Computer Vision HW3 Report

Onur YILMAZ S009604

Image Classification with Bag of Features

1. Introduction

Bag of visual words is commonly utilised in image classification. Its concept is adapted from information retrieval and NLP's bag of words.

The common idea of a bag of visual words is to describe an image as a set of features. Features consist of keypoints and descriptors. Keypoints are the "stand out" points in an image, so 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. I utilise the key points and descriptors to construct vocabularies and describe each image as a frequency histogram of features that are in the image. From the frequency histogram, later, we can find another similar image or predict the category of the image.

In this assignment, we are expected to develop an image classifier based on the Bag-of-Features model using Python. We are given a dataset which contains a variable number of instances per class. There are 4 classes which are 'Airplanes', 'Cars', 'Faces' and 'Motorbikes'. The dataset is also divided into two as training and test. We are expected to train our classifier using the training image set and test it using the test image set.

2. Implementation

1. Find a training images path.

2. Read train images, and compute SIFT descriptors.

3. Find the dictionary.

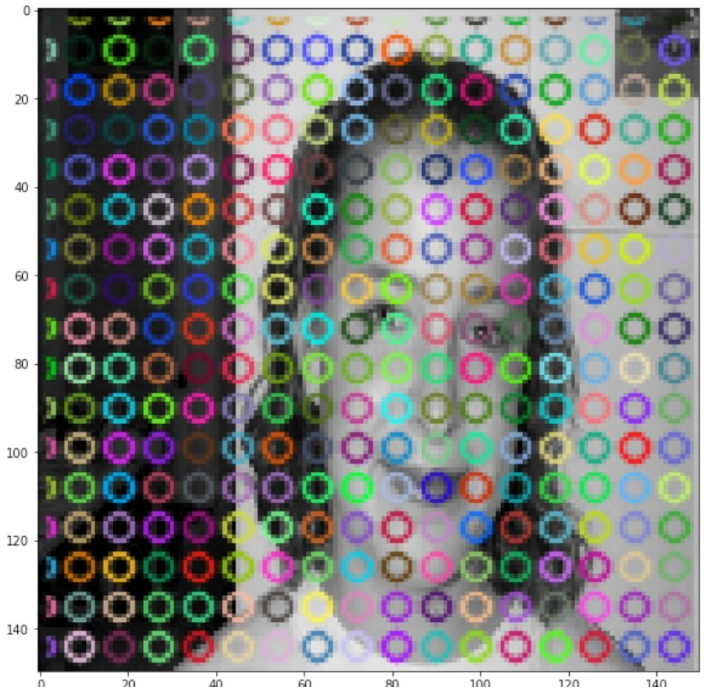
4. Quantize features.

5. Classify test images.

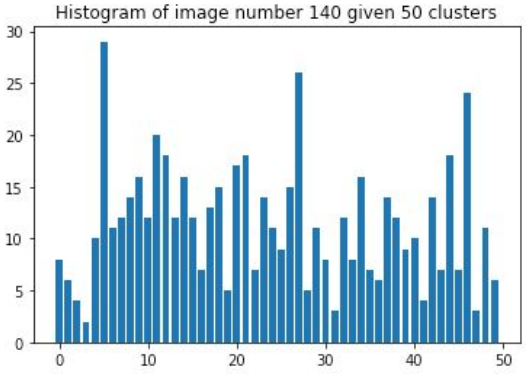
3.Experiment Outputs

- **3.1 GRID1 & SIFT Kmean: K=50**

Sample Image Divided 3*3 Boxes.

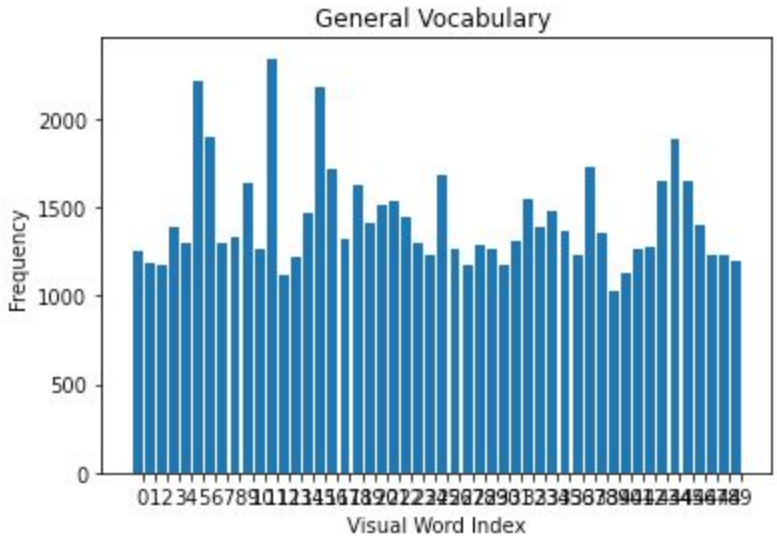


Histogram Evalation of Face Image



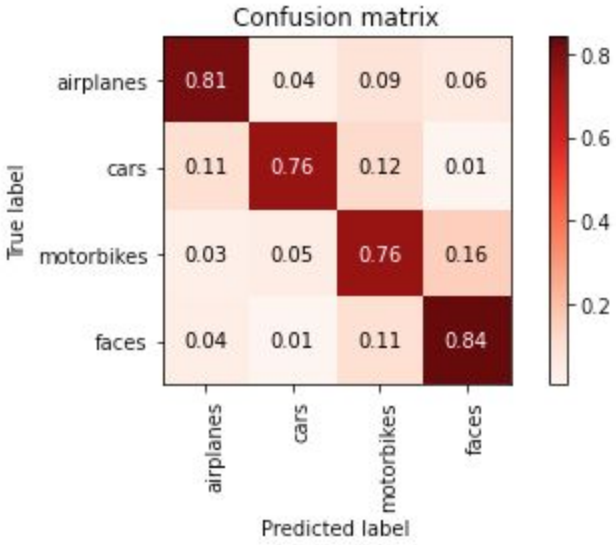

```
<matplotlib.image.AxesImage at 0x13ae3a8d0>
```

