Computer Vision
Assignment 4
Samarth Ramesh

Summary and Design Choices

Our first task was to build a dictionary of visual words. The first step to this is extracting features from each image in our training set. We decide to perform this feature description using the SIFT algorithm. The SIFT algorithm on average extracted 370 keypoints from each image. Now we would like to express our large set of keypoints as a more usable set of visual words that would serve as our vocabulary. We achieve this by K Means Clustering. We choose a vocabulary size of 200, and the algorithm took about 45 minutes to return the clusters that would operate as our visual word vocabulary.

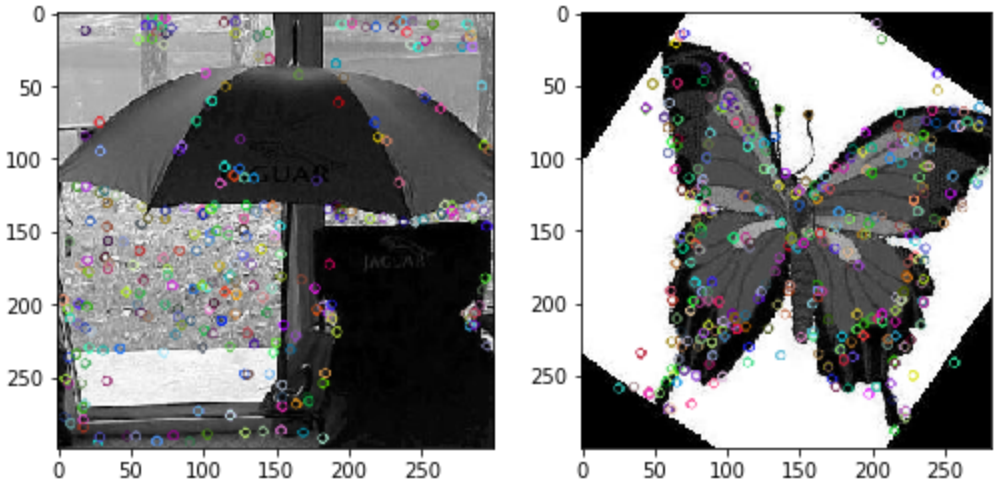
Now for each image, we built a histogram of visual words. For each keypoint in each image, we used the kmeans from the previous step to find the nearest visual word and added one to the corresponding entry in the histogram.

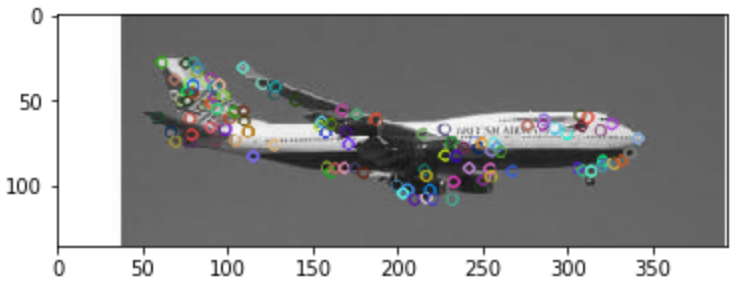
(Using kmeans.predict() is much faster than computing the closest visual word manually). Since the images were of different sizes, some images had more keypoints than others so we had to L_1 normalize the histograms.

Now we simply considered the histograms as the features for each image and performed classification using the K Nearest Neighbor Classifier and the Support Vector Machine Classifier.

Output of Interest Point Detector

We used the SIFT feature detector from the OpenCV library to extract keypoints.

