**Ball Detection using open cv**

Various approaches were tried for detecting the ball, but the basic idea was:

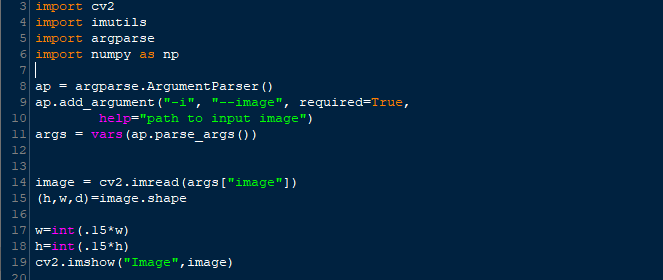
* Convert to HSV.
* Generate a Mask (with lower and upper values and a “inRange” function) to separate the given color
* This mask is later exploited with contour functions, to detect the ball

There are 2 files present for testing:

1. ballDetect.py
2. ballDetectVideo.py

Below is discussed each step in details.

1.)



All the necessary libraries are imported in line **3 – 6.**

Argparse is used to take the image input from the user in command window.

Lines 8-11 take the input from cmd window.

Rescaling of the image is done in lines 14-18



Actual image :



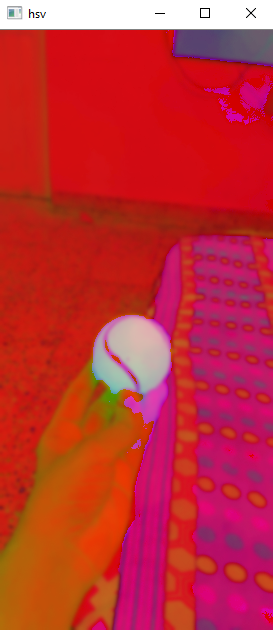
**Applying blur**





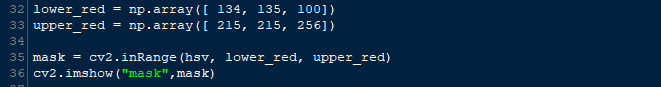
**Converting to HSV**



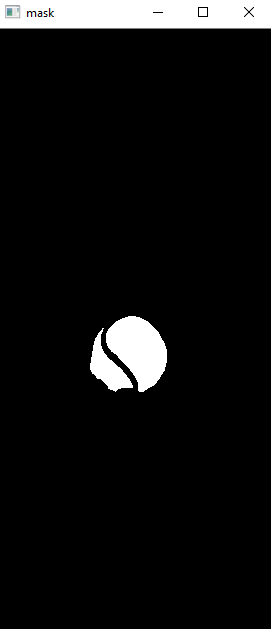


**Seperating mask.**

Using the color picker code, we pick the range of the ball. Which comes out to be –

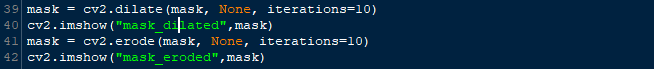


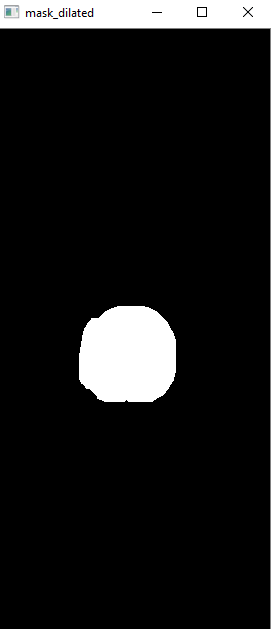
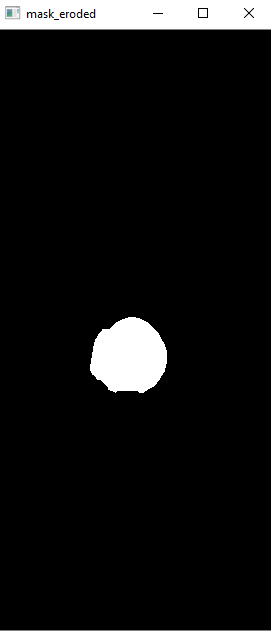
*cv2.inRange*is used to obtain the mask.



**Dialating and Eroding**

There is still some noise, which is eliminated in this process.

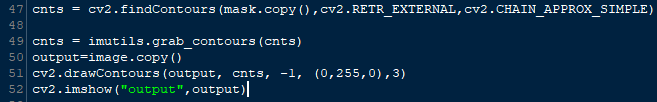


For detecting, following methods were tried.

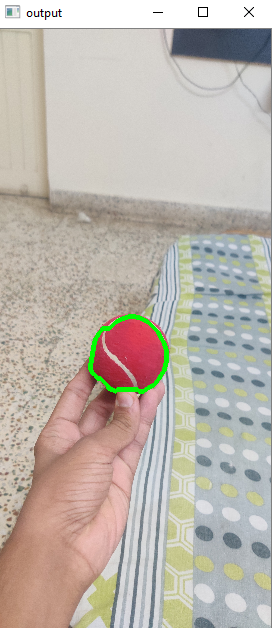
All the detailed description of contour related function is available on : <https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_imgproc/py_contours/py_contour_features/py_contour_features.html>

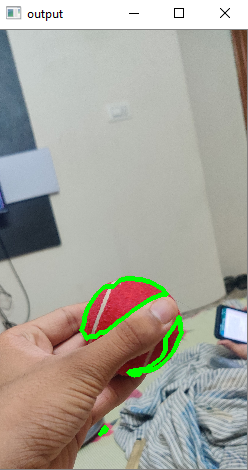
1



In this method, contour is grabbed, and is drawn on original image.