Total Image Histogram with K=50:



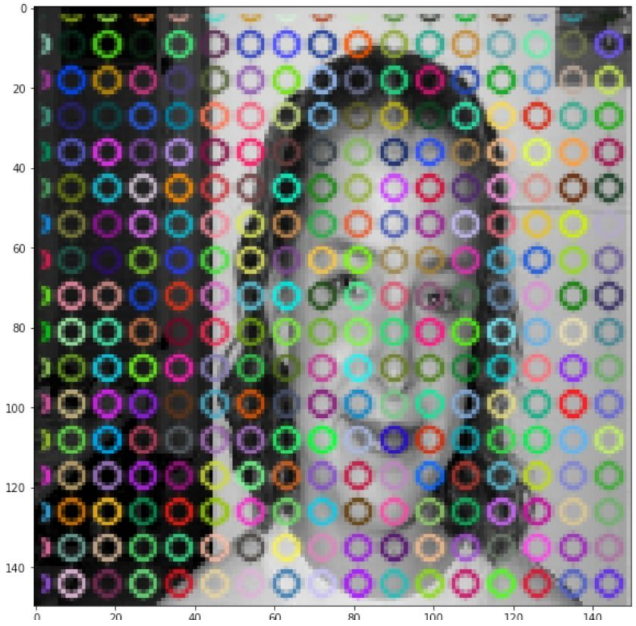
Confusion Matrix:

The Prediction accuracy of this model is 79.00%

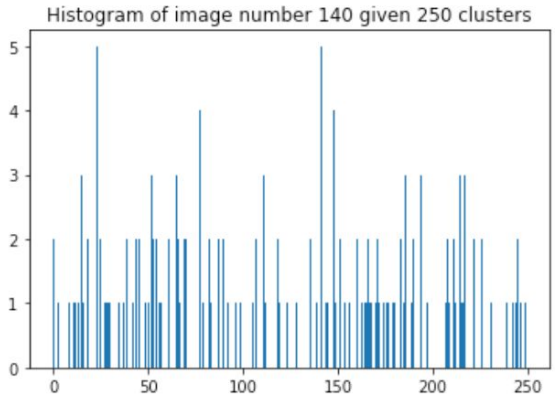


3.2 GRID1 & SIFT Kmean: K=250

Sample Image Divided 3*3 Boxes.

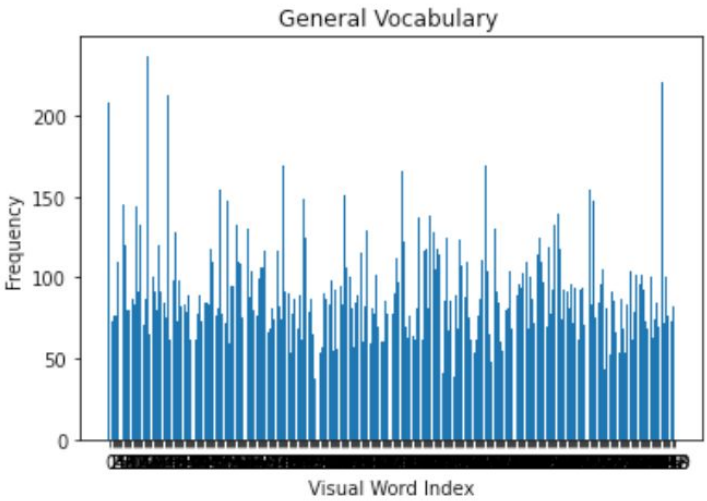


Histogram Evalation of Face Image



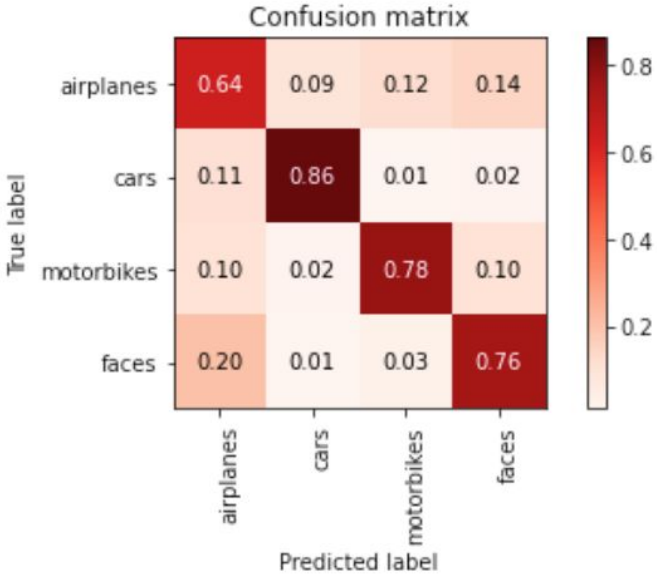

```
<matplotlib.image.AxesImage at 0x13a2d2890>
```

Total Image Histogram with K=250:

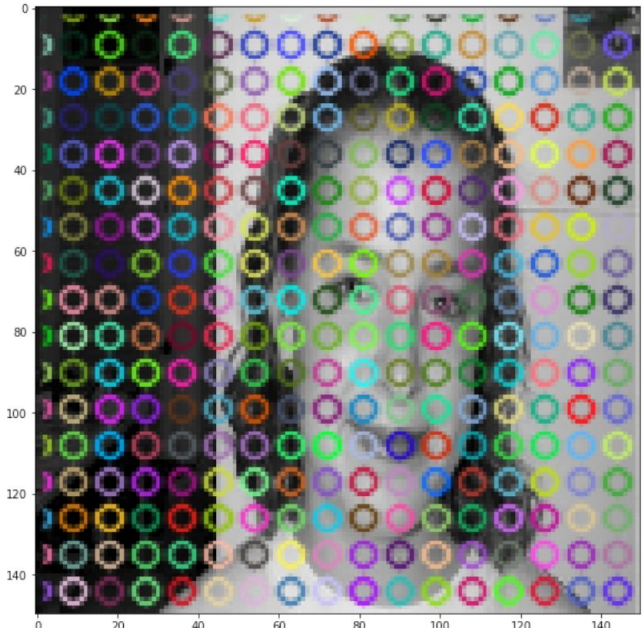


Confusion Matrix:

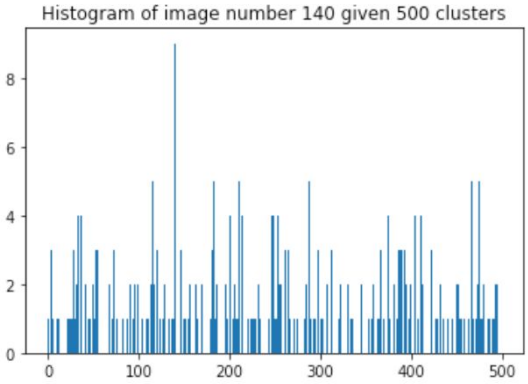
The Prediction accuracy of this model is 76.50%



3.3 GRID1 & SIFT Kmean: K=500

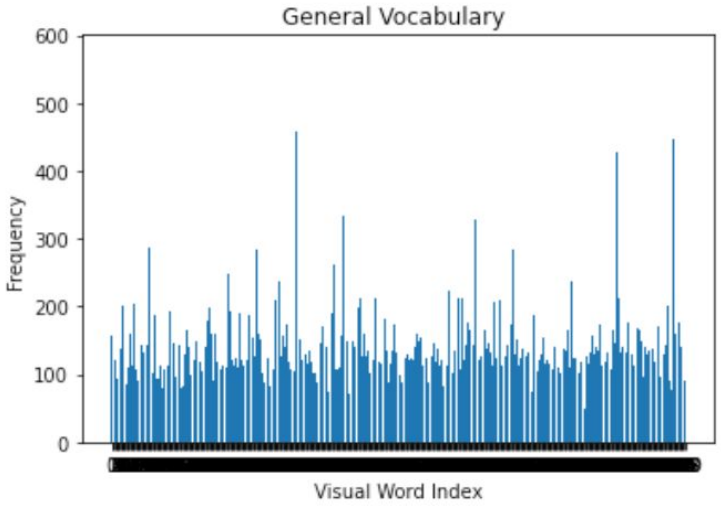


Histogram of the Sample Image:



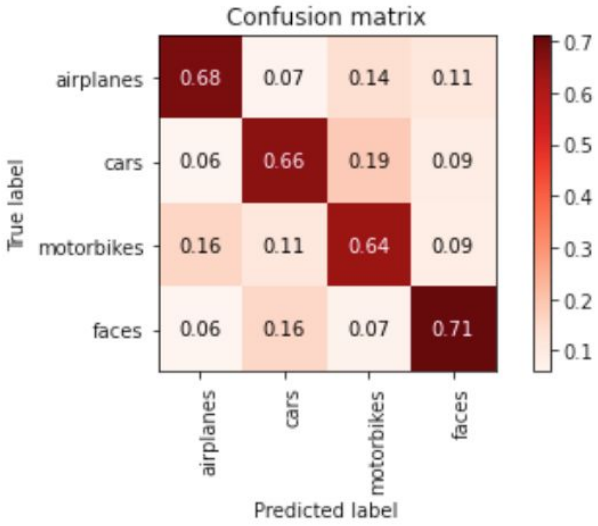

```
<matplotlib.image.AxesImage at 0x1377f71d0>
```

Total Image Histogram with K=500:



Confusion Matrix:

