Stevens Institute of Technology

Department of Computer Science

CS558: Computer Vision

Homework Assignment 3

Submitted by:
Hadia Hameed
CWID: 10440803

Submitted to:
Prof. Xinchao Wang

May 3, 2019

1 Problem 1 - Image Classification:

Using the images in the ImClass directory. The objective of this problem is to classify each image of the test set into one of three classes:

1. coast
2. forest
3. insidecity

The representation will be in the form of three separate histograms of the R, G and B color channels. Each histogram will have 8 bins. Therefore, each image will be represented by 24 numbers. These representation should be computed for all images in the training set. The class labels of the 12 images in the training set will be considered known. When computing the histograms make sure that all pixels are counted exactly 3 times, once in each color channel. Include a verification step that will be submitted with your code. During testing, each image is classified independently of all other images in the test set. Use the same function to compute the representation and assign to the test image the label of the training image that has the "nearest" representation. The "nearest" representation should be computed using the Euclidean distance in the 24D histogram space. In other words, use the 1-nearest neighbor classifier. (You can use brute force search for the nearest neighbor due to the small size of the training set.) Your code should print a sting like the following for each image:

```
Test image 1 of class 2 has been assigned to class 1.
```

Compute the accuracy of your classifier and include it in the report.

Repeat the above experiments and show results for:

1. Changing the number of bins for each histogram (originally 8) to 4, 16, 32, meaning that the each image will be represented 12, 48, 96 numbers respectively.
2. Use 8 bins for each histogram, but classify a test sample using 3-nearest neighbor classifier.

Solution:

Algorithm 1 k-means segmentation.

procedure MYKNN(NK)
  for *each point i in testData* **do**
    for *each point j in trainData* **do**
      $D(j) \leftarrow$ distance between ith and jth point.
    $clst \leftarrow$ choose nk number of nearest neighbors.
    $predicted\_label(i) \leftarrow$ use majority voting by taking mode of clst
  $accuracy \leftarrow$ proportion of true test labels that match predictions
return *accuracy*

1.1 Output:

1.1.1 nbins = 8 and nk = 1

```
>> Test image 1 of class 1 has been assigned to class 1
>> Test image 2 of class 1 has been assigned to class 1
>> Test image 3 of class 1 has been assigned to class 1
>> Test image 4 of class 1 has been assigned to class 1
>> Test image 5 of class 2 has been assigned to class 3
>> Test image 6 of class 2 has been assigned to class 3
>> Test image 7 of class 2 has been assigned to class 2
>> Test image 8 of class 2 has been assigned to class 3
>> Test image 9 of class 3 has been assigned to class 3
>> Test image 10 of class 3 has been assigned to class 2
>> Test image 11 of class 3 has been assigned to class 3
>> Test image 12 of class 3 has been assigned to class 3
-----------------------------------
Average accuracy: 66.6667% with nbins = 8 using 1-nearest neighbor classifier.
```

Figure 1: Output for $nbins = 8 and number of neighbors = 1$

1.1.2 nbins = 8 and nk = 3

```
>> Test image 1 of class 1 has been assigned to class 2
>> Test image 2 of class 1 has been assigned to class 1
>> Test image 3 of class 1 has been assigned to class 1
>> Test image 4 of class 1 has been assigned to class 1
>> Test image 5 of class 2 has been assigned to class 3
>> Test image 6 of class 2 has been assigned to class 2
>> Test image 7 of class 2 has been assigned to class 2
>> Test image 8 of class 2 has been assigned to class 2
>> Test image 9 of class 3 has been assigned to class 2
>> Test image 10 of class 3 has been assigned to class 2
>> Test image 11 of class 3 has been assigned to class 3
>> Test image 12 of class 3 has been assigned to class 3
------------------------------------
Average accuracy: 66.6667% with nbins = 8 using 3-nearest neighbor classifier.
```

Figure 2: Output for $nbins = 8 and number of neighbors = 3$

1.1.3 Overall Results

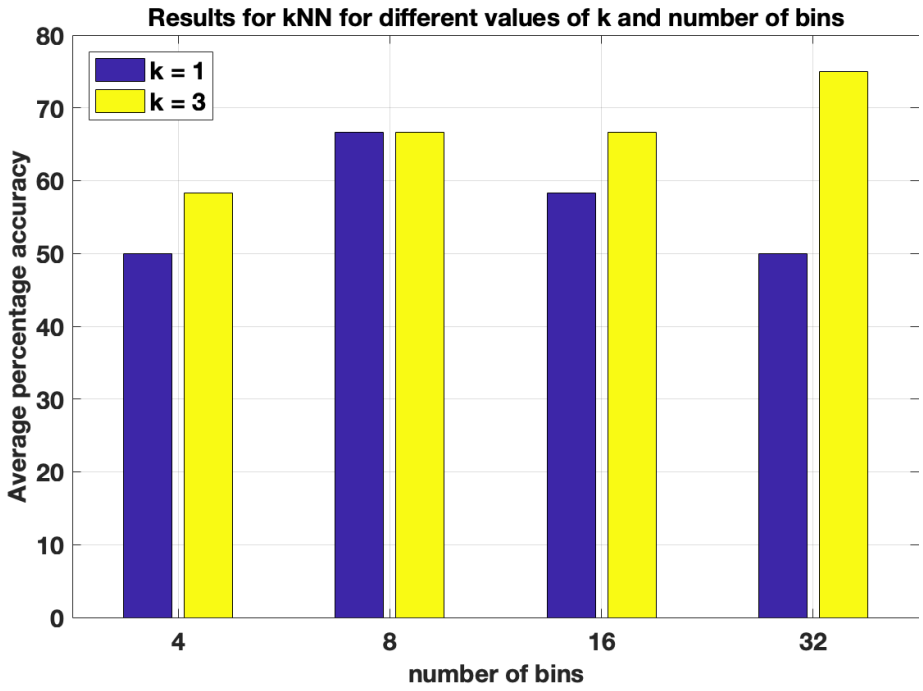

Figure 3: Output for different configurations of nbins and nk. The best performance is for $nbins = 32$ and $nk = 3$ with an average percentage accuracy of 75%

