Date: December 2nd, 2018

Computer Vision and Image Processing

Project-3

Instructor: Jungsun Yuan

Name: Varun Nagaraj

UB Person# : 50290761

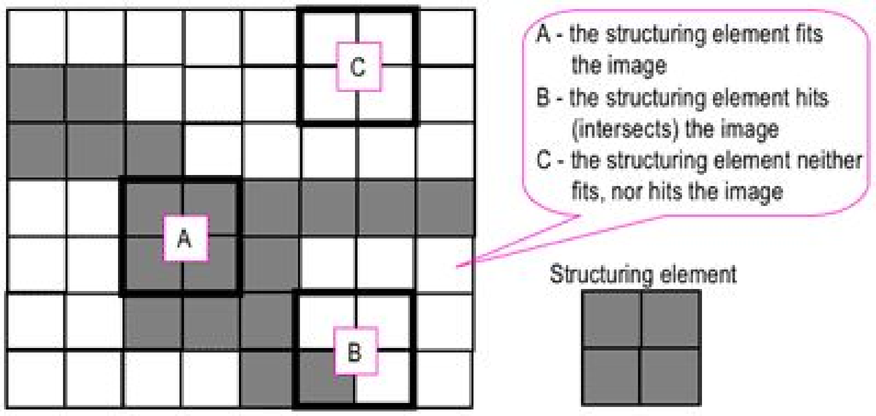
University at Buffalo

Task 1:

Morphological Image Processing

Morphological image processing is a collection of non-linear operations related to the shape or morphology of features in an image. According to Wikipedia, morphological operations rely only on the relative ordering of pixel values, not on their numerical values, and therefore are especially suited to the processing of binary images. Morphological operations can also be applied to grayscale images such that their light transfer functions are unknown and therefore their absolute pixel values are of no or minor interest.

Morphological techniques probe an image with a small shape or template called a structuring element. The structuring element is positioned at all possible locations in the image and it is compared with the corresponding neighbourhood of pixels. Some operations test whether the element "fits" within the neighbourhood, while others test whether it "hits" or intersects the neighbourhood:


A - the structuring element fits the image
B - the structuring element hits (intersects) the image
C - the structuring element neither fits, nor hits the image
Structuring element
A
B
C

The **structuring element** is a small binary image, i.e. a small matrix of pixels, each with a value of zero or one:

- The matrix dimensions specify the size of the structuring element.
- The pattern of ones and zeros specifies the shape of the structuring element.
- An origin of the structuring element is usually one of its pixels, although generally the origin can be outside the structuring element.

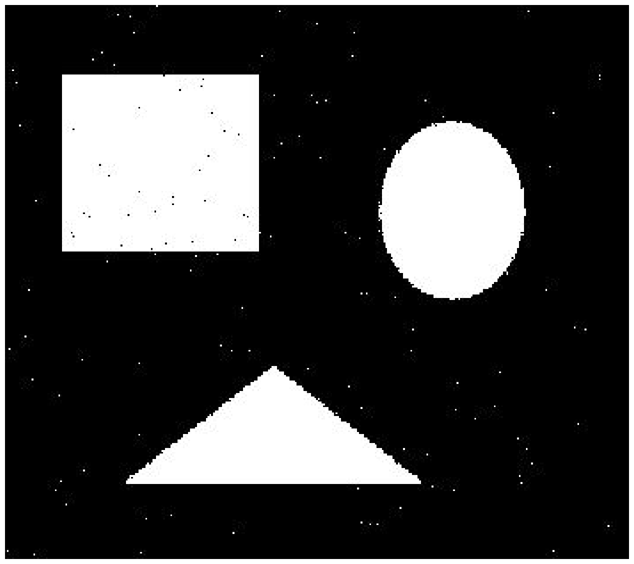
In this task we perform the morphological operations of dilation/erosion in order to remove the noise by the process of filtering and then we find the boundary of the shapes given in the image which is nothing other than finding the edges present in the image.

Erosion:
Erosion of A by B is the set of all points z such that B, translated by z, is contained in A.

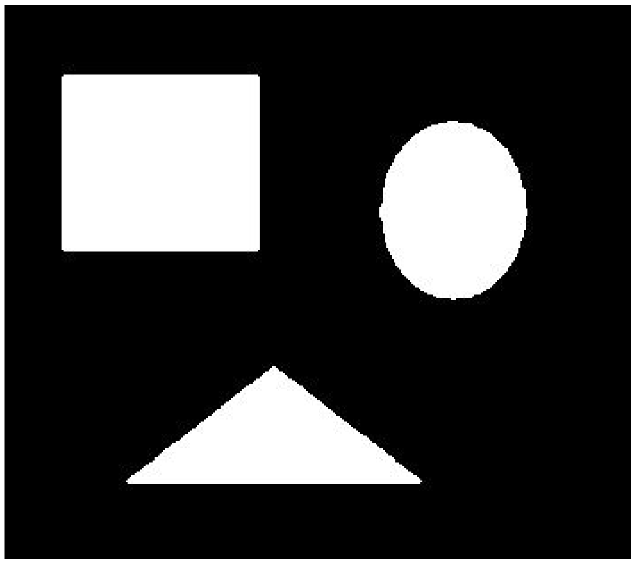
Dilation:
Dilation of A by B then is the set of all z displacements such that and A overlap by at least one nonzero element.

First subpart requires filtering the image by using 2 morphological methods. I make use of the median and mean filtering methods which are given in the class slides.
Below output images show the initial input image and the denoised output from median and mean filtering operations.

