Cairo University

Faculty of Engineering

Systems and Biomedical Engineering

Computer Vision

Task 2 Report

Submitted to:

Dr. Ahmed M. Badawi

TA/ Eng Laila Abbas.

TA/ Eng. Peter Salah.

Prepared by:

Team 8.

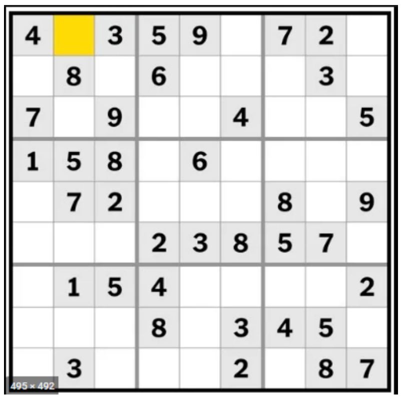
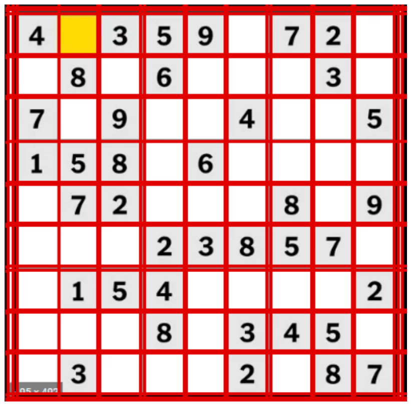
Name	Section	Bench Number
Arwa Essam	1	12
Sohaila Mahmoud	1	45
Shrouk Shawky	1	46
Maryam Megahed	2	34

Functions Implemented

1. <u>Hough Line Detection</u>

   The line equation is y = ax + b, but we first work with Cartesian coordinates so we determine the range of ρ and θ. the range of θ is [0, 180] degrees and ρ is [-d, d], where d is the diagonal length of the edge.

   - First, we create a 2D array called the accumulator with the dimensions (r, θ) and set all its values to zero.
   - Use the original Image to do edge detection by using canny function.
   - Check each pixel on the edge picture to see if it is an edge pixel. If the pixel is on edge, loop over all possible values of θ, compute the corresponding ρ, and locate the θ and ρ index in the accumulator, then increase the accumulator base on those two index.
   - Loop over the accumulator's values. and get the value of ρ and θ from the index pair, which may be transformed back to the form of y = ax + b if the value is greater than a specified threshold ,then put theses point to array called lines then through theses points we draw line over the original image

