Landmark Retrieval via Local Guidance and Global Expansion

Aimin Su[1,2] Steven Ly[3,2] Fan Yang[4,2] Ryota Hinami[4,2] Qier Meng[2] Sang Phan[2]
Yusuke Matsui[2] Zheng Wang[2] Shin'ichi Satoh[2,4]

Abstract

Google recently held a competition on the Google-Landmarks dataset, the largest worldwide dataset of images for large-scale image retrieval research. We proposed our own model, which incorporated different modern image retrieval techniques together, and finished in 7^{th} place in the competition. We present our full image retrieval model and its results, as well as go over the challenges we faced in the competition.

1. Introduction

With the explosive growth of multimedia data on the web, image retrieval is becoming a fundamental problem in computer vision. The task is straightforward: given a query image, find the similar images in a a large database. We competed recently in the Google Landmark Retrieval Challenge[*1], a large-scale image retrieval competition on a landmarks dataset. Image retrieval is especially important for landmark photos, which comprise a large percentage of the photographs taken. To tackle this challenge, we propose an approach that incorporated query expansion and database augmentation with deep local and global features. Finally, we utilized regional diffusion to do a final re-ranking of our results.

2. Dataset

The Google-Landmarks dataset comprises of 1,098,462 index and 117,704 test images, with 15,000 unique landmarks. The index set was constructed by clustering photos based on their geolocation and visual similarity. The task was challenging because of a couple of aspects of the index dataset. First, the ground-truth labels were not included with images in the index, so this made our task an unsupervised one. Also, both the index and query set contained many distractors (non-landmark photos), which ranged from things like paintings to food.

[1] University Grenoble Alpes
[2] National Institute of Informatics
[3] University of Southern California
[4] The University of Tokyo
[*1] `https://www.kaggle.com/c/landmark-retrieval-challenge`

3. Proposed pipeline

The pipeline of our image retrieval system is illustrated in Fig. 1. It consists of five key steps: **(1)** Deep local feature (DELF) search. **(2)** Sptatial verification (RANSAC) on top-100 results of (1). **(3)** Deep image retrieval (DIR) search with database-side feature augmentation (DBA). **(4)** Query expansion with top-5 results of (3) and the results of (2) with inlier > 40. **(5)** Re-ranking by regional diffusion. In the next sections, we go over the details of the different components of our pipeline, and also explain how they tie together.

3.1 Image Representation

We first explain the feature representation of images. For each gallery image, two kinds of features are extracted.

DELF is an activation from a fully convolutional network with an attention mechanism [6]. For each image, salient key points are detected. The feature is extracted as $X_n = \{\mathbf{x}_n^{(1)}, \mathbf{x}_n^{(2)}, \ldots\}$, where X_n is a set of local features for the nth image, and its ith feature is denoted as $\mathbf{x}_n^{(i)} \in \mathbb{R}^{40}$. The coordinate (x, y) of each feature point is stored as well.

DIR was recently proposed by Gordo et al. [2], and is the current state-of-the-art global feature of image retrieval. Its architecture follows R-MAC [7], which constructs a single global feature vector by mean-pooling multiple sub-region features. Each region feature $\mathbf{z}_n^{(i)}$ is extracted by max-pooling on convolutional feature map followed by normalization steps. All DIR components are optimized for image retrieval by learning with triplet loss, and can be trained in an end-to-end manner. Using the DIR pretrained model provided[*2], we extracted features from multi-scale images following [2].

3.2 DELF search

Given a query image, we first run a k-NN search on the DELF representations. We decided to use local descriptors so that we could get a small number of accurate initial results by focusing on local details of the images. As a part of preprocessing, all DELF features from all

[*2] http://www.europe.naverlabs.com/Research/Computer-Vision/Learning-Visual-Representations/Deep-Image-Retrieval

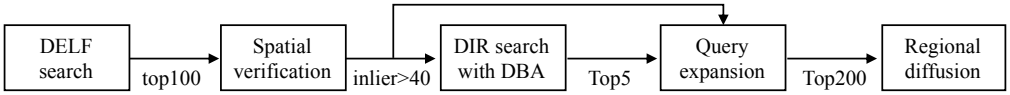

Fig. 1 Our image retrieval pipeline

images are stored in an index structure for fast search. We employed the state-of-the-art HNSW [5] + IVFPQ [4] framework with Faiss library*3.

During the search phase, query images also have their DELF features extracted: $X_q = \{\mathbf{x}_q^{(1)}, \mathbf{x}_q^{(2)}, \ldots\}$. We maintain a counter $\mathbf{c} = [c_1, \ldots, c_N]$, where c_n measures the similarity between the query and nth gallery image. The counter is first initialized by zeros: $\mathbf{c} \leftarrow \mathbf{0}$. For each query feature $\mathbf{x}_q^{(j)}$, the top-T (approximate) nearest neighbors are retrieved from all gallery features. We denote these top-T results as a set of tuples $\{(n_1, i_1), (n_2, i_2), ...\}$, where (n_t, i_t) means that i_tth feature of n_tth gallery image is tth nearest from $\mathbf{x}_q^{(j)}$.