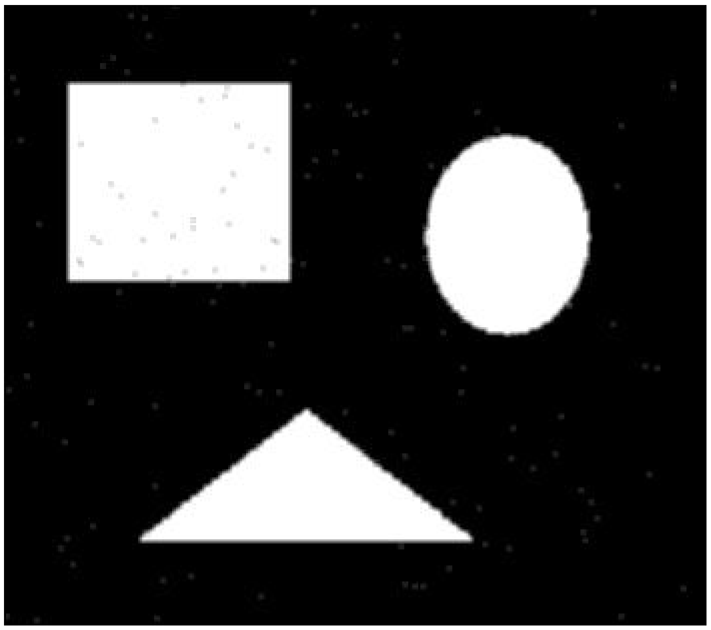
Original:



Median Filtering:



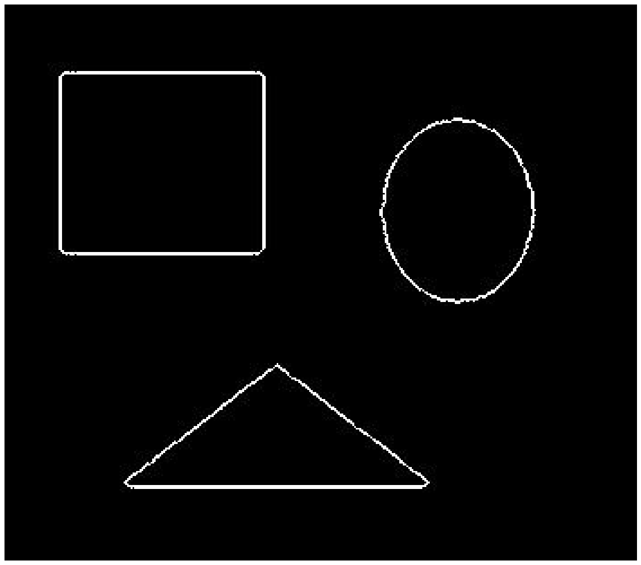
Mean Filtering:



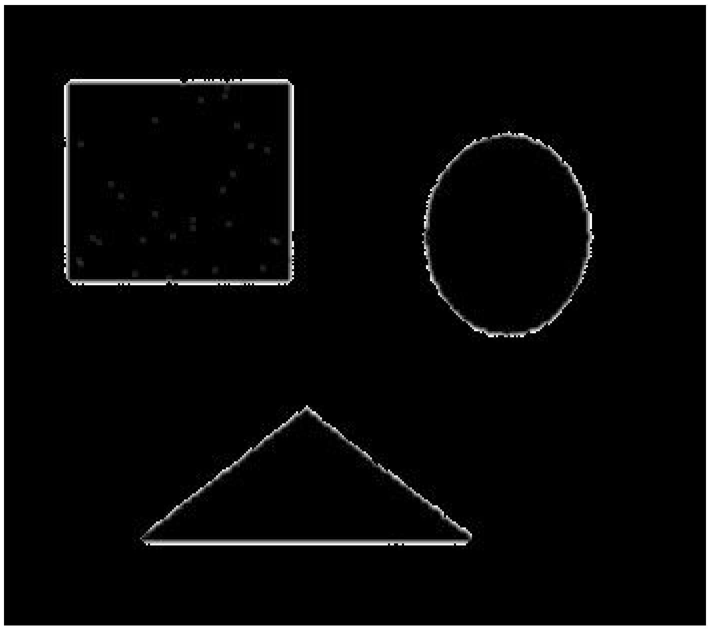
We can see that median filtering does a better job at removing the noise from the image when compared with Mean filtering. The median filter is also widely claimed to be 'edge-preserving' since it theoretically preserves step edges without blurring. However, in the presence of noise it does blur edges in images slightly. Median filtering is one kind of smoothing technique, as is Linear Gaussian filtering. All smoothing techniques are effective at removing noise in smooth patches or smooth regions of a signal, but adversely affect edges. Often though, at the same time as reducing the noise in a signal, it is important to preserve the edges.

Boundary extraction from shapes:

Using median filter with dilation and subtraction:



Using Mean filter with dilation and subtraction:



We can see that the edges are well preserved in case of Median filtering when compared to Mean filtering where we can see breaks in the boundaries of the shapes.

