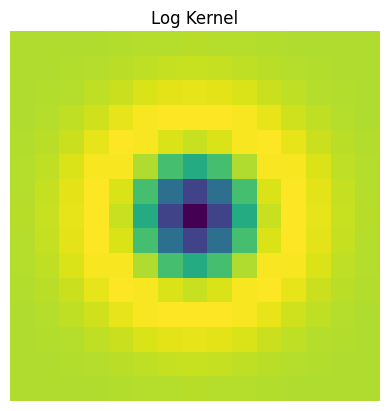
**EN3160 Assignment 2 on Fitting and Alignment**

Rathnasekara T.S. 200529H

GitHub Link : [*Click here to visit*](https://github.com/TishanSathruwan/ImageProcessing_MachineVision/tree/main/Assignments/Assignment_2)

# **Question 01**



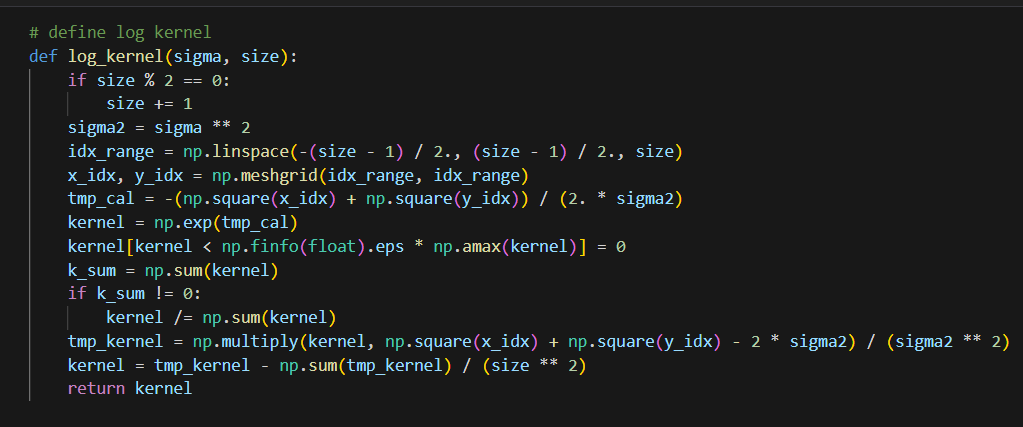
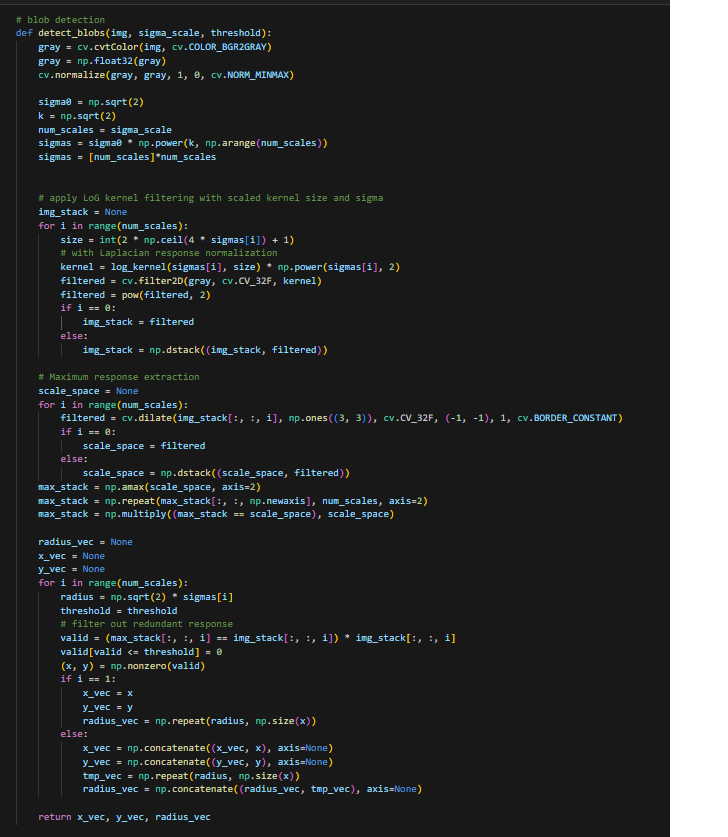


Figure : Laplacian of Gaussians Filter

Here, Laplacian of gaussian filter has been implemented for blob detection. The shape of the kernel can be shown above.

