CS783 Assignment1

Harshit Sharma (160283)

Ashish Kumar (160160)

Initial Approach

- **First** we applied the **Text Retreival Approach to Image Instance Recognition**. But we later switched to other technique because this was not working for hard test images.
- In the above technique we applied the steps as follows:-
  - First we extracted **local features** from all the images in the database using **SIFT Descriptor**.
  - After that we clustered all the local features into **8000 clusters**, where each cluster centroid represented a new visual word in our visual dictionary.
  - Using the above clustering model we extracted the number of distinct visual words present in each image.
  - Then we created a similarity vector for all the images in the database using **TF-IDF** scores.(**Dimension** : is 1 * number of visual words)
  - Now for **query image** we extracted it's **local features, then clustered it, find it's similarity vector** by using **TF-IDF** scores.
  - Used **cosine similarity** for comparing the similarity of the query image with all the images and then ranked the images by sorting their scores.
- **Local Features –** Initially used **orb** for extracting local features,even though it was faster but it was **not finding enough key points**, that's why we shifted to **SIFT** for extracting local features.
- **Clustering –** Initially we tried doing K-means for the full training images but it lead to exploding of number of local features, which almost made k-means impossible to run. So then we **cropped our training images** such that most of the chess-board part is cropped of and used **Mini-batch K-means** so that our training remains feasible for running on a normal PC.
- **Number of Clusters –** Initially we started with 1000 clusters then we decreased it but result got worsened so finally we set it to 8000.
- **Comparing the similarity –** We intuitively thought **cosine similarity** to be the most sensible option.

Final Approach

- Unsatisfied with the results of above sift based approach, we start searching for other good methods and encountered **DELF(DEep learning Local Features)**.This new feature is based on convolutional neural networks and thus is able to extract more relevant features than SIFT.
- We first extracted local features for all the database images(cropped for better result) using DELF and build a cKDTree using it.
- For a query, We extract its DELF local features and find K nearest neighbour feature for each feature by querying on cKDTree.

- We find the all database images that contained any of these K nearest feature.So we have some candidate image from database and we need to rank them.
- For ranking, we took query image and candidate image, get all matching feature between them. Then, for removing outlier features pair, we used RANSAC and used the inliers count as score. And Final ranking is based on sorted score.
- Then for ranking the residual database images,we use above sift based approach.
- The final model turns out to be better than previous sift model.

Partially Tried Approach

- We tried to finetune some pretrained model using our database, so that we can detect bounding box for the object present in image.
- We tried this using gluon-cv library to fine-tune pretrained SSD(Single Shot Detector) but were unable to do it due to lack of computation resources and time constraint.

md5 hash of our Final Model

- **ee5a08eb70cb958b686b48df81bd3687**

References

- A Text Retrieval Approach to Object Matching in Videos
- Github Repository link for DELF
- Research Paper on DELF
- Blog explaining about DELF and code related to it
- Gluon–Finetuning Pre-trained Model