# Introduction

In this assignment, the aim is to detect the void in the solder paste in the image.

Solder voids are created when gasses released from solder paste are trapped into the solder joint. Voids are like holes or empty space in the solder paste, which creates weakness which may cause cracks in solder joints. Voids cause failure in transferring heat away from a component which may cause overheating or thermal failure of the component.

Therefore, it is crucial to be able to detect these voids in the solder paste so that the software is able to recognise whether the solder paste is defective or not.

# Methodology

The image taken lacks contrast, so the first few things I tried are to try different ways to increase contrast so that binary thresholding can work more effectively.

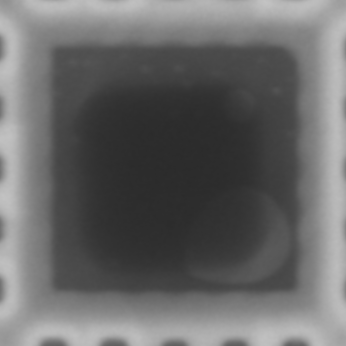
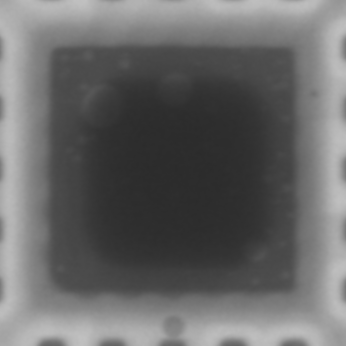
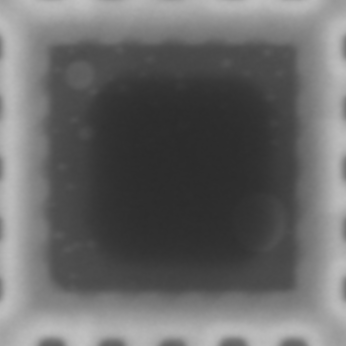
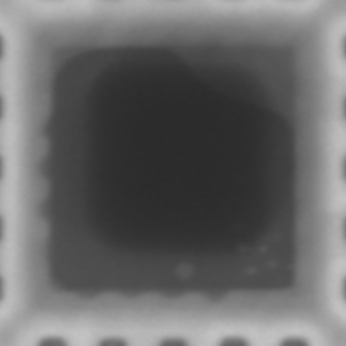
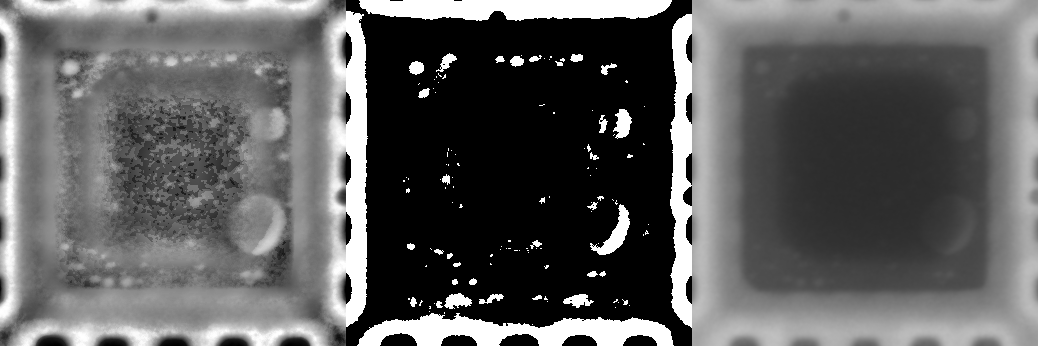