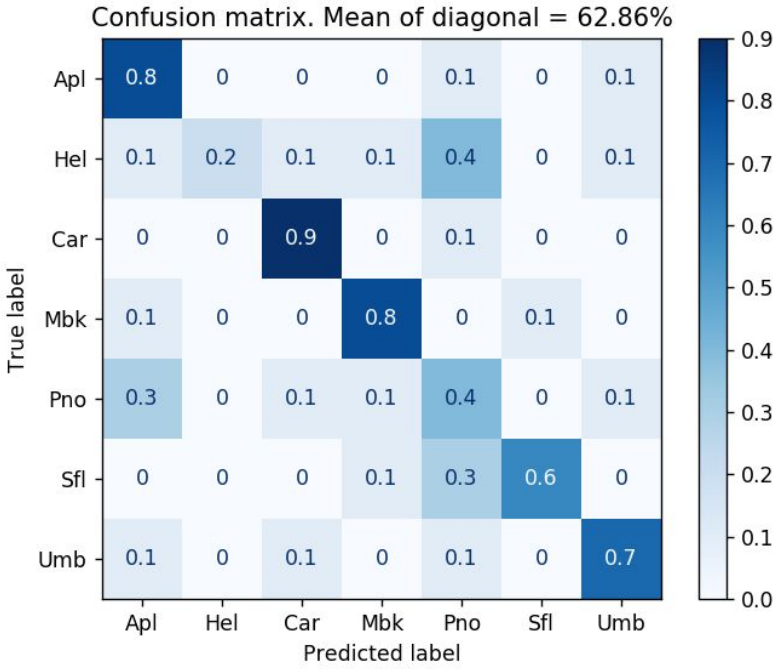




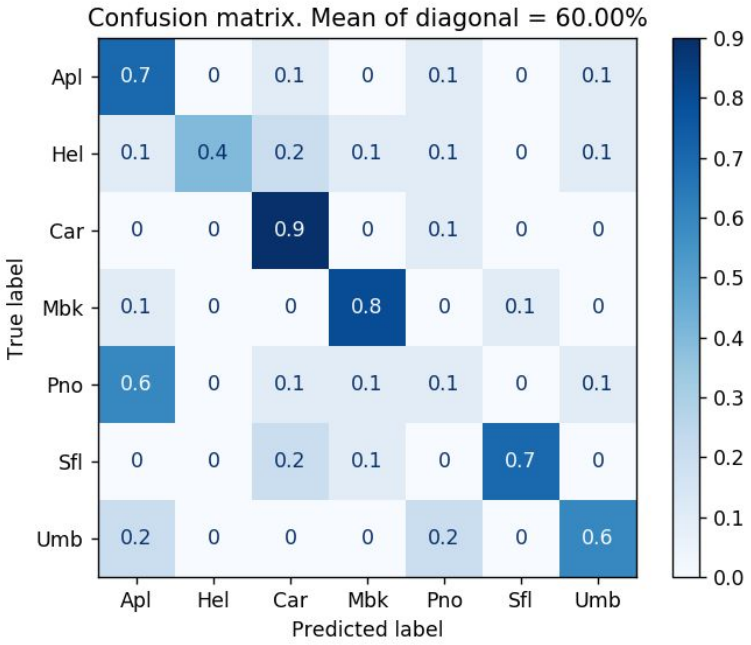
Results of K Nearest Neighbor Classifier

We experimented with a few different values of K and obtained the following results.

