Scene Recognition with Bag of Words

Yuancheng Fang (方元成), School of Computer and Science, 22221206

Implementation

I have implemented below functions:

1. Use the tiny image representation (resolution 16×16) and KNN classifier to classify images.
2. Use bags of quantized SIFT features and KNN classifier to classify images. Its workflow is, detect SIFT features, sample `vocab_size` (default is `64000`) features from them to form a vocabulary, train a kmeans cluster to generate histogram representations, and finally use KNN to classify images.
3. Analyze the influence that the vocabulary size cause to classify results.
4. Analyze different performance in different categories of both SIFT method and tiny image method.

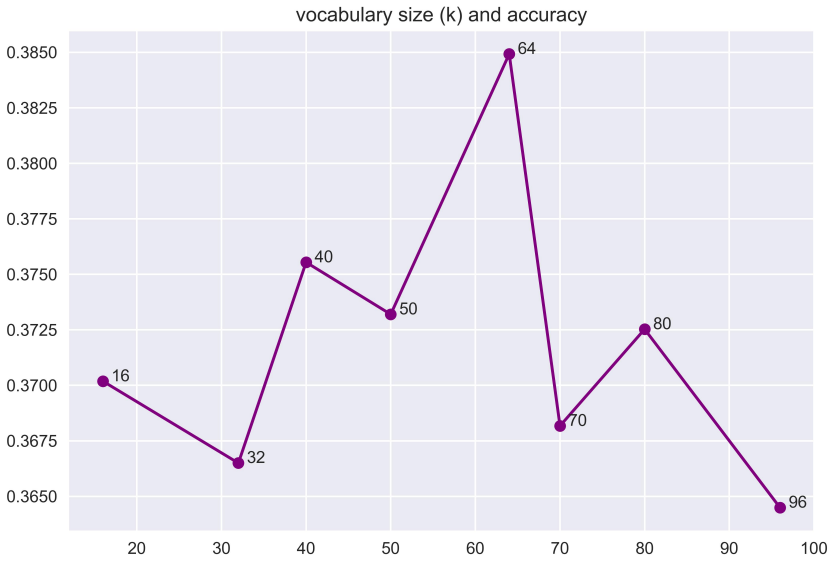
Results

The performaces of Bag of SIFT Representation (with default `vocab_size`) and Tiny Images Representation (with size 16×16) are as follows:

	SIFT	**Tiny Images**
Accuracy	38.49%	20.23%

SIFT with bag of words method is significantly better than tiny images method.

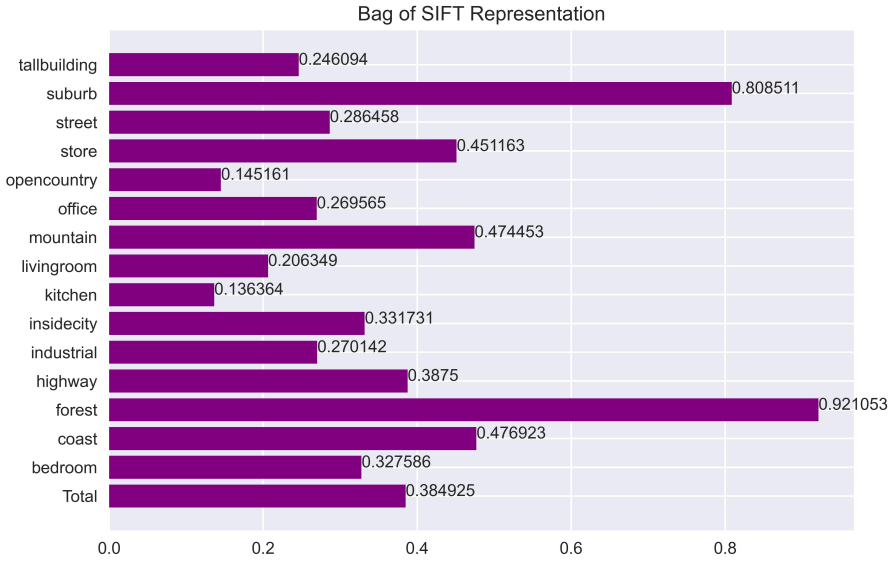
In Bag of SIFT Representation method, the relation between `vocab_size` and the result accuracy is as follows:



Among 8 sizes above, 64k performs best. As as vocabulary size grows larger, however, the accuracy do not always increase. It shows that blindly pursuing larger vocabulary is not a good choice.