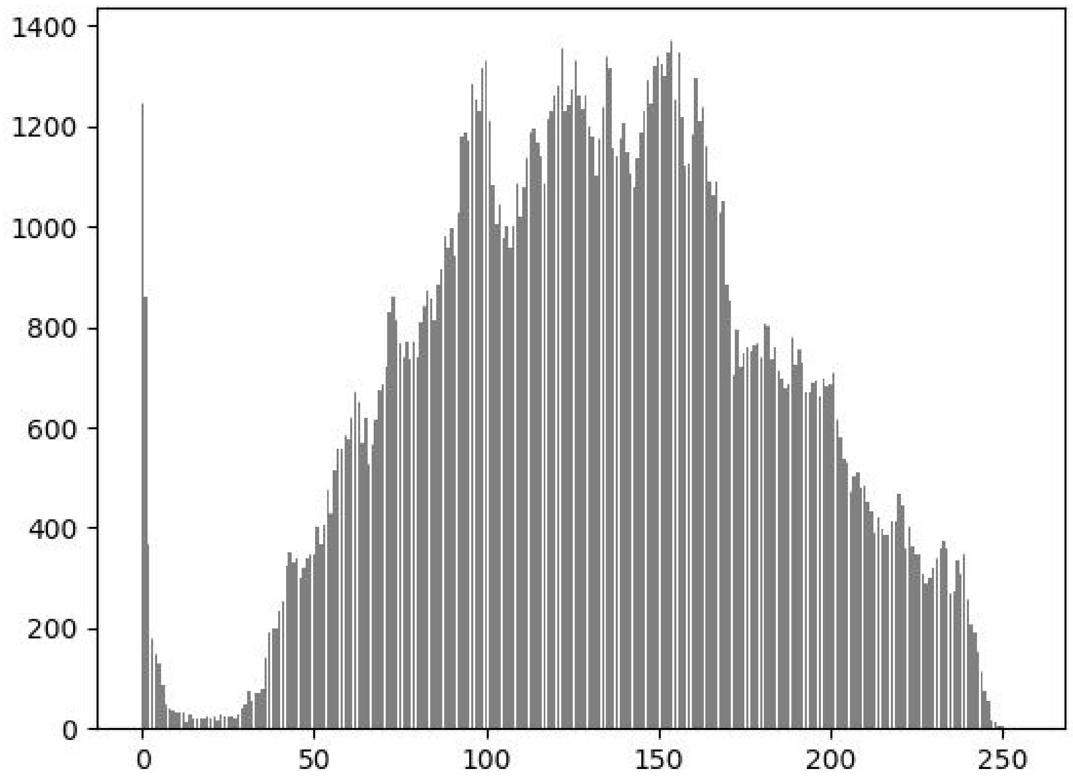
Task 2:

Image segmentation and point detection

In computer vision, image segmentation is the process of partitioning a digital image into multiple segments. The goal of segmentation is to simplify and/or change the representation of an image into something that is more meaningful and easier to analyze. Image segmentation is typically used to locate objects and boundaries (lines, curves, etc.) in images. More precisely, image segmentation is the process of assigning a label to every pixel in an image such that pixels with the same label share certain characteristics.

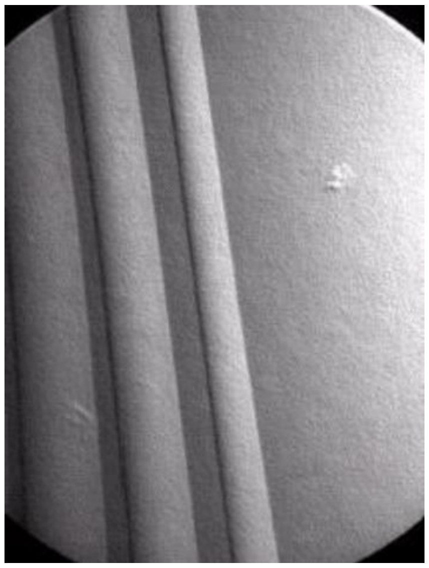
In this task we are required to do two sub tasks: One to identify the porosity of the image using point detection algorithm. And second to perform segmentation on the images containing the bones of a fish.
In the point detection algorithm, I make use of a 5x5 mask which is multiplied with the corresponding position of the image pixel value and the total sum is found for that center pixel.
I manually checked the threshold value by plotting a histogram for all the pixel counts and I found that at a threshold value of 101 gave the correct point detection with some outlier values

on the edges. Below is the histogram which shows the distribution of the values of the image intensities.

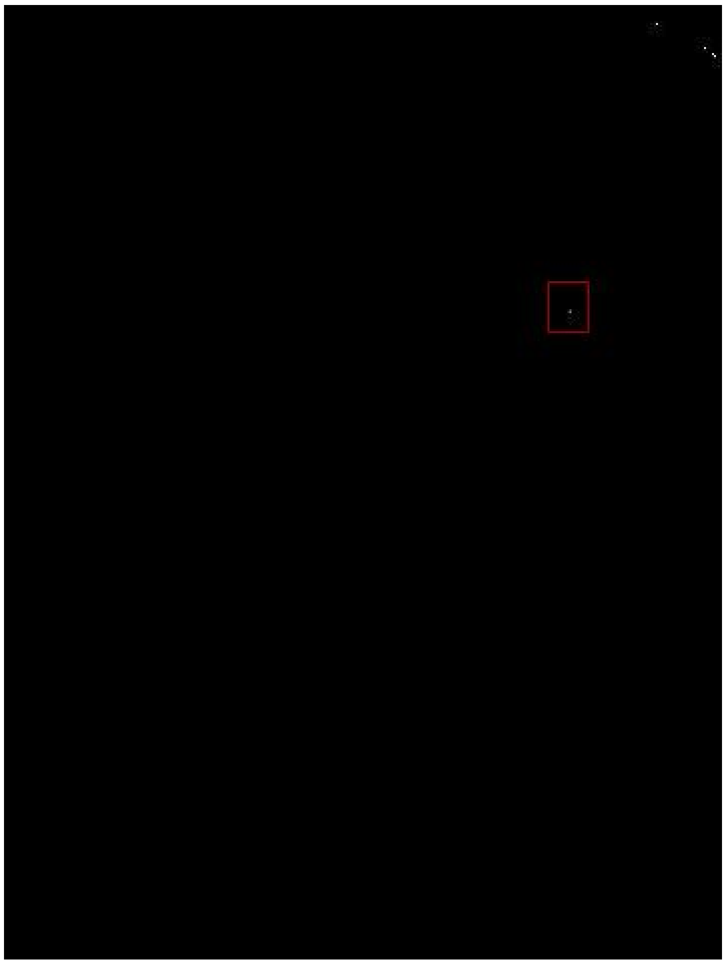


We can see that the threshold value between 100-150 would be the correct value. By trial and error I found the correct threshold value of 101.

