Harvester Bot

Task 1B – Count the Objects

Goal:

1. To learn Python and use it to:
   - Identify shapes
   - Identify colours and sizes of shapes
   - Find number of exactly similar shapes
   - Output a .csv file with an image of objects with contours and labels

Please find the *task1B_main.py* file in the folder named *Test Images*. Modify the sections of *task1B_main.py* to accomplish the following:

Given:

1. We have given a set of five images named: **test1.png, test2.png, test3.png, test4.png and test5.png** in a folder named **Test Images**. This folder is a sub-folder inside the **Task 1B** folder. Example Test Images are shown in Figure 1.

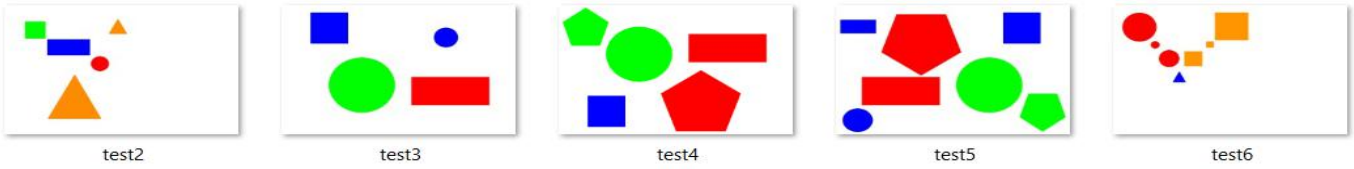

Figure 1: Test Images

2. We have also given a set of eight images named: **Sample_circle_large.png, Sample_circle_small.png, Sample_rectangle_large.png, Sample_rectangle_small.png, Sample_square_large.png, Sample_square_small.png, Sample_triangle_large.png and Sample_triangle_small.png** in a folder named **Sample Images**. This folder is a sub-folder inside the **Test Images** folder. Sample Images are shown in Figure 2.

Figure 2: Sample Images

Every image has a certain number of objects. Each object is defined by three features, viz. **Colour, Shape and Size.** Objects in a given image vary in colour, shape and size and may appear multiple times in an image. An example image is shown in Figure 3.

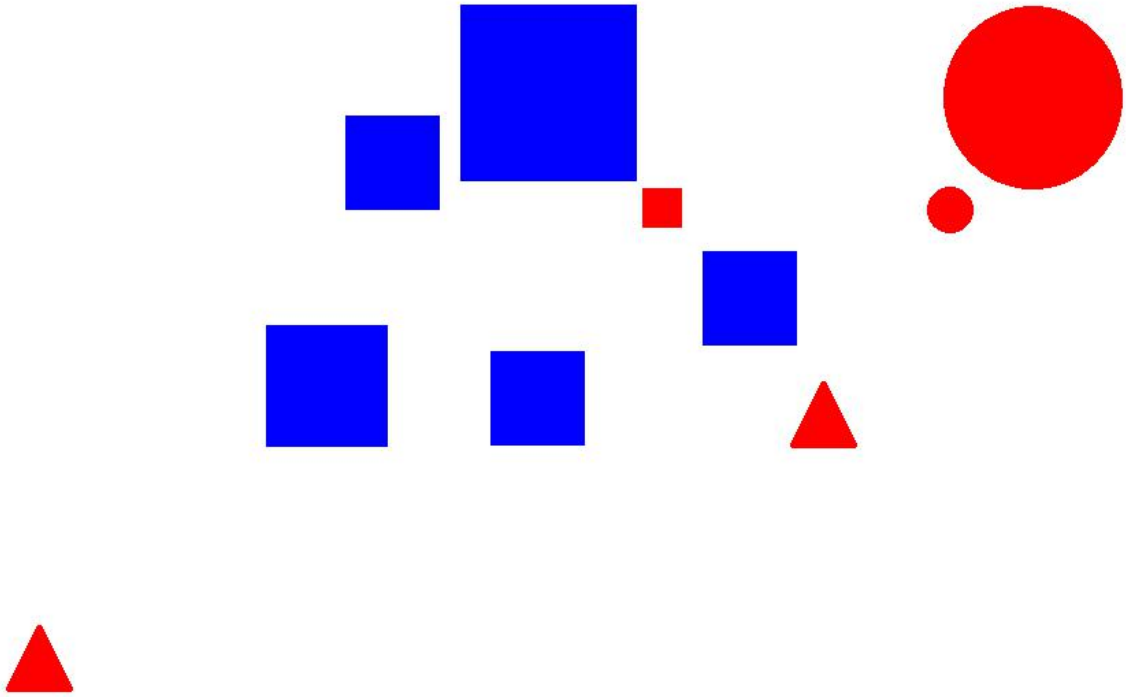

Figure 3: Example Image

We learnt about Colour and Shape detection in Task 1A; a new parameter Size has been introduced in this task.