Figure: Original images

## Enhancing Contrast (CLAHE, Histogram Equalization, PIL increase contrast)

### CLAHE





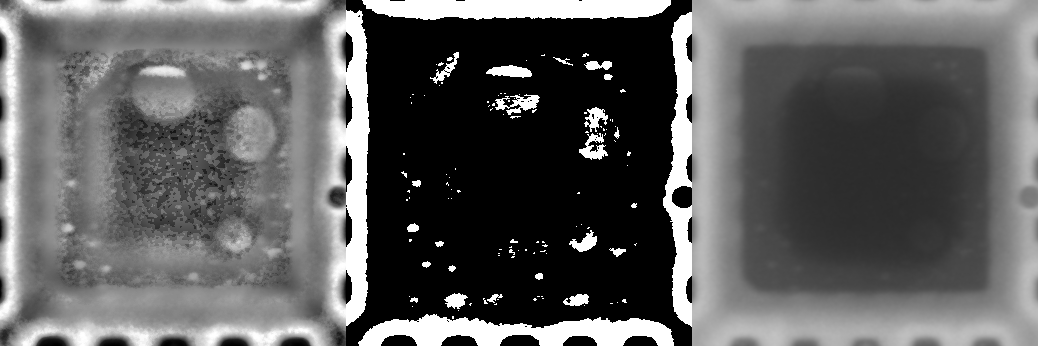
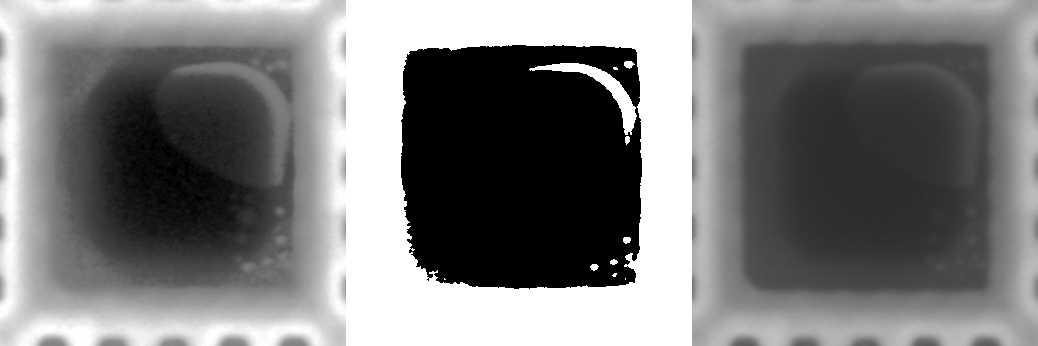
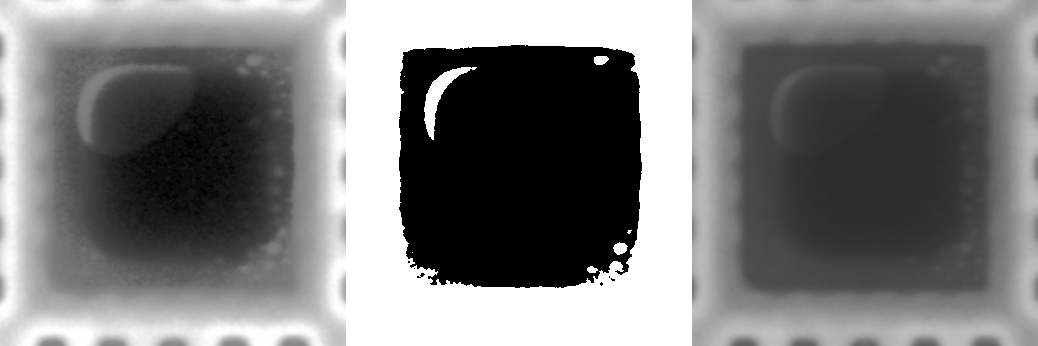
****

Figure : CLAHE (left), CLAHE with thresholding of 150 (middle), original image (right)

With CLAHE we are able to reveal most of the small voids and some part of the big voids. However, the middle becomes very noisy which the binary threshold mistakes for a void. Next we will use Histogram Equalization and see whether it provides better results than CLAHE.

