Addis Ababa Science and Technology University

Collage of Electrical and Mechanical Engineering

Department of Electrical and Computer Engineering

Computer Engineering Stream

Computer Vision

Assignment 3: Bag of Visual words For Object Classification

By

1. **Sileshi Nibret** **GSR 217/12**
2. **Gebeyaw Tigabu** **GSR 210/12**

Submitted to: Dr. Beakal Gizachew

March 3, 2021

Introduction of Bag of Visual Words

Bag of words is applicable in natural language processing to extract features based on appearance of words in a text i.e., the frequency of words in a text is taken as a feature. In Computer Vision, the same concept is used in the bag of visual words. Here instead of taking the word from the text, image *patches* and their feature vectors are extracted from the image into a bag. Features vector is nothing but a unique pattern that we can find in an *image*. Currently, there are many deep learning models that are used for image classification. No doubt these models show a very impressive state of art accuracy and have become industry standards. However, prior to the deep learning boom, we still had many classical techniques for image classification. The general idea of bag of visual words (BOVW) is to represent an image as a set of features. Features consists of key points and descriptors. No matter the image is rotated, shrink, or expand, its key points will always be the same. And descriptor is the description of the key point. We use the key points and descriptors to construct vocabularies and represent each image as a frequency histogram of features that are in the image. From the frequency histogram, later, we can and another similar images or predict the category of the image.

In this assignment, we develop an image classier based on Bag-of-Features model using Python. We download Caltech 101 dataset which contains many numbers of objects and we have selected 10 objects which contains different instances per class. There are 10 classes i.e., brain, butterfly, car_side, city, dollar_bill, flamingo, headphone, laptop, wild_cat and wrench. The dataset is also divided into two as training and test based on 80-20 (80% for training and 20 % for testing from each class).

We train the K-NN classier model using the training image set based on k-fold cross validation on the training data to find the appropriate number of neighbors for training the model and after finding the appropriate K value we trained the model and test the model on the remaining 20% test images.

Procedures followed

In this assignment we followed the following procedures to classify the objects using bag of visual words (BoVW).

1. First, we downloaded the Caltech 101 object's image dataset
2. Then, we have selected 10 objects randomly
3. After selecting objects, we split the dataset into training and testing using 80% of the images for

training and 20% of the images for testing.

4. After splitting the dataset, we extract features form images. The first step to build a bag of visual words is to perform feature extraction by extracting descriptors from image in our dataset. One of the famous descriptors is Scale-invariant feature transform (SIFT). In which is the regions on the image as a features using ***SiftDescriptors()*** which is self-defined function using OpenCV package *cv2. xfeatures2d.SIFT_create()*
5. After SIFT features are extracted from the image the descriptors of the SIFT features are computed using step 4 descriptors and vectorized. After vectoring the descriptors, descriptors are stacked vertically as row vector using ***DescriptorVstacking()***.
6. After stacking the descriptors we need to cluster the descriptors for features using KMeans clustering defined function ***DescriptorsClustering()***
7. Feature extraction for each training image and test images and storing in matrix to train the models using ***FeatureExtracter()***
8. After features are extracted training the model using k fold cross validation was done the k-folds cross validation is used to find the optimal value of K (number of neighbors for the KNN model).
9. Using optimal value of k based on the error plot from k-fold cross validation we train our model to predict the remaining test data split on step 3 to test overall accuracy of our system.

Methodology

In this assignment we have classified the objects using different parameters from them number of clusters is one. Number of clusters define the number of SIFT features to extract from the images. Small number of clusters means taking small number of features which yields small accuracy since the model didn't get best and many features its hard to classify objects exactly. However large number of clusters may yield good accuracy but need computational resource.

In our experiment we take 50,60,100 and 200 clusters to extract features and compare the accuracy score.

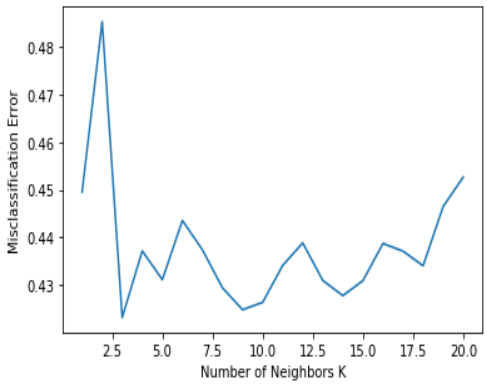
Another hyper parameter is the number of neighbors to compute distance in the KNN model. To find the optimal number of neighbors we performed k-fold cross validation on training data (80% of total data) and we have used 5-fold and 10-fold cross validation to get best K value. The best value of K (capital K) is use to test the remaining test data (20 % of total dataset).

