Weekly Report

Number:4

Adviser: Prof. Yang Wen

Student: Cheng Wensheng

Period: 2017.12.11-2017.12.17

Recap

This week I mainly put my effort on the image retrieval project. I tried sevaral solutions, including classic image processing technics and deep learning methods.

Deep learning method

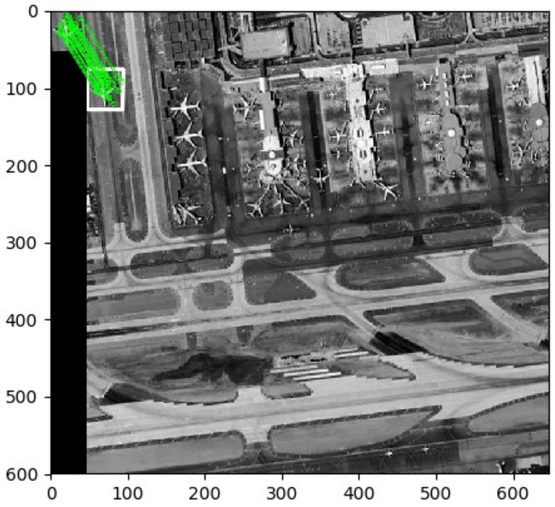
I tried codes from paper *Local Convolutional Features with Unsupervised Training for Image Retrieval* (ICCV 2015). I found 3 main issues.

1. For patch retrieval task, program input is not directly the image patches cut from larger images. In fact, the author use **Hessian-Affine detector** to extract interest points and retain color information. But the author didn't provide those codes. So I need to try to get program input in correct format.
2. The codes are Matlab/C mixed files. It's not compatible with our project in Python/Tensorflow/Pytorch.
3. For patch retrieval task, the mAP(mean Average Precision) of Patch-CKN is only about 1% better than SIFT baseline. So it seems not so attractive.

Classic method

With reasons mentioned above, I plan to use classic image processing method.

- I tried the classic SVM+BoW+knn_search solution. So I use SIFT to extract and describe images of dataset. Then I use kmeans to cluster these descriptors. Then I get these center descriptors as dictionary of BoW. For quary picture, use SIFT to get specified descriptors in BoW to represent the picture. At last, use FLANN-based kdTree to get nearest images of query picture. That's the whole pipeline.
- The problem is that the query picture is always very small. And there are few keypoints extracted, which means few matches. Besides, objects have too many poses and different scales, rotations. In my test, I cut part of the larger image as query image(*size: about 50*50*). It turns out this algorithm can't find good matches for the query picture except the original part of query image itself.



- I plan to try dense-SIFT to get more keypoints. I guess this may improve its performance.