CS663 : Digital Image Processing	Deadline: 12:00 AM, Aug-18 2021

Assignment 1

1 Question #4

Read in the images T1.jpg and T2.jpg from the homework folder using the MATLAB function imread and cast them as a double array. Let us call these images as J1 and J2. These are magnetic resonance images of a portion of the human brain, acquired with different settings of the MRI machine. They both represent the same anatomical structures and are perfectly aligned (i.e. any pixel at location (x,y) in both images represents the exact same physical entity). We are going to perform a simulation experiment for image alignment in a setting where the image intensities of physically corresponding pixels are different. To this end, do as follows:

(a) Write a piece of MATLAB code to rotate the second image by Theta = 28.5 degrees anti-clockwise. You can use the imrotate function in MATLAB to implement the rotation using any interpolation method. Note that the rotation is performed implicitly about the centroid of the image. While doing so, assign a value of 0 to unoccupied pixels. Let us denote the rotated version of J2 as J3.

```
J1 = imread("T1.jpg");
imshow(J1)
J1=J1+1;
J1 = double(J1);

J2 = imread("T2.jpg");
imshow(J2)
J2=J2+1;

J3 = imrotate(J2,28.5,'bilinear','crop');
imshow(J3)
J3 = double(J3);
```

Adding +1 to the original images so that after rotation all the pixels with 0 intensity would be the ones which are out of View and we would ignore them in our measurements

(b) Our job will now be to align J3 with J1 keeping J1 fixed. To this end, we will do a brute-force search over θ ranging from -45 to +45 degrees in steps of 1 degree. For each θ, apply the rotation to J3 to create an intermediate image J4, and compute the following measures of dependence between J1 and J4:

- the normalized cross-correlation (NCC)
- the joint entropy (JE)
- a measure of dependence called quadratic mutual information (QMI) defined as $\sum_{i_1}\sum_{i_2}(p_{I_1I_2}(i_1,i_2)-p_{I_1}(i_1)p_{I_2}(i_2))^2$, where $p_{I_1I_2}(i_1,i_2)$ represents the *normalized* joint histogram (*i.e.*, joint pmf) of I_1 and I_2 ('normalized' means that the entries sum up to one). Here, the random variables I_1, I_2 denote the pixel intensities from the two images respectively. For computing the joint histogram, use a bin-width of 10 in both I_1 and I_2. For computing the marginal histograms p_{I_1} and p_{I_2}, you need to integrate the joint histogram along one of the two directions respectively. You should write your own joint histogram routine in MATLAB - do not use any inbuilt functions for it.

First we will calculate the mean taking only those pixels which do not have intensity =0 in Image I2 (rotated image) (the ones which are in field of view)

```
%%NCC
function value =  ncc(I1,I2)
    totalI1 = 0;
    totalI2 = 0;
    [l,w] = size(I2);
    div=0;
    for i=1:l
        for j=1:w
            if I2(i,j) ==0
                continue
            else
                totalI1 = totalI1+I1(i,j);
                totalI2 = totalI2+I2(i,j);
                div=div+1;
            end
        end
    end

    meanI1 = totalI1/div;
    meanI2 = totalI2/div;
```

Now we would individually add the value corresponding to each digitally corresponding pixels which are in field of view

```
    num=0;
    den1=0;
    den2 = 0;

    for i=1:l
        for j=1:w
            if I2(i,j) ==0
                continue
            else
                num= num +(I1(i,j) - meanI1)*(I2(i,j)-meanI2);
                den1= den1 +(I1(i,j) - meanI1)*(I1(i,j)-meanI1);
                den2= den2 +(I2(i,j) - meanI2)*(I2(i,j)-meanI2);
            end
        end
    end
    value = num/sqrt(den1*den2);
   end
```

Now we would rotate J3 to form J4 and then would calculate the NCC between each image (make sure to put J4 as I2)

```
nccs = zeros(1,91);
for s = 1:91
    J4 = imrotate(J3,s-46,'bilinear','crop');
    nccs(1,s) = ncc(J1,J4);
end
```

Taking bins from [1,11),[11,21) ....[251,256) In this way we will ignore all values which are outside the field of view (value of I2 at those pixels is 0)

```
%%JE
% This function would calculate jointhistogram considering only pixels
%which are in the desired field of view in I2
%(basically the ones having greater than 0 intensity in I2 (rotated image)
function arr = jointhist(I1,I2)
    h = size(I1);
    HW = h(1)*h(2);
    arr = zeros(26,26);
    for i = 1:26
        for  j = 1:26
            arr(i,j) = sum(sum((I1 < i*10+1 & I1>= (i-1)*10+1) &
            (I2 < j*10+1 & I2>= (j-1)*10+1)));
        end
    end
end

%%The function which would calculate the total entropy
%We would add 0 where the value of hist is 0 as 0log(o) is 0

function jointent =  JE3(I1,I2)
   jointhis = jointhist(I1,I2);
   total = sum(sum(jointhis));
   jointhis = jointhis/total;
   jointent = 0;
   for h1 = 1:26
       for h2 = 1:26
           if jointhis(h1,h2) == 0  %ignoring the ones which have 0 value
               continue
           else
               jointent = jointent-jointhis(h1,h2)*log2(jointhis(h1,h2));
           end
       end
   end
end
```