Results

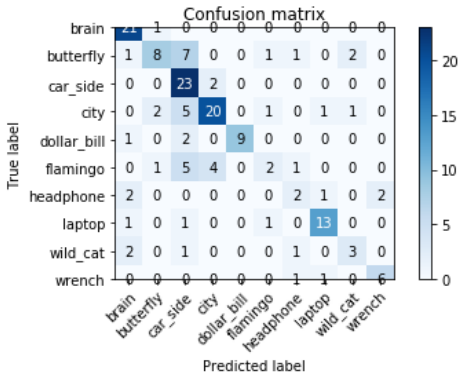
For testing the KNN mode, we take 6 scenarios to get the best of the hyperparameters.

Scenario 1: Number of clusters=50 and 10-fold cross validation

```
Maximum accuracy gained in trainig using k-fold is:57.683

Minimum Error gained in trainig using k-fold is:0.423

The optimal number of neighbors (Value of K) is 3
```




```
Training the K-NN Model using optimal k value....
Testing the K-NN Model Remaining 20% Test Data....
Test images path detected.
```

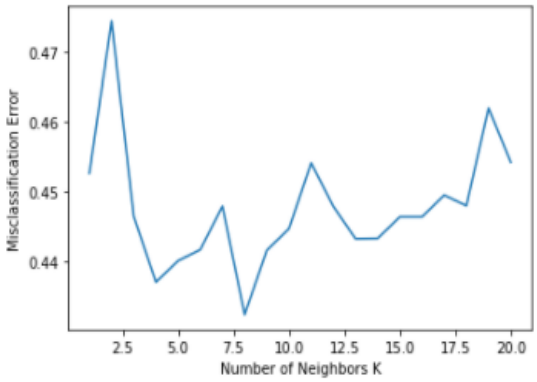



```
Accuracy score: 66.875
Total Time taken in Min: 3.897
```

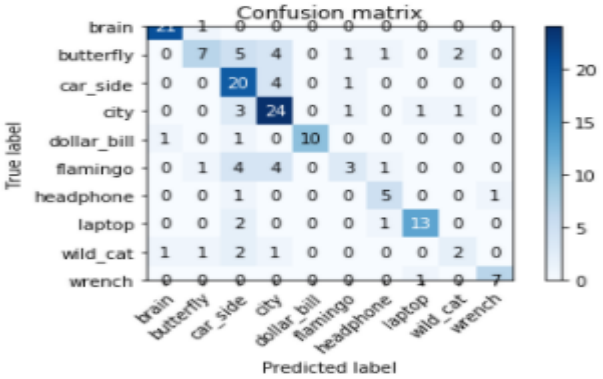
Error plot of 10-fold cross validation and Confusion matrix using optimal value of k which is 3-NN test result

Scenario 2: Number of clusters=60 and 10-fold cross validation

```
Maximum accuracy gained in trainig using k-fold is:56.752

Minimum Error gained in trainig using k-fold is:0.432

The optimal number of neighbors (Value of K) is 8
```




```
Training the K-NN Model using optimal k value....
Testing the K-NN Model Remaining 20% Test Data...
Test images path detected.
```

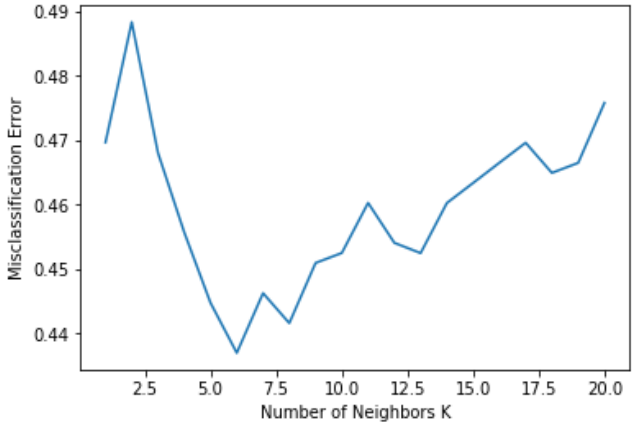



```
Accuracy score: 70.000
Total Time taken in Min: 4.688
```

