Lab Report 10 - Image Categorization

Manuel Galliker 14-921-969
manuelga@student.ethz.ch

December 13, 2019

1 Local Feature Extraction

Here the feature extraction is really simple and straight forward. The function grid points simply divides the image into a rectangular grid using the provided parameters. Then a histogram of oriented gradients is calculated for every cell in the function descriptors hog.

2 Codebook Construction

As described in the section before, features can be extracted for each image in our training set. This results in a very large number of features. It is therefore desirable, to reduce this feature space to only the relevant features. To to this, the nearest neighbor clustering using kmeans was applied. Hereby, the number of desired images after the clustering can be specified with the parameter k.
using the provided function visualize codebook, the codebook can be shown:

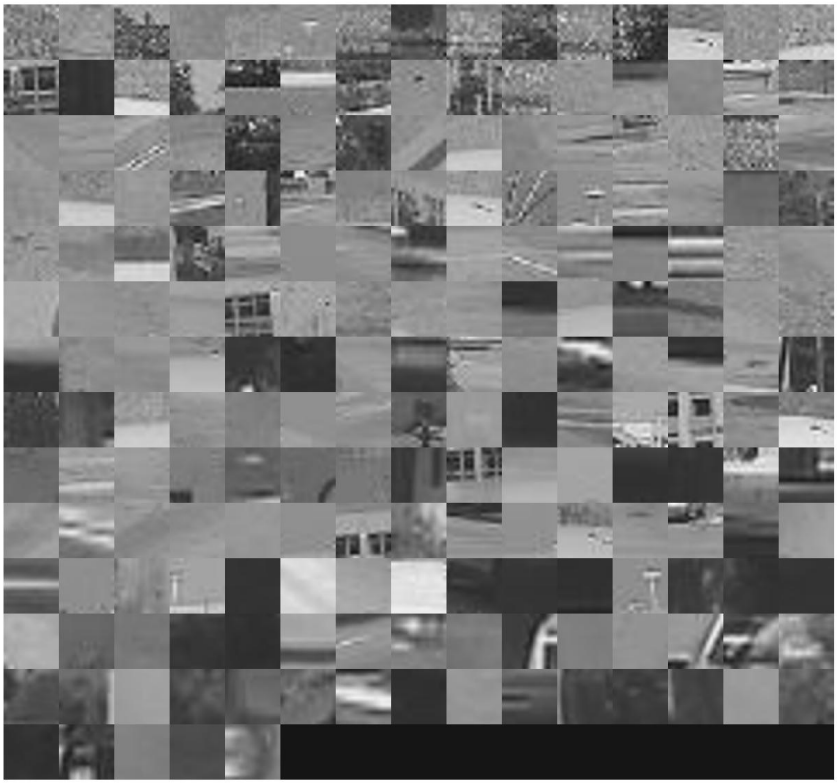

Figure 1: Codebook with $k = 200$

Note that the black squares on the bottom right do not represent features but are just there to complete the rectangle.

3 Bag of Words

Each image feature is assigned to the nearest codebook feature in the function bow histogram using knnsearch for nearest neighbour. The function create bow histograms is used to create the bag of words for all images.

4 Nearest Neighbours Classifier

The nearest neighbours classifier is fairly straight forward. Here we compare a selection of features (equal in number to the features in the codebook) to the codebooks with and without cars. if the nearest neighbour is a car we assign the car label, otherwise not.

5 Bayesian Classifier

The bayesian method is more complicated and relies on calculating probabilities instead of one nearest neighbour. In the function computeMeanStd the mean and standart deviation of the features are calculated. The goal is to estimate the probability of a car beeing present given the the histogram of the image. To solve this bayes' theorem was used under the assumption that the features are uncorrelated. The algorithm was implemented into bow recognition bayes as described. Should there be a probability of more than 50% that the image contains a car given the histogram it's classified as such.

