# author: Adrian Rosebrock

# website: http://www.pyimagesearch.com

# import the necessary packages

import cv2

def sort\_contours(cnts, method="left-to-right"):

# initialize the reverse flag and sort index

reverse = False

i = 0

# handle if we need to sort in reverse

if method == "right-to-left" or method == "bottom-to-top":

reverse = True

# handle if we are sorting against the y-coordinate rather than

# the x-coordinate of the bounding box

if method == "top-to-bottom" or method == "bottom-to-top":

i = 1

# construct the list of bounding boxes and sort them from top to

# bottom

boundingBoxes = [cv2.boundingRect(c) for c in cnts]

(cnts, boundingBoxes) = zip(\*sorted(zip(cnts, boundingBoxes),

key=lambda b: b[1][i], reverse=reverse))

# return the list of sorted contours and bounding boxes

return cnts, boundingBoxes

def label\_contour(image, c, i, color=(0, 255, 0), thickness=2):

# compute the center of the contour area and draw a circle

# representing the center

M = cv2.moments(c)

cX = int(M["m10"] / M["m00"])

cY = int(M["m01"] / M["m00"])

# draw the contour and label number on the image

cv2.drawContours(image, [c], -1, color, thickness)

cv2.putText(image, "#{}".format(i + 1), (cX - 20, cY), cv2.FONT\_HERSHEY\_SIMPLEX,

1.0, (255, 255, 255), 2)

# return the image with the contour number drawn on it

return image