

# PS6

## Report

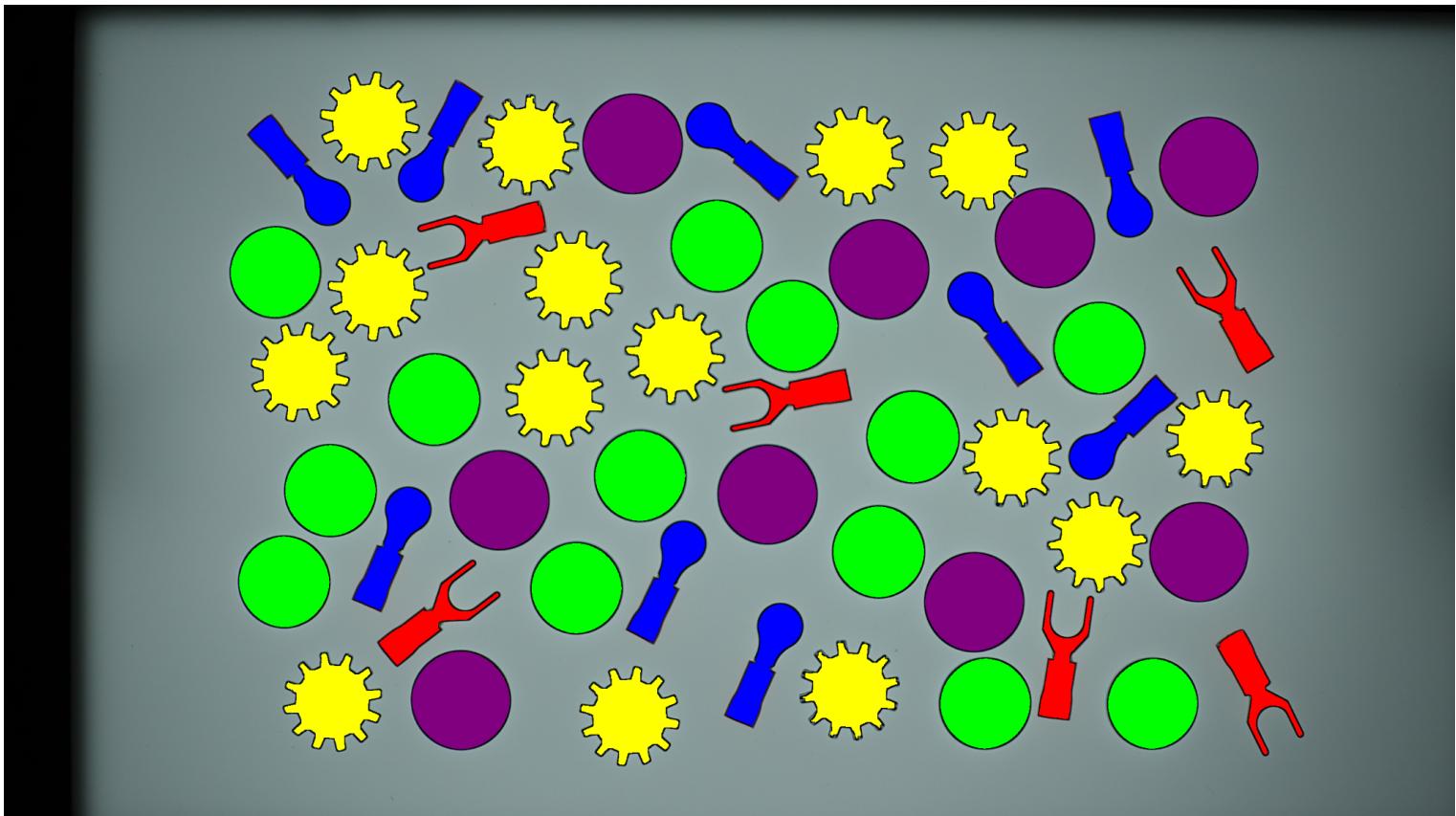
Akshay Antony  
akshayan@andrew.cmu.edu

### Q1

#### Pseudo Code

- Modify the existing code given in the question
- If the outer contour is a circle, and the inner contour is not a circle, we can classify it as an internal lock washer with purple.
- If the outer contour is not a circle, and the inner contour is a circle, it is an external lock washer and colors it yellow.

