DIRECTED RESEARCH REPORT

Project Title: Landmark Recognition

Presented By: Radhu Maradia & Vritti Ganeriwal

Tenure: Spring 2018

Presented to: Professor Saty Raghavachary

INTRODUCTION

An image retrieval system comprises of a system for browsing, searching, retrieving images from a large database of digital images.

Image retrieval can be broadly categorized into:

1. Concept Based Image Retrieval: refers to retrieval from text-based indexing of images that may employ keywords, subject headings, captions or natural language text.
2. Content Based Image Retrieval: search analyses the contents of the image which could be color, shape texture or any other information that can be derived from image.

Due to alarming growth of the internet and high volume of data, Content Based Image Retrieval is emerging to be one of the most used technology in image retrieval.

Application of Content Based Image Retrieval

Some of the important application of Image Retrieval are:

- Architectural and engineering design
- Art Collections
- Crime Prevention
- Face Finding
- Landmark Retrieval

OUR PROJECT

This project is inspired from a Kaggle Competition Google Landmark Retrieval Challenge. The aim was given a query image of a specific landmark, retrieve all the images from the database which has the same landmark.

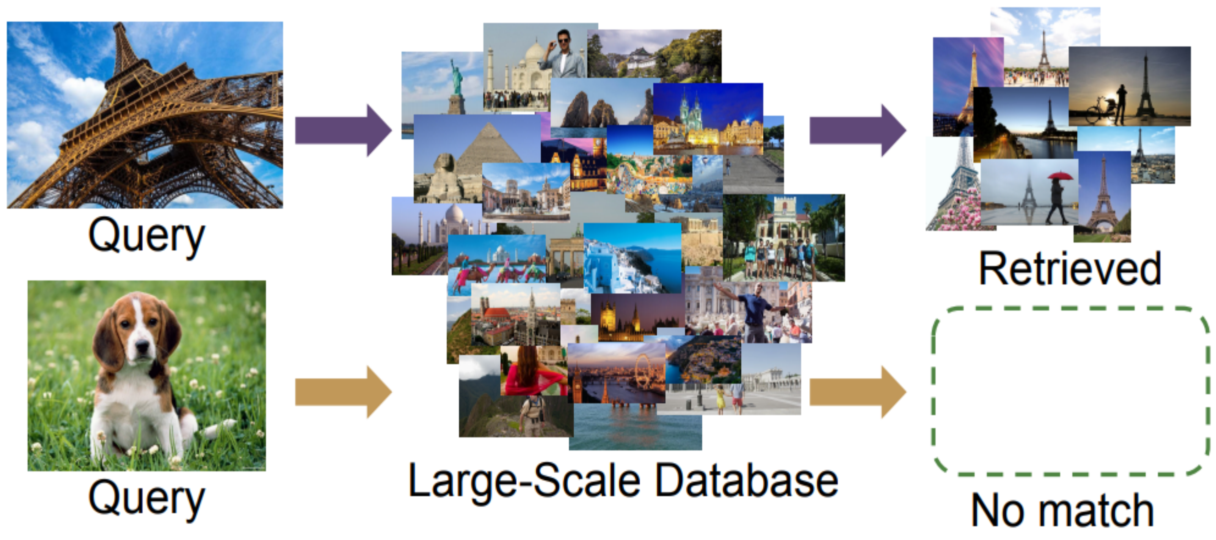


Fig 1

Dataset Used

We took images from following 20 landmarks:

Taj Mahal
Leaning Tower of Pisa
India Gate
Statue of Liberty
Eiffel tower
Twin Tower
Sydney Opera House
Pyramids of Giza
Burj Al Arab
Christ the Redeemer
Stonehenge in the English county of Wiltshire
Luxor Temple
Empire State Building
Capitol, Hill
Mount Rushmore
Golden Gate
Lotus Temple
Golden Temple
Hollywood Sign

The images were chosen with range of background colors, different sizes, with and without people, different brightness, partially out of view objects etc. to show that the algorithm works well for a wide variety of images.

Furthermore, some query images are collected to work as distractor. They belong to following 2 category:

1. Images of landmark which is not present in the database
2. Non-landmark images

These play critical role to evaluate robustness of algorithm to irrelevant and noisy queries.

