Draw quad part of an ellipse start with (-10,0)

% Define the semi-axes lengths

a = 5;

b = 3;

% Define the center of the ellipse

x0 = 0;

y0 = 0;

% Generate a set of angles to be used for plotting the ellipse

t = linspace(0, pi/2, 100);

% Compute the x and y coordinates of the ellipse

x = x0 + a \* cos(t);

y = y0 + b \* sin(t);

% Plot the quarter part of the ellipse

plot(x, y, 'r')

% Add labels and grid to the plot

xlabel('X')

ylabel('Y')

grid on

% Set the axis limits

xlim([-15 15])

ylim([-15 15])

Q2 Input the provided image and apply two consecutive scaling with the provided scaling factors (1) Sx=3, Sy=1 (2) Sx=2, Sy=1

% Load the image

img = imread('pout.tif');

% Scale the image by a factor of 3 in the x-direction and 1 in the y-direction

img = imresize(img, [size(img,1), size(img,2)\*3]);

% Scale the image by a factor of 2 in the x-direction and 1 in the y-direction

img = imresize(img, [size(img,1), size(img,2)/2]);

% Display the processed image

imshow(img);

title('Scaled Image');

Q3 Input the provided image (Figure 2) balance the intensity contrast. Validate your solution with the help of histogram

% Load the image

img = imread('pout.tif');

% Convert the image to grayscale if it's not already

if size(img, 3) == 3

img = rgb2gray(img);

end

% Stretch the image's histogram to span the entire 0 to 255 range

img = imadjust(img, stretchlim(img));

% Display the original and processed images

subplot(1, 2, 1), imshow(img), title('Original Image');

subplot(1, 2, 2), imshow(img), title('Contrast-balanced Image');

% Plot the histograms of the original and processed images

figure

subplot(1, 2, 1), imhist(img), title('Original Histogram');

subplot(1, 2, 2), imhist(img), title('Contrast-balanced Histogram');

Paper 2

Q1 Draw octant part of a circle start with (-8,0)

% Define the center and radius of the circle

x0 = 0;

y0 = 0;

r = 8;

% Define the angles for the octant (45 degrees to 135 degrees)

theta = deg2rad(45):0.01:deg2rad(135);

% Calculate the x and y coordinates of the points on the circle

x = x0 + r \* cos(theta);

y = y0 + r \* sin(theta);

% Plot the points on the circle

plot(x, y, '-r');

% Add a grid and axis labels to the plot

grid on

xlabel('X');

ylabel('Y');

% Limit the plot to a square region centered on the origin

xlim([-10 10]);

ylim([-10 10]);

% Add a marker to indicate the starting point

hold on

plot(-8, 0, 'o', 'MarkerSize', 10, 'MarkerFaceColor', 'b');

Q2 Input the provided image and apply two consecutive rotations with the provided angles (1) 30° (2) 60°

% Load the image

img = imread('pout.tif');

% Rotate the image by 30 degrees

img\_rotated = imrotate(img, 30);

% Rotate the image by another 60 degrees

img\_rotated = imrotate(img\_rotated, 60);

% Display the final result

imshow(img\_rotated)

% Display the original and rotated images

subplot(1,2,1);

imshow(img);

title('Original Image');

subplot(1,2,2);

imshow(img\_rotated);

title('Rotated Image');

Q3 Input the provided image (Figure 1) balance the intensity contrast. Validate your solution with the help of

Histogram

% Load the image

img = imread('image.png');

% Convert the image to double format

img = im2double(img);

% Stretch the histogram to balance the intensity contrast

img\_stretched = histeq(img);

% Display the original and processed images

subplot(1,2,1);

imshow(img);

title('Original Image');

subplot(1,2,2);

imshow(img\_stretched);

title('Contrast-Balanced Image');

% Plot the histograms for comparison

figure;

subplot(1,2,1);

imhist(img);

title('Histogram of Original Image');

subplot(1,2,2);

imhist(img\_stretched);

title('Histogram of Contrast-Balanced Image');

Paper 3

Q1 Draw octant part of a circle start with (0, -8))

% Define the center of the circle

x0 = 0;

y0 = 0;

% Define the radius of the circle

r = 8;

% Define the number of points to use for plotting

n = 100;

% Generate the angles for each point

theta = linspace(0, pi/4, n);

% Calculate the x and y coordinates of each point

x = x0 + r \* cos(theta);

y = y0 + r \* sin(theta);

% Plot the octant of the circle

plot(x, y);

% Add axis labels and a title

xlabel('x');

ylabel('y');

title('Octant of a Circle');

% Add grid lines

grid on;

Task 2

Input the provided image and apply two consecutive rotations with the provided angles (1) 90° (2) 90°

% Load the image

img = imread('pout.tif');

% Rotate the image by 90 degrees

img\_rotated = imrotate(img, 90);

% Rotate the image by another 90 degrees

img\_rotated = imrotate(img\_rotated, 90);

% Display the final result

imshow(img\_rotated)

% Display the original and rotated images

subplot(1,2,1);

imshow(img);

title('Original Image');

subplot(1,2,2);

imshow(img\_rotated);

title('Rotated Image');

Task 3

Input the provided image (Figure 3) balance the intensity contrast. Validate your solution with the help of histogram

% Load the image

img = imread('pout.tif');

% Convert the image to grayscale

img\_gray = im2gray(img);

% Balance the intensity contrast of the image

img\_balanced = histeq(img\_gray);

% Display the original and balanced images

subplot(1,2,1);

imshow(img\_gray);

title('Original Image');

subplot(1,2,2);

imshow(img\_balanced);

title('Balanced Image');

% Plot the histograms of the original and balanced images

figure;

subplot(1,2,1);

imhist(img\_gray);

title('Histogram of Original Image');

subplot(1,2,2);

imhist(img\_balanced);

title('Histogram of Balanced Image');