### Histogram Equalization





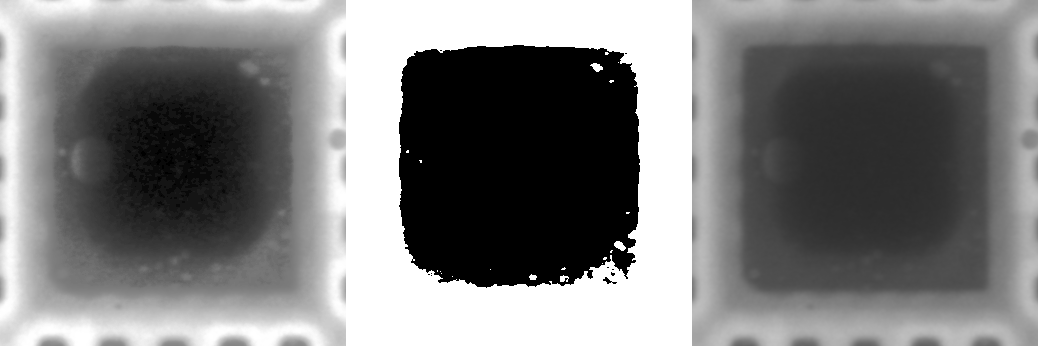
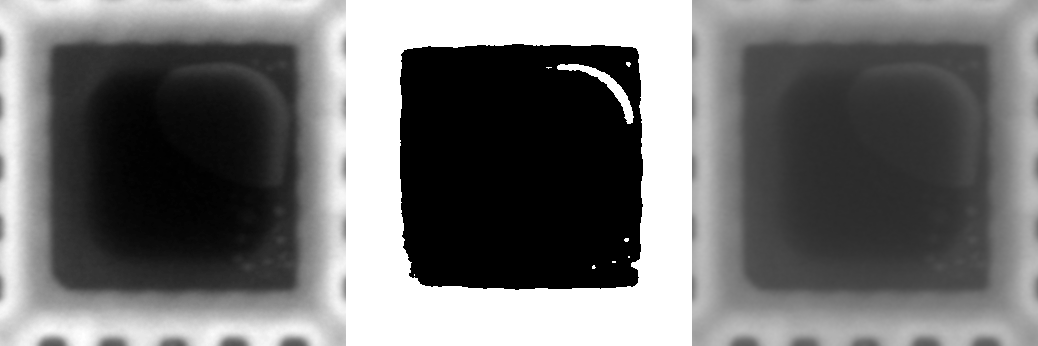
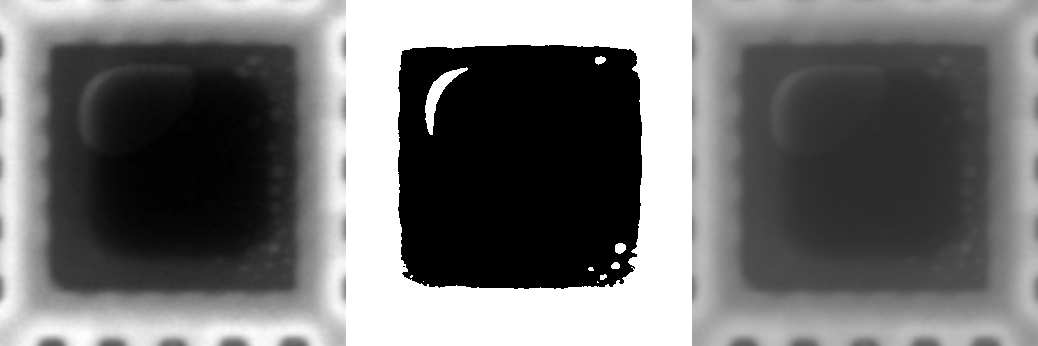


Figure : Histogram Equalization (left), Histogram Equalization with thresholding of 115 (middle), and original image (right)

The image after applying histogram equalization does have improved contrast, however it does not increase the contrast between the void and background as much as CLAHE, so the binary threshold algorithm has a harder time of segmenting out the void from the background. It still gets around 50% of the voids, but the results are not that good compared to CLAHE. Next, we will look at using PIL python library’s increase contrast function to improve the contrast.

### PIL increase contrast





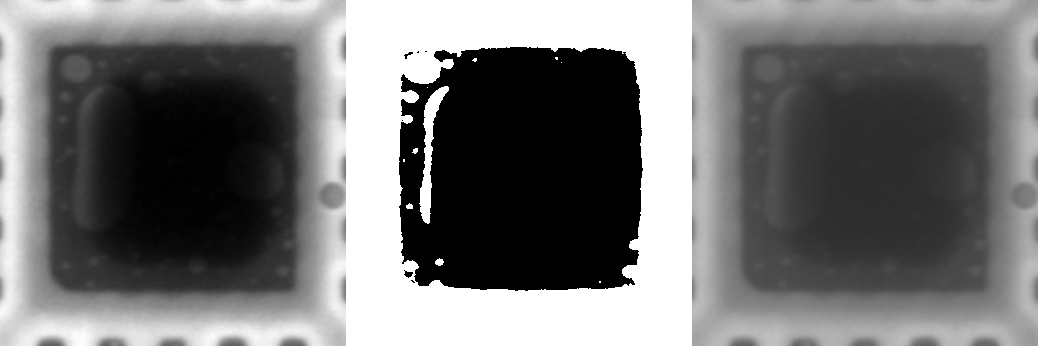


Figure : PIL enhance contrast (left), PIL enhance contrast with thresholding of 115 (middle), and original image (right)

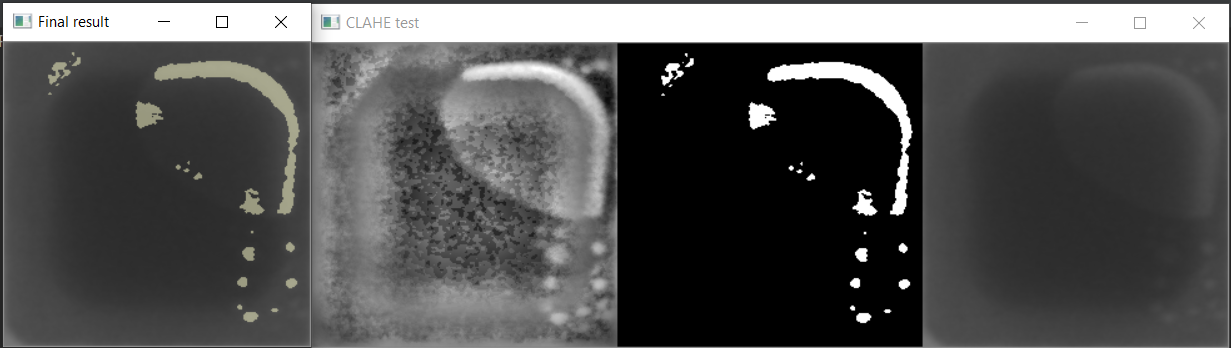
PIL enhances contrast functions by increasing the contrast by increasing blacks and increasing whites, but it does not equalise the contrast in the image. The thresholding shows almost the same results as histogram equalization.