6 Results

6.1 25 Codebook Descriptors

Figure 2: Codebook with $k = 25$

Nearest neighbour classifier:
Percentage of correctly classified images:90.909%

Bayesian classifier:
Percentage of correctly classified images:87.879%

6.2 50 Codebook Descriptors

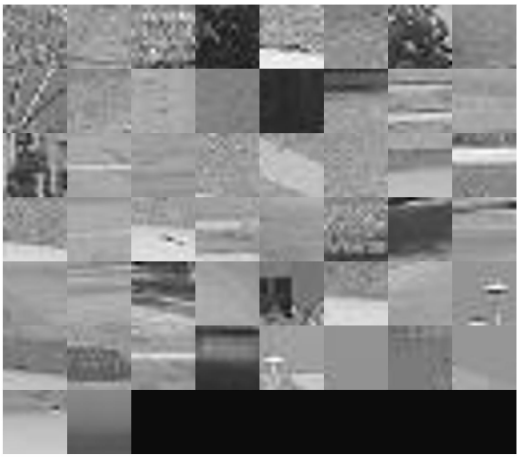

Figure 3: Codebook with $k = 50$

Nearest neighbour classifier:
Percentage of correctly classified images:94.949%

Bayesian classifier:
Percentage of correctly classified images:96.97%

6.3 100 Codebook Descriptors

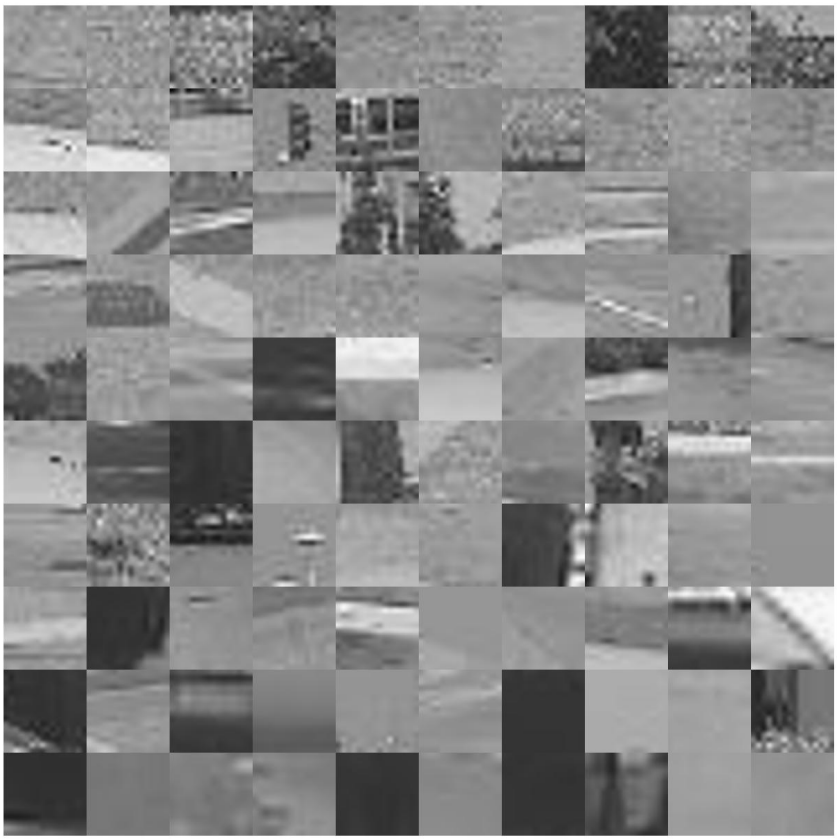

Figure 4: Codebook with $k = 100$

Nearest neighbour classifier:
Percentage of correctly classified images:93.939%

Bayesian classifier:
Percentage of correctly classified images:96.978%

6.4 200 Codebook Descriptors

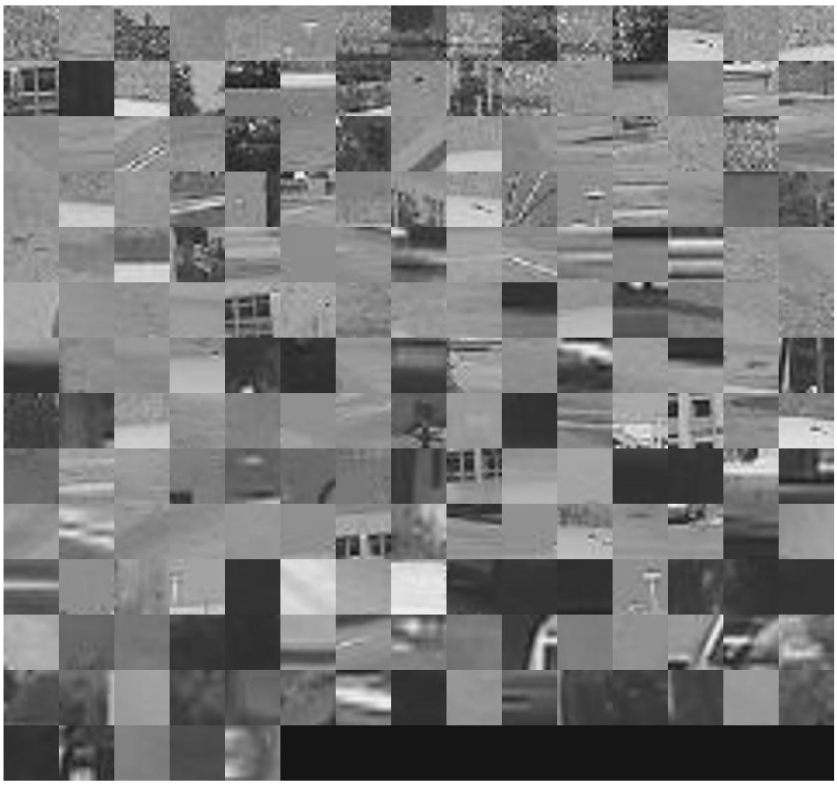

Figure 5: Codebook with $k = 200$

Nearest neighbour classifier:
Percentage of correctly classified images:91.919%

Bayesian classifier:
Percentage of correctly classified images:98.99%

6.5 400 Codebook Descriptors

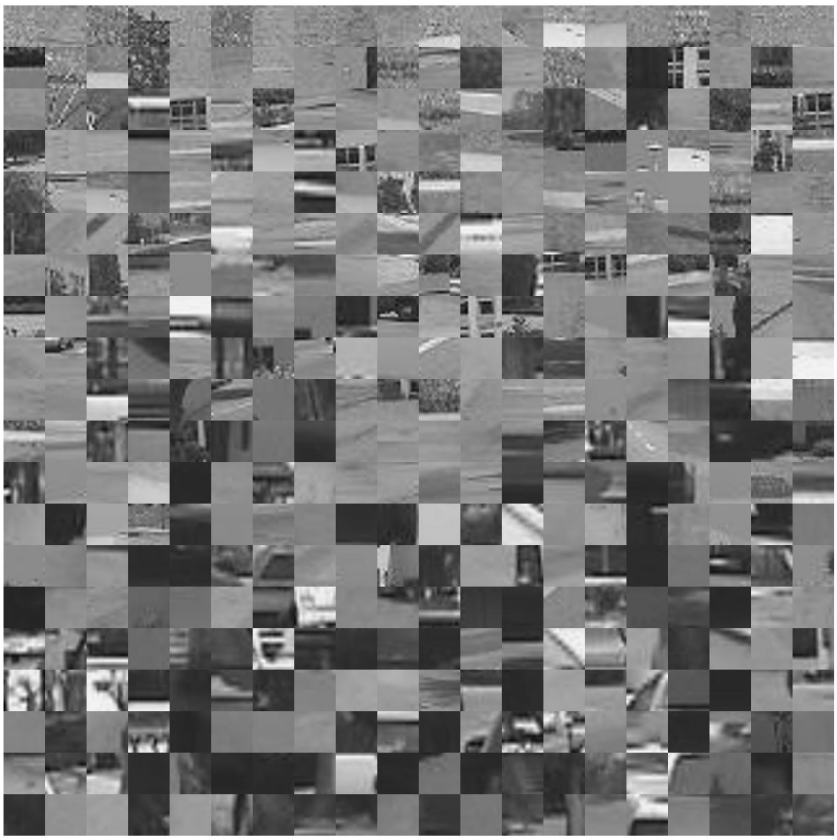

Figure 6: Codebook with $k = 400$

Nearest neighbour classifier:
Percentage of correctly classified images:94.949%

Bayesian classifier:
Percentage of correctly classified images:100%

7 Discussion

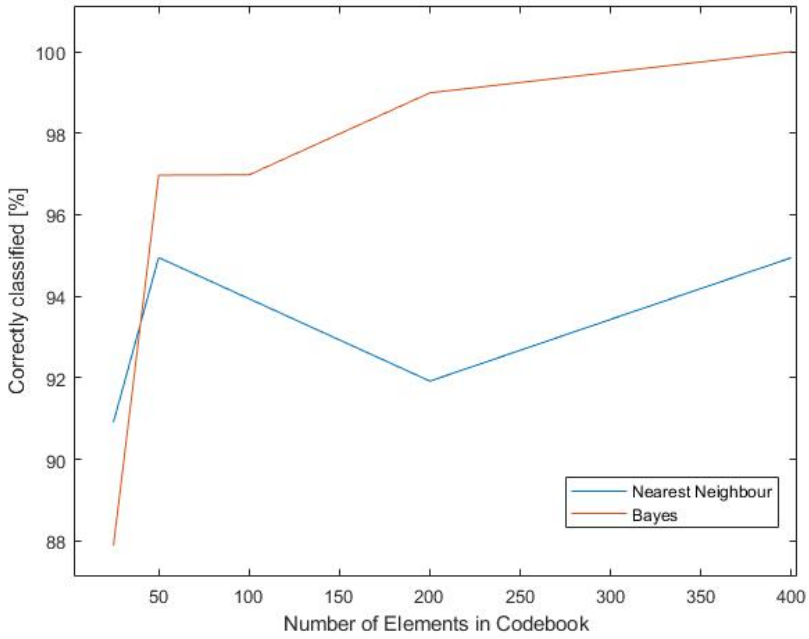

Figure 7: Performance of the two algorithms

As can be seen the Bayesian classifier outruns the nearest neighbour classifiers for all k except of at 25. This makes sense, since we feed our "model" with some prior belief about the world. As expected, the accuracy of the Bayesian classifier increases steadily which increasing Codebook size, this makes sense, since we get a more accurate description of the training data. Strangely this seems not to be strictly the case for the nearest neighbour.
Since theese measures vary a bit from run to run more testing would be needed for a statistically relevant sample.