Aastha Chaudhary
CECS 553 – Machine Vision

Lab Assignment – 4

1. Read the tiny images located in tiny folder.

The images present in the tiny folder can be read by putting the folder location in the imread command and adding the suffix i.e. the image numbers after converting them to strings after the word " im " and add the extension after the name so as to scan every image which is present in the folder. Please find the code snippet below for reference-

```
clc
clear
close all

nr_im = 150;
for i = 1:nr_im

   im = imread(['tiny/im',num2str(i),'.png']);
   All(:,:,:,i) = im;
end
All = reshape(All,size(All,1)*size(All,2)*size(All,3),nr_im);  % Change All into a matrix (each row = one img)
```

2. Create labels for the image using the labels.txt file.

The labels for the corresponding images are present in the file labels.txt which is present in the tiny folder. The file contains the range of values which should be assigned to a particular set of images like for example first 16 images should have the labels starting from 1-16, likewise the next 17 images should have labels starting from 17-34 and so on.. We have first tried to identify the no. of labels and then tried to store the labels for each and every image in the labels matrix by extracting the labels id's from the ranges which are provided in the text file. Please find the code snippet for the same below-

```
%% Read the labels
fileID = fopen('tiny/labels.txt','r');  % Open the txt file for reading
chr = fscanf(fileID,'%c');           % Read data, '%c' = Read any single character
range = sscanf(chr,'%d');            % Convert chr to numbers, '%d' = integers
NrLabels = length(range)/2;

for i = 1:NrLabels
   labels(range(2*i-1):abs(range(2*i))) = i;% Save all of the imgs in All
```

```
end

fclose(fileID);                    % Close the file
```

3. Create 10 folds to do a cross validation.

We can assign the value 10 to K which is responsible for creating the folds through cvpartition command.

```
%% K-Fold Cross Validation

% Identifying the accuracy rates for the first 9 folds

K = 10;
```

4. Leave one-fold aside for testing and the remaining 9 folds for training and validation. Explain how you did that.

We have created the partitions through cvpartition command where we have extracted all the 10 folds and then started an iteration for the first 9 folds where we are gathering the data pertaining to the training and testing matrices for each and every fold. Please find the code below-

```
all_acc = [];
row = 1;
%idx = crossvalind('Kfold', nr_im, K)
c = cvpartition(nr_im,'KFold',K);      % Define a random partition for K folds

for j = 1:9
   idxTrain = training(c,j);           % Return the training indices for repetition j
   idxTest = test(c,j);                % Return the testing indices for repetition j
   TestIDs = find(idxTest == 1);       % Find the acctual index values
   nr_test = sum(idxTest);
   nr_train = sum(idxTrain);
```

5. In your 9 folds calculate the average accuracy rate for each value of k (k = 1 to 15) in your k-NN

After starting a loop for processing the 9 folds, we have defined an additional loop to calculate the accuracy for the k values starting from 1 to 15. The loop is basically going to replace the value of the k for each and every iteration and calculate the accuracy of the algorithm every time.

At the end we are going to calculate the average accuracy rate pertaining to each value of k using the mean function. Please find the code snippet below-

```
  acc = 0;
   for k = 1:15
      for i = 1:size(TestIDs,1)
         testId = TestIDs(i);
         testIm = All(:,testId);
         dst = abs(All(:,idxTrain)-repmat(testIm,1,nr_train));   %Calculate the distance between all
training imgs and the test img
         dst = sum(dst);
         [~,mnIdx] = mink(dst,k);              % Find the k smallest elements
         Allids = 1:nr_im;
         testLab = labels(idxTrain);
         lab = testLab(mnIdx);
         vote = mode(lab);
         acc = acc + (vote == labels(testId)); % Check to see if you predicted the label correctly
      end
      all_acc(row,k) = acc;
      %all_acc = [all_acc; {}];
   end
   row = row + 1;
   idxTrain = 0;            % Return the training indices for repetition j
   idxTest = 0;
end
avg_acc = mean(all_acc);
```