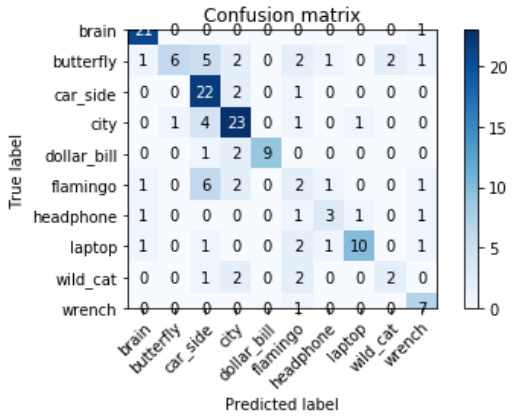
Error plot of 10-fold cross validation and Confusion matrix using optimal value of k which is 8-NN test result

Scenario 3: Number of clusters=100 and 5-fold cross validation

```
Maximum accuracy gained in trainig using k-fold is:56.298

Minimum Error gained in trainig using k-fold is:0.437

The optimal number of neighbors (Value of K) is 6
```



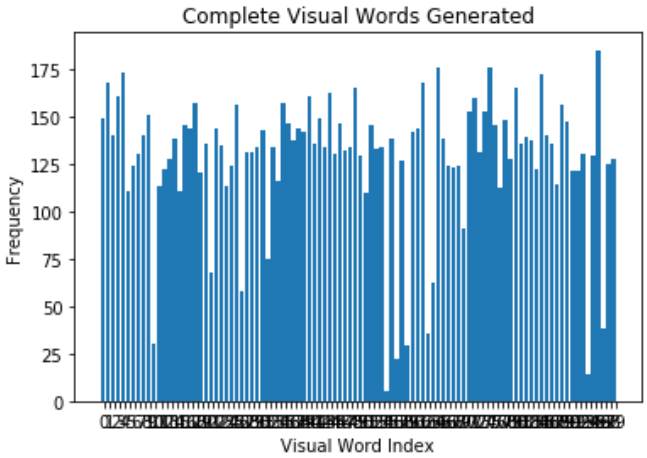

```
Training the K-NN Model using optimal k value...
Testing the K-NN Model Remaining 20% Test Data..
Test images path detected.
```



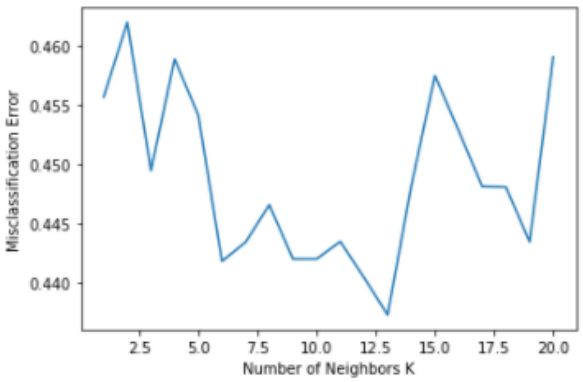

```
Accuracy score: 65.625
Total Time taken in Min: 7.864
```

Error plot of 10-fold cross validation and Confusion matrix using optimal value of k which is 6-NN test result

Scenario 4: Number of clusters=100 and 10-fold cross validation

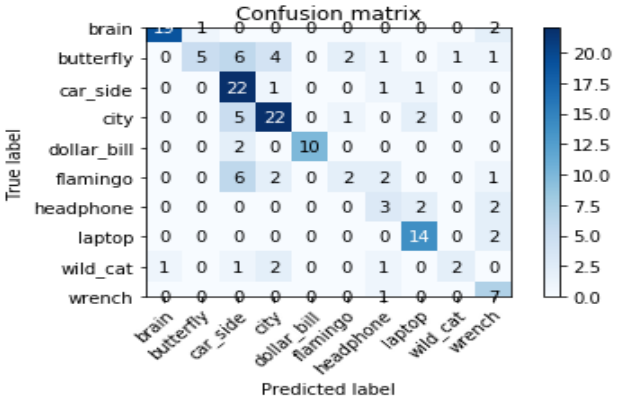



```
Maximum accuracy gained in trainig using k-fold is:56.272

Minimum Error gained in trainig using k-fold is:0.437

The optimal number of neighbors (Value of K) is 13
```