This works when whole ball is visible. But fails some times.

. 

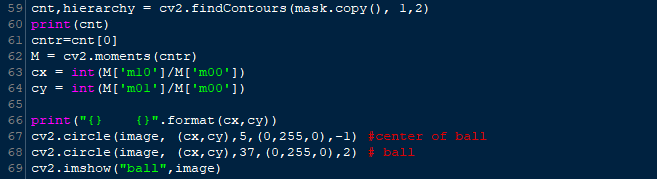


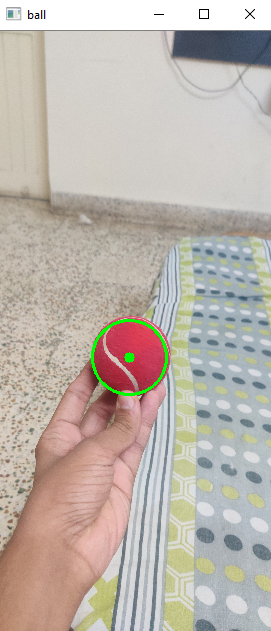


2.

Center of contour is detected, if only a single contour is found. Thius method fails if multiple contours are found

This happens because we are not choosing the biggest contour.



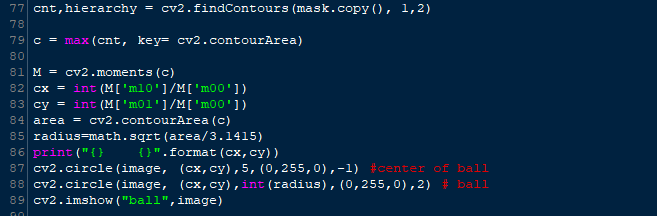


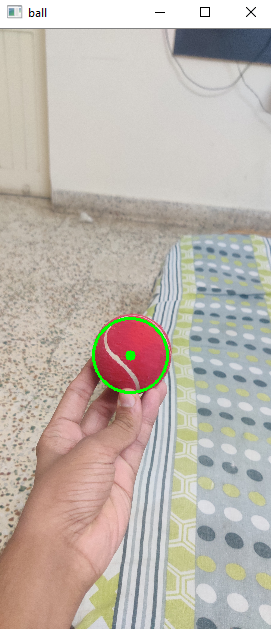


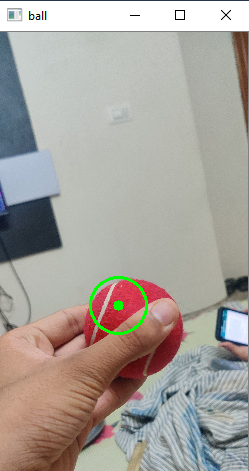
3.

Moments are used here too, but here we consider all the contours and maximum is choosen.

This works well in most of the cases. But this method gives us the centre. Radius obtained is not always accurate.



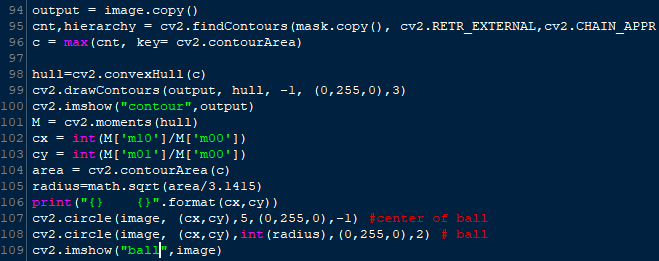


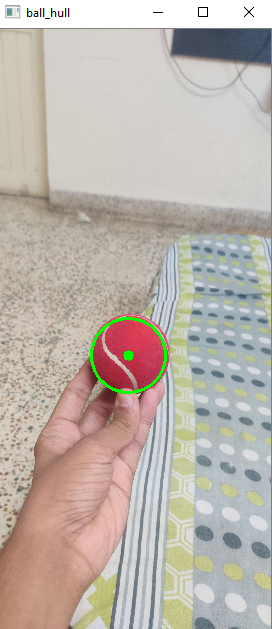
 detected in multiple contours.

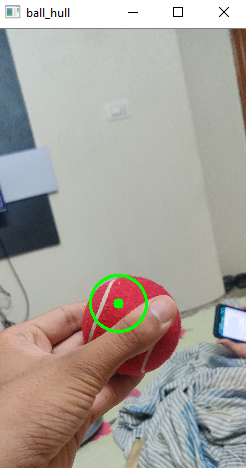


4.

Here a *hull* is used, which is a feature, that removes concave sides from contour. This helps in improving output for some explicit cases.







5. here circle radius and centre is directly obtained, without using moments.

