4.10 Image 10

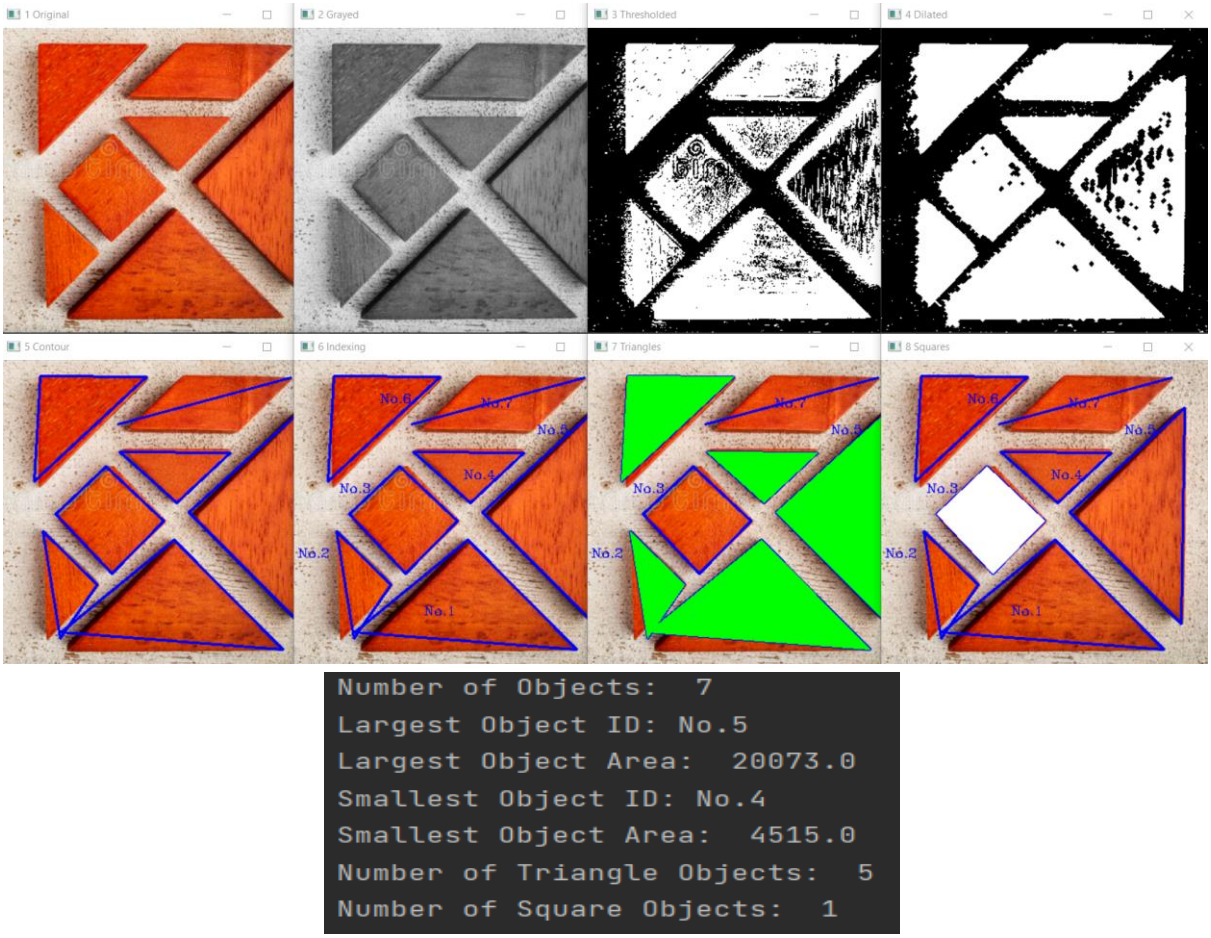

Figure 46 Final output of image10

Figure 46 shows different images after various algorithm processes starting from top left to top right and then bottom left to bottom right. As seen within the first pic, "Original", the picture has 7 objects, with 5 triangles and 1 square objects.