The Prediction accuracy of this model is 67.50%

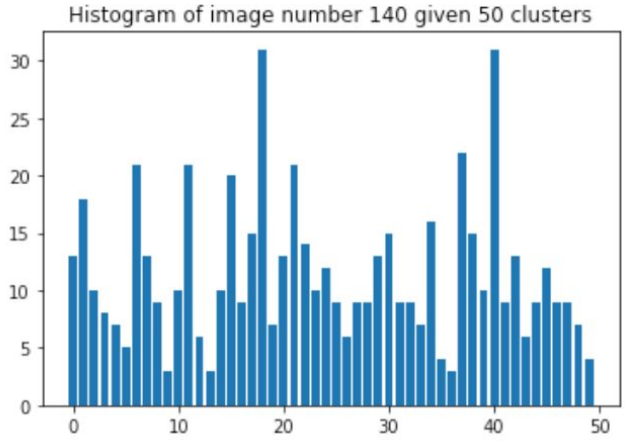


4. Experiment 2 Outputs

4.1 GRID2 & SIFT Mean: K=50

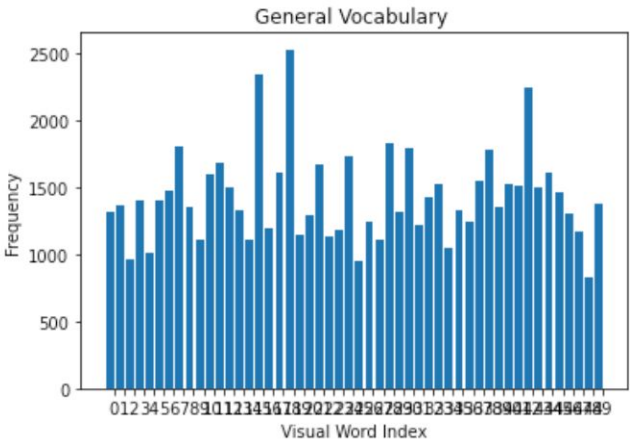
Sample Image Divided 6*6 Boxes.

Histogram of the Sample Image:



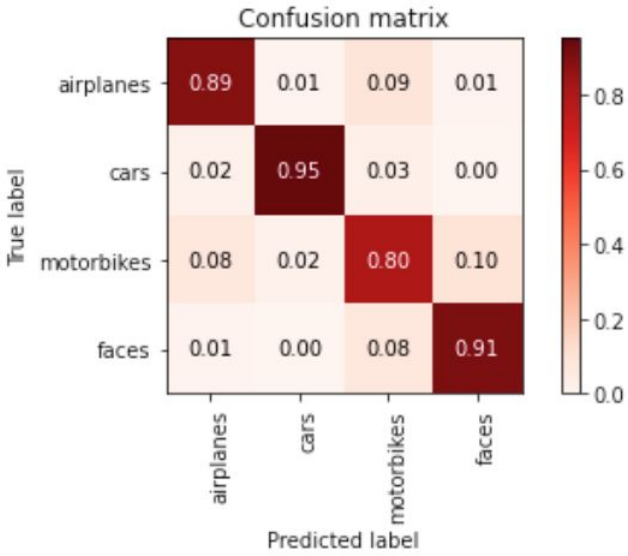

```
<matplotlib.image.AxesImage at 0x1315523d0>
```

Total Image Histogram with K=50:



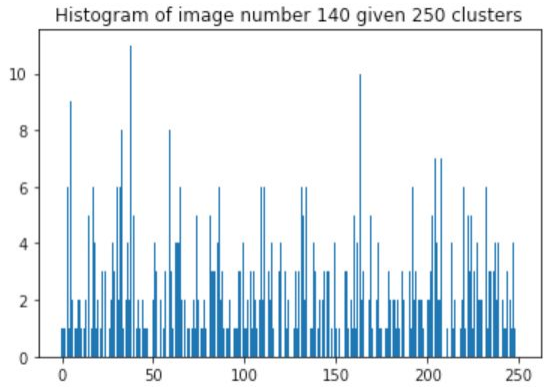
Confusion Matrix:

The Prediction accuracy of this model is 88.75%

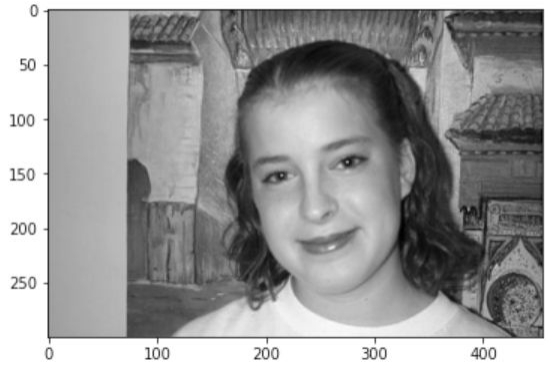


4.2 GRID2 & SIFT Mean: K=250

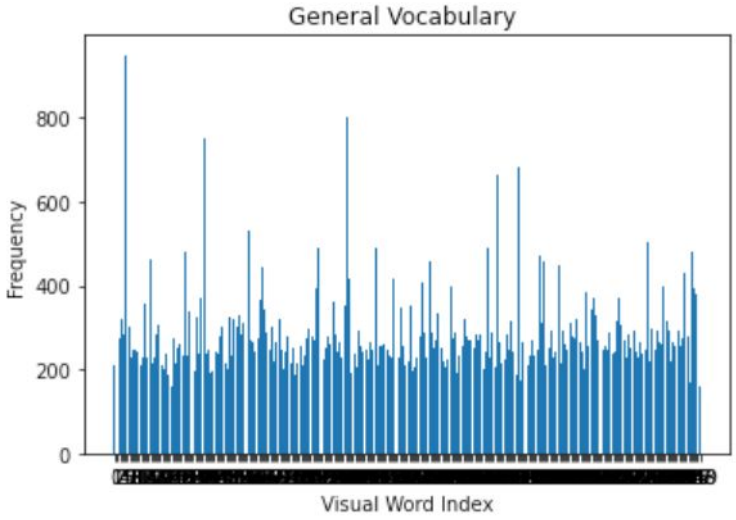
Histogram of the Sample Image:




```
<matplotlib.image.AxesImage at 0x12eb4e5d0>
```