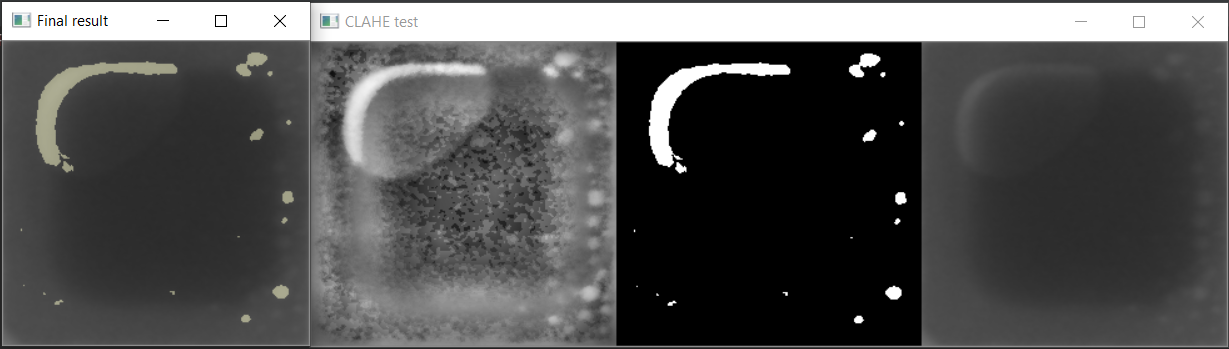
Thus, we will be using CLAHE to segment the voids and background so that we are able to detect them.

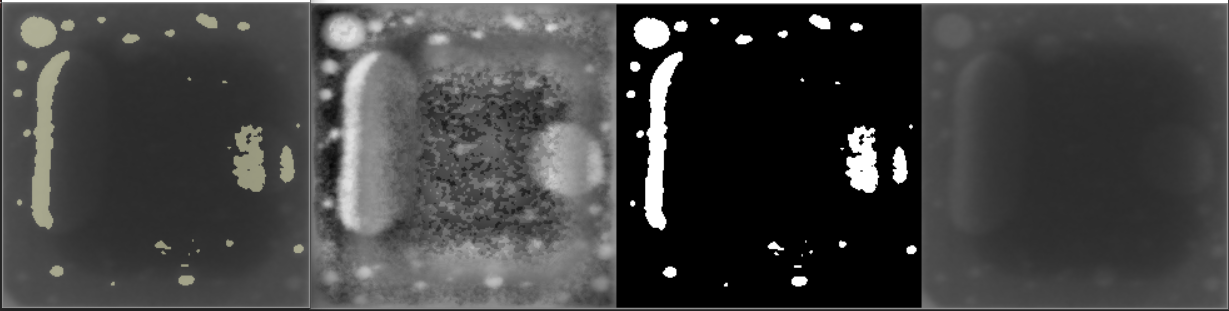
## Void Detection

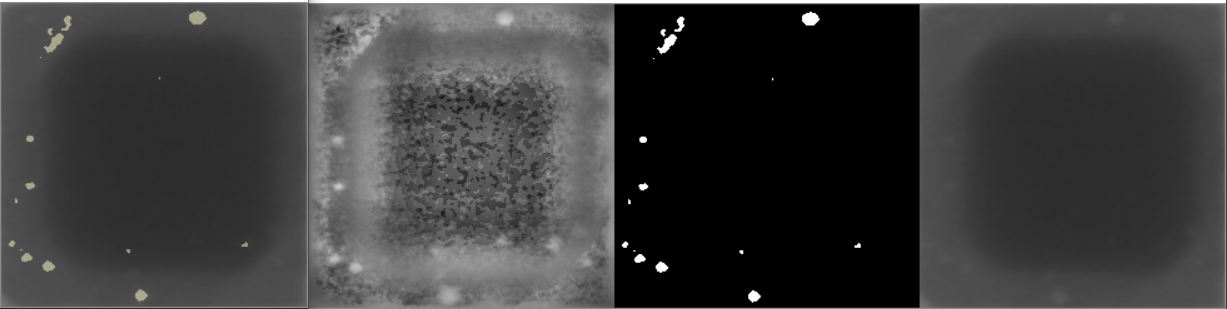
For detection of the void I will try different methods. We will use CLAHE and binary thresholding to increase contrast to improve segmentability. The first method I tried is to find contours to try to detect the void.

