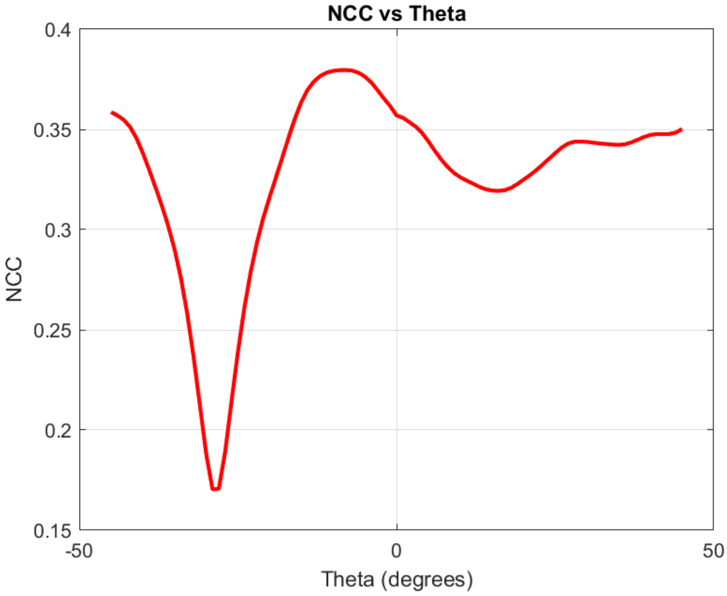
Q5

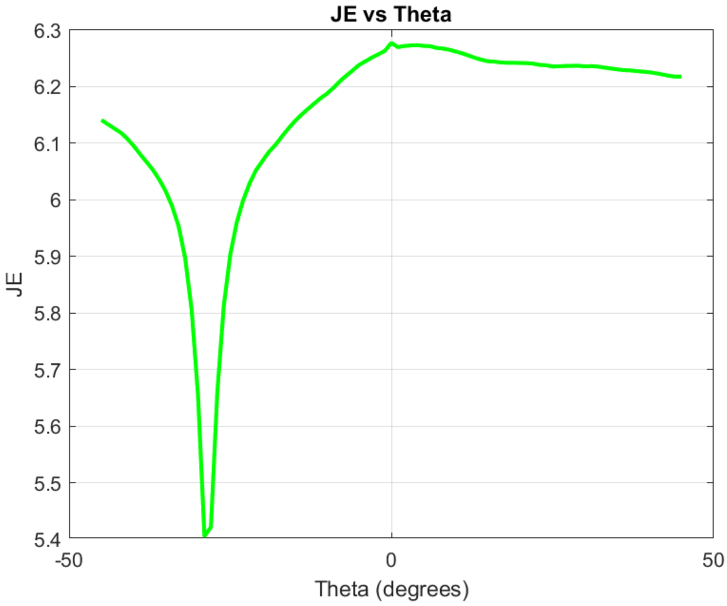
```
function QMI = calculate_QMI(joint_hist)
    % Computing Marginal histograms
    pI1 = sum(joint_hist, 2); % Sum over the second dimension
    pI2 = sum(joint_hist, 1); % Sum over the first dimension

    % Computing QMI
    QMI = 0;
    [rows, cols] = size(joint_hist);

    for i = 1:rows
        for j = 1:cols
            if joint_hist(i,j) > 0
                QMI = QMI + (joint_hist(i, j) - pI1(i) * pI2(j))^2;
            end
        end
    end
end

function JE = JointEntropy(joint_hist)
    % Calculate the joint entropy
    jointHist = joint_hist(joint_hist > 0);  % Remove zero probabilities
    if isempty(jointHist)
        JE = 0; % Return 0 if no non-zero values
    else
        JE = -sum(jointHist .* log2(jointHist));
    end
end

function joint_hist = compute_joint_histogram(I1, I2, num_bins, bin_width, range)
    joint_hist = zeros(num_bins, num_bins); % Initialize the joint histogram
    [rows, cols] = size(I1);

    % Rescale images from [0, 1] to [0, 255]
    I1_rescaled = I1 * 255;
    I2_rescaled = I2 * 255;

    for i = 1:rows
        for j = 1:cols
            % Compute bin indices
            bin1 = floor((I1_rescaled(i, j) - range(1)) / bin_width) + 1;
            bin2 = floor((I2_rescaled(i, j) - range(1)) / bin_width) + 1;

            % Ensure bins fall within valid range
            if bin1 >= 1 && bin1 <= num_bins && bin2 >= 1 && bin2 <= num_bins
                joint_hist(bin1, bin2) = joint_hist(bin1, bin2) + 1;
            end
        end
```

