EQ2425 Analysis and Search of Visual Data

EQ2425, Project 2

Hanqi Yang hanqi@kth.se

Qian Zhou qianzho@kth.se

September 28, 2022

Summary

The purpose of this project is to build a visual search system and evaluate it. First, we need to extract a few thousands SIFT features from each database image, based on the 50 building objects. Then, we build the vocabulary tree with the SIFT keypoint descriptors obtained above. Finally, client database of 50 images is used for querying. Here, we use a weighting method called TF-IDF (term frequency inverse document frequency). We send all the descriptors of each query object into the vocabulary tree and rank the database objects according to their TF-IDF scores. Moreover, to evaluate the system, both top-1 and top-5 recall are implemented. As for results, we can observe that the system's accuracy improves with more children branches and higher tree level. In addition, the top-5 recall gives us better results. When we increase the number of features, the accuracy will be slightly improved. However, when the number of features increases to a certain number, the accuracy will stop changing or even show a downward trend, which may be the result of over-fitting.

1 Introduction

In this project, we build a visual search system and evaluate it. We first use SIFT algorithm to extract features from database and query images. The database used is composed by 3 images of the same building taken from different perspectives, while the query images are taken from the same building objects in the database. Then, we use vocabulary trees as the database structures. The vocabulary tree is built by using the hierarchical k-means algorithm. With query images, the goal is to recognize buildings. That is, the visual search system retrieves the objects that are most similar to the query object. Here, TF-IDF (term frequency inverse document frequency)[1] score is used for object retrieval. We send all the descriptors of each query object into the vocabulary tree and rank the database objects according to their TF-IDF scores. In total, we test all the 50 query objects and calculate the average recall rate to evaluate system performance.

2 Problem Description

The visual search system will include three parts: image feature extraction, vocabulary tree construction, querying.

2.1 Image Feature Extraction

In this section, we need to extract the SIFT features from each database and query image. For the database images, they are stored in the server folder and there are 50 building objects in total with each building object appearing in three images. For the query images, they are stored in the client folder with each building object appearing in one query image. The 50 images are taken from the same building objects in the database.

First, we extract 1000 SIFT features from each database image because this number can minimize calculation time without affecting the normal operation of our vocabulary tree construction. Then, we combine the features of the same object and save them. Similarly, we extract 1000 SIFT features from each query image.

2.2 Vocabulary Tree Construction

The vocabulary tree is built by using the hierarchical k-means algorithm. The structure of the tree is controlled by the tree branch number and the tree depth. Therefore, we use the function $hi\_kmeans(data, b, depth)$ to generate the vocabulary tree, where data holds the SIFT features from the database objects, b is the branch number of the vocabulary tree for each level and depth is the number of levels of our vocabulary tree. We first run built-in k-means function on the original data. Then for each resulting clusters, we run built-in k-means function on them recursively.

2.3 Querying

In this part, we send all the descriptors of each query object into the vocabulary tree and rank the database objects according to their TF-IDF scores. For each query image, the extracted descriptors are sent one at a time. When reaching the final node, the absolute values of all the descriptors' score vectors are summed up. Then, the resulted score vector is ranked according to its best value to determine which one is a match. We test all the 50 query objects and calculate the average recall rate.

3 Results

3.1 Image Feature Extraction

(a) As mentioned above, we extract 1000 SIFT features from each database image. Since each building object appears in three images, the average number of features we extract per object is 3000.
(b) Similarly, the average number of features we extract per object for the query image is 1000.

3.2 Vocabulary Tree Construction

(a) In each node in the vocabulary tree, when the node has its children branch, the cluster centroid vectors and the index of children cluster are stored in the node. When the node has no children branch, it stores the number of features in each object that belong to this cluster.
(b) To compute the TF-IDF score for a node, we need to know the number of occurrences of an object in the corresponding cluster, the total number of descriptors in a database object, the number of database objects that contain the cluster, and the total number of database objects.

3.3 Querying

(a) We build three vocabulary trees by varying the settings as: $b = 4, depth = 3$; $b = 4, depth = 5$ and $b = 5, depth = 7$. For these three trees, the average top-1 and top-5 recall rates over 50 objects are shown in Table 1.

It can be observed that as the number of children branches and the level of tree increase, the average recall rate becomes larger, which means more accurate identification. As the number of optimal values taken increases, the average recall rate also increases.

Table 1: Average recall rate by varying the tree settings

branches	depth	top-1	top-5
4	3	0.02	0.1
4	5	0.12	0.24
5	7	0.62	0.86

(b) Then, we keep the parameters of the tree unchanged with $b = 5, depth = 7$. For querying, we use 90, 70 and 50 percent of the number of the query features. The average top-1 and top-5 recall rates are shown in Table 2.

When the number of features increases, the accuracy will be slightly improved, but when the number of features increases to a certain number, the accuracy will stop changing or even show a downward trend, which may be the result of overfitting.

Table 2: Average recall rate by varying the tree settings

number of features	top-1	top-5
900	0.62	0.86
700	0.64	0.86
500	0.62	0.84

(c) It can reduce computational time using hierarchical clustering. When using hierarchical clustering, we only need to calculate and compare the Euclidean distance with the clusters in the current level, and then take the cluster with the smallest value and calculate the Euclidean distance with its children clusters. However, without hierarchical clustering, we need to compute the Euclidean distance concerning all the clusters. For example, with a tree with with $b = 2, depth = 2$, with hierarchical clustering we will compute the comparison 4 times, while 6 times are needed without it.

3.4 Bonus

From the results above, we think we can extract more features and build a vocabulary tree with higher levels and more children branches. For instance, when we set the parameters of the tree as $b = 7, depth = 9$, we can obtain an accuracy of 80% on top-1 recall, and of 92% on top-5 recall. When we extract 1500 SIFT features from each database and query image and we set the parameters of the tree as $b = 5, depth = 7$, we can obtain an accuracy of 66% on top-1 recall, and of 86% on top-5 recall.

However, this two methods will greatly increase the calculation time. Moreover, too many features can cause over-fitting, which will lead to a wrong evaluation in the end.

4 Conclusions

In conclusion, we build a visual search system by using vocabulary tree. With more children branches and higher tree level, the accuracy of this system increases. In addition, the top-5 recall shows higher accuracy. When we increase the number of features, the accuracy will be slightly improved. However, when the number of features increases to a certain number, the accuracy will stop changing or even show a downward trend, which may be the result of over-fitting.

Appendix

Who Did What

Both of us have the same contribution to the completion of this project.

References

[1] Markus Flierl, EQ2425 Analysis and Search of Visual Data, Lecture Slides, 2022