Histogram of 100 features plotted using the frequency (tf-idf) and Error plot of 10-fold cross validation

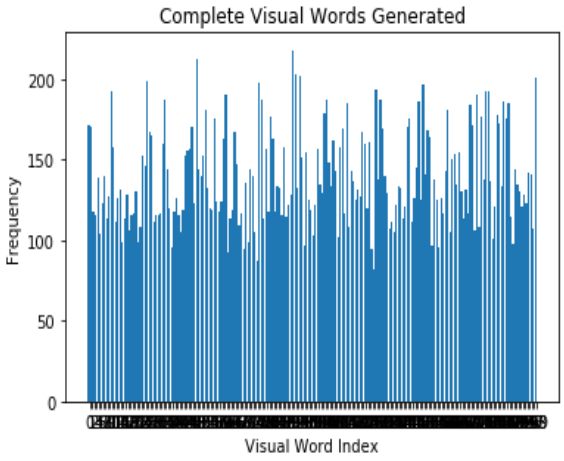
```
Training the K-NN Model using optimal k value....
Testing the K-NN Model Remaining 20% Test Data....
Test images path detected.
```



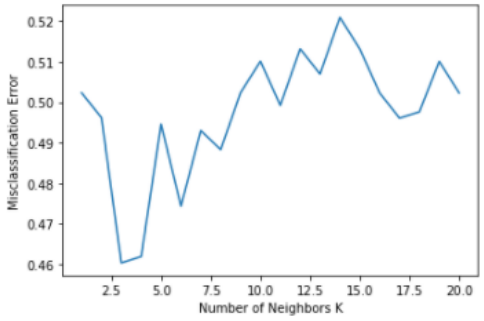

```
Accuracy score: 66.250
Total Time taken in Min: 7.391
```

Confusion matrix and total accuracy gained using optimal value of k which is 13-NN test result

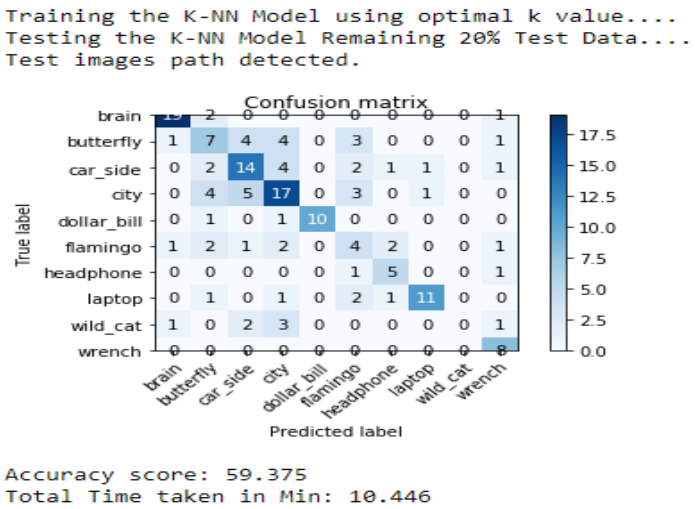
Scenario 5: Number of clusters=200 and 5-fold cross validation




```
Maximum accuracy gained in trainig using k-fold is:53.966

Minimum Error gained in trainig using k-fold is:0.460

The optimal number of neighbors (Value of K) is 3
```

