```python
# 24-678 Computer Vision for Engineers
# Ryan Wu (ID:weihuanw)
# PS06-1 Part Identification and Classification
# Due 11/10/2023 (Fri) 5 pm

# import the necessary packages
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

# check 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__":
#
    parser = argparse.ArgumentParser(description='Hough Circles')
    parser.add_argument('-i', '--input', default='all-parts.png')

    args = parser.parse_args()
    # Read image
    img = cv2.imread(args.input)

    # Convert to grayscale
    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)

    # find contours with hierarchy
    cont, hier = cv2.findContours(dst, cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE)

    # filter out small contours based on area
    cont = [c for c in cont if cv2.contourArea(c) > 100]

    # each contour
    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:
                    # single circle
                    img = cv2.drawContours(img, cont, i, (187,41,187), -1)
            else:
                # 1 child and shape bounding box is not square
                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)
                # 2 children and shape bounding box is square
                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)
    cv2.imwrite('all-parts-output.png', img)
    cv2.waitKey(0)
    cv2.destroyAllWindows()
```