There are three types of Sizes viz., Large, Medium and Small.
Sample Images of Large and Small objects are given in **Sample Images** folder.

Small Object: an object **smaller than or equal to** the size of small sample images in **Sample Images** folder namely, *Sample_circle_small.png, Sample_rectangle_small.png, Sample_square_small.png and Sample_triangle_small.png*

Large Object: an object **larger than or equal to** the size of large sample images in **Sample Images** folder namely, *Sample_circle_large.png, Sample_rectangle_large.png, Sample_square_large.png and Sample_triangle_large.png*

Medium Object: an object that is **neither Small nor Large**

Problem Description:

For each image in the **Test Images** folder, add your methods or functions to the Python script file in order to find the Colour, Shape, Size and Count of similar objects in that image.

Count refers to the number of **exactly similar** objects in an image.

Your code should return a List of Lists; where each individual List contains - image name followed by List of **Colour-Shape-Size-Count** of each object contained in that image, in a comma separated values List.

You have to find out the **Colour, Shape, Size and Count** of each object in the image and concatenate them into one string in the format given below and then store it in the list.

Considering the image **"example.png"** shown in Figure 2, output would be as follows:

[“example.png”,[“Red-Triangle-Small-2”],[“Blue-Square-Medium-4”],[“Red-Square-Small-1”],[“Red-Circle-Small-1”],[“Red-Circle-Large-1”],[“Blue-Square-Large-1”]]

You should also return an image of objects with **Contours** and **Labels**. A **Contour** in Image processing is simply an outline along the boundary of the Object. Contours are used for shape analysis, object detection and recognition. **Label** is text with color and shape of the object. An example is shown in Figure 5.

Note:

1. An object will be one of the five colours, viz. **Red, Green, Blue, Yellow and Orange.**
2. An object will be one of the four shapes, viz. **Triangle, Rectangle, Square and Circle**.

Required Output:

1. The Python script file named *task1B_main.py* with the function ***“main”*** populated with your logic of solving the task

2. A **.csv** file with the Lists of each image on each row i.e. five rows of data for a total of five test images in the Test Images folder. For our example, .csv file would look like Figure 4.

example.png	Red-Triangle-Small-2	Blue-Square-Medium-4	Red-Square-Small-1	Red-Circle-Small-1	Red-Circle-Large-1	Blue-Square-Large-1

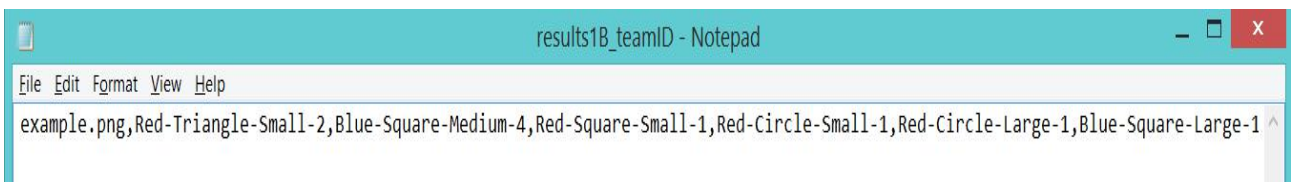

Figure 4: CSV output

3. For each image in the *Test Images* folder create a corresponding output image with Contours and Labels; each output image should be named as **output1.png** (for test1.png), **output2.png** (for test2.png) and so on, in the *Test Images* folder itself.