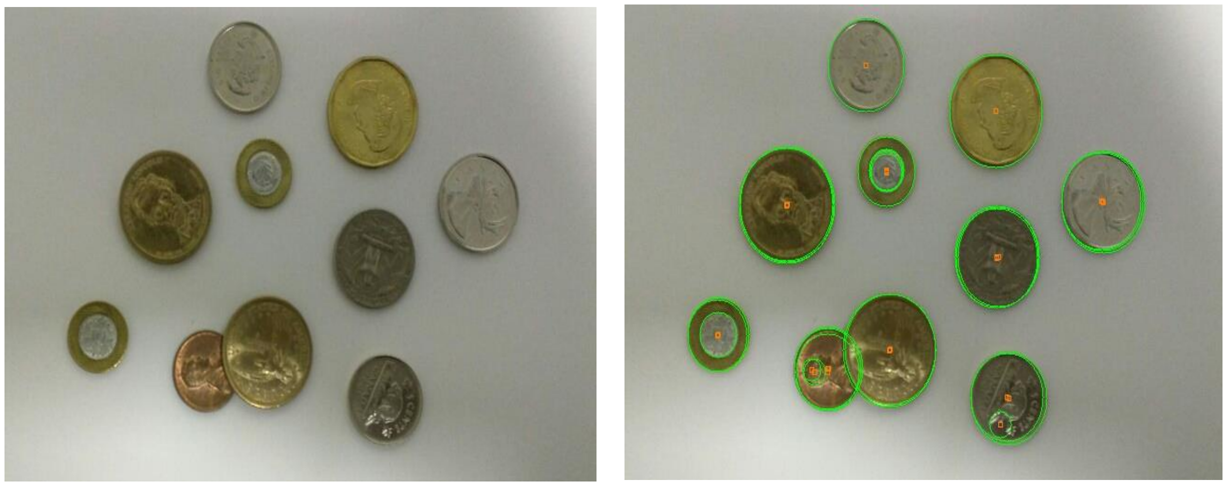



2. <u>Hough Circle Detection</u>

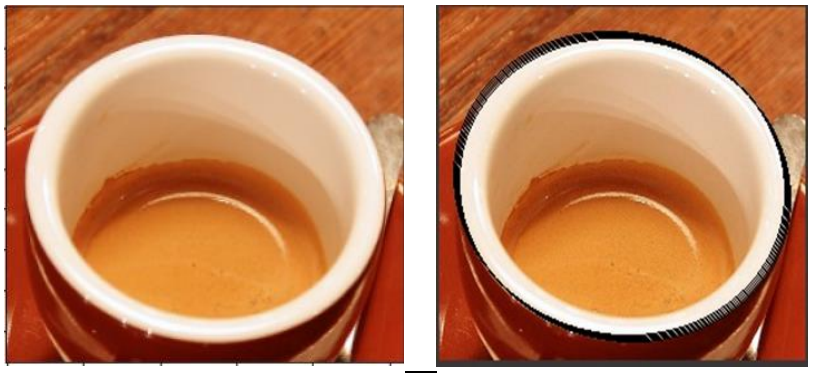
   Circle can be represented as (x-a)2 + (y-b)2 = r2 where a, b represents the circle center and r is the radius. So, we require 3 parameters (a,b,r) to completely describe the circle. So, Here is the pseudo code for the implemented function:

   - Find the edge image using any edge detector
   - For r= 0 to diagonal image length
   - For each edge pixel (x,y) in the image

- For Θ = 0 to 360
- a = x – r*cosΘ
- b = y – r*sinΘ
- incrementing the accumulator
- Find the [a,b,r] value(s), where the accumelator_cell_max is above a suitable threshold value.



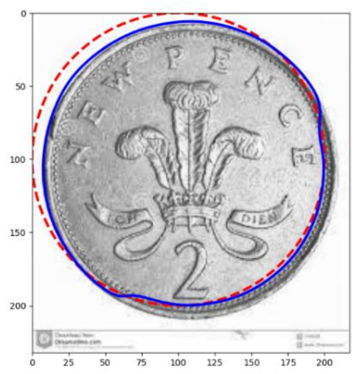
3. <u>Hough Ellipse detection</u>
    - First, create an edge image using canny function.
    - We calculate Euclid distance map which generated from the edged image.
    - Center of candidates are detected by voting mid-points of every two edge points and Slope candidates are detected by voting perpendicular bisectors of every two edge points.
    - Axes candidates are detected by the result of voting the length of axes for every edge point.
    - We create a list of candidate ellipses and sort it in descending order.
    - Determine the detected ellipses by thresholding the sorted list of chosen ellipses based on normalized vote counts.



4. <u>Active Contour</u>

This function takes prior-initialized points for the snake and returns the final snake points after minimizing its total energy using Greedy algorithm (an algorithm that builds up a solution piece by piece, constantly selecting the component that provides the most evident and immediate benefit as the next step), which is summation of internal and external energy, based on the parameters given by the user, which are:

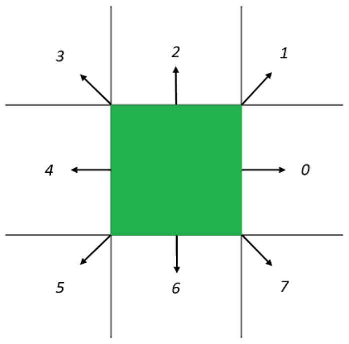
1. External Energy: depends on input image features
    - Gamma: determines the weight of energy that forces the snake toward or away from edges using image gradient magnitude and direction of the input image, where higher gamma values pulls snake points towards contour more.
    - W-line: determines the weight of attraction force to brightness of image where, negative values attracts snake toward dark regions, while higher values attracts it toward brighter regions.
    - W-edge: determines the weight of attraction to edges where, negative values repel snake from edges.
2. For Internal energy: depends on number of snake points
    - Alpha: determines the elasticity of the snake, where higher values of alpha makes snake contract in a faster rate, as alpha is the weight of how much is the effect the first-order derivative (using finite central difference between snake points).
    - Beta: determines the stiffness of the snake, where higher values of alpha makes snake curvatures more smooth, as beta is the weight of how much is the effect of the second-order derivative.

