Shape detection

The algorithm:

First of all I read the picture then applied canny on it as shapes is all about edges and in order to remove the noise I used cv2.filter2D , I applied dilation using cv2.dilate function to increase the edges area so the contours can pick it up easier and some thresholding using cv2.threshold function ,

After all these preparations I found the contours using cv2.findcontours function the for each contour I converted it to an approximate polygon(nearest shape to it) and starting drawing them on every shape using cv2.drawcontours function , then to write the name of the shape I used approximate. ravel function it returns the very starting point at the left side of the shape so I can start writing from it, moreover the approximate polygon has length function that count its vertices so I used it as it’s a unique identifier for each shape 3 for triangle, 4 for square and rectangle and I differentiated between them using the aspect ratio (w/h) if its near 1 so it’s a square if not it is a rectangle , more than 5 so it has many vertices

therefore, it’s a circle.

Of course, after all these operations I showed the image using cv2.imshow function and within the loop I counted each shape occurrence and printed it as required.