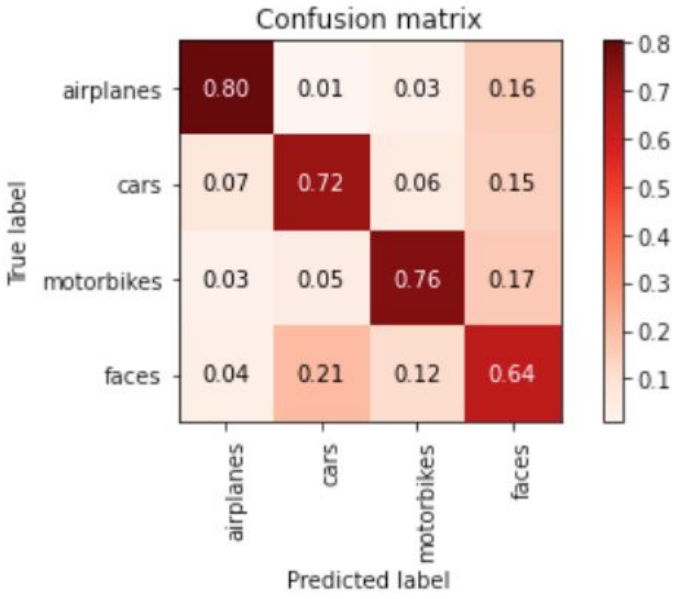


Total Image Histogram with K=250:



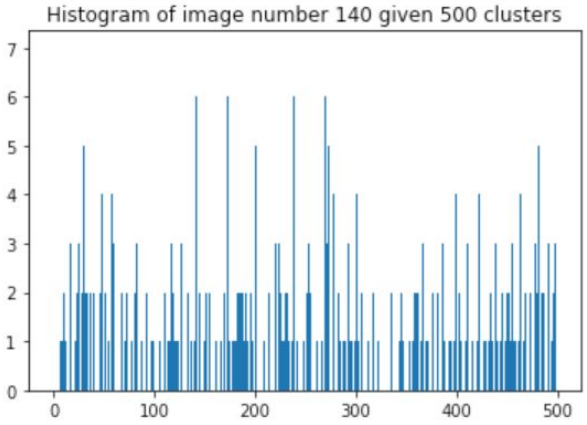
Confusion Matrix:

The Prediction accuracy of this model is 73.75%



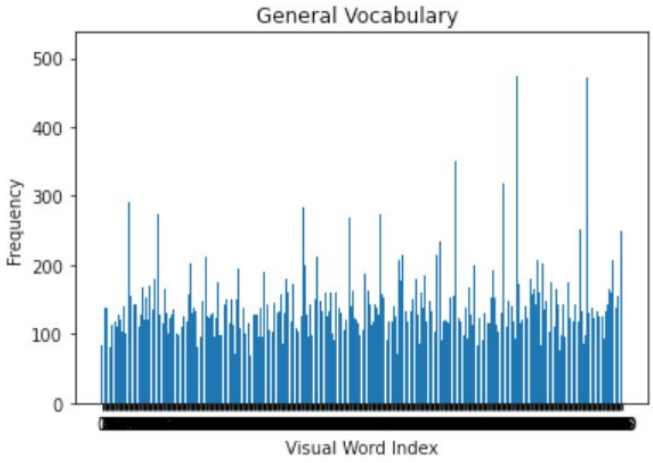
4.3 GRID2 & SIFT Mean: K=500

Histogram of the Sample Image:



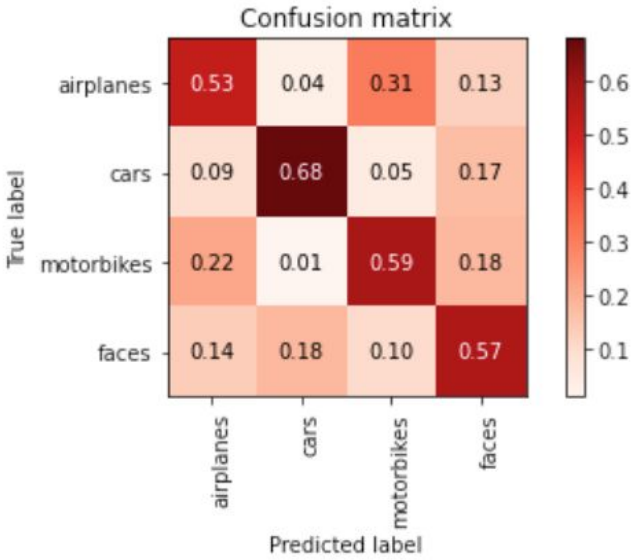

```
<matplotlib.image.AxesImage at 0x135a28090>
```

Total Image Histogram with K=500:



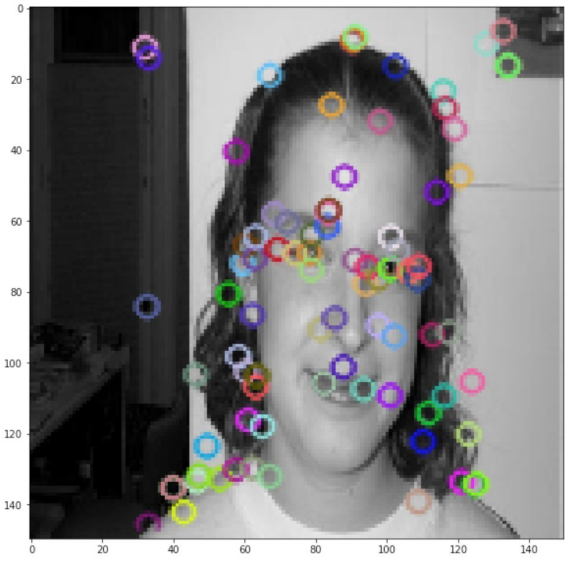
Confusion Matrix:

The Prediction accuracy of this model is 59.50%

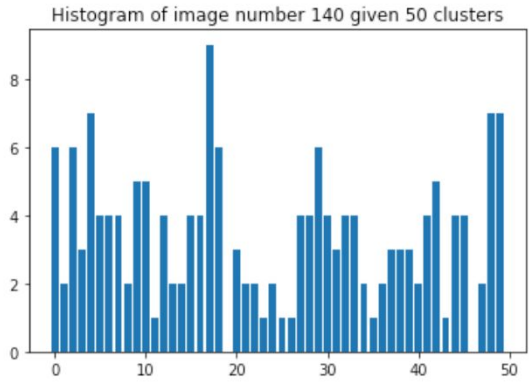


5. Experiment 3 Outputs

5.1 KeyPoint & SIFT Mean: K=50

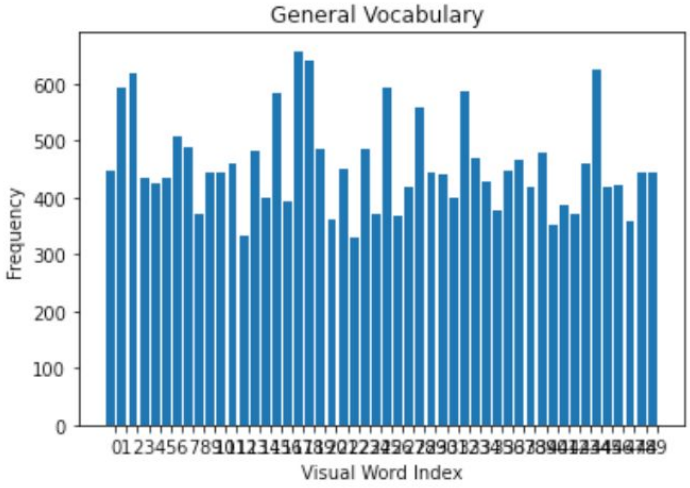


Histogram of the Sample Image:



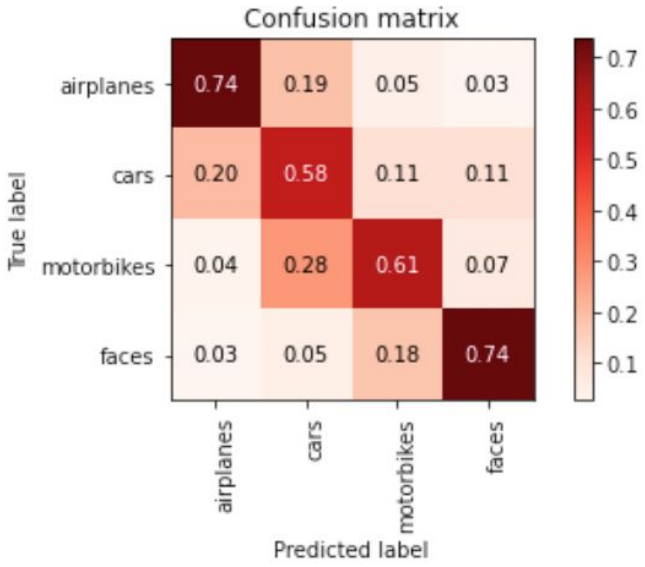

```
<matplotlib.image.AxesImage at 0x13308dd50>
```

Total Image Histogram with K=50:

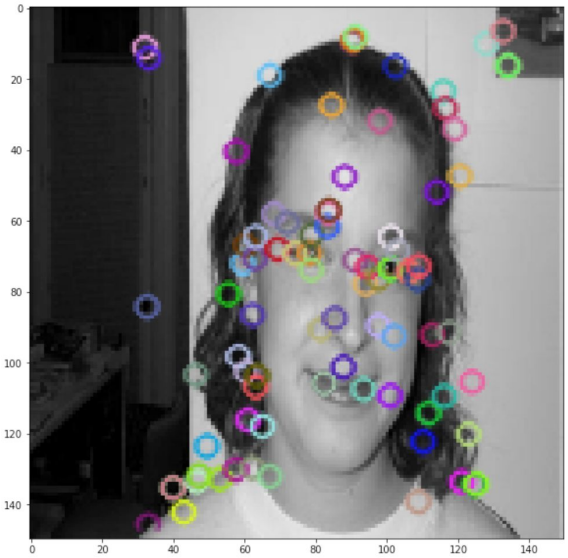


Confusion Matrix:

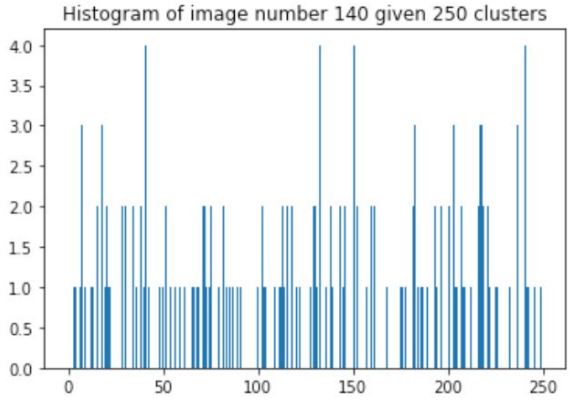
The Prediction accuracy of this model is 67.50%



