

## Assignment 2

### Automotive Sensor Systems

Name:- Shubham Sunil Chavan

Student Number:- 110091852

#### Q1: Line Detection

Driver Code to call the function:

```
line = line_detection("test1.jpg")
```

Function line\_detection:

```
function line = line_detection(I)
    %Reading the image from directory
    input_image = imread(I);
    %grayscale conversion of image
    Img_grayscaled = rgb2gray(input_image);
    % Eddge detection of Img_grayscaled
    identified_edges = edge(Img_grayscaled, "canny");

    imshow(identified_edges);
    title('Edge Detected image');

    % STEP: PARAMETERIZATION

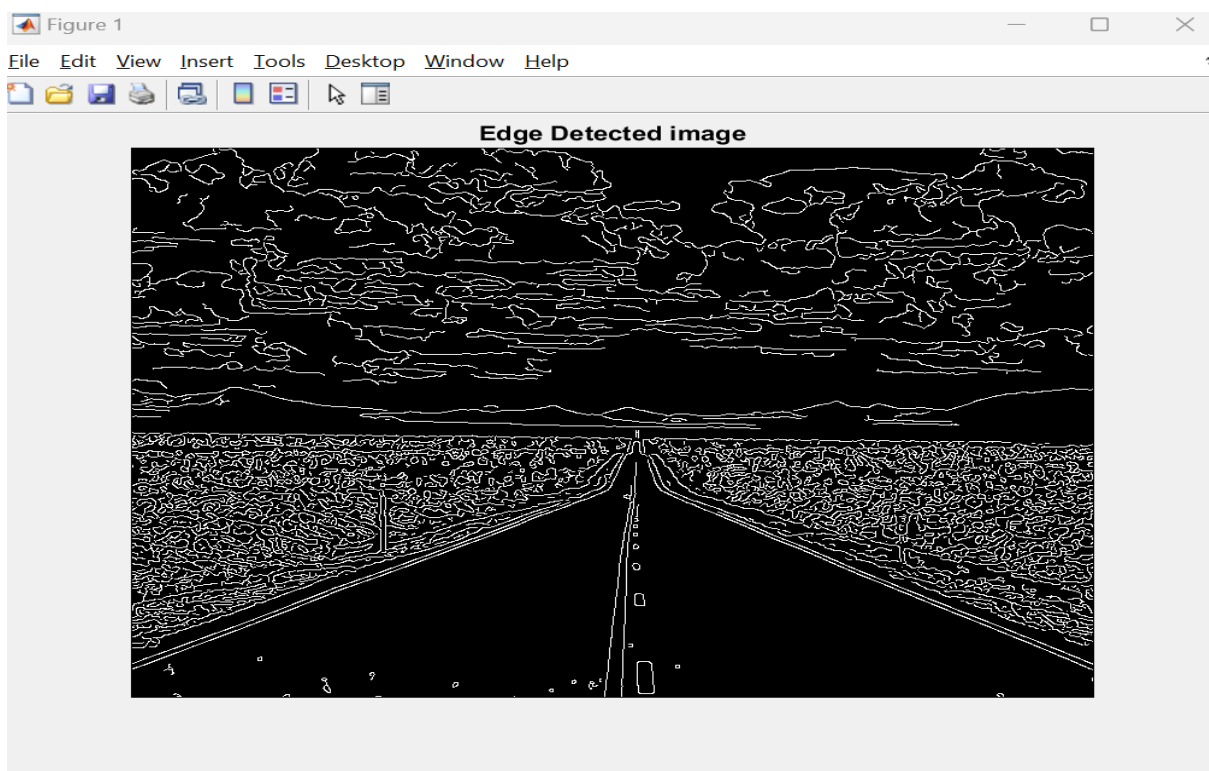
    %Quantization process
    th_freq = 0.01;
    %image dimension
    [x,y] = size(identified_edges);
    %Diagonal detection of image
    diagonal_det = norm([x,y]);
    diag = (-diagonal_det:1:diagonal_det);
    %Gradiant of Radius
    theta = (0:th_freq:pi);
    theta_iter = numel(theta);
    iter_diags = numel(diag);
    %Accumulator i.e Hough Space
    hs_a = zeros(iter_diags, theta_iter);
    % Start HS transform
    for i = 1:x
        for j = 1:y
            if identified_edges(i,j) == 1
                for theta_num = 1:theta_iter
                    % Gradiant angle for each theta
                    th = theta(theta_num);
                    % dia representation
                    r = i*cos(th) + j*sin(th);
                    diag_id = round(r + iter_diags/2);
                    hs_a(diag_id, theta_num) = hs_a(diag_id, theta_num) + 1;
                end
            end
        end
    end
    % Display HT
```