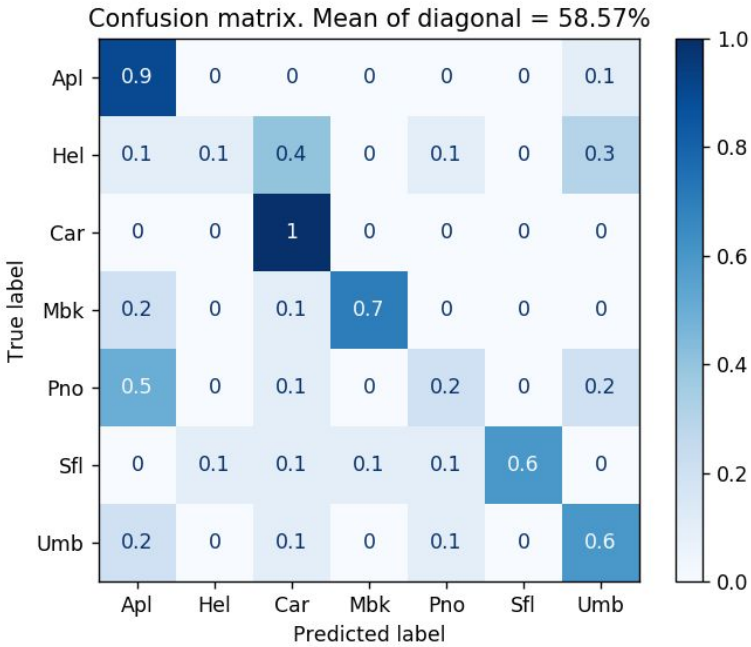
K=1



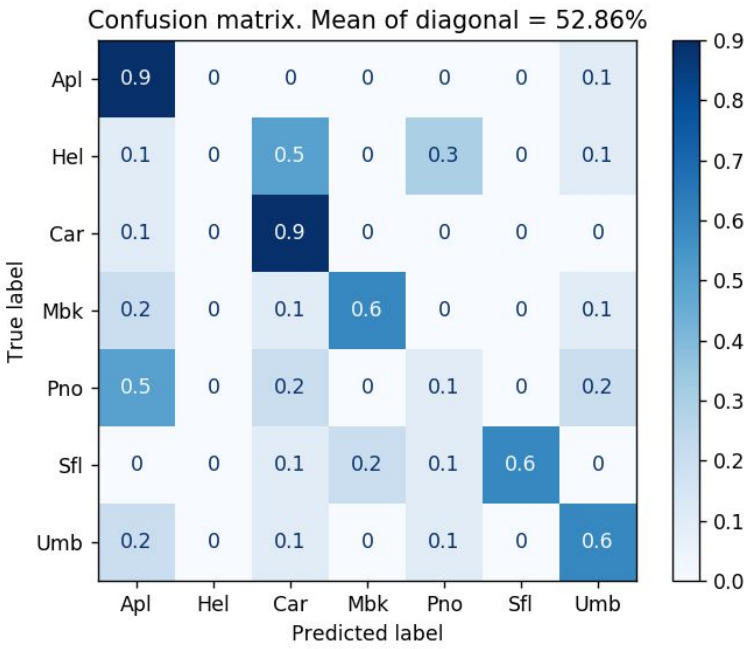
K = 5


Confusion matrix. Mean of diagonal = 60.00%
Apl 0.7 0 0.1 0 0.1 0 0.1
Hel 0.1 0.4 0.2 0.1 0.1 0 0.1
Car 0 0 0.9 0 0.1 0 0
Mbk 0.1 0 0 0.8 0 0.1 0
Pno 0.6 0 0.1 0.1 0.1 0 0.1
Sfl 0 0 0.2 0.1 0 0.7 0
Umb 0.2 0 0 0 0.2 0 0.6
Apl Hel Car Mbk Pno Sfl Umb
True label
Predicted label
0.9 0.8 0.7 0.6 0.5 0.4 0.3 0.2 0.1 0.0

K = 10


Confusion matrix. Mean of diagonal = 58.57%
Apl 0.9 0 0 0 0 0 0.1
Hel 0.1 0.1 0.4 0 0.1 0 0.3
Car 0 0 1 0 0 0 0
Mbk 0.2 0 0.1 0.7 0 0 0
Pno 0.5 0 0.1 0 0.2 0 0.2
Sfl 0 0.1 0.1 0.1 0.1 0.6 0
Umb 0.2 0 0.1 0 0.1 0 0.6
Apl Hel Car Mbk Pno Sfl Umb
True label
Predicted label
1.0 0.8 0.6 0.4 0.2 0.0

K = 20

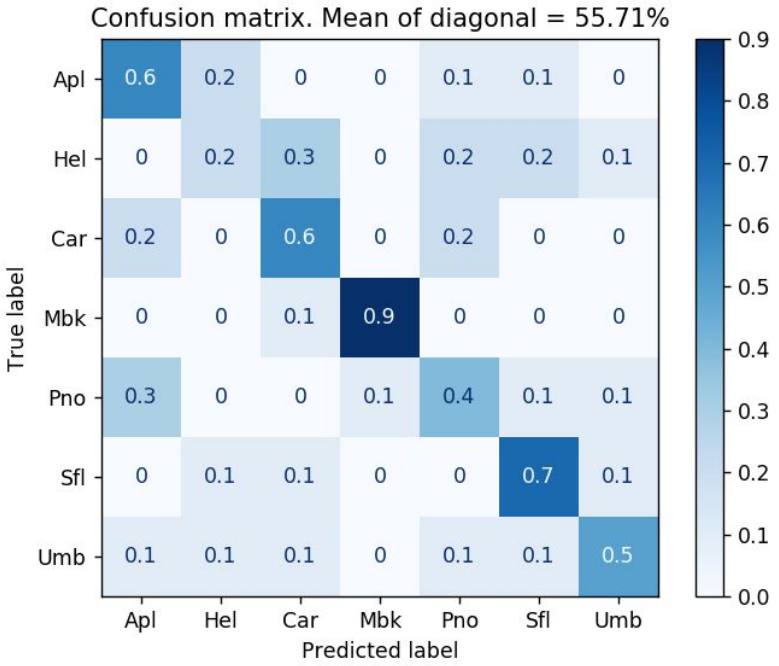


We observe that K = 1 has the best performance of 62.86%.

