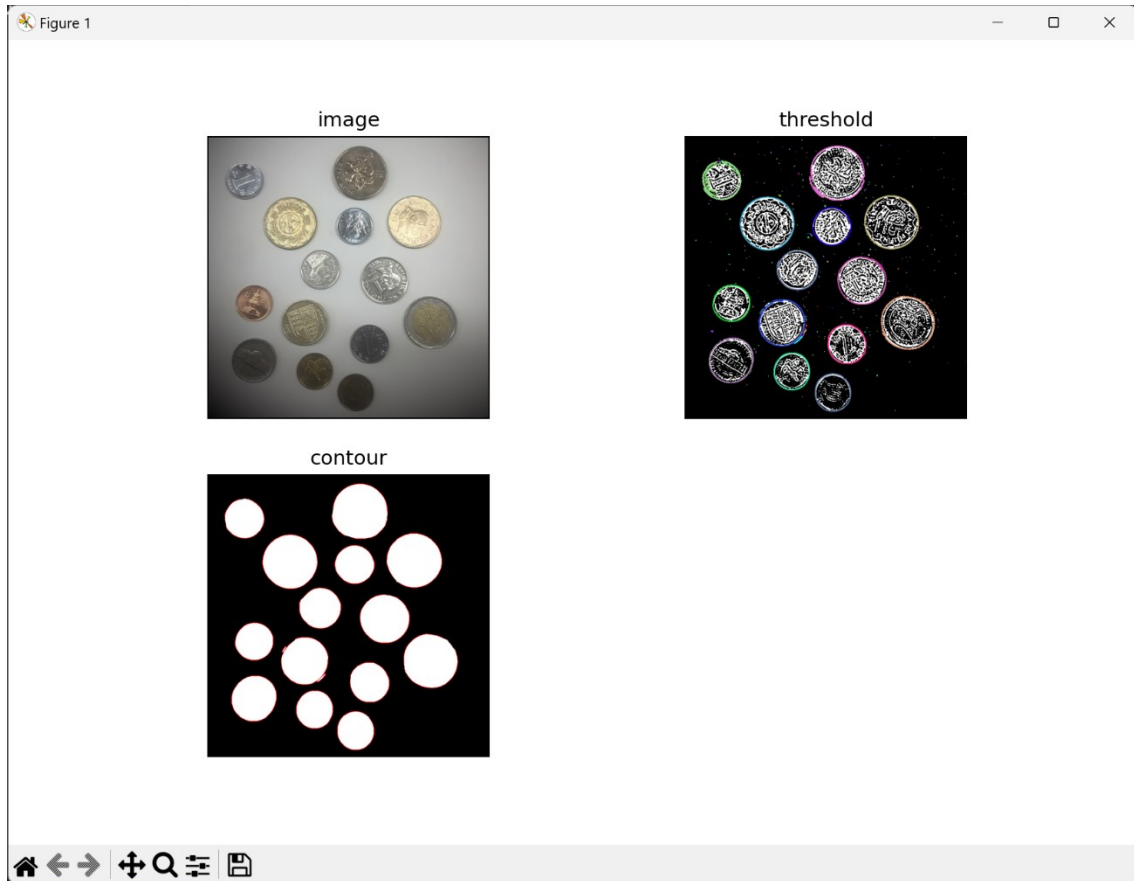
■ 실행 결과

- 각각 다른 3장의 동전 사진을 넣어본 결과들이다.



■ 느낀점

처음에 과제를 수행하면서 threshold된 바이너리 영상을 화면에 출력해봤는데, 내가 원하는 검정색과 흰색이 아닌 다른 색으로 출력이 되는 것이었다. 그래서 인터넷에 찾아보니 plt.imshow(binary) 부분에서 인자로 cmap='gray' 를 추가하면 되는 것이었다. 이렇게 처음 닥친 문제를 해결하고, 과제를 이어나갔다. 그런데



이 사진처럼 threshold 부분에 테두리가 나오고, contour 부분에는 흰색으로 채워지긴 했지만, 테두리 색이 다 동일하게 나오는 것이었다. 이를 해결하기 위해 원래 findContours 함수 내부에 있던 랜덤 색 부분을 main함수 부분으로 빼오고, 다른 코드들도 수정해 가면서 제대로 원하는 출력이 나오게 했다. 그렇게 원하는 이미지가 나오긴 했지만, 이미지들을 한 줄로 이어지게 출력하고 싶은데 위의 사진처럼 격자로 출력이 되는 것이었다. 그래서 이것도 인터넷에 찾아보니 plt.subplot(1, 2, 1)의 두번째 인자를 바꿔주면 되는 것이었다. 이렇게 해서 문제를 모두 해결하고, 과제를 전부 수행해나갔다. 이렇게 오늘도 파이썬의 함수들의 사용법을 익혀나갔다.