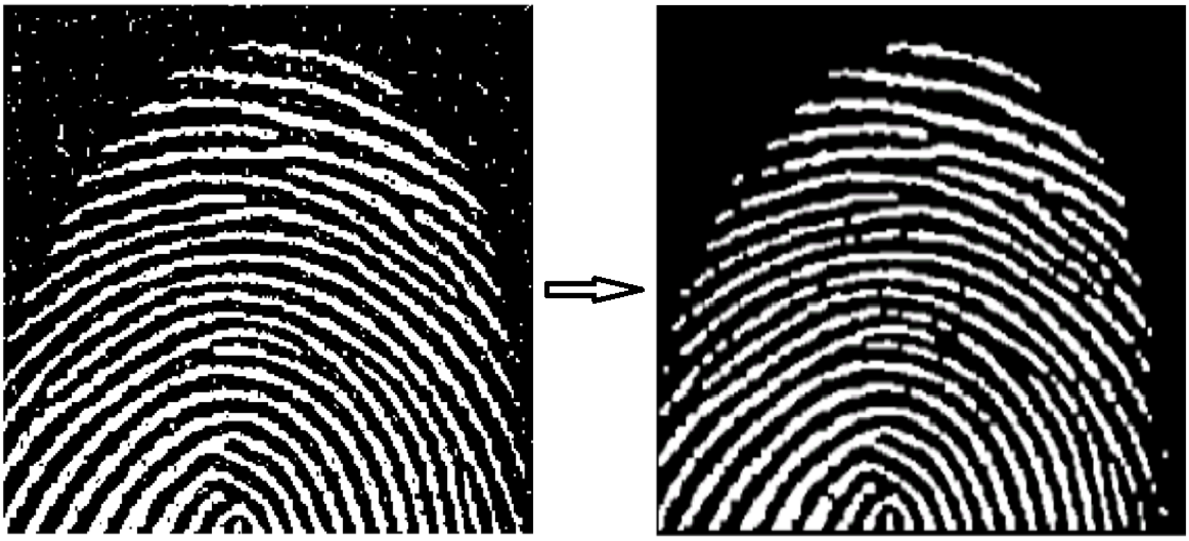
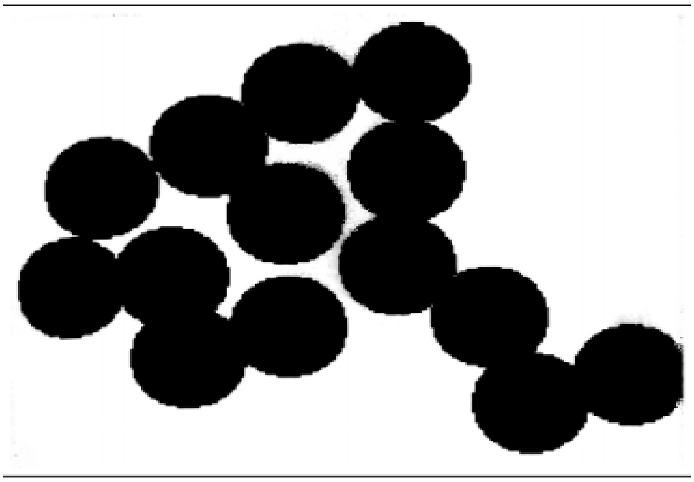
1Q. Develop a program that would use appropriate hand coded morphological operations to enhance the given **fingerprint.png** image as the resultant output image.



2Q. Implement a program to count the number of disks in the image **disks.png**, which has a number of touching black disks on a white background.



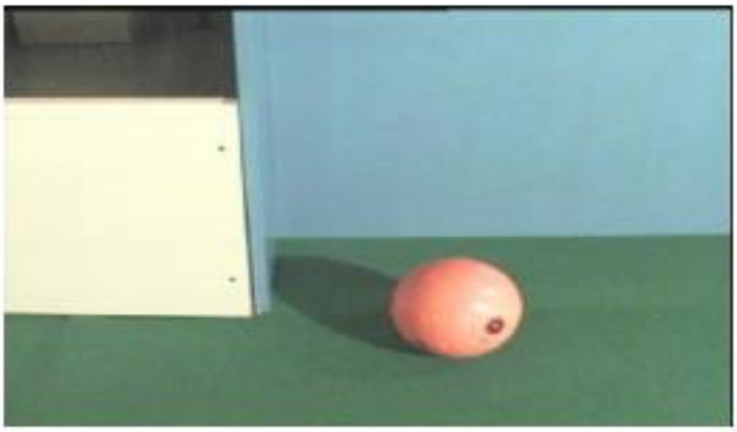
3Q. The task here is to help a robot to identify a bright orange ball in its surrounding. The **ball.bmp** is an image obtained from a camera mounted on the robot.