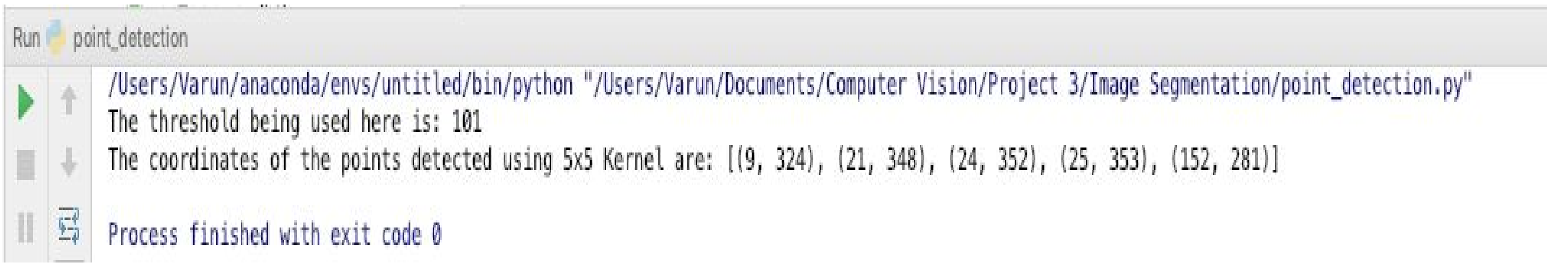
Original Image:



Point Detected Output:



The point present in the box is the point which is supposed to be detected and it has been detected along with 4 other points which share the same pixel intensity.
The output of the code when I ran the code is as shown below.



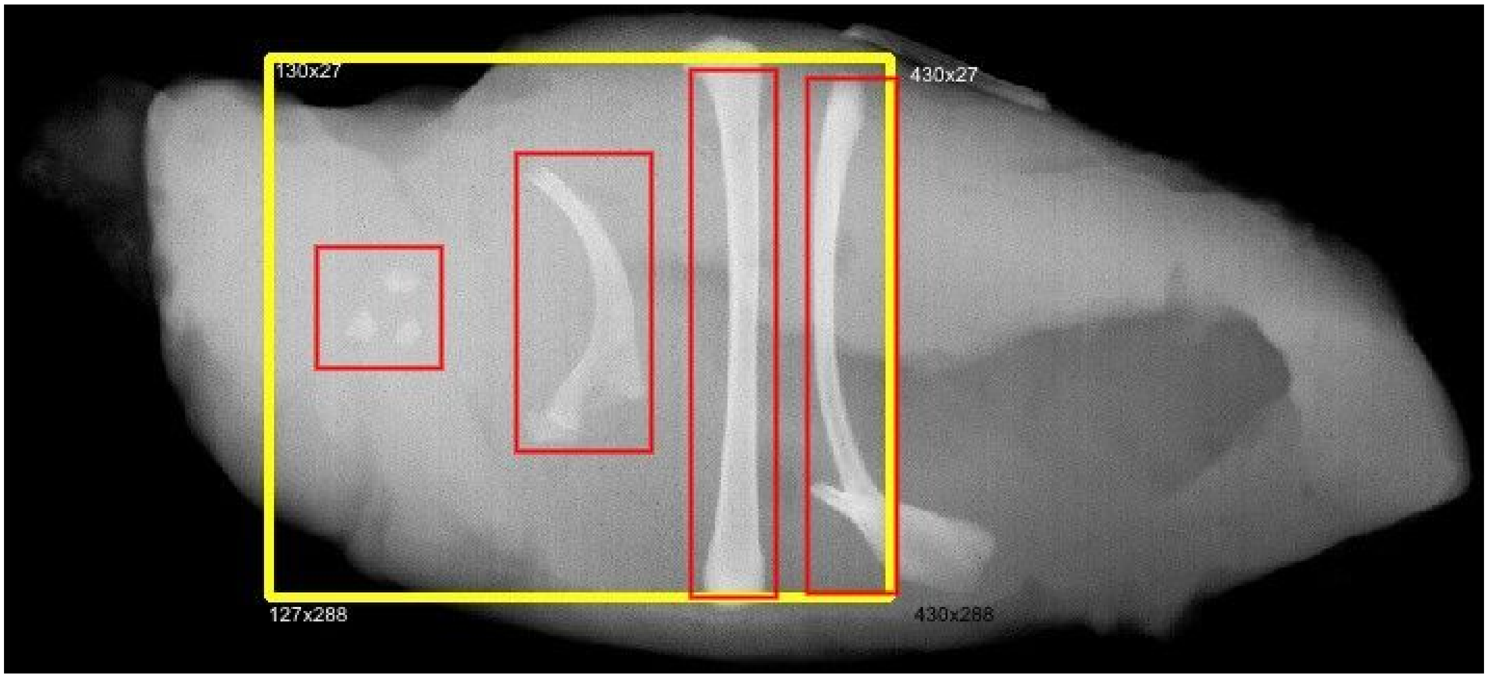
```
/Users/Varun/anaconda/envs/untitled/bin/python "/Users/Varun/Documents/Computer Vision/Project 3/Image Segmentation/point_detection.py"
The threshold being used here is: 101
The coordinates of the points detected using 5x5 Kernel are: [(9, 324), (21, 348), (24, 352), (25, 353), (152, 281)]

Process finished with exit code 0
```