This is the code for blob detection. First, the image was converted into gray scale and then it was passed through the “LoG” filter to get filtered images. By using that filter we can enhance the areas where the features which are similar to filter, are present.

So, after filtering , we extracted coordinates of the points which show the maximum response and respective radius were calculated to detect the blob.

Then, this function returns the coordinates of the local maximum responses and radii were calculated according to the sigma values.

**Sigma values range is from 2 to 9.**



Figure : Code for blob detection

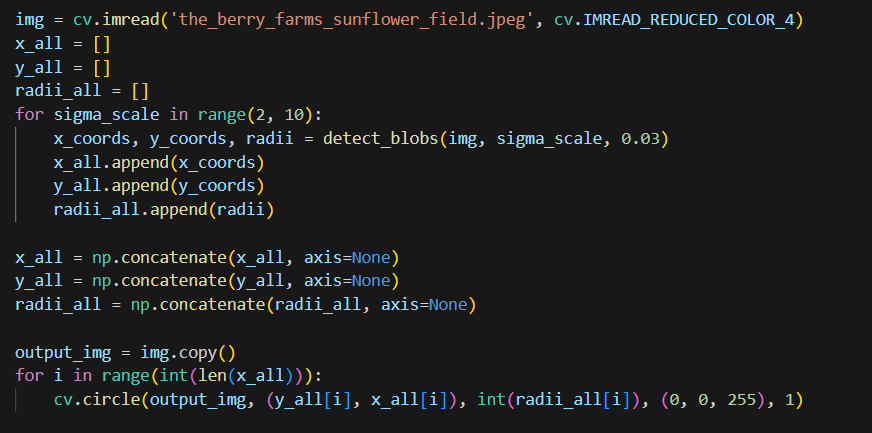


Figure : Resultant Images

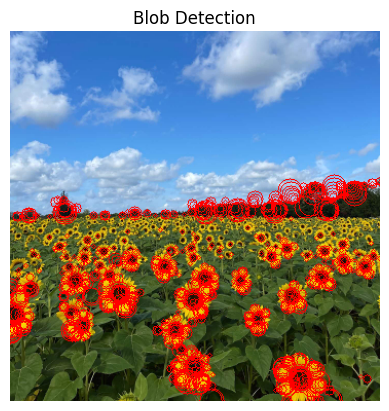


Figure : Blobs detection of various radii

According to the code, blobs of various radii were detected using this code and using OpenCV, circles were drawn on bob to identify. The resulting images can be seen above.

