CS 558: Homework Assignment 4 -- Visual Recognition
Program Language: MATLAB
Name: Lan Chang
Date: 04/27/17

1. Image Classification

First, I compute the histogram for each RGB channel in each training and testing image, while I also check that all pixels are counted exactly 3 times. Then, I use knn search to compute “nearest” representation. Finally, I display the result after classification and the accuracy, and the result after checking the verification.

I changed the number of bins several times, and I got the result that:

if bin=4, 8, accuracy=0.833,

if bin=2, 16, 32, 64, accuracy=0.75,

if bin=128, 256, accuracy=0.667,

if bin=1, accuracy=0.333.

Result:

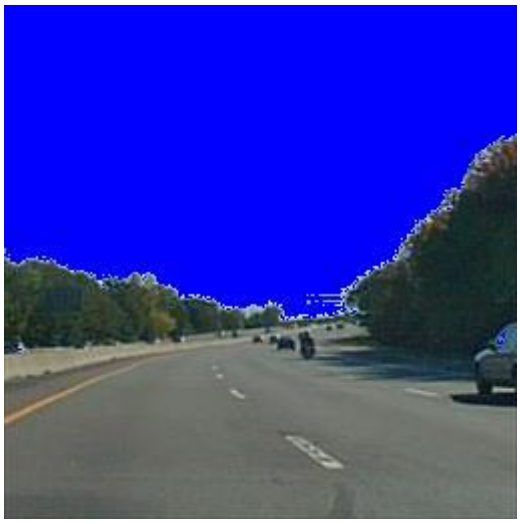
```
>> homework4_1
Test image 1 of class 1 has been assigned to class 1.
Test image 2 of class 2 has been assigned to class 2.
Test image 3 of class 3 has been assigned to class 2.
Test image 4 of class 1 has been assigned to class 1.
Test image 5 of class 2 has been assigned to class 2.
Test image 6 of class 3 has been assigned to class 2.
Test image 7 of class 1 has been assigned to class 1.
Test image 8 of class 2 has been assigned to class 2.
Test image 9 of class 3 has been assigned to class 3.
Test image 10 of class 1 has been assigned to class 1.
Test image 11 of class 2 has been assigned to class 2.
Test image 12 of class 3 has been assigned to class 3.
The accuracy of the classifier is 0.83333.
All pixels are counted exactly 3 times.
```

2. Pixel Classification

First, I use the original image and the mask to separate the sky and non-sky pixels. Next, I get ten visual words for each class using k-means and combine them together using one and zero to make difference. Then, for each pixel in each testing image, I find the nearest word and classify it as sky or non-sky. Finally, I paint the pixel blue if it is sky.

I think it works because in the training and testing image, the sky is almost blue and others are not. And I think it fails because some clouds are white (image 2 and 4), so they cannot classify as sky, and also because the color of the flowers and the mountain are close to the color of sky (image 3), so they also classify as sky. As a result, if we want to improve the accuracy, we can use more training image with more situations, we also can increase the number of k in knn searching.

Result:



3. MATLAB Code

(1) homework4_1.m

```
clear all;

% Problem 1: Image Classification
% parameter
class={'coast','forest','insidecity'};
bin=8;

% initialize
train_histogram=zeros(12,bin*3);
test_histogram=zeros(12,bin*3);
train_label=zeros(12,1);
test_label=zeros(12,1);

idx1=1;
v=0;
for i=1:4
    for j=1:3
        % training histogram
        train=imread(['ImClass/' class{j} '_train' num2str(i) '.jpg']);
        h11=histogram(train(:,:,1),bin);
        h12=histogram(train(:,:,2),bin);
        h13=histogram(train(:,:,3),bin);
        train_histogram(idx1,:)=[h11 h12 h13];
        train_label(idx1)=j;

        % testing histogram
        test=imread(['ImClass/' class{j} '_test' num2str(i) '.jpg']);
        h21=histogram(test(:,:,1),bin);
        h22=histogram(test(:,:,2),bin);
        h23=histogram(test(:,:,3),bin);
        test_histogram(idx1,:)=[h21 h22 h23];
        test_label(idx1)=j;

        % verification
        v1=sum(h11)+sum(h12)+sum(h13)-size(train,1)*size(train,2)*3;
        v2=sum(h21)+sum(h22)+sum(h23)-size(test,1)*size(test,2)*3;
        if v1~=0||v2~=0
            v=1;
        end

        idx1=idx1+1;
    end
```

```
end

idx2=knnsearch(train_histogram,test_histogram,'k',1,'Distance','euclidean');

count=0;
test_num=size(test_histogram,1);
for i=1:test_num
    % display the result
    idx3=num2str(i);
    class_name1=num2str(test_label(i));
    class_name2=num2str(test_label(idx2(i)));
    disp(['Test image ' idx3 ' of class ' class_name1 ' has been assigned to class '
class_name2 '.']);

    % count the correct
    if test_label(i)==train_label(idx2(i))
        count=count+1;
    end
end

% display the accuracy
disp(['The accuracy of the classifier is ' num2str(count/test_num) '.']);

% display the verification
if v==0
    disp('All pixels are counted exactly 3 times.');
else
    disp('All pixels are not counted exactly 3 times.');
end
```

(2) homework4_2.m

```
clear all;

% Problem 2: Pixel Classification
% load the image
image1=imread(['sky/sky_train.jpg']);                % original image
image2=imread(['sky/sky_train_mask.jpg']);        % mask
image1=double(image1);
image2=double(image2);

% separate sky from non-sky
sky=[];
non_sky=[];
idx1=1;        % sky index
idx2=1;        % non_sky index
for i=1:size(image1,1)
      for j=1:size(image1,2)
            % RGB=(255,255,255) color=white
            if image2(i,j,1)==255&&image2(i,j,2)==255&&image2(i,j,3)==255
                  sky(idx1,:)=image1(i,j,:);
                  idx1=idx1+1;
            else
                  non_sky(idx2,:)=image1(i,j,:);
                  idx2=idx2+1;
            end
      end
end

% get ten visual words for each class using k-means
k=10;
[~,sky_word]=kmeans(sky,k,'EmptyAction','singleton');
[~,non_sky_word]=kmeans(non_sky,k,'EmptyAction','singleton');
word=[ones(k,1) sky_word;zeros(k,1) non_sky_word];

% testing
for n=1:4
      % load the image
      test1=imread(['sky/sky_test' num2str(n) '.jpg']);
      s1=size(test1,1);
      s2=size(test1,2);
      s3=size(test1,3);

      % reshape the matrix
      test2=double(reshape(test1,s1*s2,s3,1));
```

```
    % find the closest word
    idx3=knnsearch(word(:,2:end),test2,'k',1,'Distance','euclidean');
    test3=word(idx3,1);

    % getting x and y
    [x,y]=ind2sub([s1 s2],1:s1*s2);

    for i=1:s1*s2
        % paint the pixel blue if it is sky
        if test3(i)==1
            test1(x(i),y(i),1)=0;
            test1(x(i),y(i),2)=0;
            test1(x(i),y(i),3)=255;
        end
    end

    % output
    figure,imshow(test1);
    imwrite(test1, ['output' num2str(n) '.jpg']);
end
```

(3) histogram.m

```
function result=histogram(image,bin)
s=double(image(:));

for i=1:bin
    if i==1
        result(i)=size(find(s<256/bin*i),1);
    else
        result(i)=size(find(s<256/bin*i),1)-sum(result(1:i-1));
    end
end

end
```