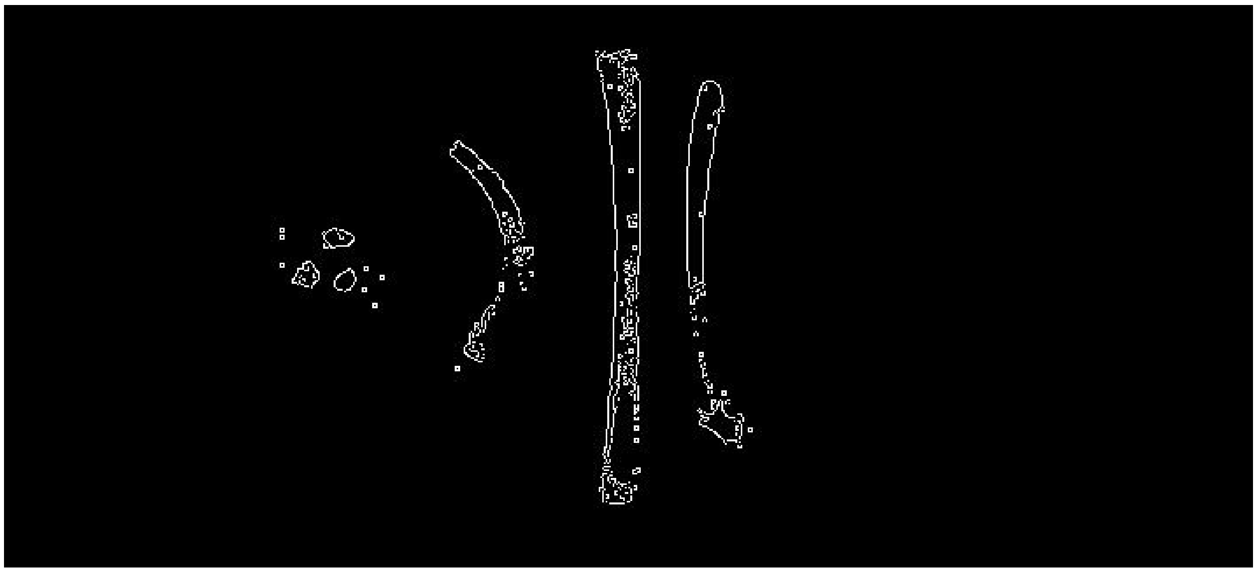
The position of the point is (152,281).

The second sub task that has to be done is the separation of foreground from background. We choose an optimal thresholding algorithm and segment the foreground from the background.
This can be done by checking each pixel value if the value is greater than the threshold. If so then we update the new image array by 255, otherwise we put 0 in the new image array. After

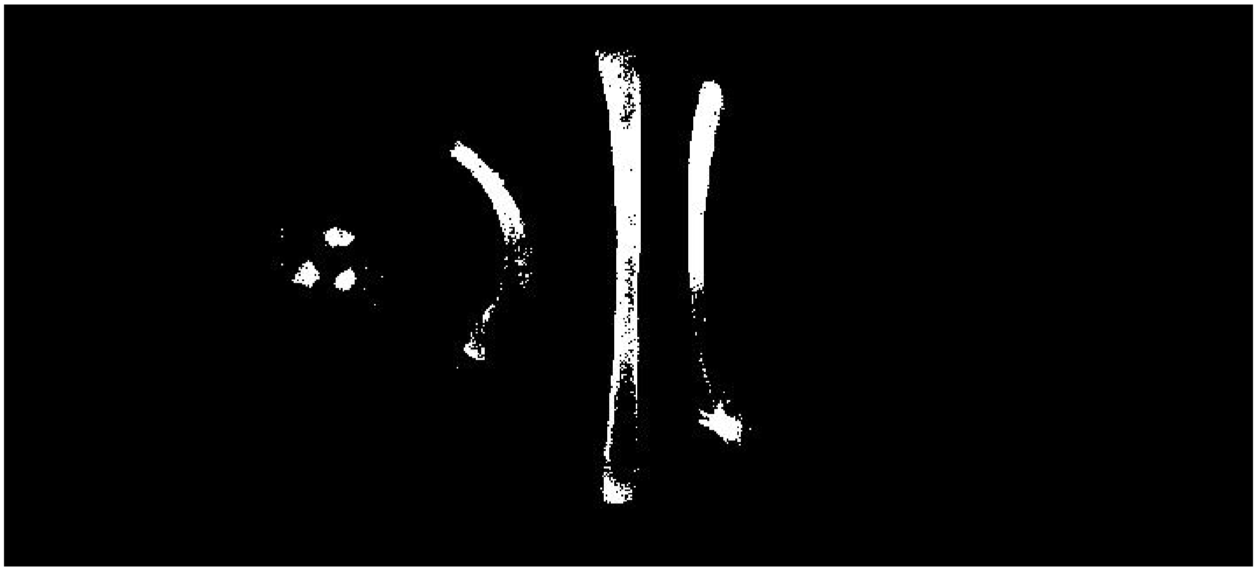
this is done I perform an edge detection operation using Canny operator. Using this we find the extreme points for the box drawing ,i.e separation of foreground from background.



In the image shown above we can see a yellow box which is got by separating the foreground and the background. The extreme points are depicted as shown.
(130x27) - top left, (430x27) - top right, (127x288) - bottom left, (430x288) - bottom right.