# **Question 02**

## Best fitting Line Detection Using RANSAC

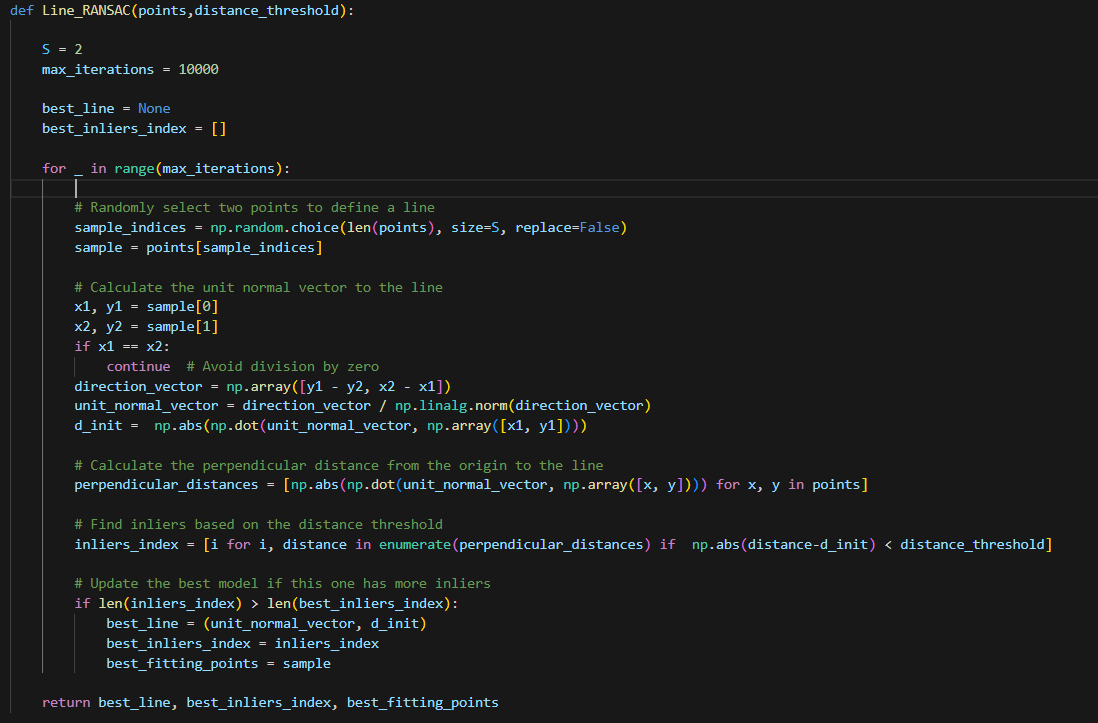


Figure : RANSAC algorithm for line estimation

For the detecting inliers in given data points first two random points were detected and then line was estimated using these two points.

Then I defined the distance threshold to select the best inliers and run this loop several times to get best inliers from given data set.

## Circle Estimation Using RANSAC

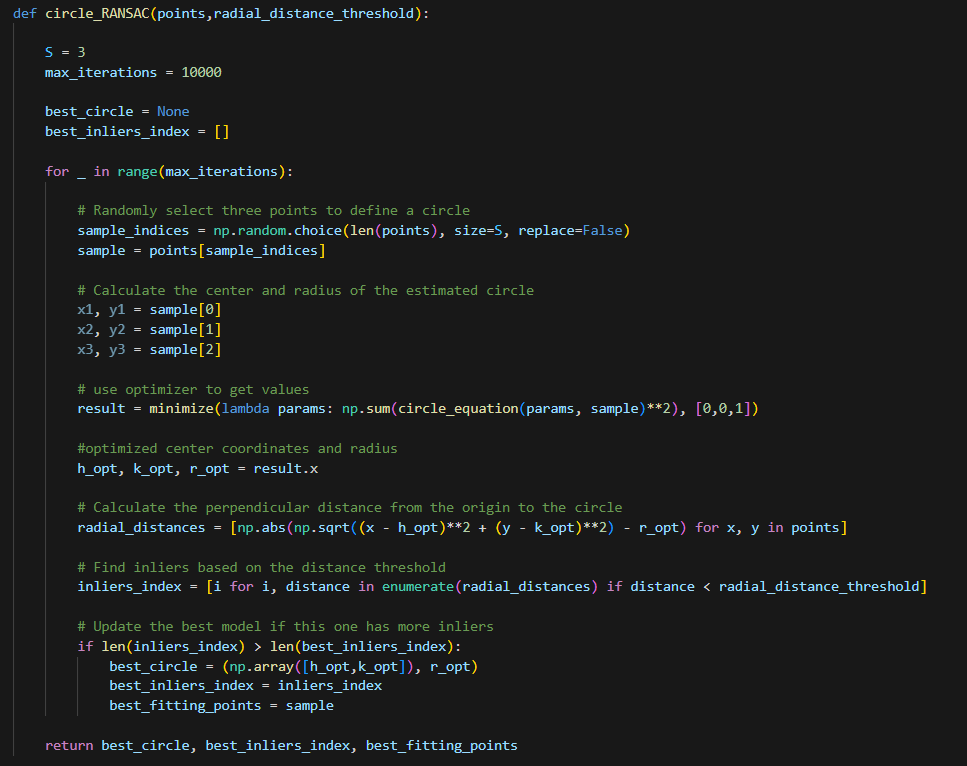


Figure : Circle Estimation code

This algorithm is also same as above line detection code. Instead of using two points, here I used three points for estimating the best fitting circles. Apart from that part, all the other parts are same as before.