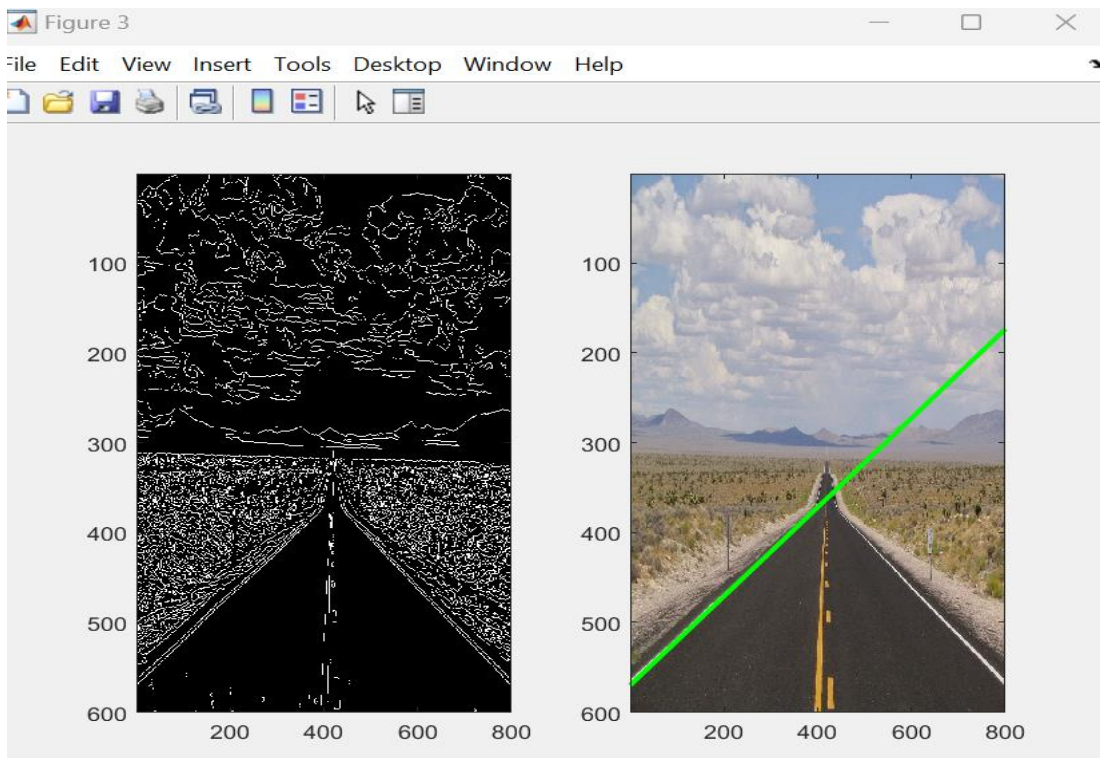
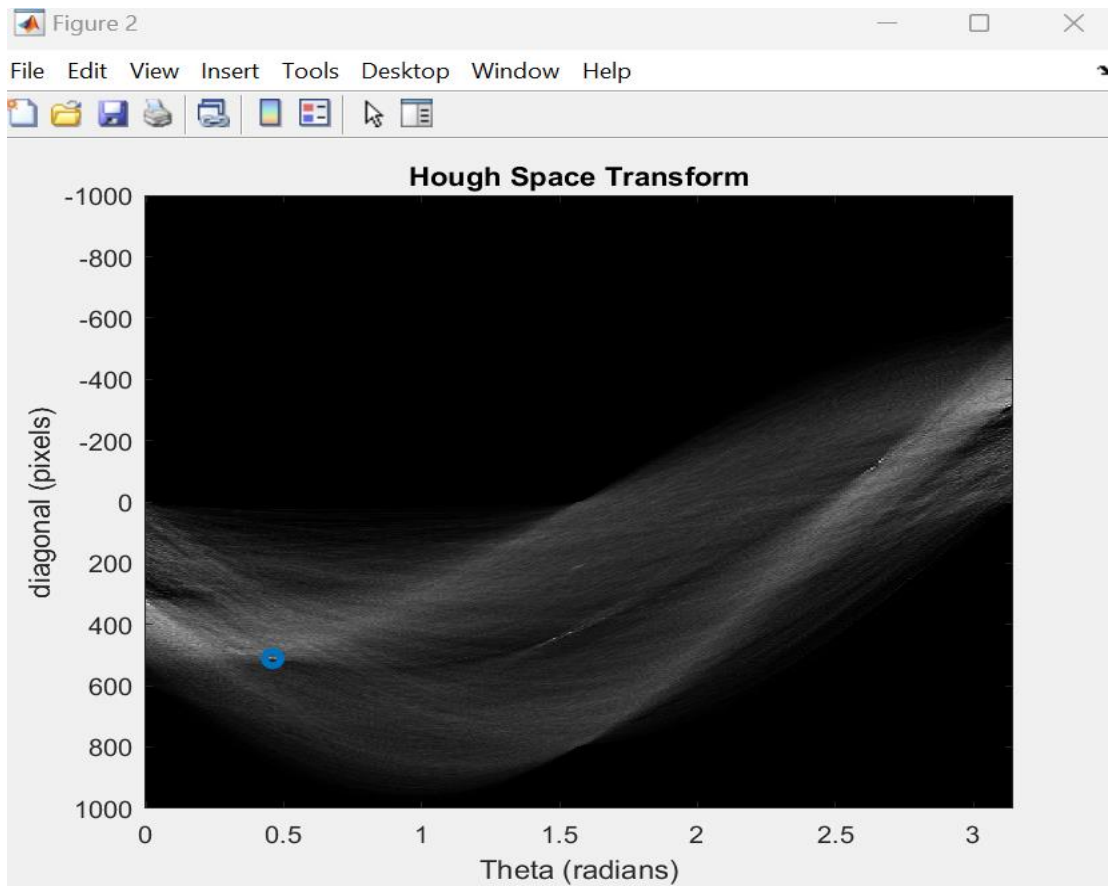
```

figure;
imagesc(theta,diag,hs_a);
title('Hough Space Transform');
xlabel('Theta (radians)');
ylabel('diagonal (pixels)');
colormap('gray'); hold on;
% Extracting all the params
[M, I] = max(hs_a(:));
%Coordinates in HT Space
[diag_id, theta_num] = ind2sub(size(hs_a), I);
% Plotting
plot(theta(theta_num),diag(diag_id),'o','LineWidth',3);
hold off;
% calculating the line coordinates
m = -(cos(theta(theta_num))/sin(theta(theta_num)));
b = (diag(diag_id)) / sin(theta(theta_num));
x = 1:x;
y = m*x+b;
% Superimposing detected line on image
figure;
subplot(1,2,1)
imagesc(identified_edges);
colormap(gray);
hold on;
subplot(1,2,2)
imagesc(input_image);
colormap(gray);
hold on;
line= plot(y,x,'g', 'LineWidth',2)
end