The above image shows the edge detection output for the bones image given after thresholding has been applied.



The above image shows the segmented image which clearly separates the bones from the background.

In the bounding boxes image we can see the red boxes which are individually segmented images of the bones.

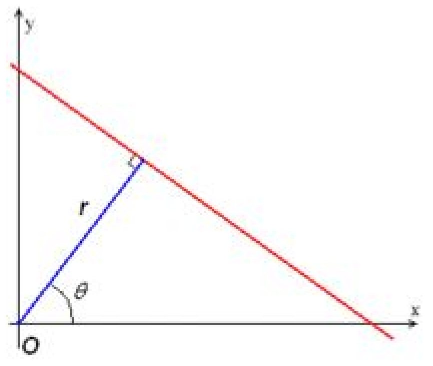
Task 3:

Hough Transform

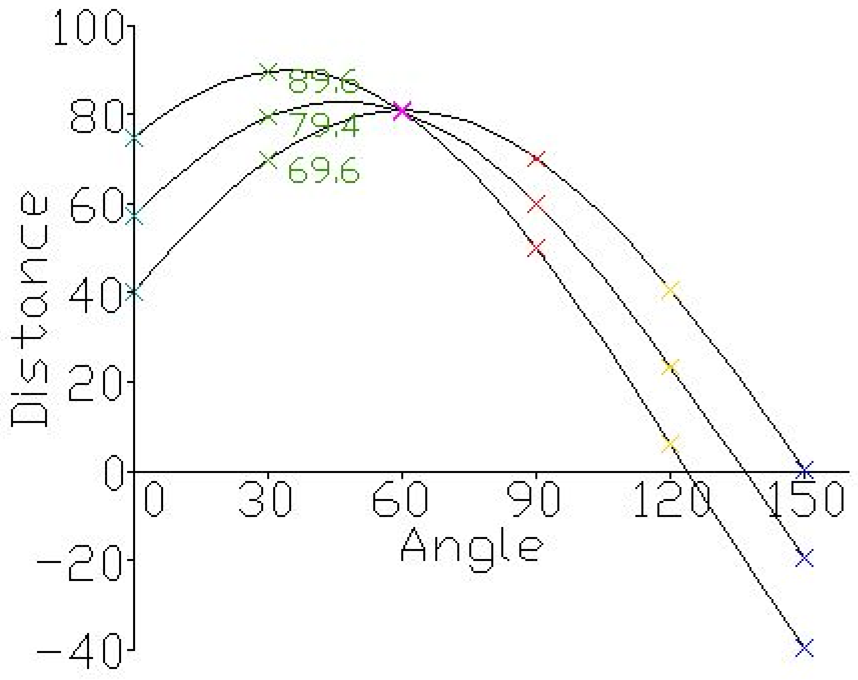
The Hough transform is a feature extraction technique used in image analysis, computer vision, and digital image processing. The purpose of the technique is to find imperfect instances of objects within a certain class of shapes by a voting procedure. This voting procedure is carried out in a parameter space, from which object candidates are obtained as local maxima in a so-called accumulator space that is explicitly constructed by the algorithm for computing the Hough transform.
The simplest case of Hough transform is detecting straight lines. In general, the straight line y = mx + b can be represented as a point (b, m) in the parameter space. However, vertical lines pose a problem. They would give rise to unbounded values of the slope parameter m.

$$r = x\cos\theta + y\sin\theta$$

is therefore used to avoid the vertical lines issue which would result in infinite slope if using a straight line equation to solve for hough lines.

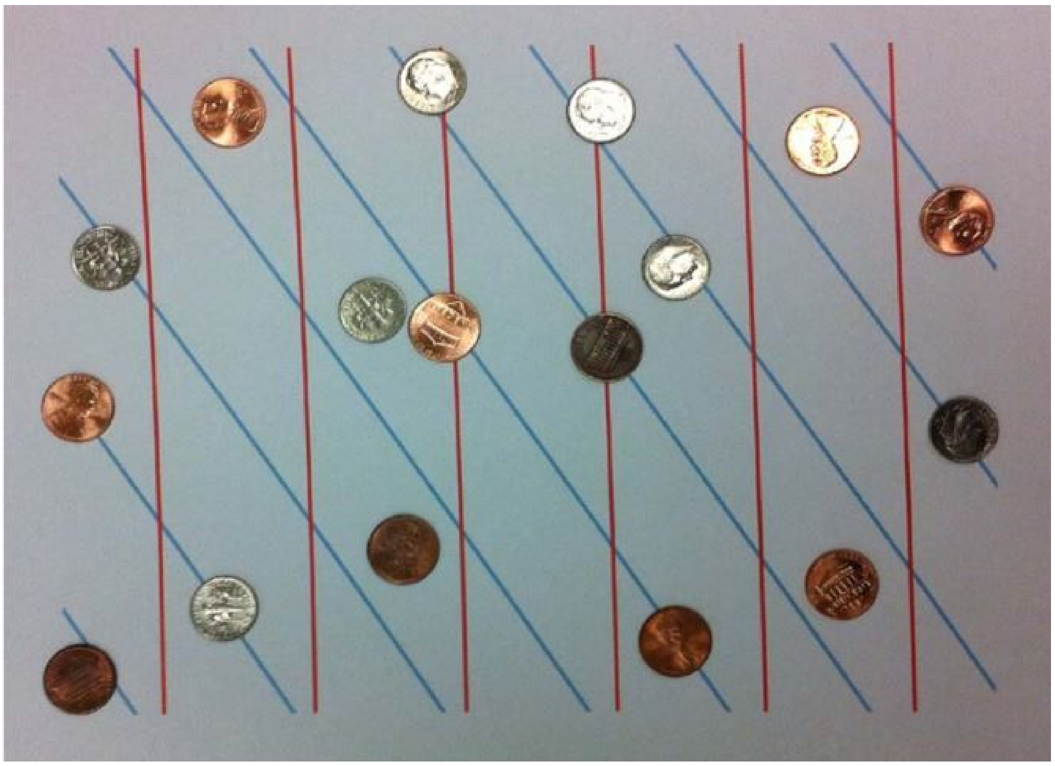


r defines the distance from the origin and theta defines the angle at which the line is tilted. Using these two we can find the line which is needed.