### Find Contour on CLAHE binary thresholding of 157









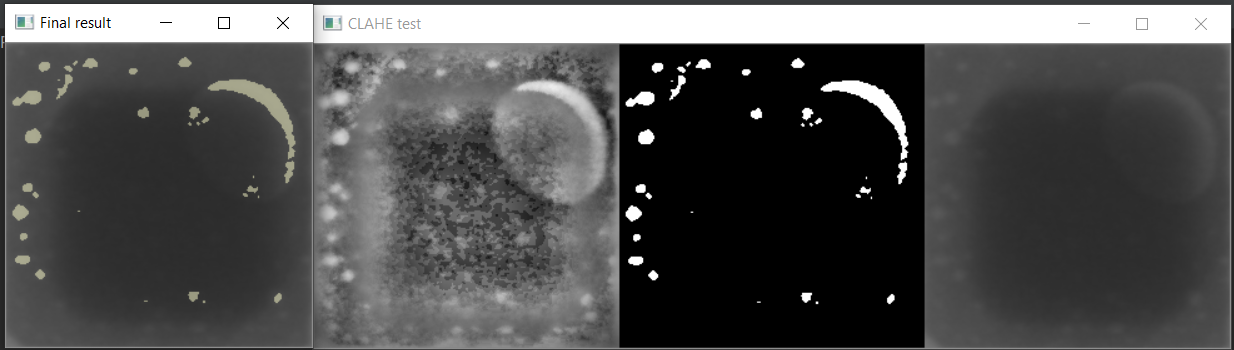


Figure: 1. Original Image with detected void

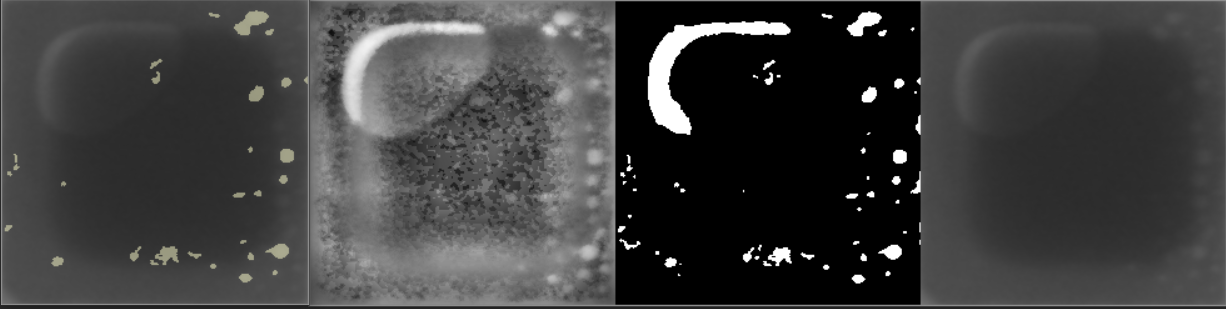
2. CLAHE

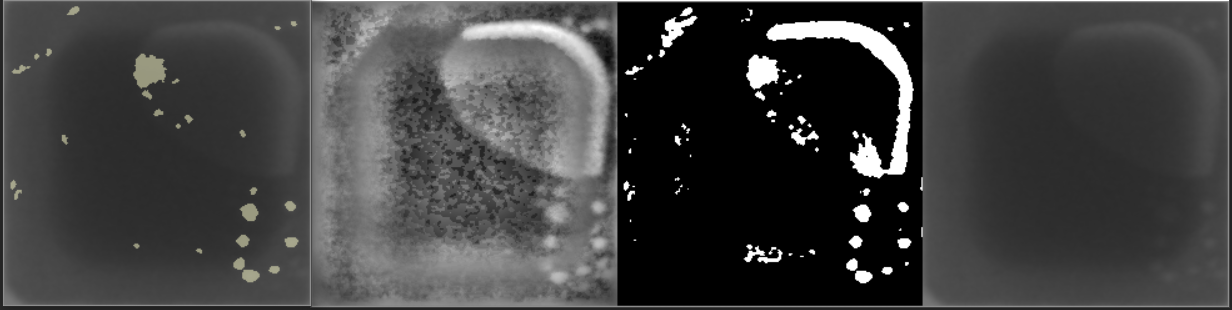
3. CLAHE and binary thresholding of 157

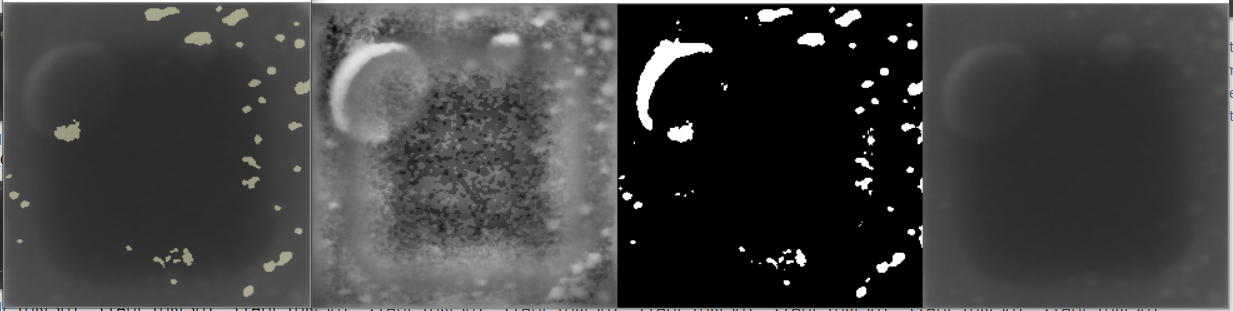
4. Original Image

The result is decent. Most of the small voids are detected, with some darker once undetected. The big voids that overlap the inner darker square also cannot be detected completely. Thus, it is decided that different methods will be used to detect the small voids and the bigger overlapped voids. Next, I tried to lower the threshold to 140, which shows more voids and other noise to be shown, then I will use circularity and contour area to filter out the noise as much as possible.

### Find Contour on CLAHE binary thresholding of 140, then filtered with area and circularity.







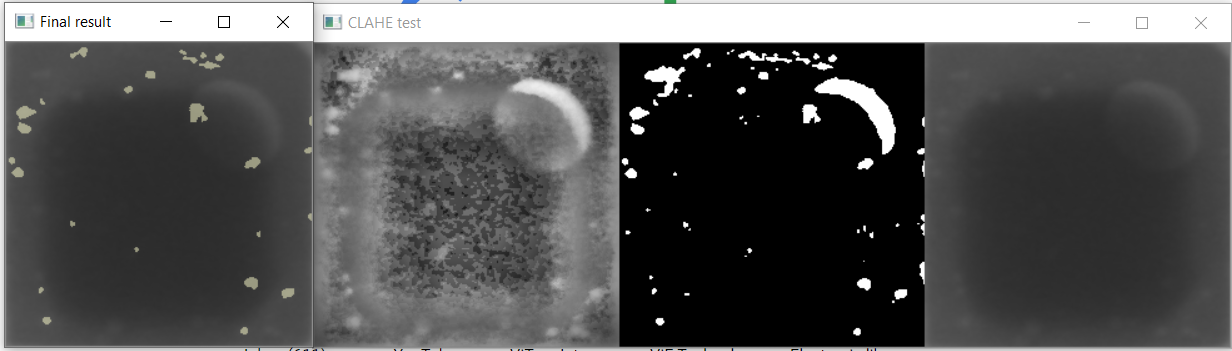


Figure : 1. Left: Original Image with detected void filtered with area and circularity

2. 2nd left: CLAHE

3. 2nd right: CLAHE and binary thresholding of 140

4. Right: Original Image