Deep Local Features

To compare 2 images we have used Deep Local Features (DeLF). It is based on Convolution Neural Network which is trained with weak supervision using image level class labels. Among other methods used for image retrieval, Deep Local features gives significant optimized and accurate results for large scale datasets.

Image Retrieval with DeLF

Large scale image retrieval system using DeLF can be broken into 4 important steps.

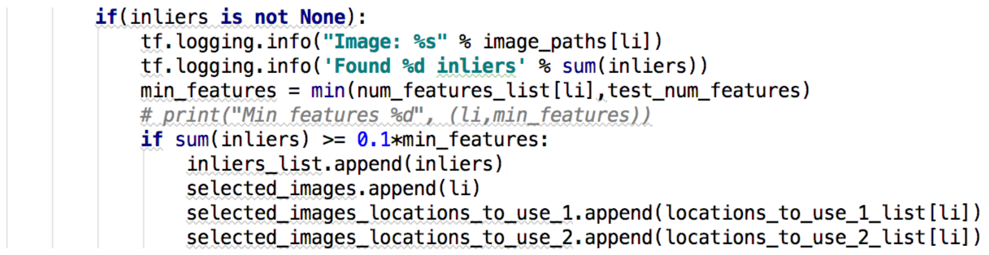
1. Dense Localized Feature Extraction
   Extraction of dense features from an image by applying a fully convoluted network (FCN), which is constructed by using the features extraction layers of a CNN. The obtained feature maps are regarded as a dense grid of local descriptors.

2. Key point Selection
   Instead of using densely extracted features directly for image retrieval, we design a technique to effectively select a subset of the features. Since a substantial part of the densely extracted features are irrelevant to our recognition task and likely to add clutter, distracting the retrieval process, keypoint selection is important for both accuracy and computational efficiency of retrieval systems.

3. Dimensionality Reduction
   We reduce the dimensionality of selected features to obtain improved retrieval accuracy. First, the selected features are L2 normalized, and their dimensionality is reduced to 40 by PCA.

4. Indexing and Retrieval
   We extract feature descriptors from query and database images, where a predefined number(minimum number of features from both images) of local features with the highest attention scores per image are selected. When a query is given, we perform approximate nearest neighbor search for each local descriptor extracted from a query image. Then for the top K nearest local descriptors retrieved from the index, we aggregate all the matches per database image. Finally, we perform geometric verification using RANSAC and employ the number of inliers as the score for retrieved images. Many distractor queries are rejected by this geometric verification step because features from distractors may not be consistently matched with the ones from landmark images.
   We then take the matched images and filter based on a 0.1% threshold of the matched

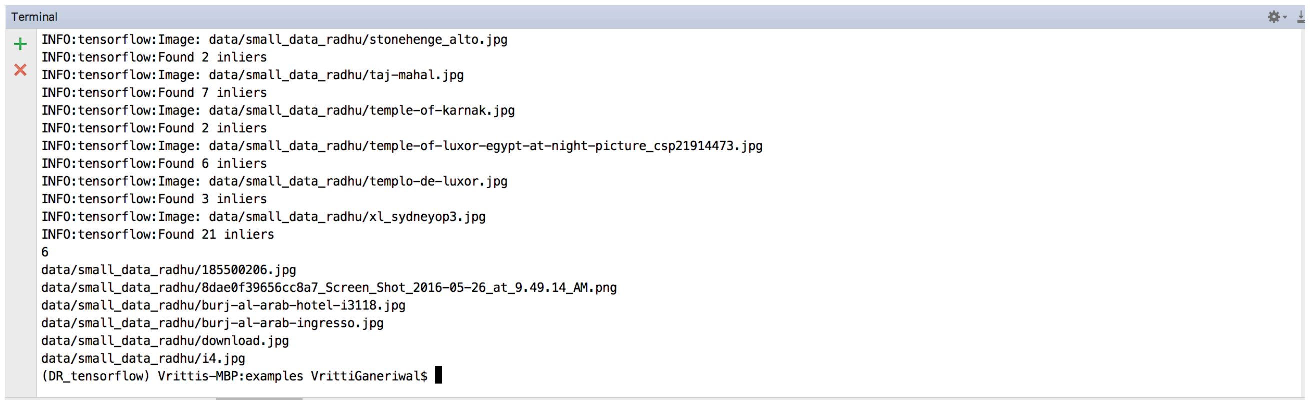
features. If the number of inliers of the image are greater than the threshold, then we append it to the selected images list.



```
if(inliers is not None):
    tf.logging.info("Image: %s" % image_paths[li])
    tf.logging.info('Found %d inliers' % sum(inliers))
    min_features = min(num_features_list[li],test_num_features)
    # print("Min features %d", (li,min_features))
    if sum(inliers) >= 0.1*min_features:
        inliers_list.append(inliers)
        selected_images.append(li)
        selected_images_locations_to_use_1.append(locations_to_use_1_list[li])
        selected_images_locations_to_use_2.append(locations_to_use_2_list[li])
```