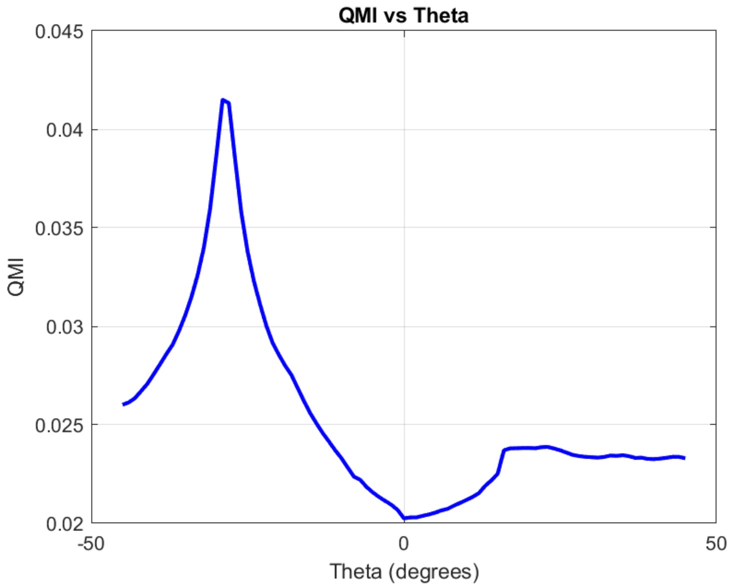
```
    end
    joint_hist = joint_hist / sum(joint_hist(:));
end

function NCC = normalizedCrossCorrelation(I1, I2)
    % Calculate the mean of each image
    mean_I1 = mean(I1(:));
    mean_I2 = mean(I2(:));
    numerator = sum((I1(:) - mean_I1) .* (I2(:) - mean_I2));
    denominator = sqrt(sum((I1(:) - mean_I1).^2) * sum((I2(:) - mean_I2).^2));
    if denominator == 0
        NCC = 0; % If denominator is zero, set NCC to zero
    else
        NCC = numerator / denominator;
    end
end
% Read the images
J1 = im2double(imread('C:\Users\Admin\OneDrive - Indian Institute of Technology Bomba
J2 = im2double(imread('C:\Users\Admin\OneDrive - Indian Institute of Technology Bomba

% Angle of rotation
theta = 28.5;
bin_width = 10; % Bin width
anrange= -45:1:45; % Range of angles
range = [0,255];
num_bins = ceil((range(2) - range(1) + 1) / bin_width);
n = length(anrange);
NCC = zeros(1,n);
JE = zeros(1,n);
QMI = zeros(1,n);

% We use bilinear interpolation and crop the image to the original
J3 = imrotate(J2, theta, 'bilinear', 'crop');
for i = 1:length(anrange)
    theta = anrange(i);
    J4 = imrotate(J3, theta, 'bilinear', 'crop');
    NCC(i)=normalizedCrossCorrelation(J4,J1);
    joint_hist =  compute_joint_histogram(J4, J1, num_bins, bin_width, range);
    JE(i)=JointEntropy(joint_hist);
    QMI(i) = calculate_QMI(joint_hist);
    %fprintf('NCC for theta = %f: %f\n', theta, NCC(i));
    %fprintf('JE for theta = %f: %f\n', theta, JE(i));
end

% Plot NCC vs Theta
figure;
plot(anrange, NCC, 'r', 'LineWidth', 2);
xlabel('Theta (degrees)');
```

```
ylabel('NCC');
title('NCC vs Theta');
grid on;

% Plot JE vs Theta
figure;
plot(anrange, JE, 'g', 'LineWidth', 2);
xlabel('Theta (degrees)');
ylabel('JE');
title('JE vs Theta');
grid on;

% Plot QMI vs Theta
figure;
plot(anrange, QMI, 'b', 'LineWidth', 2);
xlabel('Theta (degrees)');
ylabel('QMI');
title('QMI vs Theta');
grid on;


% Find the index of the minimum JE value
[~, optimal_index] = min(JE);

% Extract the optimal rotation angle
optimal = anrange(optimal_index);
%fprintf(optimal)
J4 = imrotate(J3, optimal, 'bilinear', 'crop');

% Compute the joint histogram for the optimal rotation
joint_hist_optimal = compute_joint_histogram(J1, J4, num_bins, bin_width, range);

% Plot the joint histogram using imagesc
figure;
imagesc(joint_hist_optimal);
colorbar;
xlabel('Bin Index for J1');
ylabel('Bin Index for J4');
title(sprintf('Joint Histogram for Optimal Rotation \\theta = %f degrees', optimal_th
axis equal; % Ensure that the aspect ratio is equal
grid on;
```

c.