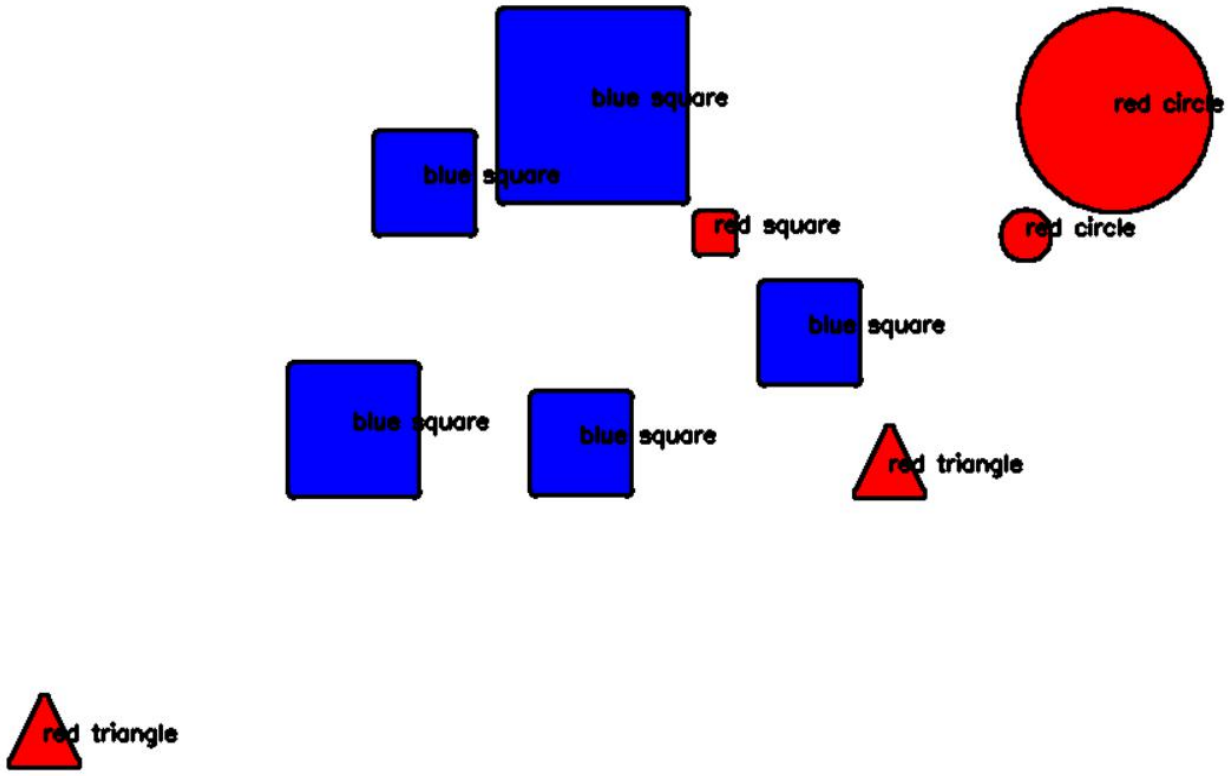

Figure 5: Output Image with Contours and other properties enlisted, overlaid on the object

To do:

1. Open **task1B_main.py** located in the folder named **Test Images**. It has two functions:
   - **main():** It returns the required Lists for the **writecsv()** function.
     It should also write the Output Images and save it in the *Test Images* folder as explained in point 3 under *"Required Output"* section above.
   - **writecsv():** This function expects four parameters as arguments and writes the elements of List of Lists one at a time into a **"results1B_teamid.csv"** file. **Do not edit this function. However you can edit the global variable - "filename" with your teamid in the *task1B_main.py* Python code file.** Please use your eYRC Team ID while actually naming the file.
     IMPORTANT: Do NOT change names of any of these functions.

Rules:

1. You need to write a **generic program**. Your code should be capable enough to detect **any number** of objects in an image and extract their properties. In addition your code will be tested on several **undisclosed images when you submit your code.**

2. Use basic knowledge of geometry to differentiate between the shapes of objects.

3. **Matching of images should be independent of orientation of the objects.** Objects in *image* might be rotated w.r.t. each other.

Result of rotation:

- For Circles , there is no difference in the images. Rotation does not affect these images.

- Triangles, Rectangles and Squares can be in any orientation. An example rotation for a Triangle is shown in Figure 6 below:

Figure 6a. Original orientation

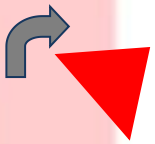

Figure 6b. Rotated Triangle

4. In case no object is present in the image, **return nothing or an empty List.**

Happy Learning!

All The Best!!!