Here, algorithms tries to obtain maximum numbers of inliers.

Output of above function were represented as follows.

## Plotting Estimations (d) What will happen if we fit the circle first ?

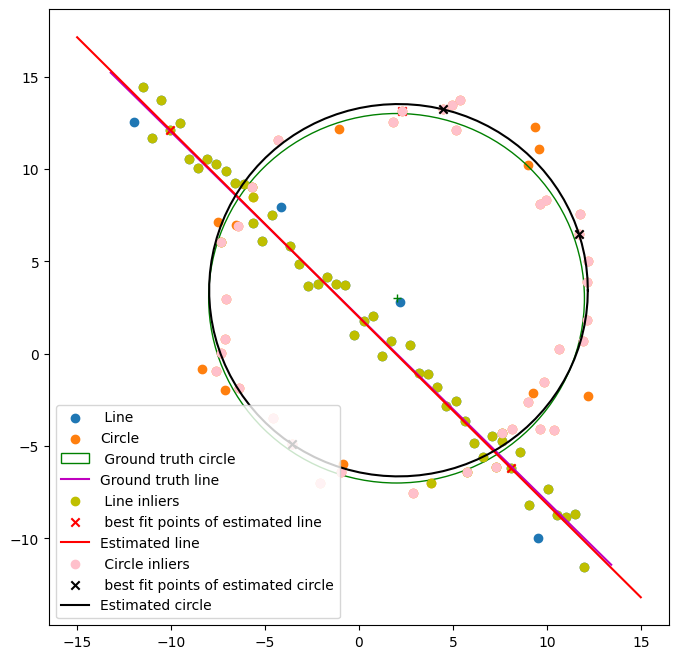
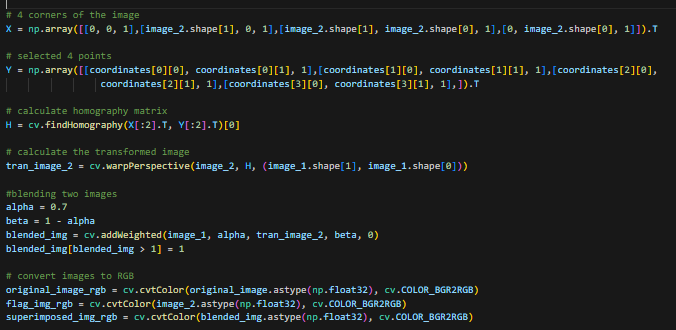


Figure :Estimated line and circle

If we first fit the circle, then it uses some points which belong to the line and selected all inliers are not from the circle. So, this gives wrong prediction. As well as , after calculating circle, if we subtract the inliers of circle then line points will be separated into 3 groups and that leads to wrong prediction.

# **Question 03**

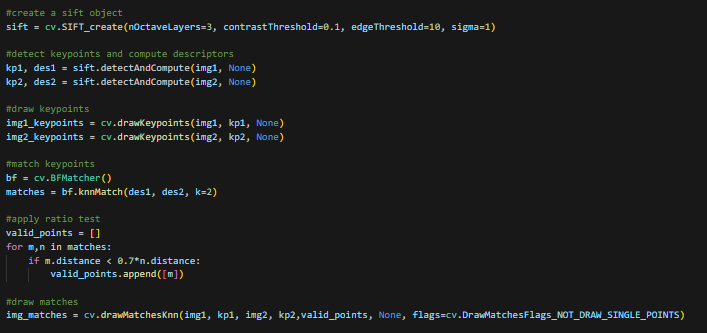
Figure : Superimposed image and code



Here I used selected four points and four points of image 2 to calculate homography matrix and then do warping using in built function of OpenCV . After these images were blended to obtain superimposed images.

# **Question 04**

## (a)



## (b) , (C)



Homography matrix

Here, I used RANSAC algorithm to find inliers to generate homography matrix and then do warping to get warped image for blending with image 2.