a) Implement a **Myrgb2hsi** function to convert red-green-blue (RGB) colors to hue saturation intensity (HSI). The function prototype should be:

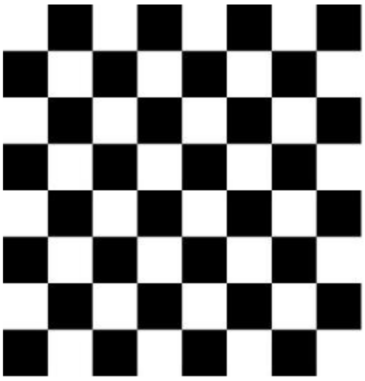
function [H, S, I] = Myrgb2hsi(Im)

where Im is the original color image, and H, S, and I are the normalized hue value, saturation-value, and intensity value in the HSI color space, respectively. That is, H, S, and I should be in the range of [0, 1]. Load and convert **ball.bmp** to HSI color space by calling the **Myrgb2hsi** function. Display the three images in figures 1 to 3 with the appropriate titles. Call an appropriate OpenCV/Matlab function to do the similar conversion and display the three images (i.e., Hue, Saturation, and Intensity) in figures 4 to 6 with appropriate titles. Display the difference images between your results and the Matlab's results in figures 7 to 9 with appropriate titles. Explain the reason for these differences and the visual differences between your results and the Matlab's results.

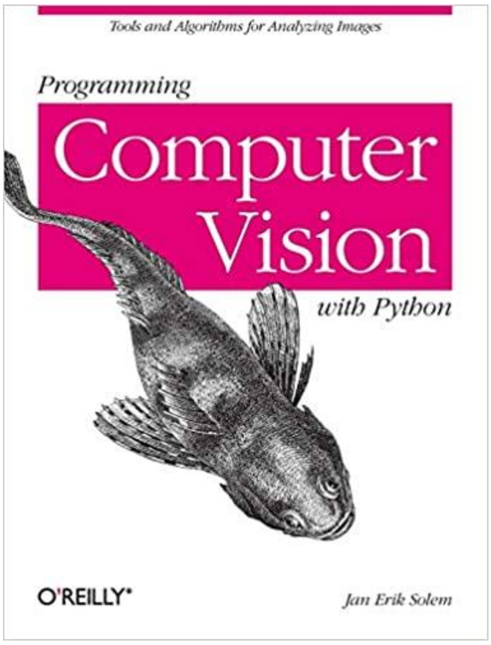
b) In H-space, find a threshold for the ball. Find the centroid of the ball and indicate its location by a cross on the original color image.

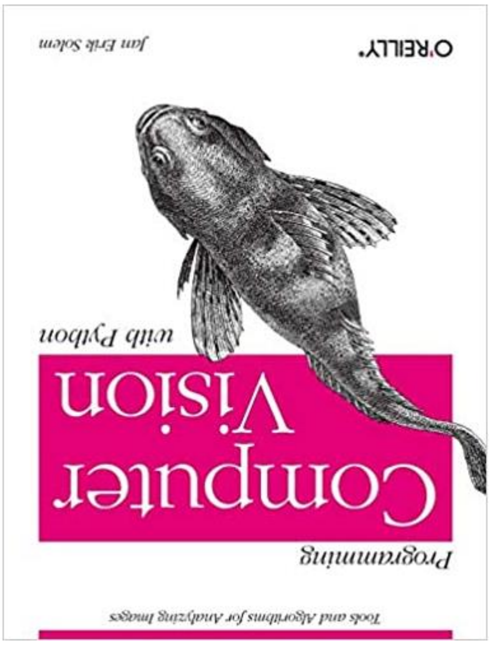


4Q. Develop a function to implement Harris corners detection algorithm. Use the function to detect corner points in the given **corner_test.png** image. Compare the hand crafted function result with built in cv2.cornerHarris() function result to test your code.



5Q. Write a program to perform robust image matching using RANSAC algorithm and Harris Corner features. Use **FM_img1.jpg** and **FM_img2.jpg** to test the program.


Tools and Algorithms for Analyzing Images
Programming
Computer
Vision
with Python
O'REILLY®
Jan Erik Solem


Tools and Algorithms for Analyzing Images
Programming
Computer
Vision
with Python
O'REILLY®
Jan Erik Solem