#### Results



*all-parts-output.png*

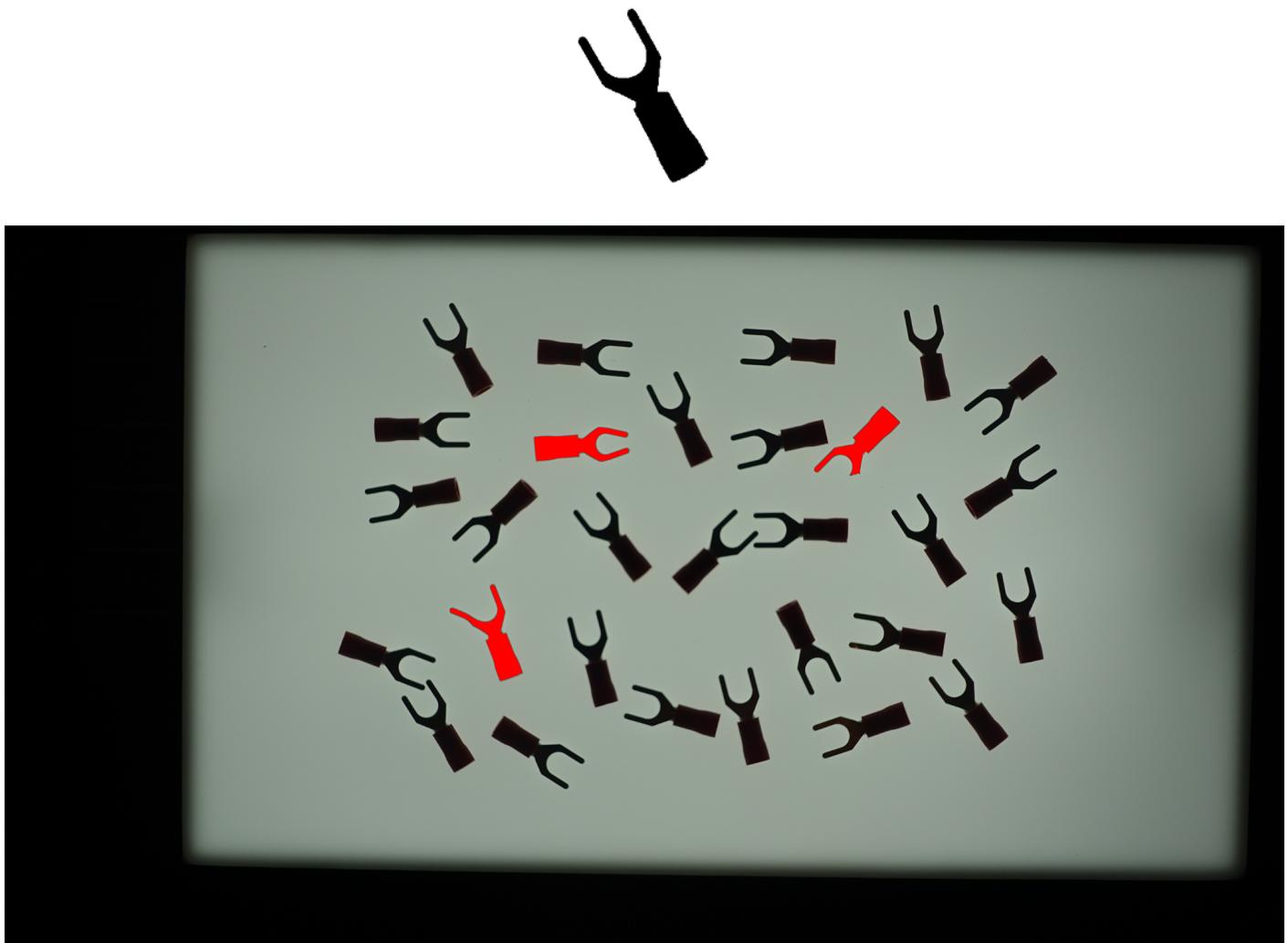
## Q2:

### Pseudo Code

- Read the image
- Convert a copy of the image to GRAY
- Threshold the image using 60 as the threshold value
- Apply 2 erosions and 3 dilations to enhance the input image
- Find all the contours
- Take a single contour as the template contour, here contours[29] is taken as the template
- Compare all contours to the template while storing similarity values and corresponding indices contours in a 2d np array
- Sort the 2d array according to the similarity value and reverse the array.
- So the first three values in the array will represent the most different contours to the template.
- Draw these three contours on the original image with red color