The area of the contours are limited to between 5 to 1000 px^2 and the circularity is limited to between 0.3 and 1.2. More of the void is detected by this method, however some of the noise bypasses the filter and is detected as void. Overall, it is able to detect more voids compared to the first method but it shows a lot more false positives, so this method is less accurate when detecting the small voids. After that, I decided to use canny edge detection after thresholding with threshold value of 110 to find the voids, and find all the contours to detect the void.

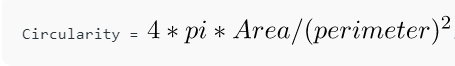


Figure : Formula for Circularity

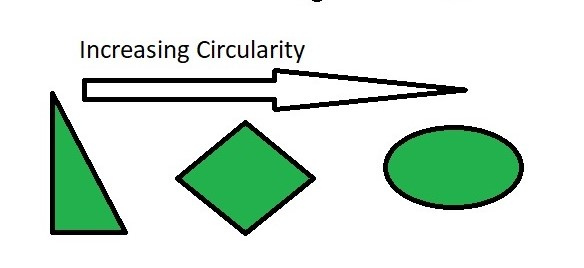


Figure : Circularity examples

### Edge detection then find contour on CLAHE binary thresholding of 110

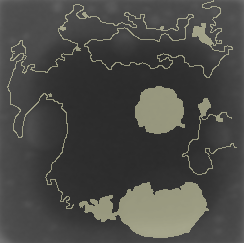


Figure : Image with edges and voids detected

As we can see here canny edge detector did detect the edges of these voids, but the edges are not great, so most of the voids are not seen as contours by OpenCV’s findContour function. Some voids are detected but most of the voids are not detected. This method shows the worst result and cannot be used to detect the voids. A suggestion to improve is to use adaptive thresholding before edge detection as adaptive thresholding does enhance edges which may be able to improve the edge detection and better results may be obtained.