Now we would rotate J3 to form J4 and then would calculate the JE between each image (make sure to put J4 as I2)

```
jnts = zeros(1,91);
for s = 1:91
    J4 = imrotate(J3,s-46,'bilinear','crop');
    jnts(1,s) = JE3(J1,J4);
end
```

```
%%QMI
function information = QMI(I1,I2)
   jointhis = jointhist(I1,I2);
   total = sum(sum(jointhis));
   jointhis = jointhis/total;
   his1 = sum(jointhis,2);
   his2 = sum(jointhis,1);
   information = sum(sum((jointhis - his1*his2).*(jointhis - his1*his2)));
end
```

Now we would rotate J3 to form J4 and then would calculate the JE between each image (make sure to put J4 as I2)

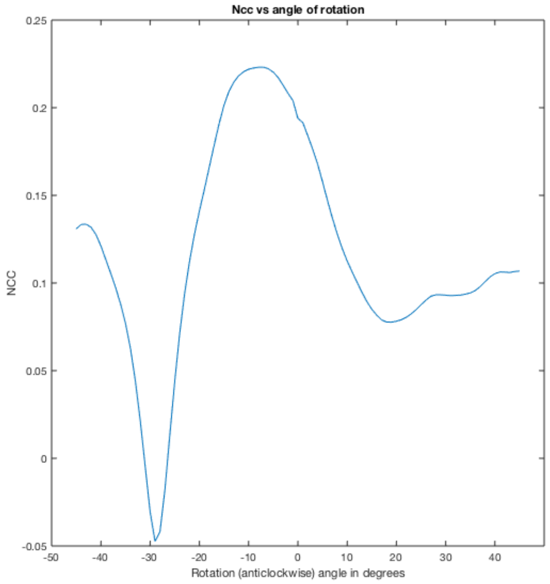
```
qmis = zeros(1,91);
for s = 1:91
    J4 = imrotate(J3,s-46,'bilinear','crop');
    qmis(1,s) = QMI(J1,J4);
end
```

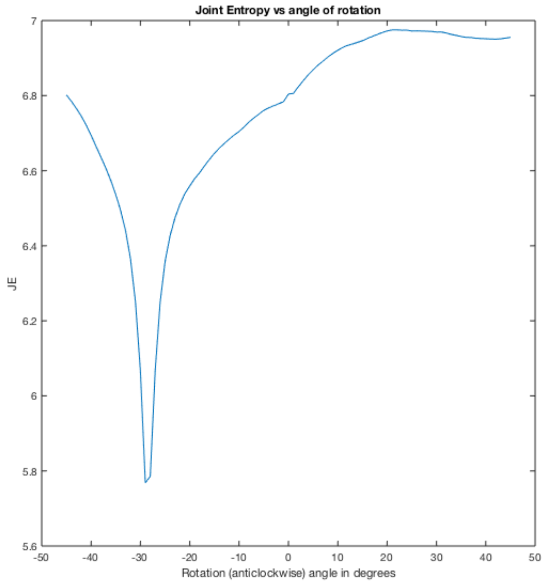
(c) Plot separate graphs of the values of NCC, JE, QMI versus and include them in the report PDF.

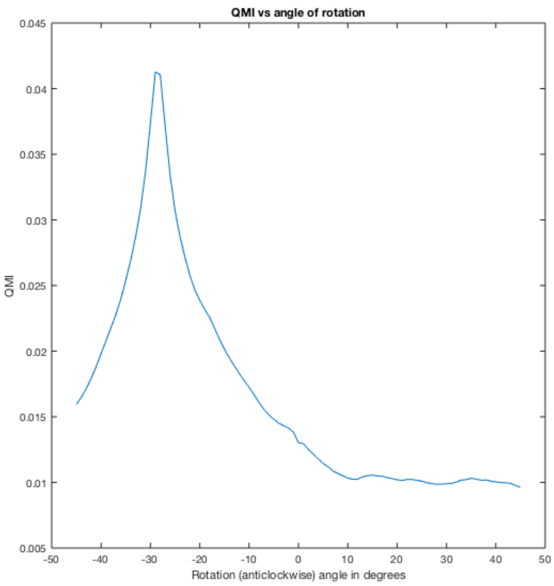
```
x = -45:45;
plot(x,nccs)
title("Ncc vs angle of rotation")
ylabel("NCC")
xlabel("Rotation (anticlockwise) angle in degrees")

plot(x,jnts)
ylabel("JE")
title("Joint Entropy vs angle of rotation")
xlabel("Rotation (anticlockwise) angle in degrees")

plot(x,qmis)
ylabel("QMI")
title("QMI vs angle of rotation")
xlabel("Rotation (anticlockwise) angle in degrees")
```