Fig. 2

Figure 3 depicts the images and their corresponding that satisfy the particular threshold along with their inliers detected.



```
INFO:tensorflow:Image: data/small_data_radhu/stonehenge_alto.jpg
INFO:tensorflow:Found 2 inliers
INFO:tensorflow:Image: data/small_data_radhu/taj-mahal.jpg
INFO:tensorflow:Found 7 inliers
INFO:tensorflow:Image: data/small_data_radhu/temple-of-karnak.jpg
INFO:tensorflow:Found 2 inliers
INFO:tensorflow:Image: data/small_data_radhu/temple-of-luxor-egypt-at-night-picture_csp21914473.jpg
INFO:tensorflow:Found 6 inliers
INFO:tensorflow:Image: data/small_data_radhu/templo-de-luxor.jpg
INFO:tensorflow:Found 3 inliers
INFO:tensorflow:Image: data/small_data_radhu/xl_sydneyop3.jpg
INFO:tensorflow:Found 21 inliers
6
data/small_data_radhu/185500206.jpg
data/small_data_radhu/8dae0f39656cc8a7_Screen_Shot_2016-05-26_at_9.49.14_AM.png
data/small_data_radhu/burj-al-arab-hotel-i3118.jpg
data/small_data_radhu/burj-al-arab-ingresso.jpg
data/small_data_radhu/download.jpg
data/small_data_radhu/i4.jpg
(DR_tensorflow) Vrittis-MBP:examples VrittiGaneriwal$
```

Fig. 3

RESULTS

Figure 4,5,6 depict the selected images based on the threshold along with the features that are matched with the query image.