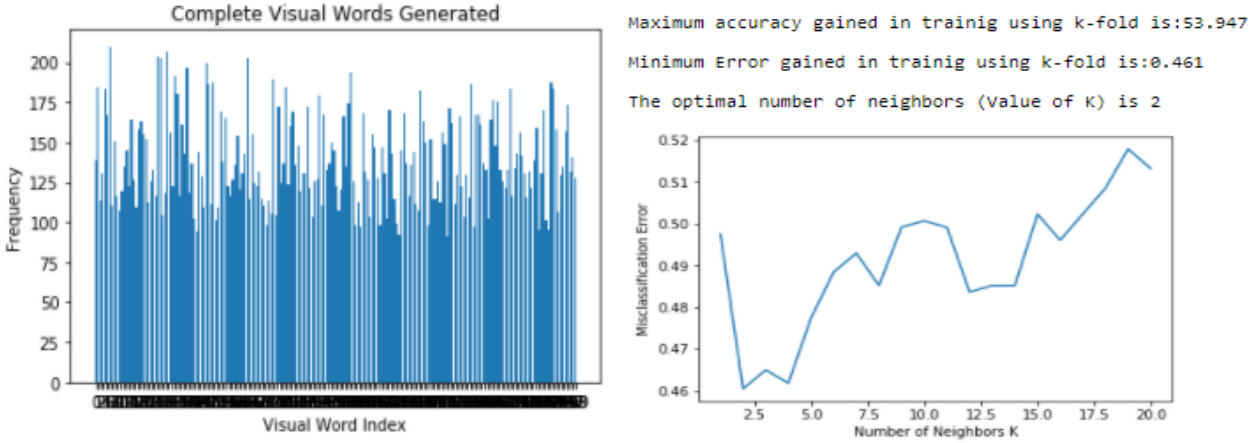


Histogram of 100 features plotted using the frequency (tf-idf) and Error plot of 5-fold cross validation



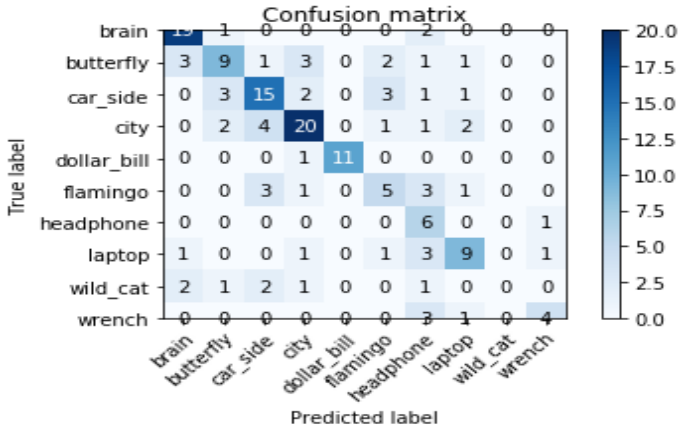
Confusion matrix and total accuracy gained using optimal value of k which is 3-NN test result

Scenario 6: Number of clusters=200 and 10-fold cross validation



Histogram of 200 features plotted using the frequency (tf-idf) and Error plot of 10-fold cross validation

```
Training the K-NN Model using optimal k value....
Testing the K-NN Model Remaining 20% Test Data....
Test images path detected.
```




```
Accuracy score: 61.250
Total Time taken in Min: 10.373
```

Confusion matrix and total accuracy gained using optimal value of k which is 2-NN test result

Summary of Results

Number of clusters	k-fold (cross validation)	Optimal K	Accuracy (%)
50	10	3	66.875
50	5	9	68.120
60	10	8	**70.00**
60	5	6	65.625
100	5	6	65.623
100	10	13	66.250
200	5	3	59.372
200	10	2	61.250

Conclusion

we classify test images according to training images. We have used K nearest neighbor (KNN) using k-fold cross validation on the training images. Total of 643 training images are classified into 10 classes Since there exists different number of train images for each class. The unbalance images in the classes may affect the model in its decision process. Selecting optimal value of clusters for clustering and selecting best K value for the model in addition to k-fold cross validation is done. Therefore, the optimal number of clusters is gained is **60** with **10-fold** cross validation to yields **70%** correct classification. This accuracy is gained using 8 nearest neighbors (8-NN). Increasing number of clusters does not increase the accuracy. Which is because increasing number of features beyond certain optimal point does not improve the model's accuracy.

We have compared our result to the assignment report which are reported in 2019 to Stanford university that used 7 classes from Caltech 101 dataset and the reported result was 50.1 % accuracy. The main difference of there method and ours is finding the optimal values of the hyperparameters. For example, we train our model using k-fold cross validation to find best value of K and cluster the descriptors using different number of clusters.

The model can be improved by using better clustering of descriptors, extract features using other key point extractor algorithms and fine tuning the hyper parameters of the classifier model.