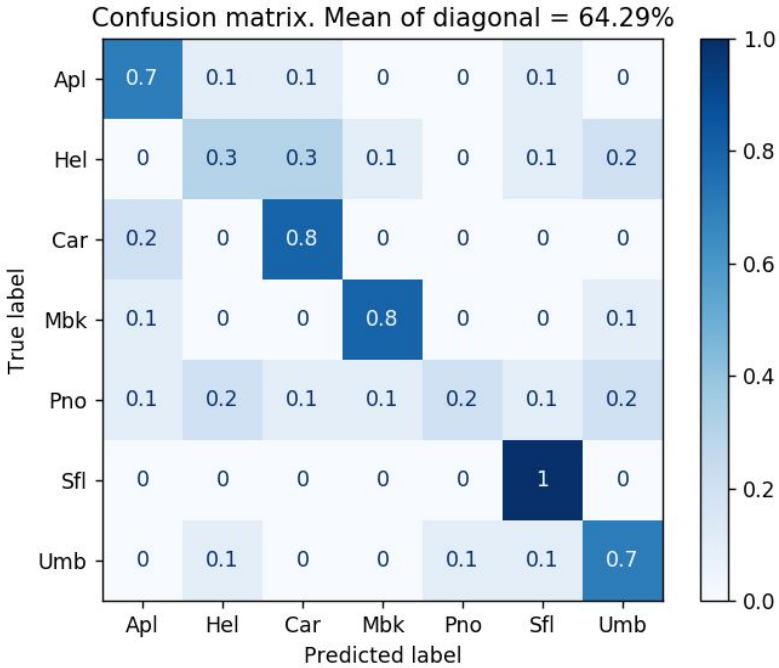
Results of SVM Classifier

We experimented with a few different values of the regularization parameter lambda and obtained the following results.

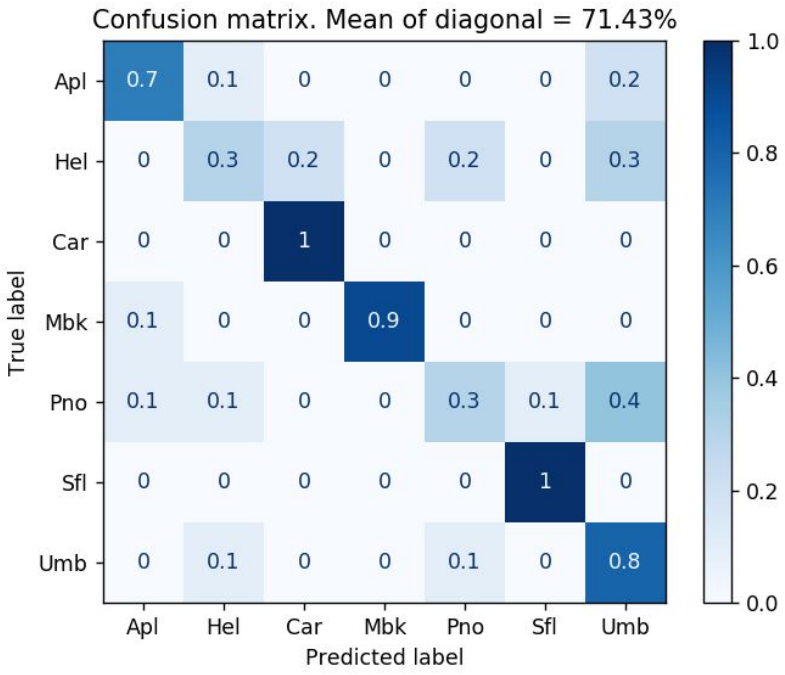
lambda = 10



lambda = 100



lambda = 500



We observe that lambda = 500 has the best performance of 71.43%.

Overall we observe that the SVM Classifier with lambda = 500 has the best performance with an accuracy of 71.43%.

We could justify this by considering that our categories are fairly distinct in appearance. In this kind of scenario, an

algorithm with a higher bias like SVM would do better than an algorithm with higher variance like KNN.