

CS663: Digital Image Processing - Homework 1

Harsh | Pranav | Swayam

August 23, 2024

1 Homework 1 - Question 5

(a)

The MATLAB code for this question is as follows:

```
1 J3 = imrotate(J2, 28.5, 'bilinear', 'crop');  
2 J3(isnan(J3)) = 0;  
3  
4 % Displaying and saving the image  
5 figure, imshow(J3), title('Rotated Image J3')  
6 imwrite(J3, 'J3.jpg', 'jpg')
```

The rotated image is as follows:

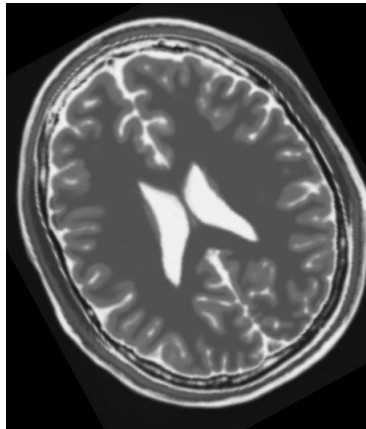


Figure 1: Rotated Image J3

(b)

The MATLAB code for this question is as follows:

```
1 angles = -45:1:45;
2
3 ncc = zeros(size(angles));
4 je = ncc;
5 qmi = ncc;
6
7 for i = 1:length(angles)
8     J4 = imrotate(J3, angles(i), 'bilinear', 'crop');
9     J4(isnan(J4)) = 0;
10
11     normJ4 = (J4 - mean(J4(:))) / std(J4(:));
12     normJ1 = (J1 - mean(J1(:))) / std(J1(:));
13
14     ncc(i) = sum(normJ1(:) .* normJ4(:)) / numel(J1);
15
16     jointHist = histcounts2(normJ1(:), normJ4(:), 7);
17     ↪ % 256^(1/3) bins approximately
18     jointProb = jointHist / sum(jointHist(:));
19     je(i) = -sum(jointProb(jointProb > 0) .* log2(
20         ↪ jointProb(jointProb > 0)));
21
22     P1 = sum(jointProb, 1);
23     P2 = sum(jointProb, 2);
24     P1P2 = zeros(length(P1));
25     for j = 1:length(P1)
26         for k = 1:length(P2)
27             P1P2(j, k) = P1(j) * P2(k);
28         end
29     end
30     qmi(i) = sum(sum((jointHist - P1P2).^2));
31 end
```

(c)

The plots for the three dependence measures are as follows:

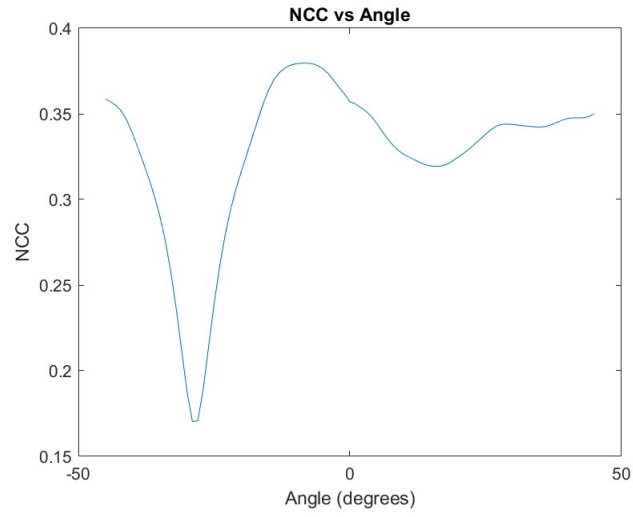


Figure 2: Normalized Cross Correlation

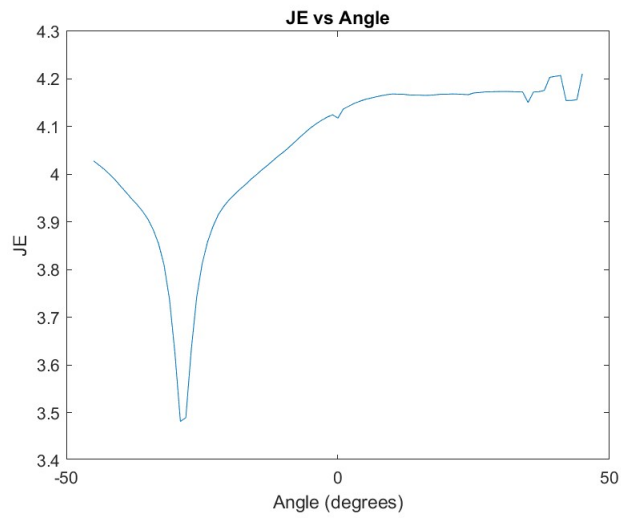


Figure 3: Joint Entropy

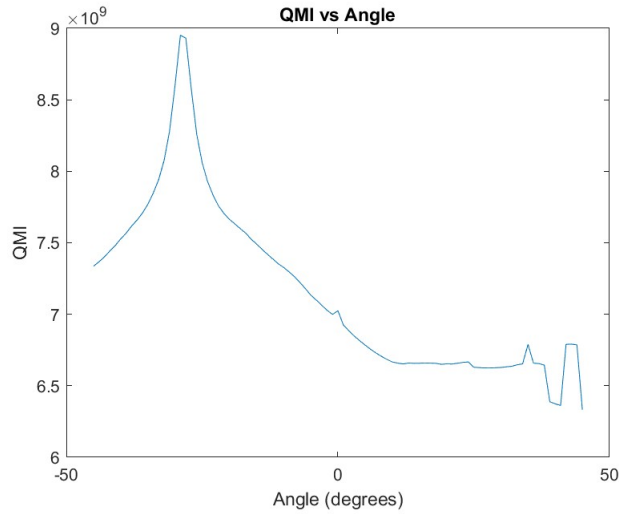


Figure 4: Quadratic Mutual Information

(d)

