Ant Bot

Task 1.2 - To detect coloured objects and ArUco IDs from an Image

Goal:

To Use Python and OpenCV to:

1. Identify the ID of the ArUco marker whose image is given in JPEG format.
2. Identify the colour and shape of the object in the sequence given indicated by the ID.

Please find the **Task1.2.py** file in the nested folder named **"2. Code"**.

Modify the sections of **Task1.2.py** marked for the same, to accomplish the task given in problem description below.

Given:

Each team is given five images (refer to folder **"3. Images"**) which contain ArUco markers whose IDs are encoded within the ArUco image. The images are named **"Image1.jpg"** to **"Image5.jpg"**. The number in the image name eg. **"1"** in the image name **"Image1.jpg"** is **not the ID of the ArUco**, it is the input image number of the given set of test images.

Each input test image consists of:

- An ArUco marker located at the top-left hand corner of the input test image is a monochrome pattern, whose ID is **NOT** specified. However, which dictionary it belongs to is given in ***Table 1*** below.

and

- Four colour objects of different shapes.

The objects can be any of four shapes:

1. Circle
2. Square
3. Ellipse
4. Triangle

The shapes can be any of the five colours:

1. Red
2. Blue
3. Green
4. Yellow
5. Orange

No two objects of the same shape will have the same colour i.e. if there are two circle objects in the image, then both circles will be of different colour.

However, two different objects (for example. a triangle and a square) within the same image can have the same color.

There will be a minimum of one object and a maximum of two objects to be detected for any given image. Also, you have to detect objects which are only of Red, Blue or Green colour.

If object 1 or object 2 is **specified** as "**None**", it means no object is to be detected for object type 1 and/or object type 2 respectively.

Problem Description:

Each team will be given five images:

1. Images named **Image1.jpg** and **Image2.jpg** contain ArUcos belonging to a dictionary of **4x4** Bit size with **100** combinations.
2. Images named **Image3.jpg** and **Image4.jpg** contain ArUcos belonging to a dictionary of **5x5** Bit size with **1000** combinations.
3. Image named **Image5.jpg** contain an ArUco belonging to a dictionary of **7x7** Bit size with **250** combinations.

The team is expected to detect the ID and angle of the ArUco marker using these designated Bit size and combination dictionaries of the ArUco marker. The above information in table format is given below:

Table 1: ArUco Dictionary Specification

Image Name	Bit Size	Combinations
Image1.jpg	4x4	100
Image2.jpg	4x4	100
Image3.jpg	5x5	1000
Image4.jpg	5x5	1000
Image5.jpg	7x7	250

Once the ArUco ID and angle are detected, it should then perform colour and shape detection of objects as mentioned in Table 3 below. Also, outline these objects with the indicating outlining colour specified in Table 2 below.

Table 2: Outline Colour Specification

Colour of Object	**Outline Colour**
RED	GREEN
GREEN	BLUE
BLUE	RED

The sequence of colour objects associated with the particular ArUco marker is given in Table 3 below:

Table 3: ArUco marker - Colour Object Sequence Specification

Image Name	Colour of Object 1	Shape of Object 1	Colour of Object 2	Shape of Object 2
Image1.jpg	Red	Square	Blue	Square
Image2.jpg	Blue	Triangle	Green	Circle
Image3.jpg	None	None	Red	Triangle
Image4.jpg	Green	Square	Red	Circle
Image5.jpg	Blue	Circle	Green	Triangle

Consider an example Image1.jpg is as shown in Figure 1.

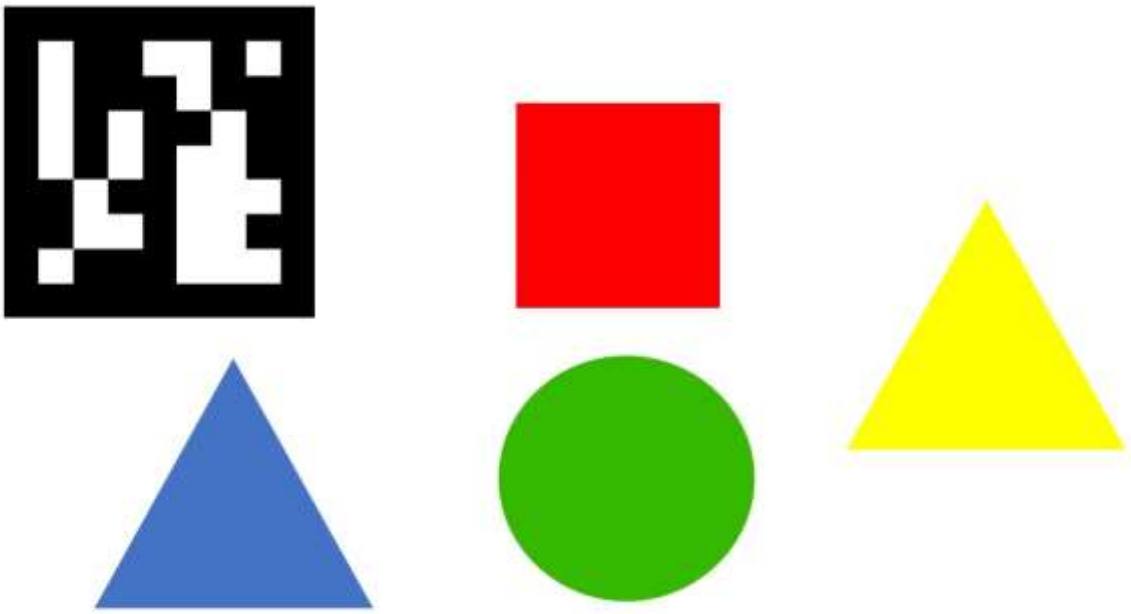

Figure 1: Sample input Image1.jpg

Now, if for this input image, the **1st Object** is a **Red Square** and the **2nd Object** is **Blue Triangle**; then the output result image should look like that shown below:

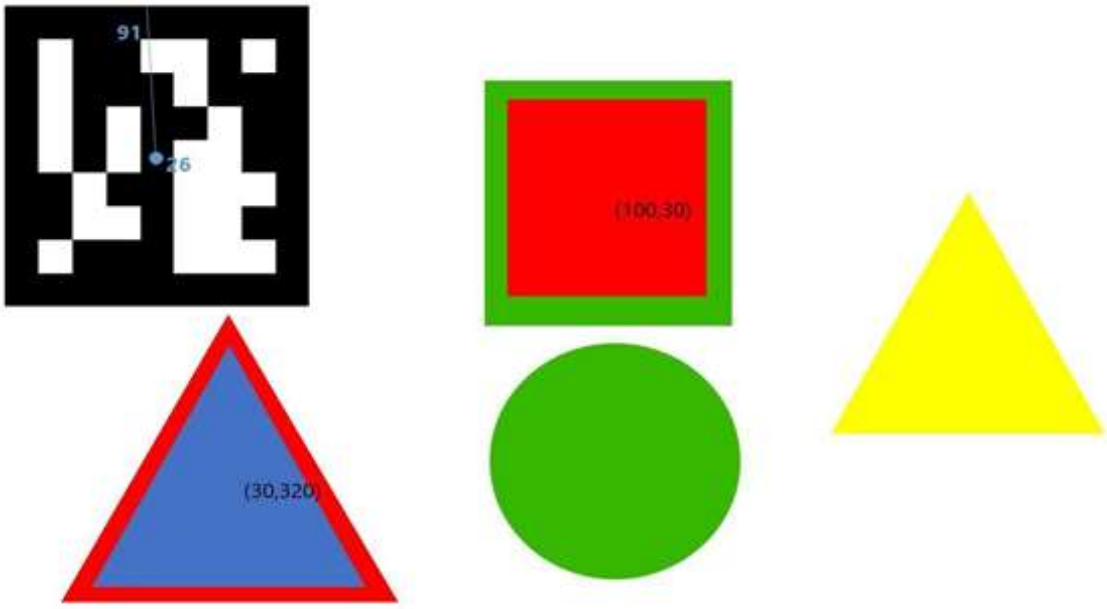

Figure 2: Sample Output Image

Required Output:

1. A **jpg** image containing the ArUco detected with its ID and the highlighted objects with the required colour outline of 25 pixels border per input ArUco jpg. Save this output image within the folder named **"Images"** within a submission zip folder named **"<TeamID>_Task1.2.zip"** i.e. If your team ID is 1001, the folder should be named **"1001_Task1.2.zip"**
2. Code file that contains the algorithm to solve this problem in a folder named **"Code"** within a submission zip folder named **"<TeamID>_Task1.2.zip"** i.e. If your team ID is 1001, the folder should be named **"1001_Task1.2.zip".**
3. A **.csv file** containing the **ArUco ID, object's centroid coordinates** in **(x-coordinate,y-coordinate).** A sample csv format for Sample input image in Figure 1 is as shown in Table 4:

Table 4: Sample Output .csv

Image Name	ArUco ID	(x,y) Object-1	(x,y) Object-2
ArUco1.jpg	26	(100,30)	(30,320)

Thus your result folder named **"<TeamID>_Task1.2.zip"** should contain **2 subfolders** and a **csv** file as listed here-

1. **Images**
2. **Code**
3. **<TeamID>_Task1.2.csv**

To do:

Open **Task1.2.py** located in the folder named **"Code"**.

It has three functions:

- **main():** which calls the function to detect ArUco markers given in each input image.
- **aruco_detect():** This function expects one parameter as arguments - name of the input image file with full path.
- **color_detect():** This function expects one parameter as arguments - the OpenCV Image file. Call this method only once the ArUco ID is detected. You have to factor in this logic in the **"aruco_detect"** function. In this method please write the code for populating the required output csv.

Edit these functions as per your generalised algorithmic code.

Save this modified code with your **TeamId** in the name of the code file as **<TeamID>_Task1.2.py**. Please use your eYRC team id while actually naming the file i.e. if your TeamID is **1001**, then the name of the code file should be **1001_Task1.2.py**.

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 the specified objects in an image and enlist their mentioned properties vis. ID of ArUco, and Positions of the centroid of coloured shape objects. 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. Objects are not rotated or differently oriented with respect to each other.
4. In case, the specified objects are found in the image, return the List of **(x-coordinate,y-coordinate)** of the centroid of the image in addition with **image name** and **ArUco ID** in a csv. Thus the csv should contain 6 rows. The first row should be the title of the row and remaining 5 rows should contain that particular data for that input image (as depicted in Table 4 above).
5. You are permitted to separate the ArUco marker section of the image while detecting ID if this is easier for you to do. However, the resulting image should contain the Aruco at the original position and orientation with the detection overlaid as shown in Figure 1 above.

Happy Learning

All The Best!!!