```

output:





## Q2: Circle Detection

Driver Code to call the function:

```
R = 32
C = circle_detection("test2.jpg",R)
```

Function circle\_detection:

```
function C = circle_detection(I,R)
    r=R;
    %Reading the image
    input_image= imread(I);
    %grayscale conversion of image
    Img_grayscaled=rgb2gray(input_image);
    % Eddge detection of Img_grayscaled
    identified_edges = edge(Img_grayscaled,'canny');
    %Display the edge image
    imshow(identified_edges)
    title('Edge image');
    % STEP - PARAMETERIZATION

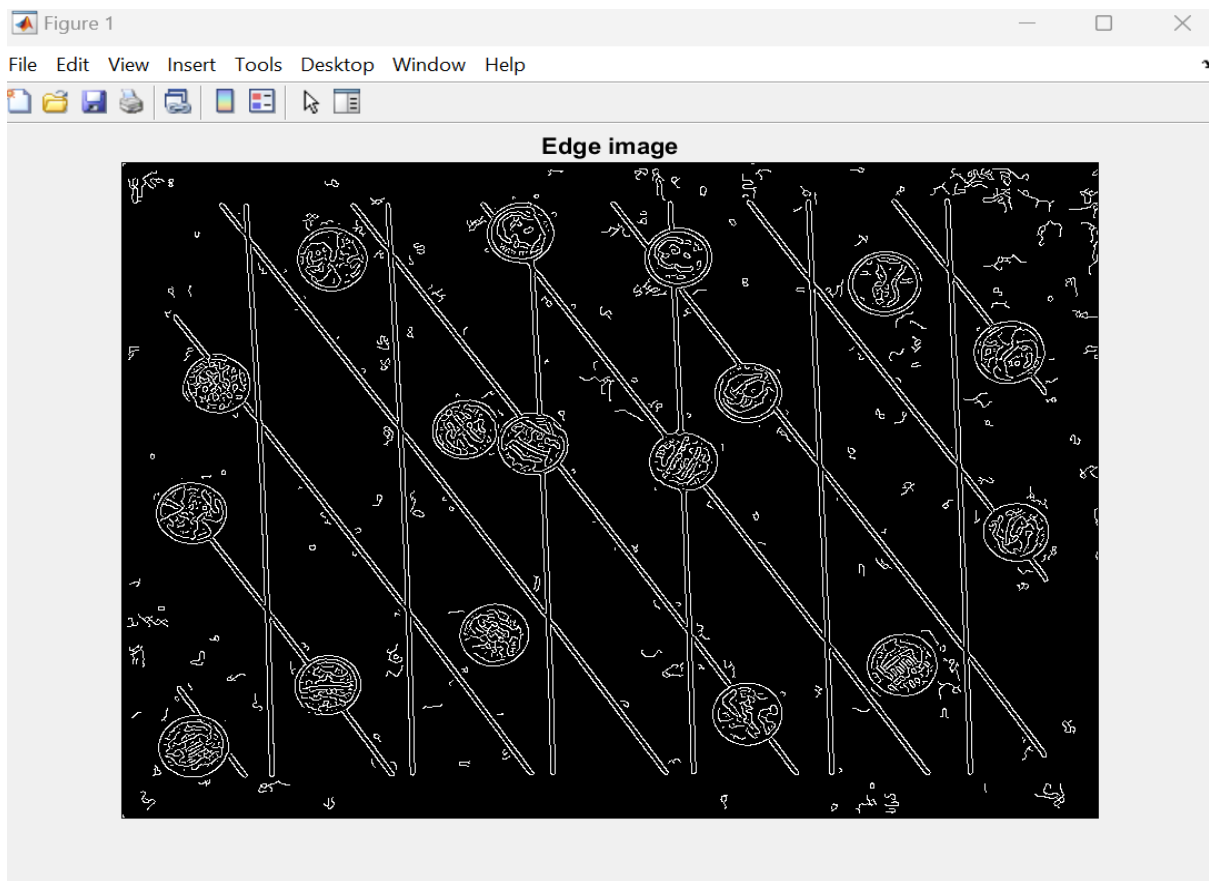
    %Quantization process
    th_freq= 0.01;
    %Gradient calculation in Radius
    theta=(0:th_freq:2*pi);
