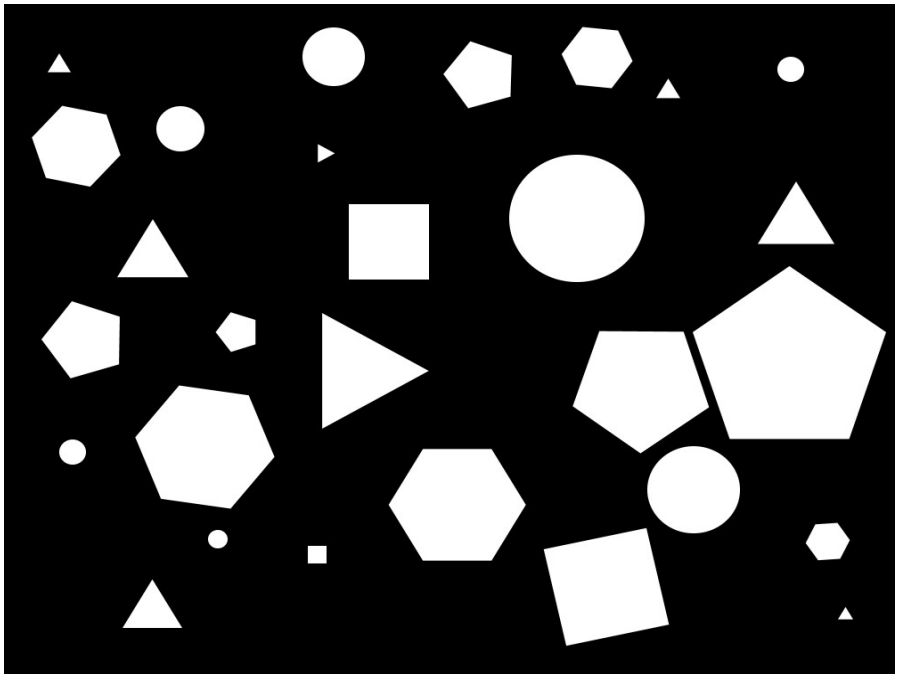
Intenseye Computer Vision Engineering Task

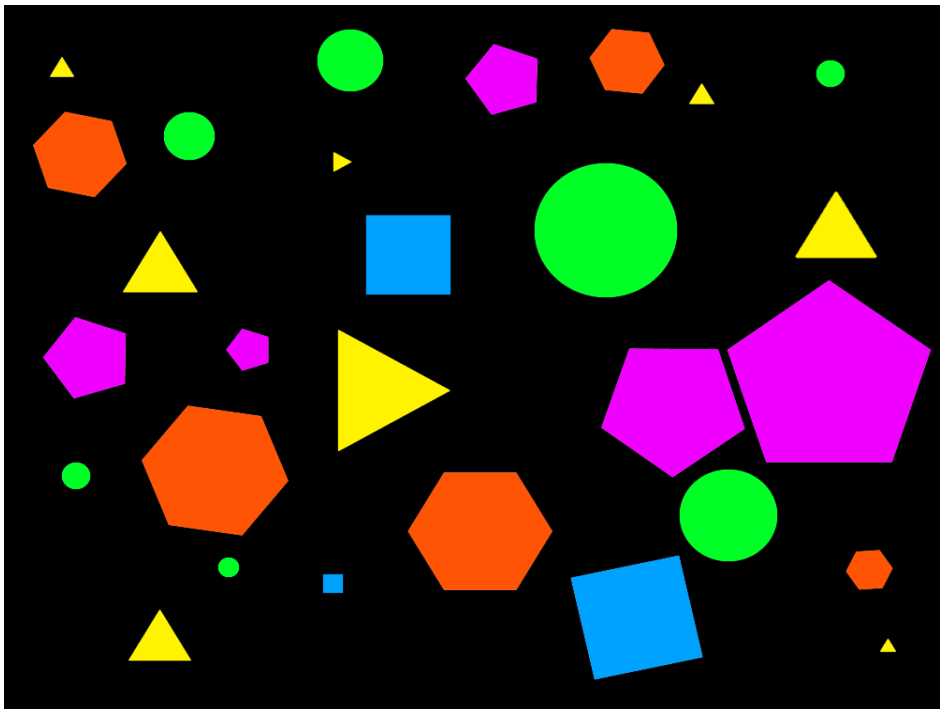
Shape Detector

Consider a binary image that consists of a couple of shapes including only the following ones: circle, triangular, square, pentagon and hexagon.

All shapes are randomly located into the image as all are regular and equilateral and with varying sizes and randomly rotated like shown on the image below. They do not intersect or touch each other. The color of shapes is pure white and the background color is pure black.



Write a program in Python or C++ that detects and categorizes the shapes and outputs an information about each shape with instance count. Also output an auto-generated, shape based color labeled image like below.



- Your program will be tested with similar images generated by us, so prepare your program in this way and describe its usage briefly.

- Shapes with a minimum area of 50 pixels can be ignored if exist.

- You can use OpenCV like libraries for basic operations like loading image to memory or access to its pixels or get contours. On the contrary, you are not allowed to use any built-in OpenCV detection algorithms or any other library or helper methods written by others. Implement your own detection algorithm and be genuine.

- Prefer as possible as the latest version of libraries you use and if possible avoid using more than one dependent library.