Analysis

I have taken a close look on how it performs on different categories. In Bag of SIFT Representation method, the performance is as follows:

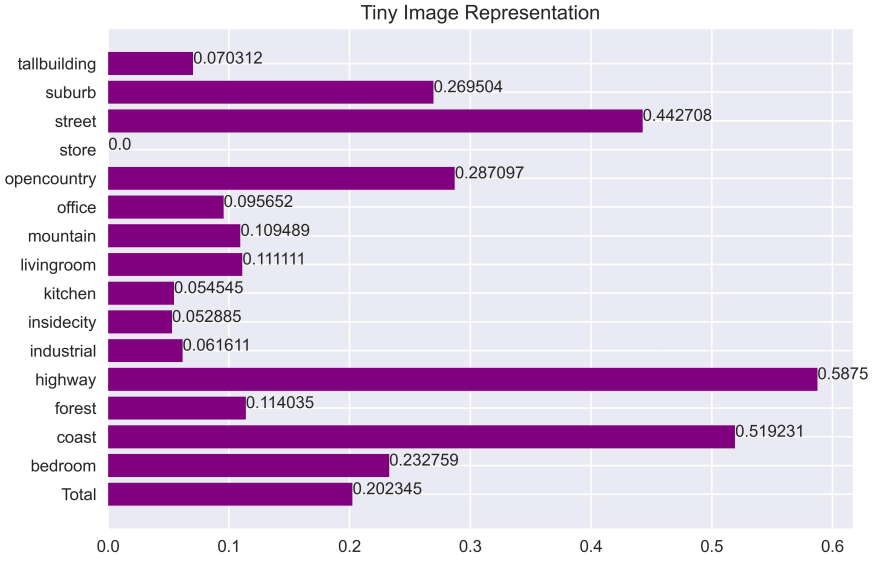


The best two are `forest` and `suburb`. Here are some images I sampled from the training data, and I have mark the keypoints that SIFT have detected on the images:

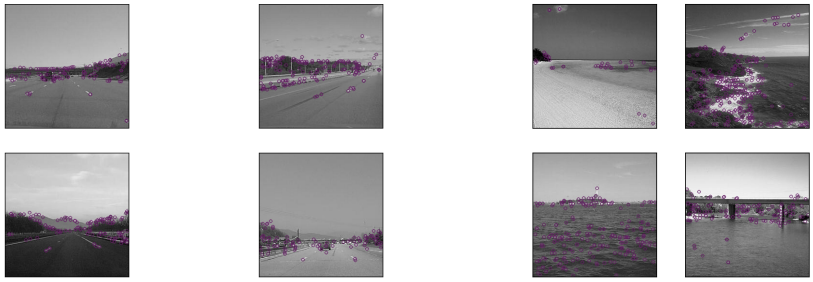
The amount of keypoints is quite big, and the keypoints are densely distributed. In contrast, the worst two are `kitchen` and `opencountry`, and their samples are as follows:

Apparently, the keypoints of worse performed categories are fewer and sparser.

Then, the results of Tiny Image Representation are as follows:



The best two are `highway` and `coast`. Here are some images I sampled:



The composition of the above pictures is simple. So when the images are zoomed into 16 ×16 size, part of structure of them are somehow keeped. And I think that is why they perform well on Tiny Images Representation.

In comparison, The worse performanced ones are as follows:

The composition of the above pictures is complicated. Once the images are zoomed into small size, the information of them are massively lossed. I think that is why the Tiny Images Representation fails on this case.

Run codes

See `codes/RADME.md` for details.