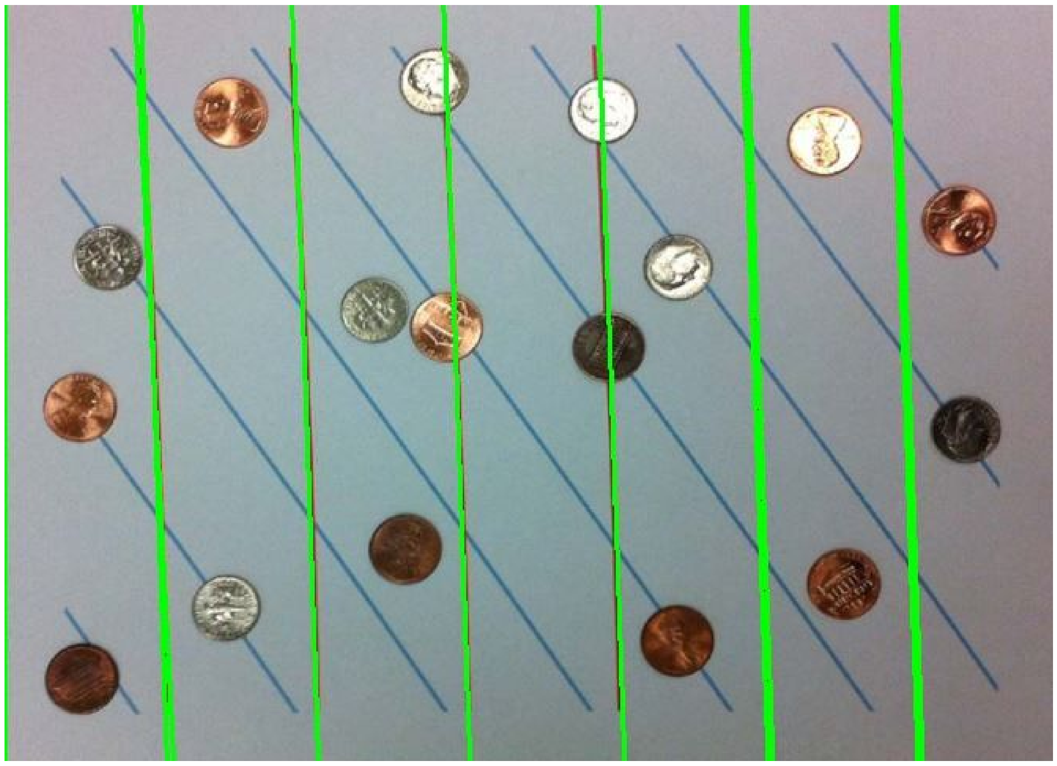


The above image shows the hough space where we get sinusoidal curves and the intersection point will be the major point through which a line passes.

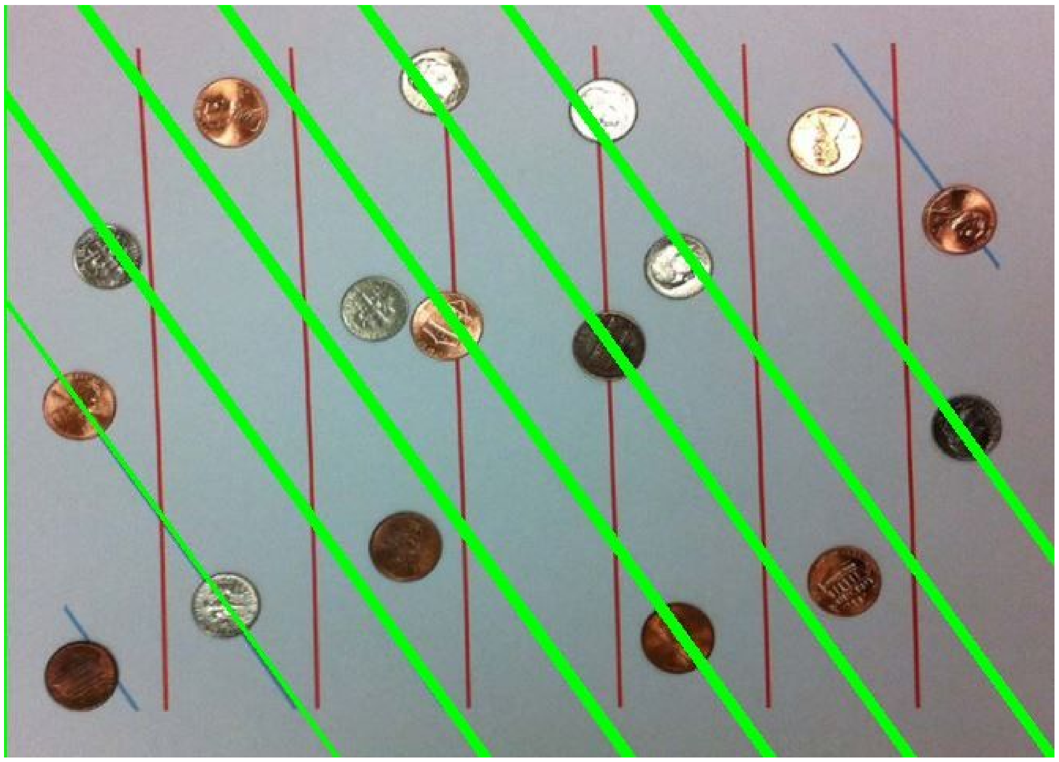
Original Image:



Red Lines:



Blue Lines:



Bonus Task

Hough Circles

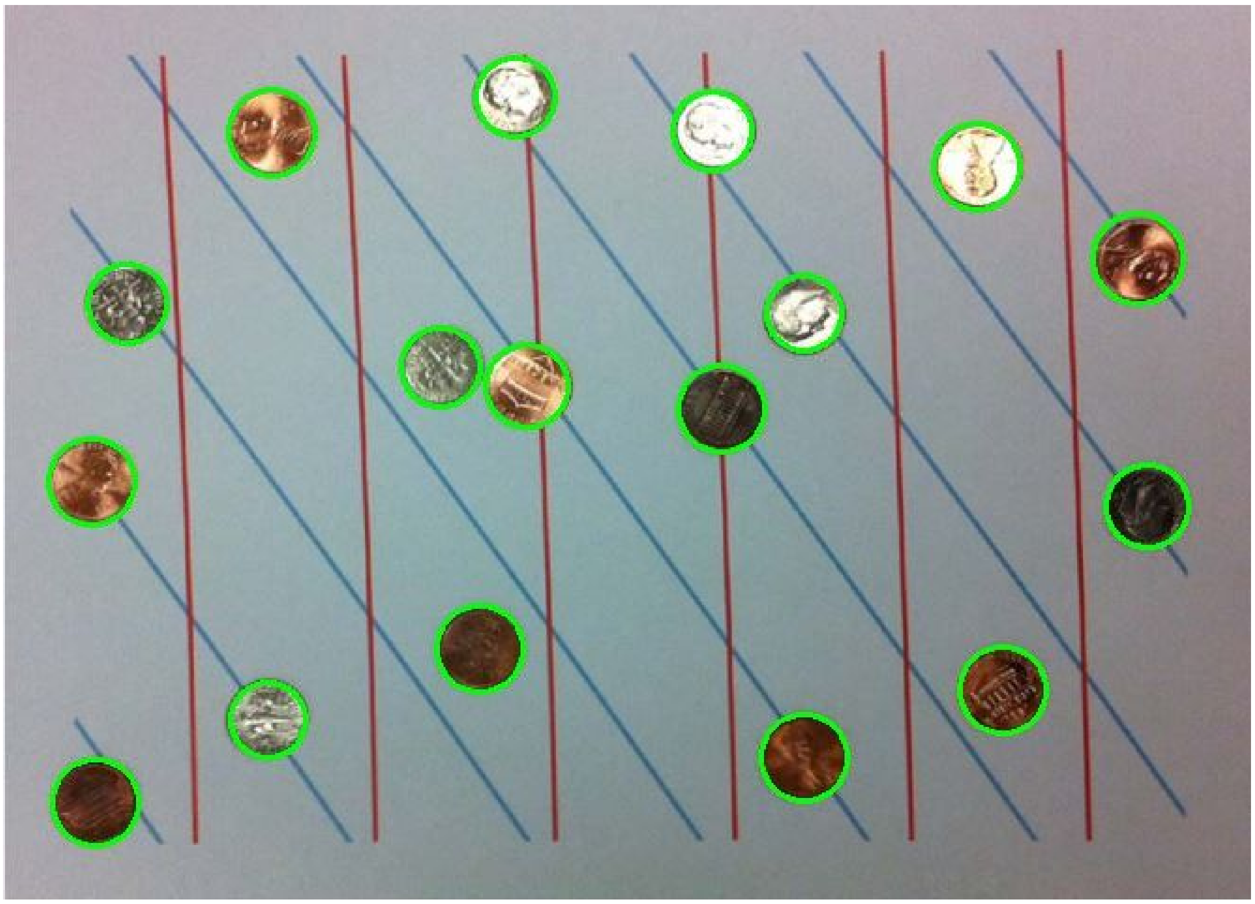
The Circle Hough Transform (CHT) is a feature extraction technique for detecting circles. It is a specialization of Hough Transform. The purpose of the technique is to find circles in imperfect image inputs. The circle candidates are produced by “voting” in the Hough parameter space and then select the local maxima in a so-called accumulator matrix.

$$(x-a)^2 + (y-b)^2 = r^2 \qquad (1)$$

This is the equation of the circle which is made use of to determine the location of the circles in the image. If a 2D point (x,y) is fixed, then the parameters can be found according to (1). The parameter space would be three dimensional, (a, b, r). And all the parameters that satisfy (x, y) would lie on the surface of an inverted right-angled cone whose apex is at (x, y, 0).
An accumulator matrix is used for tracking the intersection point. In the parameter space, the voting number of points through which the circle passing would be increased by one. Then the local

maxima point (the red point in the center in the right figure) can be found. The position (a, b) of the maxima would be the center of the original circle.

Output:



References:

1. Hough Transform:
   https://rosettacode.org/wiki/Hough_transform#Python
2. https://nabinsharma.wordpress.com/2012/12/26/linear-hough-transform-using-python/
3. Thresholding:
   https://www.meccanismocomplesso.org/en/opencv-python-otsu-binarization-thresholding/
4. https://en.wikipedia.org/wiki/Circle_Hough_Transform