5.2 KeyPoint & SIFT Mean: K=250

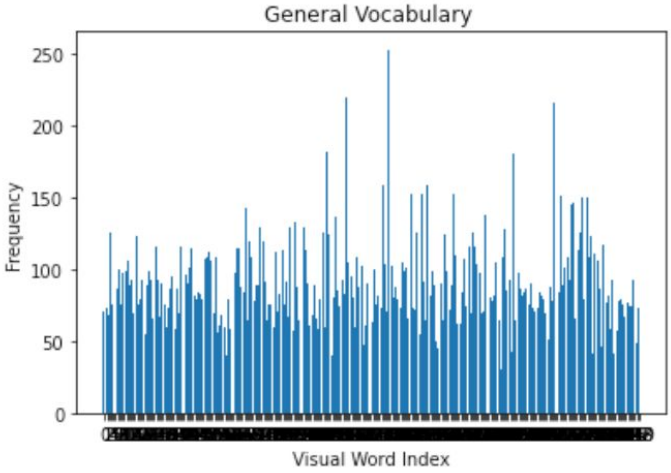


Histogram of the Sample Image:



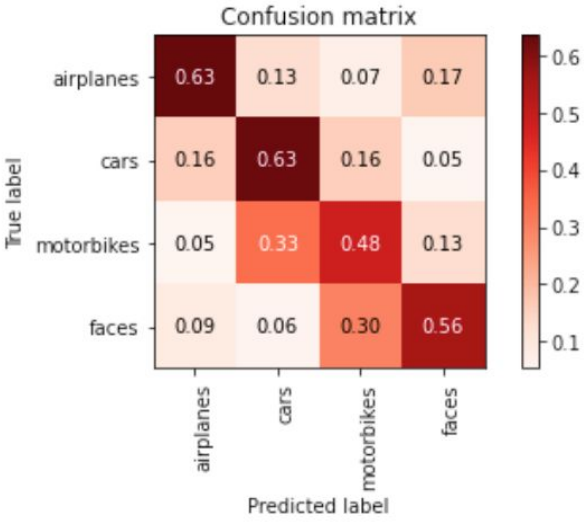

```
<matplotlib.image.AxesImage at 0x13120f8d0>
```

Total Image Histogram with K=250:

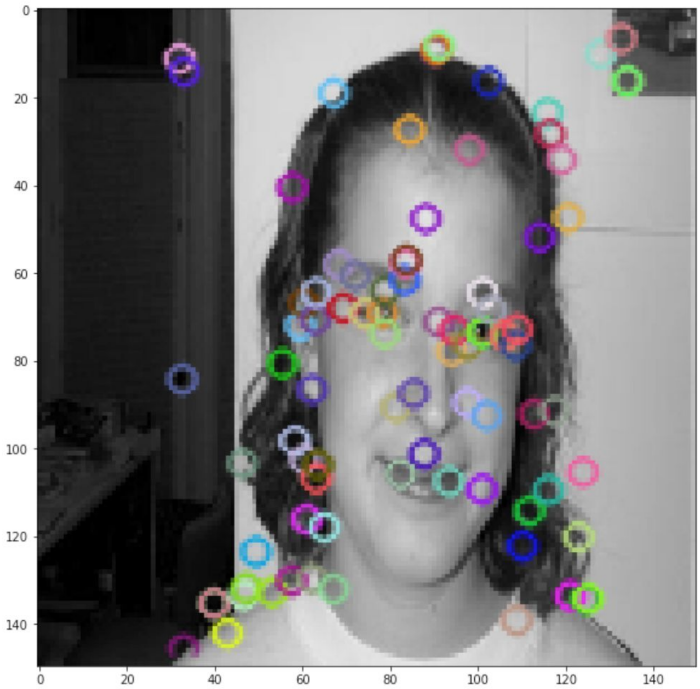


Confusion Matrix:

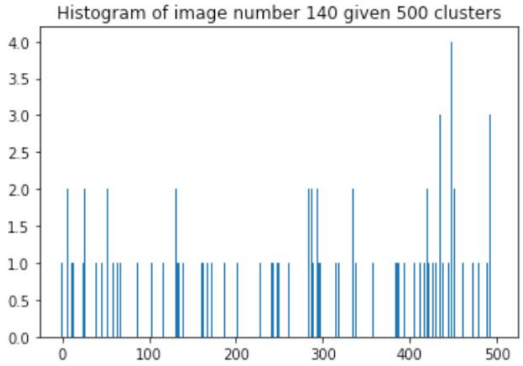
The Prediction accuracy of this model is 57.50%



5.3 KeyPoint & SIFT Mean: K=500

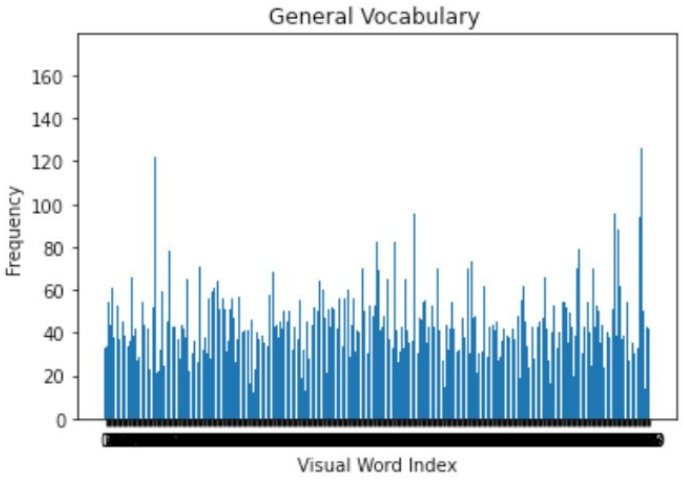


Histogram of the Sample Image:



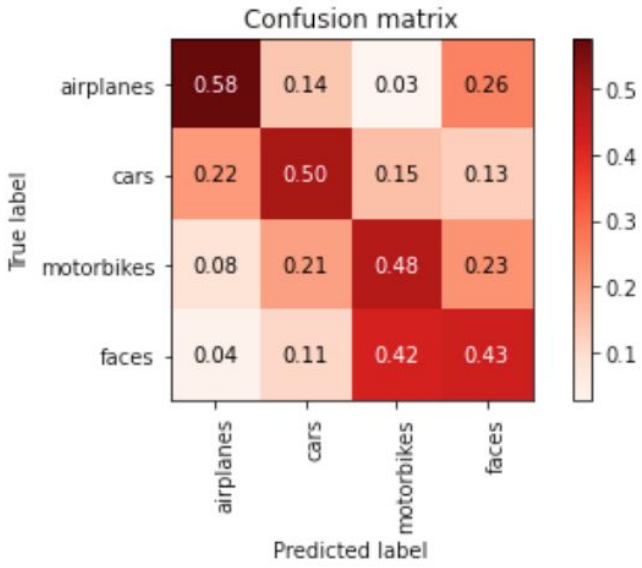

```
<matplotlib.image.AxesImage at 0x12dc7b4d0>
```

Total Image Histogram with K=500



Confusion Matrix:

The Prediction accuracy of this model is 50.25%



Classifier Training Result Using SVM:

In [49]:

```
warnings.filterwarnings("ignore")
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
#Standardizing the histogram matrix
train_featn = scaler.fit_transform(train_feat)
test_featn = scaler.fit_transform(test_feat)

label3 = test_label.copy()

start = time.process_time()
clf = svm.LinearSVC(random_state=0, C = 4.6)    #Using lambda = 4.6
clf.fit(train_featn, train_label)
time5 = time.process_time() - start
print("Time Consumption in Training (Bag of SIFT Representation + one-vs-all SVMs) : ",time5 , "seconds")

start = time.process_time()
pred3 = clf.predict(test_featn)
time6 = time.process_time() - start
print("Using Lambda = 4.6")
print("The Prediction accuracy of this model is {:.2f}%".format(accuracy_score(label3,pred3)*100))
print("Time Consumption in Prediction (Bag of SIFT Representation + one-vs-all SVMs) : ",time6 , "seconds"
```

```
Time Consumption in Training (Bag of SIFT Representation + one-vs-all SVMs) :  0.35426599999999553 seconds
Using Lambda = 4.6
The Prediction accuracy of this model is 80.50%
Time Consumption in Prediction (Bag of SIFT Representation + one-vs-all SVMs) :  0.004112000000020544 seconds
```

Mean SHIFT - K Value Predidiction:

```
:
1  bandwidth = estimate_bandwidth(train_x, n_jobs=-1)
2  ms = MeanShift(bandwidth=bandwidth, bin_seeding=True)
3  ms.fit(train_x)
4  labels = ms.labels_
5  cluster_centers = ms.cluster_centers_
6
7  #Predicting k number by mean sift
8  labels_unique = np.unique(labels)
9  n_clusters_ = len(labels_unique)
10 print("Number of estimated clusters : %d" % n_clusters_)
```

```
Number of estimated clusters : 1
```

Mean Shift predict k value as 1 using train dataset.

6. Results and Discussion

In this homework, we consider features of images as raw pixels. This is the simplest form of features we can get from an image.While doing this, also we are expected to use Support Vector Machine, different image grid size and keypoint parameters to feature exraction.

Training images are classified into 4 classes which are ‘Airplanes’, ‘Cars’, ‘Faces’ and ‘Motorbikes’. Each classes has 40 train images and 100 test images. However, original dataset images did not divided train and tes folder as same name, therefore, Before starting this project, I adjusted to train and test folder images name.

Beside of those, in this homework we are expected to cluster image descriptors into 50, 250 and 500 using KMeans. Moreover, homework divided 2 experiment, in the first experiment, I tested image by dividing image two different grid size. I chosed 3*3 and 6*6 division size. When we look at the results, Grid & SIFT combination gives higher result than KeyPoint & SIFT results. Furthermore, when we use original image size, clustering them takes too much time. In order to save from time, I resized all input images into (150 x 150). By doing this, clustering image

descriptors takes much less time. However it is a trade-off, when we resize images, number of descriptors for each image and accuracy of testing decreases, but we save from time.

Additionally, in this homework, I use Linear kernels for SVM classifier, which are linea. As can be seen from the outputs above. When using SVM, we have a hyperparameter C that is tunable, which can yields different performances for the prediction on the testing set. Thus, it's necessary to try different C and report the one that yields the best performance. The accuracy varies from 70% to 80% depending on the value of C used in the classifier. In problem 3, with C = 4.6 (roughly) results in the accuracy of 67.73%.

Furthermore, in this homework, we cluster image descriptors into 50, 250 and 500 clusters. For different cluster sizes, our training accuracy differs. Higher cluster size means higher training accuracy. From this point of view, we are expected to say that if we increase cluster size, our testing accuracy increases too. But according to outputs listed above, testing accuracy sometimes decreases, sometimes increases. Of course, it can sourced from KMeans, descriptors etc.

Finally, thanks to this homework, I had hand-on experience on image classification using Support Vector Machines. Also, I had a chance to think on how to increase accuracy, or how to decrease computation time while doing this homework.