Computer Vision Exercise 10: Image Categorization

Soomin Lee (leesoo@student.ethz.ch)

December 20, 2020

1. Local Feature Extraction

Feature detection In this task, we take points on a grid as feature points. One thing we need to consider is that we will construct a descriptor based on the surrounding cells of each point, and hence we need a margin around the edge of an image when defining the grid. Accordingly, we define the grid with a margin of 8 around the edge and extract $10 \times 10 = 100$ feature points.

Feature description We use the histogram of oriented gradients (HOG) descriptor. The descriptor is defined over 4×4 cells around a grid point, and each cell has again 4×4 pixels. We create a histogram with 8 bins per each cell, using the image gradients at 16 pixels in each cell. Therefore, the descriptor has $4 \times 4 \times 8 = 128$ dimensions per each feature point.

2. Codebook construction

We construct a visual vocabulary, or a *codebook*, with the local descriptors we obtained from the training images. The K-means++ clustering algorithm is used, and the function is provided by Matlab. The difference from the classical K-means algorithm is the initialization of the centers. It chooses each center point sequentially based on weighted probability distribution, computed using the distance from the points that are already chosen. As the performance of K-means algorithm can depend heavily on the initialization, it is a method to improve the initialization, and the rest of the algorithm remains identical. The resulting codebooks are shown in Figure 1.

3. Bag-of-words image representation

For each image, we compute a bag-of-words histogram. The histogram counts each visual word that is present in an image. To count this, we assign the descriptors to the cluster centers, or *visual words*, and count how many descriptors are assigned to each cluster.

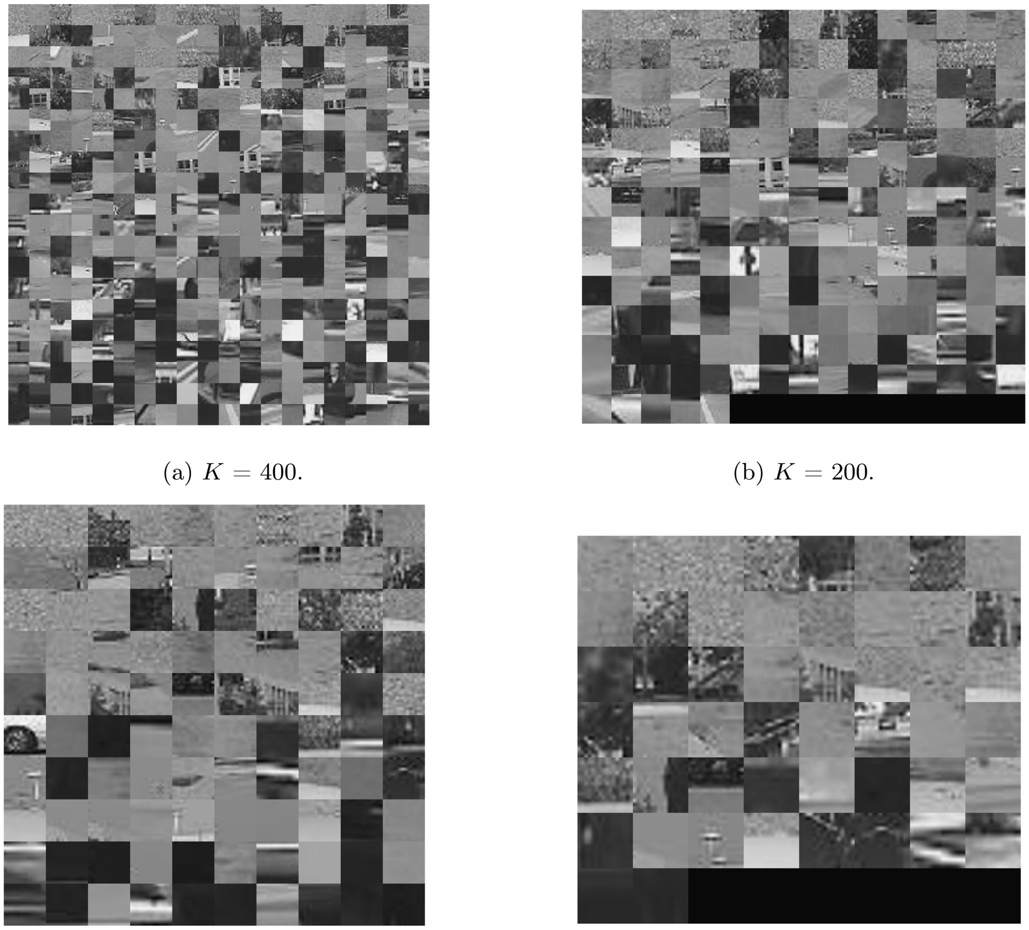
4. Nearest Neighbor Classification

With the bag-of-words image representations we obtained, we can now classify images. First, we classify it using the nearest-neighbor algorithm. Given a test image, we compute the distance to the nearest neighbor both in the positive training examples and the negative training examples. If the distance to the nearest neighbor in the positive training examples is smaller than the distance to the nearest neighbor in the negative training examples, label 1 (positive) is assigned. Otherwise, label 0 (negative) is assigned.