Same as the other images, it is converted to greyscale and thresholded as shown in the second and third image output. The 4th image output is the 3rd image going through a morphological operation called closing, which is dilation followed by erosion, known for its effectiveness in closing hole within objects. The 5th image is showing the usage of cv2.findContour and cv2.approxPolyDP to roughly outline the objects, approximating the outline to be a particular shape. The 7th image output colours the object which fulfils the condition in which the total corner of the object is equal to 3, same as a triangle. The 8th image output however colours the object white if the corner of the shape is equal to 4. Additionally, the area of each object is also compared to discover the smallest and biggest object.

As you can see the object No.7 does not have a shape. This is because the cv2.approxPolyDP is set to only accommodate the triangles and square, unfortunately incapable of outlining the rhombus shape. However, it is still considered as an object. Moreover, No.1 and No.5 has the same area physically, but based on how it is placed, No.5 is bigger within the picture hence no errors there. Next, No.4 and No.2 has the same physical size and has only an extremely small difference between the two in the image. However, due to the interruption of noises, the area of those two objects was detected to have a large difference of 500 pixels. Hence it is to be emphasized despite the correct output.

4.11 Image 11

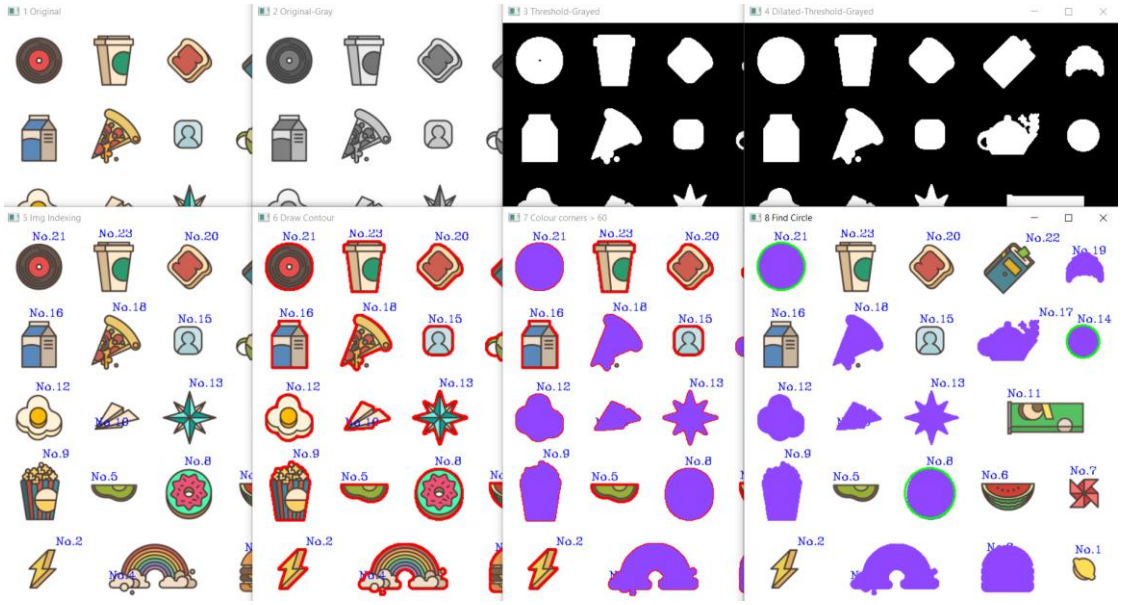

Figure 47 Final output of Image11

Figure 47 has the same sequence as Figure 46. The image is converted to greyscale as well. Then, the greyscale is also inversely thresholded with the threshold value of 254, followed up with closed morphological operation, however the operation is less effective here. The 6th image find contour with very precise outlining as compared to Figure 46. The 7th image attempts to find circle, by colouring objects with more than 60 corners purple. However, it is clearly shown that this will not work as there are other objects with more corners than all the circles. For example, object No.14 has around 62 corners only. The 8th image shows cv2.HoughCircles which outlines circles only with minimum radius of 10 to maximum of 140 in green.

Throughout the procession of image11, it has been discussed that the circle shaped objects can't be individually determined using cv2.approxPolyDP. Hence, cv2.HoughCircle is utilized. However, the param2 in the cv2.HoughCircle's parameter must be adjusted in a specific range so that objects that are not circle shaped, will not be detected as a circle. Overall, the result has been satisfactory as all the circle shaped objects have been identified.