### Find Contour on CLAHE adaptive gaussian thresholding of block size 5

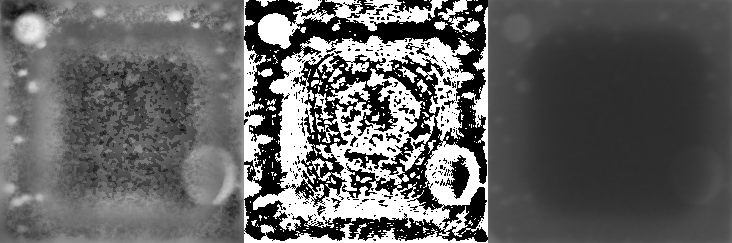
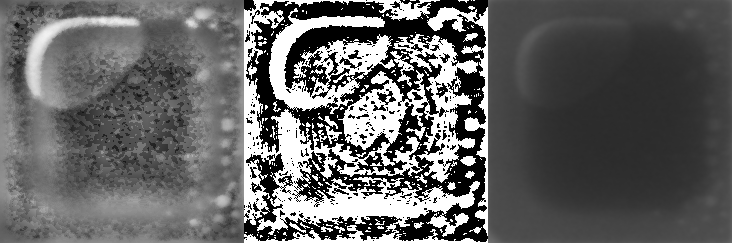
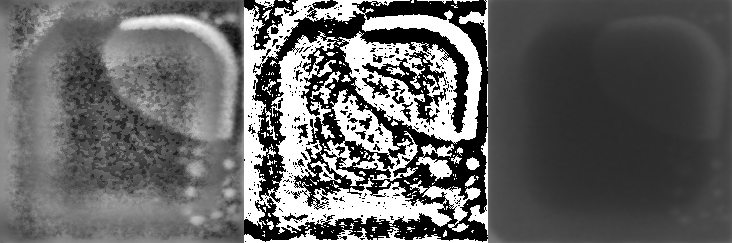
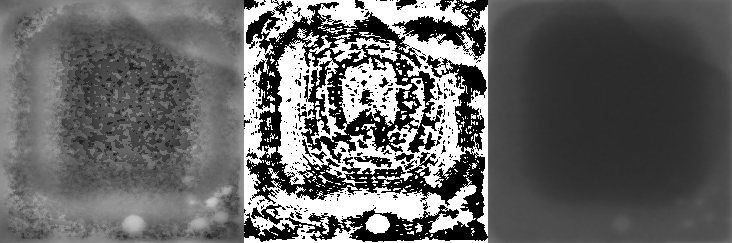


Figure: 1. Left: CLAHE

2. Middle: CLAHE with adaptive thresholding with block size 5

3. Right: Original image

Some voids can be seen segmented properly, while some voids still are only partially segmented. The noise in the middle square is also segmented, which makes detecting the void with findContour. There is a parameter that can be changed for adaptive thresholding which is block size, which determines the block size of the kernel used by the algorithm to apply local thresholding. I have tried larger block size, which minimizes the noise being segmented, but the voids are only segmented partially or even not segmented at all, thus it is sub optimal to find Contour with this image. Thus, we will try the image that has undergone histogram equalization because it is less noisy than CLAHE and does improve contrast between background and the voids compared to the original image.

### Find Contour on Histogram Equalized image with adaptive gaussian thresholding of block size 101

### 

Figure: 1. Left: Histogram Equalized image

2. Middle: Histogram Equalized image with adaptive thresholding with block size 101

3. Right: Original image

There is still some noise being segmented but it is much better than CLAHE + adaptive thresholding. Thus we will use this image to detect voids using find contour, then filter using area and circularity. I combined the contour detected by this method and the ones detected using CLAHE + binary thresholding of 157.