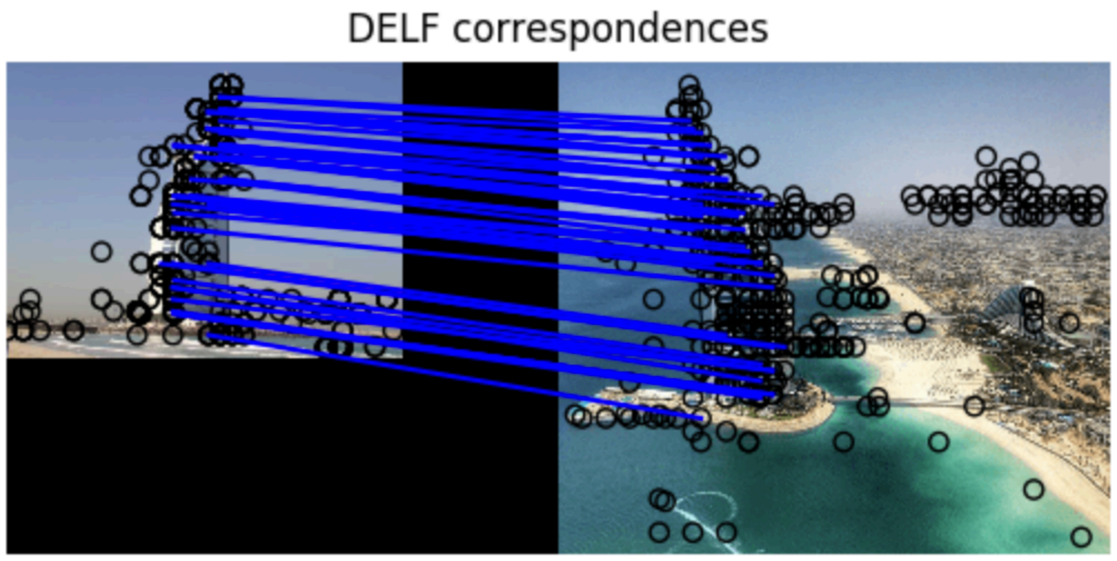


Fig 4

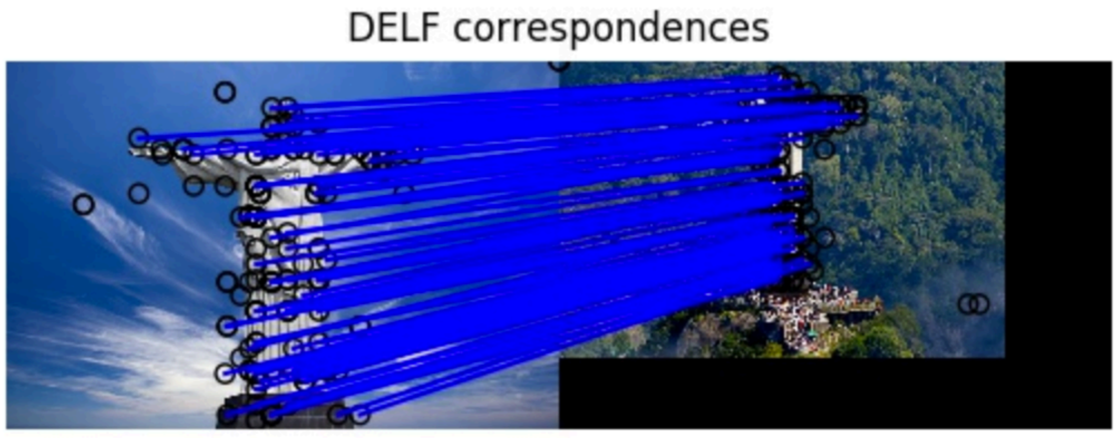


Fig 5

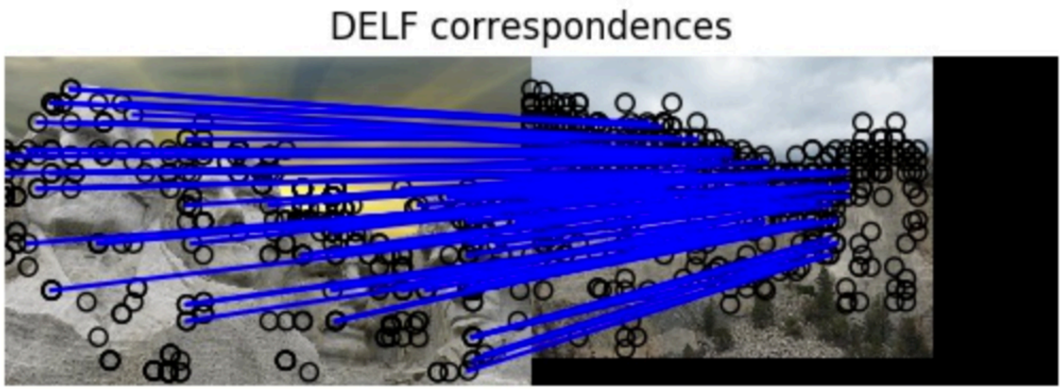


Fig 6

FUTURE WORK

1. Instead of looping over every image in database to match the query image, the images in the database can be preprocessed and clustered. This would improve the time complexity significantly.
2. Currently threshold is taken to be 0.1 based on observation from the database and the query. However, this method is extremely dependent on data. It could be improved to be adjusted dynamically.
3. It is highly possible that some landmarks can have its copy at different locations in the world. In that case, our project retrieves all images from all location. It can be further improved by attaching GPS locations to the image and making corresponding changes in the codebase.