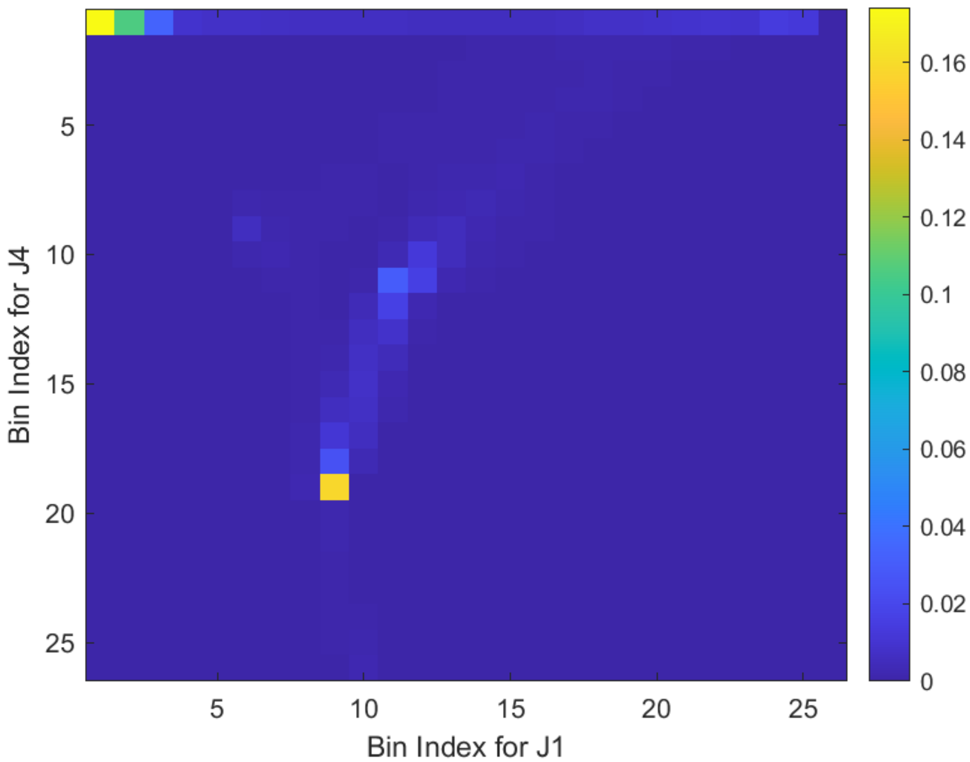

NCC vs Theta
0.4
0.35
0.3
0.25
0.2
0.15
NCC
-50
0
50
Theta (degrees)


JE vs Theta
6.3
6.2
6.1
6
5.9
5.8
5.7
5.6
5.5
5.4
JE
-50
0
50
Theta (degrees)



d. We see that the graphs for all the three plots of QMI, JE, and NCC give us the optimal rotation to be at 29 degrees so that the images align.
We determine the optimal rotation at the angle where NCC which is the measure of similarity and the Joint Entropy which measures the amount of randomness at the minimum value.We measure the QMI at the maximum value as it measures the dependence of the the two images and the values if peaks up at means the images are very close to being aligned.We also see that all the three parameters gives us the optimal rotation to be at the same angle of 29 degrees which is very close to the value we rotated at 28.5 degrees.

e.



f. Two random variables I_1 and I_2 are considered statistically independent if their joint distribution factorizes into the product of their marginal distributions:

$$p_{I_1 I_2}(i_1, i_2) = p_{I_1}(i_1) \cdot p_{I_2}(i_2)$$

Quadratic Mutual Information quantifies the dependency between two random variables. When the variables are independent, the QMI value is zero, indicating no additional shared information beyond their individual distributions. QMI is calculated by summing the squares of the similarities of all pairs of random variables, treating the bins of the distributions as random variables. To compute QMI between two images, one evaluates the statistical dependence of corresponding bins. The maximum dependence value indicates the angle of alignment between the images, reflecting their shared information and suggesting a strong correlation when the QMI is high, making it useful in image analysis.