### Results:

*template.png the contours[29] mentioned above*



*spade-terminal-output.png*

## Screenshots of files

q1.py

```
import cv2
import numpy as np
import argparse

# check size (bounding box) is square
def isSquare(siz):
    ratio = abs(siz[0] - siz[1]) / siz[0]
    #print (siz, ratio)
    if ratio < 0.1:
        return True
    else:
        return False

# chekc circle from the arc length ratio
def isCircle(cnt):
    (x,y),radius = cv2.minEnclosingCircle(cnt)
    len = cv2.arcLength(cnt, True)
    ratio = abs(len - np.pi * 2.0 * radius) / (np.pi * 2.0 * radius)
    #print(ratio)
    if ratio < 0.1:
        return True
    else:
        return False

if __name__ == "__main__":
    # Read image
    img = cv2.imread("/home/akshay/Downloads/CV/ps6/all-parts.png")

    # Convert to gray-scale
    gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
    # Binary
    thr,dst = cv2.threshold(gray, 60, 255, cv2.THRESH_BINARY)

    # clean up
    for i in range(1):
        dst = cv2.erode(dst, None)
    for i in range(4):
        dst = cv2.dilate(dst, None)

    cv2.imshow("modified", dst)

    # find contours with hierachy
    cont, hier = cv2.findContours(dst, cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE)
    # each contoure
    for i in range(len(cont)):
```

site: Ready, 0 tabs, Line 24, Column 1

```
# find contours with hierachy
cont, hier = cv2.findContours(dst, cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE)
# each contoure
for i in range(len(cont)):
    c = cont[i]
    h = hier[0,i]
    if h[2] == -1 and h[3] == 0:
        # no child and parent is image outer
        img = cv2.drawContours(img, cont, i, (0,0,255), -1)
    elif h[3] == 0 and hier[0,h[2]][2] == -1:
        # with child
        if isCircle(c):
            if isCircle(cont[h[2]]):
                # double circle
                img = cv2.drawContours(img, cont, i, (0,255,0), -1)

            else:
                img = cv2.drawContours(img, cont, i, (128,0,128), -1)
        else:
            # 1 child and shape bounding box is not squire
            if not isSquare(cv2.minAreaRect(c)[1]) and hier[0,h[2]][0] == -1 and hier[0,h[2]][1] == -1:
                img = cv2.drawContours(img, cont, i, (255,0, 0), -1)

            elif isCircle(cont[h[2]]):
                img = cv2.drawContours(img, cont, i, (0,255,255), -1)

cv2.namedWindow("Image",cv2.WINDOW_NORMAL)
cv2.imshow("Image", img)

if cv2.imwrite("all-parts-output.png", img):
    print("Saved Successfully")

cv2.waitKey()
```

## q2.py

```
import os
import cv2
import numpy as np

if __name__ == '__main__':
    #loading the input image
    filename = "/home/akshay/Downloads/CV/ps6/spade-terminal.png"
    input_image = cv2.imread(filename)

    #CONVERTING TO BINARY image
    gray = cv2.cvtColor(input_image, cv2.COLOR_BGR2GRAY)
    #thresholding the image using binary threshold
    thr, image = cv2.threshold(gray, 80, 255, COLOR_BGR2GRAY int)
    kite Docs

    #applying two erosions
    for i in range(2):
        image = cv2.erode(image, None)

    #applying 3 dilations
    for i in range(3):
        image = cv2.dilate(image, None)

    #extracting the contours from the images
    contours, hierarchy = cv2.findContours(image, cv2.RETR_TREE, cv2.CHAIN_APPROX_NONE)

    #taking the template as contour number 29, which is a non-defective contour
    #it will be shown in the write-up
    req_contour = contours[29]

    #drawing the template contour on white screen
    white_screen = np.uint8(np.full(image.shape, 255))
    white_screen = cv2.drawContours(white_screen, contours, 29, (0,0,0), -1)
    cv2.imwrite("template.png", white_screen)
    similarity = np.empty((len(contours),2))

    for i, contour in enumerate(contours):
        #iterating through all the contours and store the results returned from matchShapes
        # and the corresponding index
        similarity[i] = [cv2.matchShapes(contour, req_contour, cv2.CONTOURS_MATCH_I2, 0), i]

    #sort based on the the similarity and reverse it
    similarity = similarity[np.argsort(similarity[:,0])][::-1]
```

```
#sort based on the the similarity and reverse it
similarity = similarity[np.argsort(similarity[:,0])][::-1]

#take the first three
for i in range(3):
    input_image = cv2.drawContours(input_image, contours, int(similarity[i,1]), (0,0,255), -1)

#save the image
if cv2.imwrite("spade-terminal-output.png", input_image):
    print("Saved Successfully")

cv2.namedWindow("threshold", cv2.WINDOW_NORMAL)
cv2.imshow("threshold", input_image)
cv2.waitKey(0)
```

## Readme.txt

For q1

```
1 Python version: Python 3.9.6
2 OpenCV Version: 4.5.2
3 Operating System: Linux 20.02
4 IDE: Sublime text, run via terminal
5 Almost spend 2 hours for this problem
```

For q2

```
1 Python version: Python 3.9.6
2 OpenCV Version: 4.5.2
3 Operating System: Linux 20.02
4 IDE: Sublime text, run via terminal
5 Almost spend 3 hours for this problem
```