CSE 5524 HW7

**Problem 1**

I first tried 100 super pixels with default compactness (10). Here is the result:

Graphical user interface, application

Description automatically generated

Then I decreased the value of compactness to 5. Here is the result:

Graphical user interface, application

Description automatically generated

As you can see, the super pixels adhere to boundaries slightly better.

Then I changed the number of super pixels to 1000 under this compactness, here is the result:

Graphical user interface, application

Description automatically generated

As you can see, the details of the original image are captured mostly.

Finally, I increase the value of compactness to 100 and expect to have square-liked super pixels:

Graphical user interface, application

Description automatically generated

As you can see, the expectation is met and it looks like the super pixel evenly partitions the image.

**Problem2**

Here is the plot for NCC scores:

Chart, line chart

Description automatically generated

Here is the template image for comparison:

A picture containing graphical user interface

Description automatically generated

Image for 1st match: image for 2nd match:

A picture containing graphical user interface

Description automatically generated A picture containing graphical user interface

Description automatically generated

As you can see, the best match looks pretty well, and there are tiny shifts in 2nd match

Image for 5th match: image for 10th match:

Graphical user interface

Description automatically generated A picture containing graphical user interface

Description automatically generated

Image for 100th match: image for 500th match:

A picture containing graphical user interface

Description automatically generated A picture containing application

Description automatically generated

As you can see, up to 100th match, it is still acceptable where we can find the elephant. When it comes to the 500th match, we cannot find the elephant at all. This makes sense if looking at the plot of the scores where the NCC is almost zero at 500.

Code:

%% Problem 1

A = imread('touma.jpg');

[L, NumLabels] = superpixels(A, 1000,'Compactness',100);

figure

BW = boundarymask(L);

imshow(imoverlay(A,BW,'cyan'),'InitialMagnification',67)

outputImage = zeros(size(A),'like',A);

idx = label2idx(L);

numRows = size(A,1);

numCols = size(A,2);

for labelVal = 1:NumLabels

redIdx = idx{labelVal};

greenIdx = idx{labelVal}+numRows\*numCols;

blueIdx = idx{labelVal}+2\*numRows\*numCols;

outputImage(redIdx) = mean(A(redIdx));

outputImage(greenIdx) = mean(A(greenIdx));

outputImage(blueIdx) = mean(A(blueIdx));

end

figure

imshow(outputImage,'InitialMagnification',67)

%% Problem 2

% load the image

template = im2double(imread('template.png'));

search = im2double(imread('search.png'));

% set up the score matrix

[r,c,l] = size(search);

scores = zeros(r,c,l);

% compute mean and standard deviation for each channel of template

meanT = computeMean(template);

stdT = computeStd(template);

%% compute the NCC score

for i = 1:3

for r = 24:277

for c = 35:366

P = search(r-23:r+23,c-34:c+34,:);

T = template;

meanP = computeMean(P);

stdP = computeStd(P);

val = 0;

for x = 1:47

for y = 1:69

val = val+ ((P(x,y,i)-meanP(i))\*(T(x,y,i)-meanT(i))/(stdP(i)\*stdT(i)));

end

end

val = val/(47\*69-1);

scores(r,c,i) = val;

end

end

end

%% best match

scores = mean(scores,3);

k1 = max(max(scores));

[a,b] = find(scores ==k1);

imagesc(search(a-23:a+23,b-34:b+34,:));

%% plot the NCC scores

ls = reshape(scores,1,[]);

ls = sort(ls,'descend');

plot(ls,'-r');

xlabel('k');

ylabel('NCC');

%%

[a,b]= find(scores == ls(500));

imagesc(search(a-23:a+23,b-34:b+34,:));

%% helper function

% compute the mean of each channel in the image

function result = computeMean(image)

meanR = mean(image(:,:,1),'all');

meanG = mean(image(:,:,2),'all');

meanB = mean(image(:,:,3),'all');

result = [meanR,meanG,meanB];

end

function result = computeStd(image)

stdR = std(image(:,:,1),0,'all');

stdG = std(image(:,:,2),0,'all');

stdB = std(image(:,:,3),0,'all');

result = [stdR,stdG,stdB];

end