5. Bayesian Classification

As an alternative to the nearest neighbor classifier, we classify images in a probabilistic manner using Bayes' theorem. The posterior probability of a car given a histogram can be formulated as in Equation 1, and $P(!Car|hist) = 1 - P(Car|hist)$.

$$P(Car|hist) = \frac{P(hist|!Car) * P(Car)}{P(hist|!Car) * P(Car) + P(hist|!Car) * P(!Car)} \tag{1}$$



(a) $K = 400$.

(b) $K = 200$.

(c) $K = 100$.

(d) $K = 50$.

Figure 1: Codebooks with different number of cluster centers, denoted as K.

We can further simplify the calculation of the posterior probabilities because only the relative values of $P(Car|hist)$ and $P(!Car|hist)$ matter in the end to assign a label. Therefore, we can leave out the denominator and Equation 1 becomes Equation 2.

$$P(Car|hist) \propto P(hist|Car) * P(Car) \tag{2}$$

Furthermore, we assume $P(Car) = P(!Car) = 0.5$ for simplicity. As conclusion, we only need to calculate $P(hist|Car)$ and $P(hist|!Car)$ to decide whether $P(Car|hist)$ is bigger than $P(!Car|hist)$.

Then we make another assumption that the distribution of the counts for each visual word follows a normal distribution. The mean μ and the standard deviation σ are estimated from the training examples. Treating each sample as independent, $P(hist|Car)$ can be formulated as follows:

$$logP(Car|hist) \propto \sum_{i=1}^{K} logP(U_i|N(\mu_i, \sigma_i)) \tag{3}$$

Logarithm is used for numerical stability and U_i denotes the i-th value of the histogram of each test image. The same thing applies to $P(!Car|hist)$, and since $P(Car|hist) + P(!Car|hist) = 1$, we can decide if $P(Car|hist) > 0.5$ by comparing the relative magnitude of the two values.

In some cases, we cannot determine the mean and the variance because there is no data to do so. In this case, we skip the case which is equivalent to adding $0 = log1$ to the logarithmic values in Equation 3.

6. Result

The accuracy of the nearest neighbor classifier and the Bayesian classifier is shown in Figure 2. Each data point marked with × represents one experiment, and 5 experiments are conducted for each $K = 50, 100, 200, 400$. Note that some of the data points are overlapping. The solid line shows the trend in the average accuracies over different Ks. The randomness originates from the K-means algorithm when we make codebooks.

One can observe that the accuracy of the nearest neighbor classifier does not differ much between different number of cluster centers, while the accuracy of the Bayesian classifier differs significantly. This is because of how we model the probability as a normal distribution. When K is large, it means that we often do not have enough data for each visual word to estimate the mean and the variance reliably to model the normal distribution. Therefore, the probabilities are not accurate enough to build a reliable classifier. On the other hand, when K is small, each entry of bag-of-words representations is more likely to be filled with meaningful values which lead to a better estimation of the probabilities. Accordingly, one can improve the accuracy of the Bayesian classifier if one can model the probabilities better. For instance, specifying a minimum value for the variance of each normal distribution can be helpful.

Also, having too small or too large K can have negative effects because small K will make it hard to distinguish between the examples, and large K makes the clustering less meaningful. One can think about the extreme cases where all the examples belong to one cluster or all the clusters contain only one example each. Therefore it is reasonable to assume that one should choose the right K in order to maximize the accuracy of the classifiers.

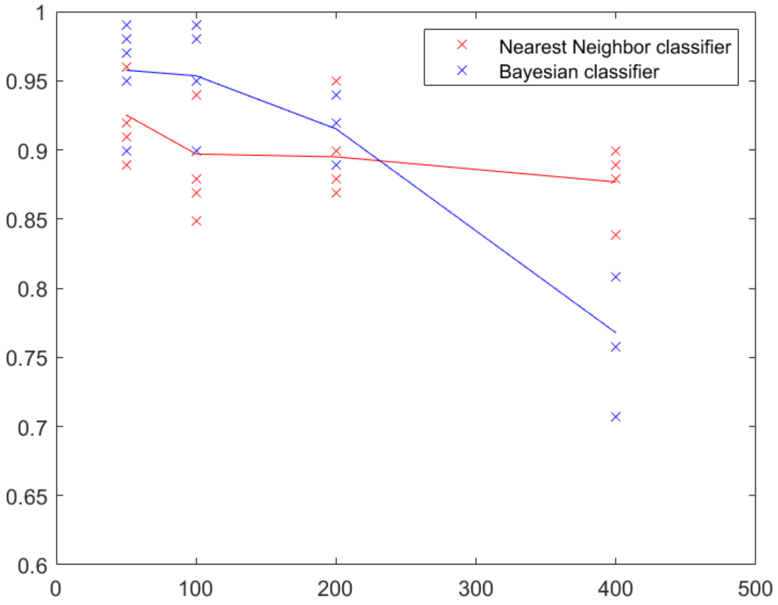


Figure 2: Accuracy for each classifier. The solid line indicates the mean value of 5 different experiments for each $K = 50, 100, 200, 400$.