**Assignment #1**

**Automatic pieces detection and quantification in thermographic image**

* **Libraries used**

1. OpenCV Python Library
2. NumPy Python Library
3. MatplotLib Python plotting Library

* **Algorithm**
  1. **Resize the original image**

The original input image is resized while maintaining aspect ratio so that parameters of Hough circles function are the same

* 1. **Convert to grayscale**

convert resized image to grayscale

* 1. **Apply GaussianBlur**

Apply GaussianBlur with 5\*5 kernel to gray image to reduce noise and improve contour detection

* 1. **Apply HoughCircles to detect circles**

Apply Houghcircles function to blurred image to obtain all circles even partial circles since votes parameter is not returned

* 1. **Apply Sobel Y filter**

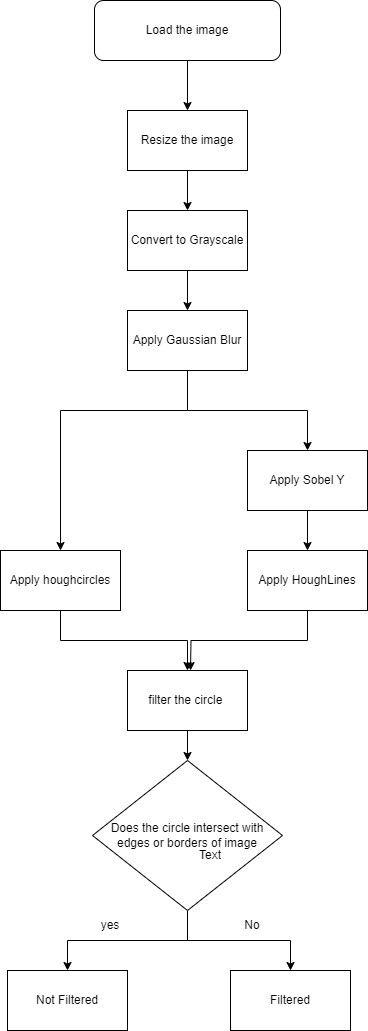
Apply Sobel Y filter to blurred image to detect horizontal edges used later to detect intersection of edges with circles since main edge is horizontal

* 1. **Apply Hough Lines to Sobel Y filter**

Apply HoughLinesP to obtain line segments in Sobel Y convolved image

* 1. **Detect and filter the circle**

filter the circle only if it is not intersected with any edge obtained from Sobel Y convolved image or an image border by comparing it’s radius with perpendicular distance to the edge or border



* **Visual outputs**

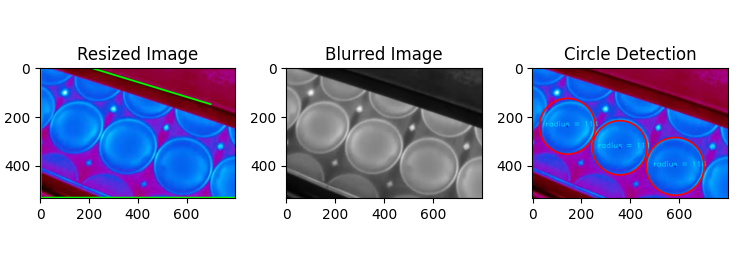
Run **test\_per\_image.py** for each of the input images to get the following results.

# Load image

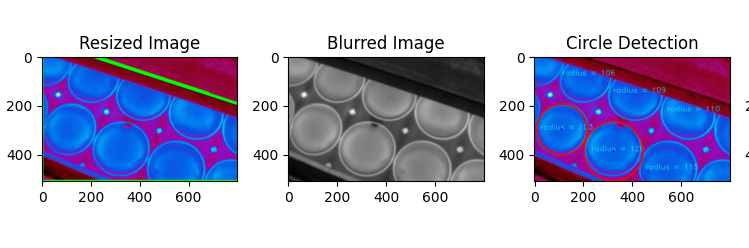
image = cv2.imread("images/image1.png")

The edge lines are shown in the resized image.

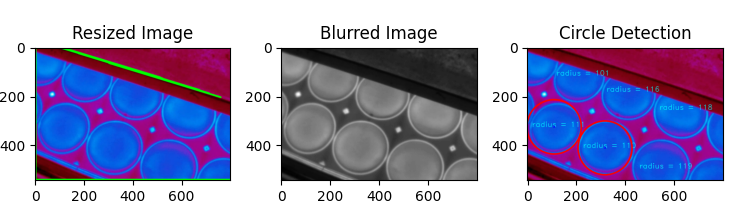
* 1. **image1.png**



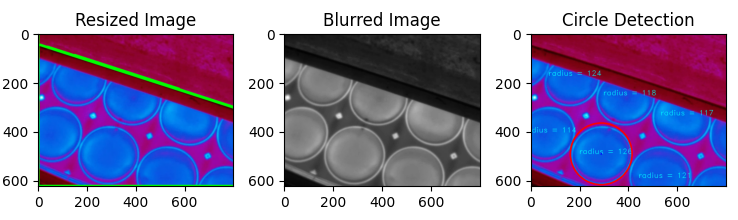
* 1. **image2.png**



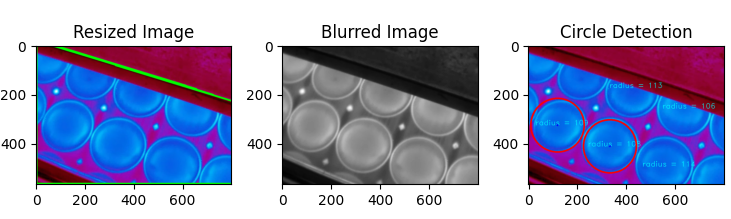
* 1. **image3.png**



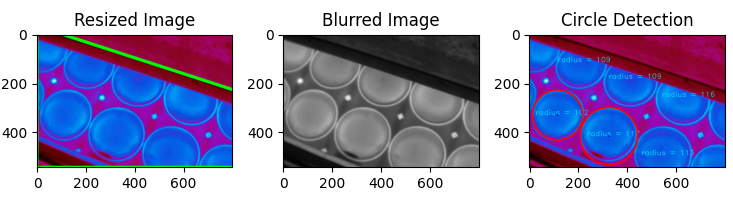
* 1. **image4.png**



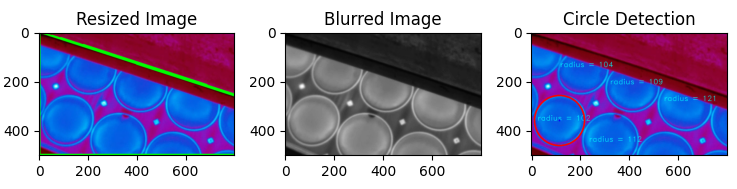
* 1. **Image5.png**



* 1. **Image6.png**



* 1. **Image7.png**



Run **Assignment1.py** to get the results for all input images in one figure