### CLAHE binary thresholding of 157 and Histogram Equalization adaptive thresholding of block Size 101

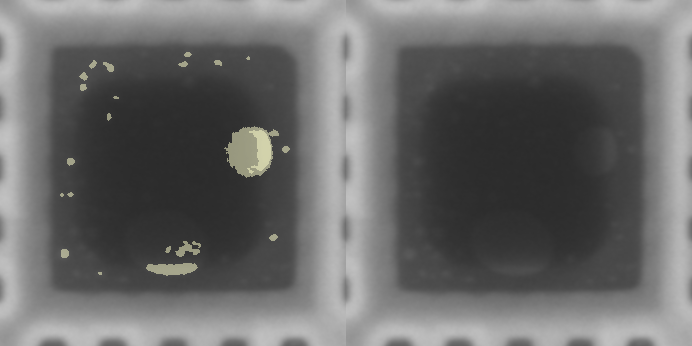
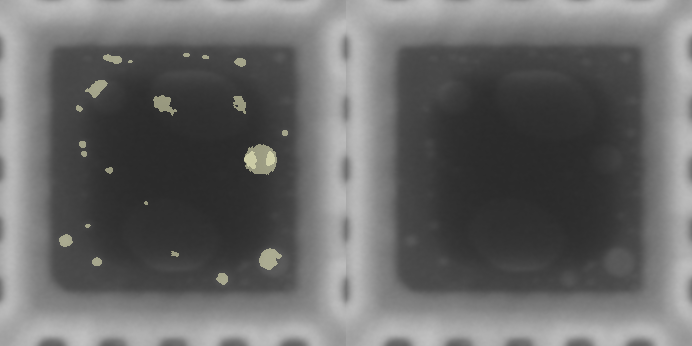
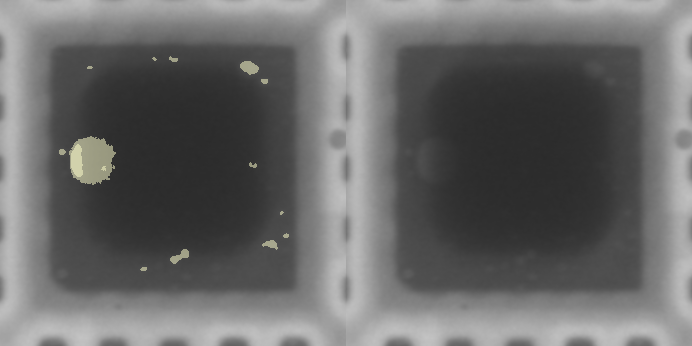
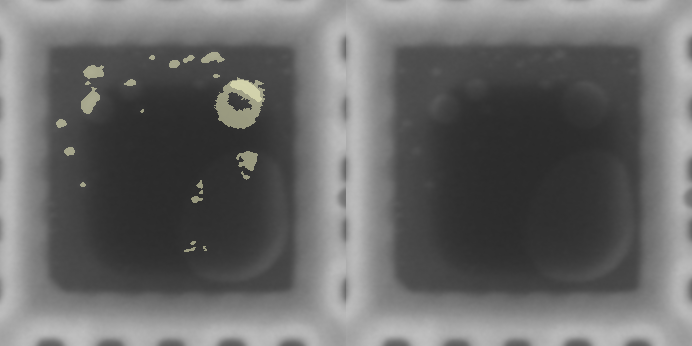
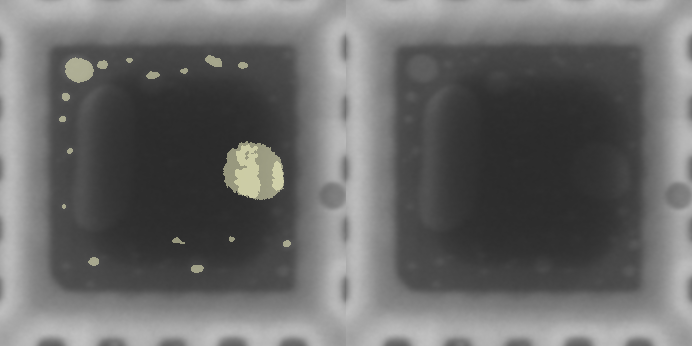


Figure: 1. Left: clahe binary + histogram equalisation adaptive image

2. Right: Original Image

This method does detect more blobs, but it is still not robust enough to detect most of the voids in the image. So far, the best we've got is this method.

# Conclusion

1. We've enhanced the contrast of the image, with the best result being CLAHE since it reveals the most voids, even though it also shows a lot of noise in the process.
2. We used binary thresholding to filter out the noise and try to get as many voids as possible from the CLAHE image, and we found that a binary threshold with threshold value 150 produces the best results. Most of the voids are detected, however there are some voids which overlap the bright and dark area which makes it hard to be segmented through binary thresholding.
3. We tried canny Edge detection and it is able to outline the voids, but it also includes the background, which leads to problems with finding blobs using OpenCV's findContour function.
4. After that we tried adaptive thresholding, because by applying local thresholding to one area at a time does improve the thresholding for blobs in varying background conditions. As we can see after applying adaptive Gaussian thresholding we are able to see most of the voids and a lot of noise. So to reduce noise, we will use the histogram equalised image which has less noise and contrast compared to CLAHE, and the result is much better. Some of the voids are not completely one blob but more of a ring, thus making it harder to filter them solely based on circularity. After filtering with area and circularity, some of the blobs are being detected that cannot be detected by using binary thresholding. Finally I combined binary thresholding and was able to detect around 70% of the voids in the image.