

## Solution 1 (OpenCv):

There are different approaches for this problem

As there were no training data for this problem I took an unsupervised approach using opencv2 library.

- Created a function which accepts only image extension in the path and pass it to cv2 imread method to read an image and create a copy
- Then the image is converted from default BGR to HSV format because R, G, B in RGB are all co-related to the color luminance( what we loosely call intensity),i.e., We cannot separate color information from luminance. HSV or Hue Saturation Value is used to separate image luminance from color information. This makes it easier when we are working on or need luminance of the image/frame.
- Then we specify range of color which we want to detect that is red, green and yellow.
- Then I have used specifically HoughCircles method in Cv2 which detect circles in the images.
- With the sliding window we will check for the color ranges and will mark a circle on it and put text with respect to color on it
- And will write the image with the help of cv2 imwrite library

## Solution 2 (TensorFlow):

- I tried taking TensorFlow approach taking my own training and testing sample and using an image labeller.
- Check requirement.txt for the library used in the project
- Used google colab for this approach, this program can be run on windows too.
- Used a transfer training model **SSD MobileNet V2 FPNLite 320x320** as per my own choice because of the speed as it takes 22 ms and with 22.2 precision considering the tradeoff for the detection and the model latency. For references [https://github.com/tensorflow/models/blob/master/research/object\\_detection/g3doc/tf2\\_detection\\_zoo.md](https://github.com/tensorflow/models/blob/master/research/object_detection/g3doc/tf2_detection_zoo.md)
- If we consider for raspberry pi this model is suitable for the application.
- Clone the models from <https://github.com/tensorflow/models>.
- Changed the config of the model as per our application.
- And then trained the model and then evaluated on test set.

Training was successful and the model could detect some images not all as the image used for training purpose was very few.

There is a chance of improvisation by training using lots of data