6. Plot a bar plot of all average accuracy rates using different k values and save it as png.

Please find the code snippet for plotting the average accuracy array for the corresponding k values.

```
bar(avg_acc);
title('k vs accuracy');
saveas(gcf,'k_vs_accuracy.png');

[mValue , vIndex] = max(avg_acc);
```

Please find image obtained below-

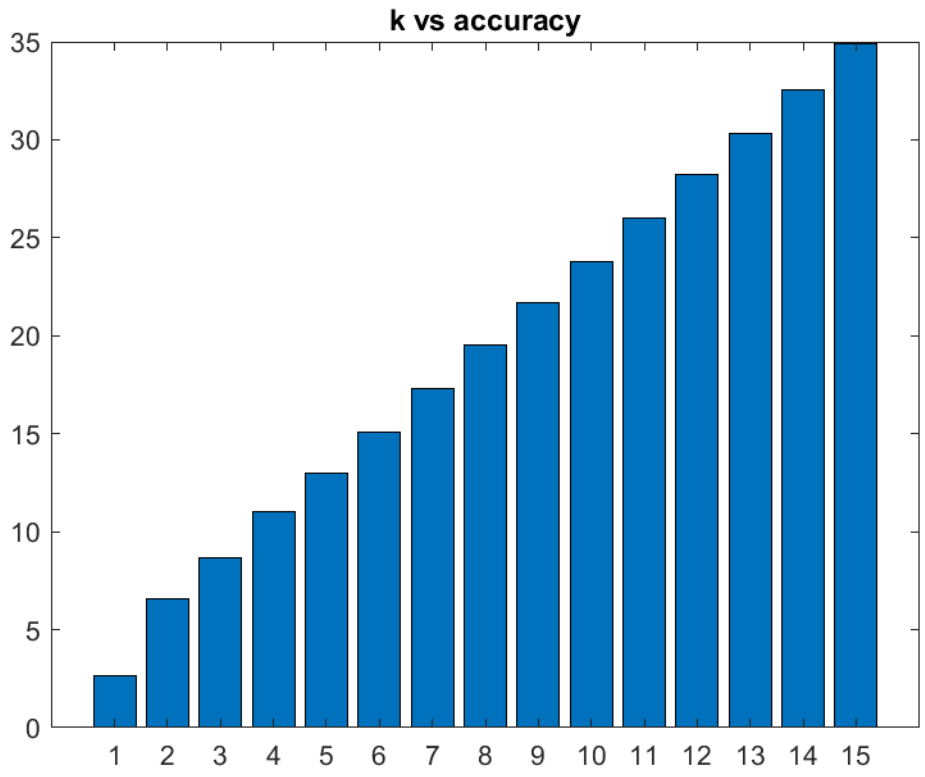

Fig 1. The bar plot for k values vs accuracy rates

7. Calculate the accuracy rate of your designed k-NN using the best selected k on the testing dataset from step four.

We have obtained the maximum value of k and the corresponding index for the same and then passed the same value for identifying the 10 fold's accuracy rate. Please find the code snippet for the same below-

```
%% Finding the accuracy of the testing fold

k = vIndex;
idxTrain = training(c,10);          % Return the training indices for repetition i
idxTest = test(c,10);               % Return the testing indices for repetition i
TestIDs = find(idxTest == 1);       % Find the acctual index values

nr_test = sum(idxTest);
nr_train = sum(idxTrain);

acc = 0;
for i = 1:size(TestIDs,1)
```

```
    testId = TestIDs(i);
    testIm = All(:,testId);
    dst = abs(All(:,idxTrain)-repmat(testIm,1,nr_train));   %Calculate the distance between all training
imgs and the test img
    dst = sum(dst);
    [~,mnIdx] = mink(dst,k);              % Find the k smallest elements
    Allids = 1:nr_im;
    testLab = labels(idxTrain);
    lab = testLab(mnIdx);
    vote = mode(lab);
    acc = acc + (vote == labels(testId));              % Check to see if you predicted the label correctly
end

fprintf('The accuracy of the testing fold is ');
disp(acc/size(TestIDs,1));
```

The accuracy obtained for this run is 0.13.

8. Upload your code (.m/.py files) and one pdf file that contains your code, your answers to questions, and the resulting images.

****