    theta_iter=numel (theta);
    %image dimension
    [x,y] = size (identified_edges);
    size (identified_edges)
    X1 = (1:x);
    Y1 = (1:y);
    %Accumulator i.e Hough Space
    hs_a = zeros(x,y);
    % Start HS transformation
    for i=1:x
        for j=1:y
            if identified_edges(i,j)==1
                for theta_id= 1:theta_iter
                    % Gradient angle for each theta_id
                    th=theta(theta_id);
                    % the a and b representation
                    a=round(i-r*cos(th));
                    b=round(j-r*sin(th));
                    if(a>0&&a<=x&&b>0&&b<=y)
                        hs_a(a,b) = hs_a(a,b)+1;
                    end
                end
            end
        end
    end
    % Display HT
    figure;
    imagesc(hs_a);
    title('Hough Transform');
    xlabel('a');
    ylabel('b');
    colormap('gray');hold on;
```

```

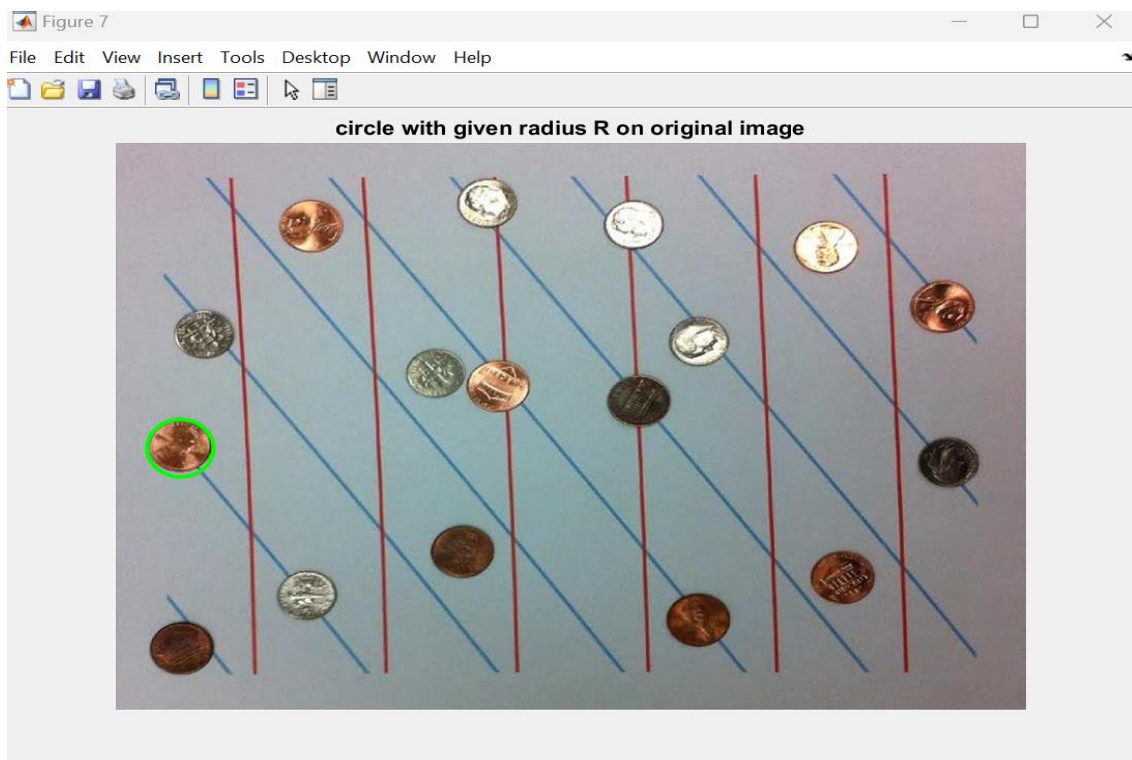
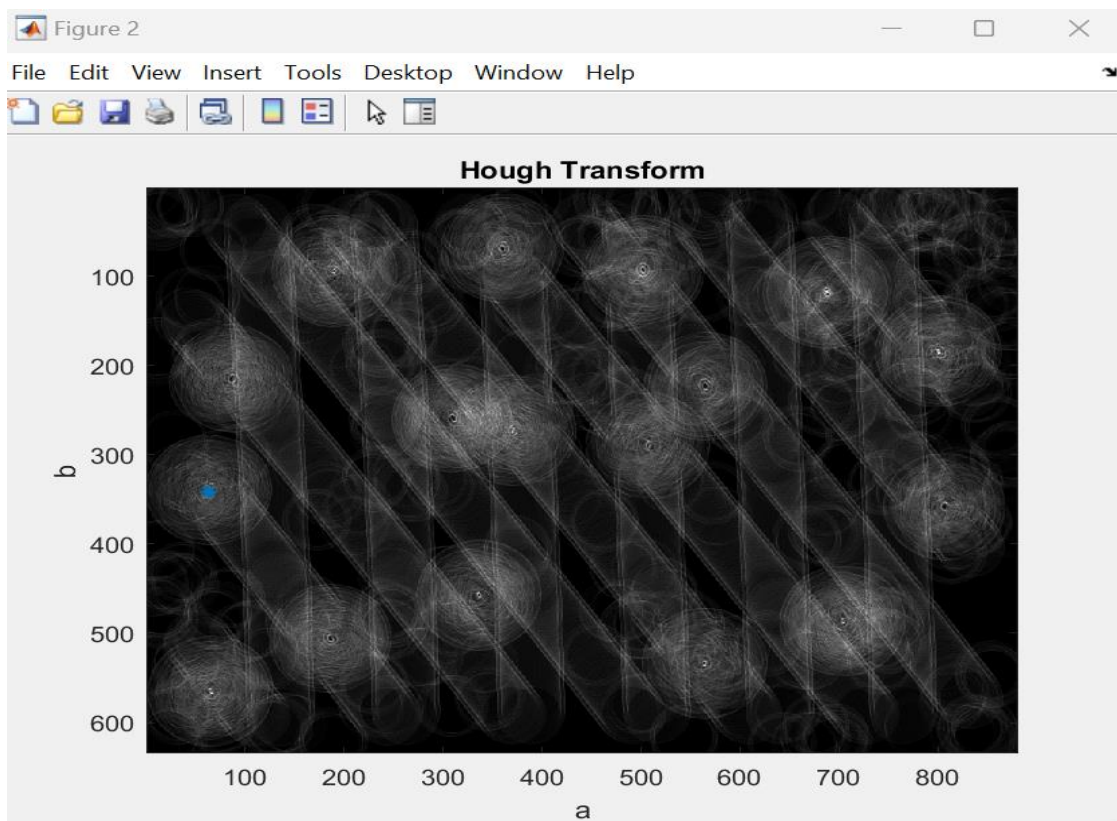
% Extracting all the params
[M, I_1]= max(hs_a (:));
%Coordinates in HT Space
[a, b] = ind2sub (size (hs_a), I_1);
plot(Y1(b), X1(a), '*', 'LineWidth',1);
hold off;
% Center co-or of circle
C = [Y1(b), X1(a)];
% plotting
figure;
imshow(input_image);
hold on;
theta = 0 : 0.01 : 2*pi;
y = R* cos(theta) + X1(a);
x = R* sin(theta) + Y1(b);
plot(x, y, 'g', 'LineWidth',2);
title({'circle with radius ' + R ' on original image'});
end