The MATLAB code for this question is as follows:

```

1 [~, nccIndex] = min(ncc);
2 [~, jeIndex] = min(je);
3 [~, qmiIndex] = max(qmi);
4
5 fprintf('Best NCC angle: %.2f degrees (NCC value: %.4f
   ↳ )\n', angles(nccIndex), ncc(nccIndex));
6 fprintf('Best JE angle: %.2f degrees (JE value: %.4f)\
   ↳ n', angles(jeIndex), je(jeIndex));
7 fprintf('Best QMI angle: %.2f degrees (QMI value: %.4f
   ↳ )\n', angles(qmiIndex), qmi(qmiIndex));

```

From the plots, the best angle for the three dependence measures comes out to be 29° **counter-clockwise**. We observe that at this angle, the NCC value is minimized, the JE value is minimized and the QMI value is maximized. Thus all the three dependence measures agree on the best angle of rotation.

Terminal Out:

```

Best NCC angle: -29.00 degrees (NCC value: 0.1703)
Best JE angle: -29.00 degrees (JE value: 3.4815)
Best QMI angle: -29.00 degrees (QMI value: 8951113300.4131)

```

(e)

The optimal angle of rotation is 29° counter-clockwise. The MATLAB code for this question is as follows:

```
1 optimJ4 = imrotate(J3, angles(jeIndex), 'bilinear', '
    ↪ crop');
2 optimJ4(isnan(optimJ4)) = 0;
3
4 bins = 37; % 256/7 approximately
5
6 histJ1 = histcounts(J1(:), bins);
7 histJ4 = histcounts(optimJ4(:), bins);
8
9 jointHist = zeros(bins);
10
11 for i = 1:size(J1, 1)
12     for j = 1:size(optimJ4, 2)
13         intensity_bin1 = floor(J1(i, j) * (bins - 1))
14             ↪ + 1;
15         intensity_bin2 = floor(optimJ4(i, j) * (bins -
16             ↪ 1)) + 1;
17         jointHist(intensity_bin1, intensity_bin2) =
18             ↪ jointHist(intensity_bin1, intensity_bin2
19             ↪ ) + 1;
20     end
21 end
22
23 % Normalize the joint histogram to create a joint
24     ↪ probability distribution
25 jointProb = jointHist / sum(jointHist(:));
26
27 % Plot the joint histogram
28 figure; imagesc(jointProb);
29 colormap jet;
30 colorbar;
31 xlabel('Intensity in J4 (Rotated)');
32 ylabel('Intensity in J1');
33 title('Joint Histogram between J1 and J4 (Optimal JE)'
34     ↪ );
35 axis xy;
36
37 % Save the plot as an image
38 saveas(gcf, 'JointHist_Optimal_JE.jpg');
```

The joint histogram between the original image J1 and the optimally rotated image J4 is as follows:

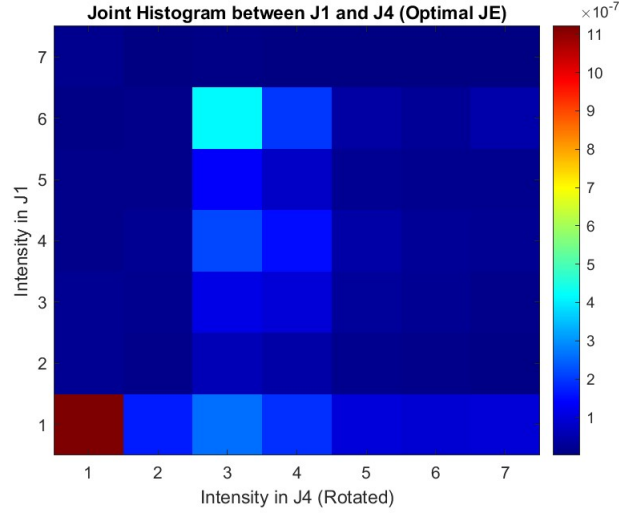


Figure 5: Joint Histogram between J1 and J4 (Optimal JE)

(f)

Quadratic Mutual Information (QMI) measures the statistical relationship between the two random variables, such as pixel intensities in images. It compares the actual joint distribution of these variables to the expected distribution if they were independent.

If the variables are independent, their joint distribution will be similar to the product of their individual distributions, resulting in a low QMI value. But if the variables are dependent, the joint distribution differs leading to a much higher QMI value.

Hence, QMI helps identify and quantify the dependence between two variables, capturing details that might not be evident with other measures.