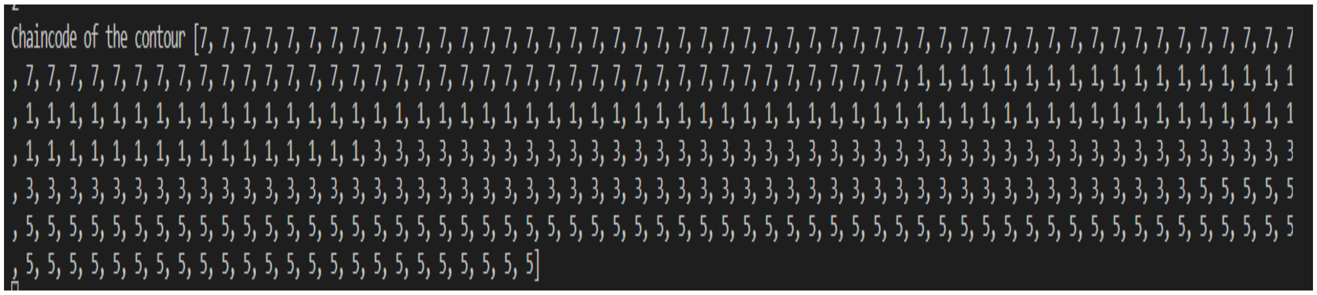


5. Chain code representation

Chain code is a method of describing the shape of the contour of an object in an image.

The chain code function takes an array of snake contours. and checks for many cases of two neighbor points. For example, if the y coordinate of the next point is more than the first one and the x coordinate of the two points are equal, the chain number equal to 2. The function returns a list of the contour of the image.





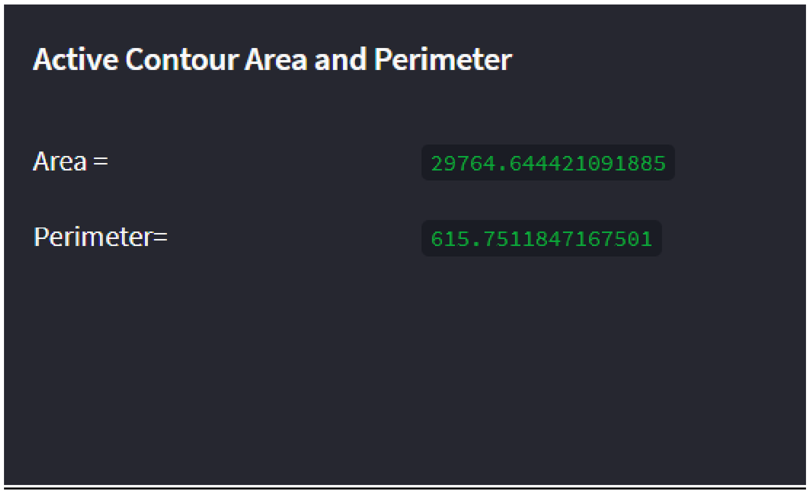
```
Chaincode of the contour [7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7
, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1
, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1
, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3
, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 5, 5, 5, 5, 5
, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5
, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5]
```

6. perimeter function

The function takes an array of snake contour and calculates the distance between every two points and summate these distances to get the total perimeter of the contour.

7. Area function

The function takes an array of snake contour and calculates the dx and dy for all points in the array and calculates the area as the summation of the 0.5*(y0*dx - x0*dy).



Active Contour Area and Perimeter

Area = 29764.644421091885

Perimeter= 615.7511847167501