```

output:

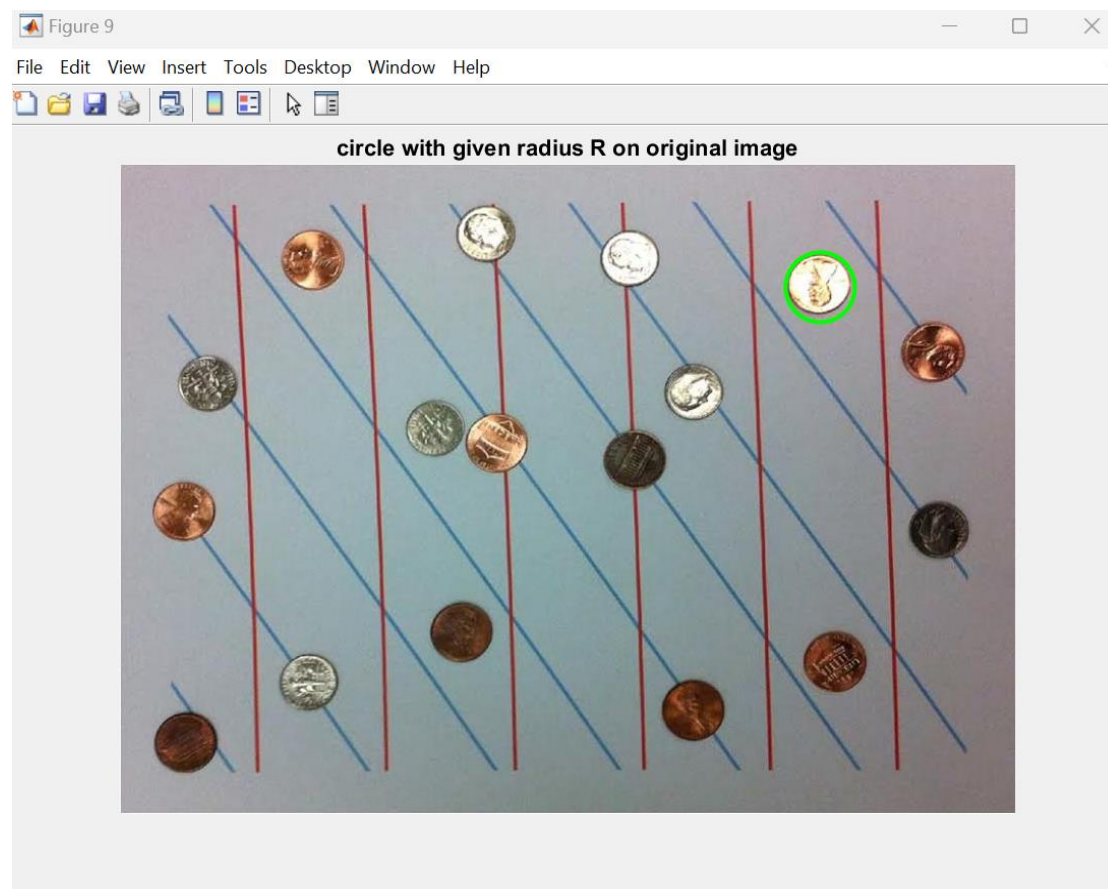
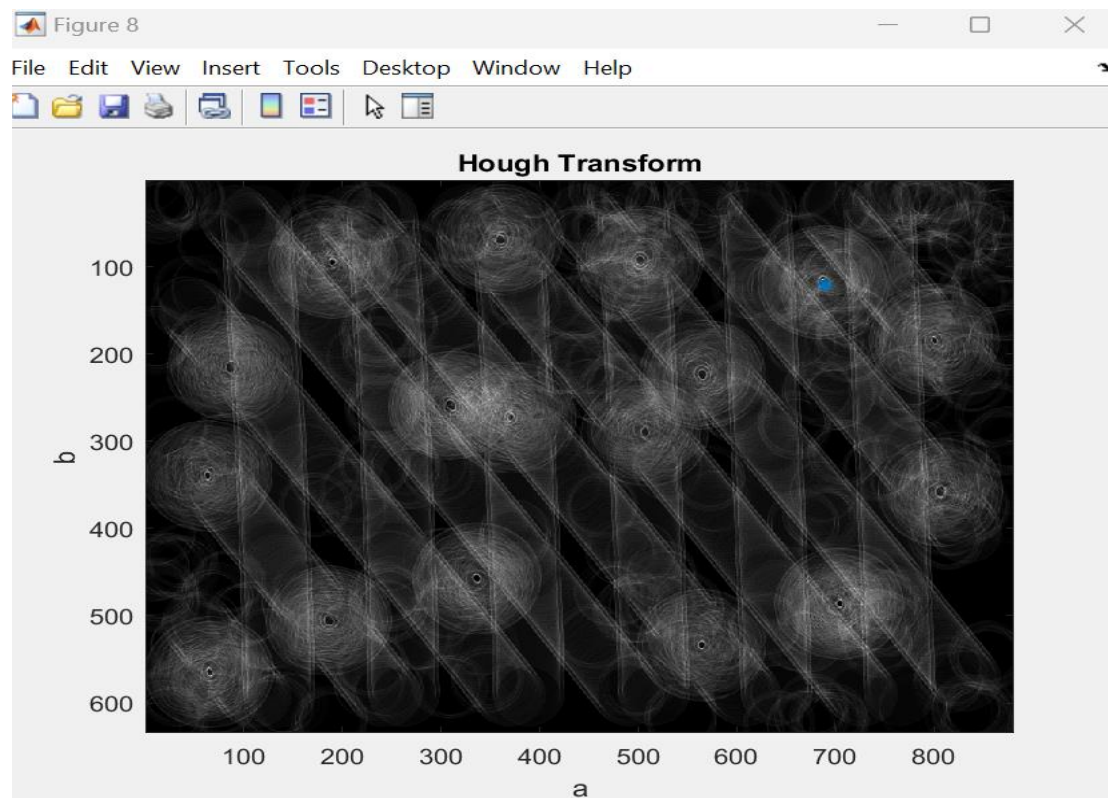


Circle with radius 32





Circle with radius 34



## Circle with radius 30