Ncc vs angle of rotation
NCC
Rotation (anticlockwise) angle in degrees
0.25
0.2
0.15
0.1
0.05
0
-0.05
-50
-40
-30
-20
-10
0
10
20
30
40


Joint Entropy vs angle of rotation
JE
Rotation (anticlockwise) angle in degrees
7
6.8
6.6
6.4
6.2
6
5.8
5.6
-50
-40
-30
-20
-10
0
10
20
30
40
50


QMI vs angle of rotation
QMI
Rotation (anticlockwise) angle in degrees
0.045
0.04
0.035
0.03
0.025
0.02
0.015
0.01
0.005
-50
-40
-30
-20
-10
0
10
20
30
40
50

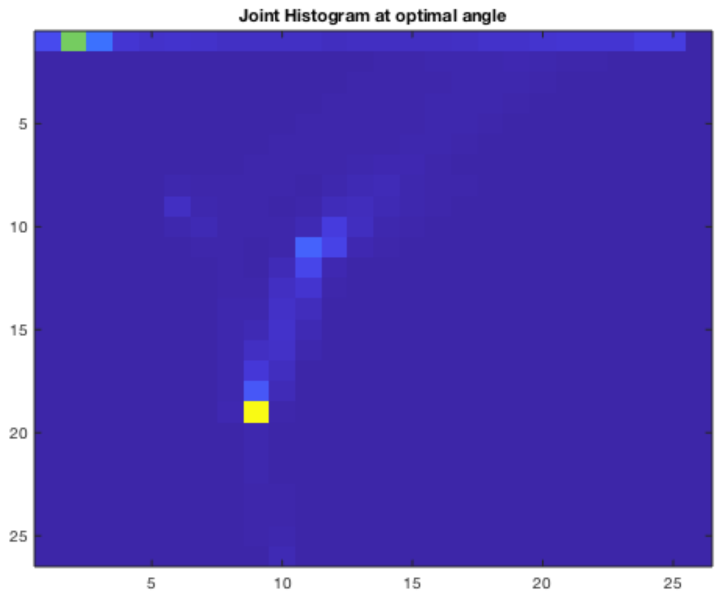
(d) Determine the optimal rotation between J3 and J1 using each of these three measures. What do you observe from the plots with regard to estimating the rotation? Explain in the report PDF.

```
%using NCC
[M1, I1] = min(nccs);
disp("optimal value using NCC in degree anticlockwise is")
disp(I1-46 )
%using JE
[M2, I2] = min(jnts);
disp("optimal value using JE in degree anticlockwise is")
disp(I2-46 )
%using QMI
[M3, I3] = max(qmis);
disp("optimal value using QMI in degree anticlockwise is")
disp(I3-46 )
```

The Optimal value obtained using all 3 method was 29 degree clockwise.
In the NCC and Joint entropy, the optimal value is found using mimimum value,
whereas in the Quadratic Mutual Information, the optimal value is found using the maximum value.
In this problem, all three measures gave accurate answer,
Although following points are noticeable:

- The Joint entropy curve decreased very steeply at the correct angle of rotation
- All these measures worked even when the original images were obtained in different settings

(e) For the optimal rotation using JE, plot the joint histogram between J1 and J4 using the imagesc function in MATLAB along with colorbar. Include it in the report PDF.



(f) We have studied NCC and JE in class. What is the intuition regarding QMI? Explain in the re- port PDF. (Hint: When would random variables I1 and I2 be considered statistically independent?)

We know from probability that if 2 events are independent then there intersection probability is given by multiplication of probability of each individual.

If A and B are 2 independent events then P(A$\cap B) = P(A) * P(B)$

Considering the same in matrix the joint histogram is basically the intersection of 2 events, whereas marginal histograms could be seen as happening of single event

If MI = P(A$\cap B) - P(A) * P(B)$

Looking at formula, take value of i1 and i2 as 1 and 2 respectively Now, pI1I2 (i1, i2) this is the probability where pixels of image I1 are in range [0,10) and where pixels of image 2 are in range [10,20). this is P(A$\cap B$)

whereas pI1 (i1) is just the probability where pixels of image I1 are in range [0,10), this is P(A) whereas pI1 (i2) is just the probability where pixels of image I1 are in range [10,20), this is P(B) So now if value of pI1I2 (i1, i2) is close to multiplication of pI1 (i1) and pI1 (i2), then the value of QMI would be low and could be considered independent and if the images are statistically independent, then we could infer that they are significantly different from each other.