WORKING LOGIC

1. Create brightness adjusted image:

Iterate thru all the pixels in the image and add 50 to each of the RGB values of each pixel. If any of the pixel values goes over 255, make them equal to 255 (as 255 is the max intensity a pixel can have).

1. Converting from RGB to HSV color space:

Make a copy of the original image. Iterate thru all pixels of the original image and for each pixel calculate the H, S and V values. Then replace the existing pixel values of the copy with the obtained the HSV values. In OpenCV, the values need to be stored in the order V, S, H to get the desired (expected) results.

1. Object detection on RGB:

Make a black matrix of the same dimensions as the image. Iterate thru all pixels and for each pixel:

1. Check if the R, G, B values are in the range (150,255), (0,80), (0,80) respectively. The thresholds for the brighter image are 50 more than the normal thresholds, i.e. (200,255), (0,130), (0,130).
2. Mark the pixel white in the black matrix and append the pixel coordinates in an array if it satisfies the above condition.
3. Finding radius:

Method 1: Find the difference between the minimum value j in the array (j is the column number, and this gives the j value of the pixel closest to the edge of the screen) and the j value of the center. This will give us an approximate of the radius.

Method 2: Find the difference between the minimum and maximum j values. This will give us the diameter of the apple, which can be divided by 2 to get the radius.

Object detection based on HSV:

Make a black matrix of the same dimensions as the image. Iterate thru all pixels and do RGB to HSV conversion. Iterate thru all the pixels of the HSV image and for each pixel:

1. Check if the H, S, V values are in the range (0,5), (150,255), (0,255) respectively. The thresholds for the brighter image are the same as the hue and saturation don’t change.
2. Mark the pixel white in the black matrix and append the pixel coordinates in an array if it satisfies the above condition.
3. Finding radius:

Method 1: Find the difference between the minimum value j in the array (j is the column number, and this gives the j value of the pixel closest to the edge of the screen) and the j value of the center. This will give us an approximate of the radius.

Method 2: Find the difference between the minimum and maximum j values. This will give us the diameter of the apple, which can be divided by 2 to get the radius.

1. Turning red apple into green and vice versa:

Iterate thru all pixels in the image and for each pixel, swap r and g values of the pixel.