These results are aggregated to the final score as follows. For each (n_t, i_t), the counter is updated as $c_{n_t} \leftarrow c_{n_t} + 1$ iff (1) this (n_t, i_t)-feature has not appeared yet for all X_q, and (2) features from n_tth image have not appeared yet when we are considering the result of $\mathbf{x}_q^{(j)}$. The first condition suppresses unreasonably strong features that match well with many other features. The second condition alleviates the burstiness problem where many similar features are extracted from the same image. We performed this procedure for all query features, and the top-T highest values are selected from the counter $\mathbf{c}$. These nearest images (IDs of the highest entries of $\mathbf{c}$) are used in the next step.

3.3 Spatial Verification with RANSAC

Spatial verification measures the spatial consistency between the query image and each of its retrieved results. RANSAC is one of the most commonly used and robust algorithm to perform such geometric verification. The number of inliers indicates how many features are matching for a given query and its retrieved images, and can be considered a metric for measuring query quality. Since there exists a large amount of distractors in both the training and testing set, RANSAC can help to filter out distractors by filtering out images that have very few matches. Figure 2 shows some matching examples between query and database side images using DELF features and RANSAC. In our model, we found that filtering out images that had at least 40 matching inliers worked best.

3.4 Global Expansion

Using the search results produced from DELF+RANSAC, we ran global feature-based search to find additional relevant images. We performed k-nearest neighbor search on the DIR feature space, using the DIR descriptors as the global image representation. However, instead of using the original query/gallery feature vectors, we employed average query expansion (AQE) and database-side feature augmentation (DBA) as enhancements to the original features.

3.4.1 Database-side Feature Augmentation

Database-side feature augmentation (DBA), proposed by Arandjelović and Zisserman [1], replaces the feature of every gallery image by a combination of itself and its neighbors. The objective of DBA is to improve the quality of image representation by augmenting features using nearest neighbor images, which is based on the assumption that the same landmark is appeared in nearest neighbor images. Specifically, the feature of each gallery image $\mathbf{y}$ is replaced by the weighted average of its k-NN features in database $\{\mathbf{y}_1, .., \mathbf{y}_k\}$ as

$$\mathbf{y}' = \frac{1}{k} \sum_{r=1}^{k} \frac{k-r}{k} \mathbf{y}_r, \tag{1}$$

$\frac{k-r}{k}$ is the weight of rth nearest neighbor feature. Larger weights are assigned to the features that are close to the original vector. This simple process significantly improves the scores without increasing the search cost. In our model, we applied DBA during the DIR search process, using the top 10 nearest neighbors of gallery features as described in [2].

3.4.2 Average Query Expansion

Average query expansion (AQE) is a common form of query expansion used in image retrieval. The top-k results from search are taken from the original gallery, and they are then averaged to form a new feature vector. This feature vector is then appended to the original query's feature vector. In our model, we used AQE twice. **1)** We first use the DELF results filtered by RANSAC in 3.3 as inputs to AQE for DIR search with DBA. **2)** We then select the top-5 results from **1)** and concatenate it with results of 3.3 to form an input for a second AQE that goes through the k-NN search process with DIR features again.

3.5 Reranking by Regional Diffusion

Diffusion, originally reported in [8], uses the neighborhood graph constructed on the dataset to search each query on the manifold. It was further developed to handle both global and regional features. Moreover, by us-

*3 https://github.com/facebookresearch/faiss

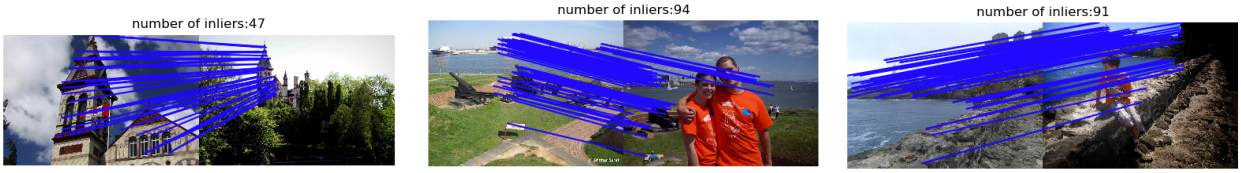

Fig. 2 Pairwise feature correspondences of images in query and database. Blue lines connect the center of matching DELF local features. In these examples, RANSAC can correctly match the landmarks despite of varying illumination, landmark scale, different points of view, etc.

ing scaling up methods such as truncation on the affinity matrix, we can conduct diffusion on even a large-scale dataset [3].

We apply diffusion for re-ranking based on regional DIR features in our task. Firstly, we construct the neighborhood graph for the whole gallery off-line. Each node in the graph stands for a feature vector of a certain image in the gallery, while each edge represents the similarity between a pair of features. Given a query denoted by $Z_q = \{\mathbf{z}_n^{(1)}, \mathbf{z}_n^{(2)}, \ldots\}$, a set of regional features, we obtain a ranking list $\{Z_1, \cdots, Z_k\}$ containing the top-k retrieved results that were produced by the aforementioned methods. Since we only diffuse on the subset of the gallery that covers the ranking list for reranking, we use just a small part of the whole neighborhood graph. In particular, a variant of Laplacian $\mathcal{L}_\alpha$ proposed in [3] is constructed by this truncated neighborhood graph. Similar to query expansion, we concatenated the regional features of the original query and top-5 gallery images on the ranking list to form a new query.

Then we obtained the initial kNN search results for each feature of the new query over gallery features belonging to the subset. The scores of initial results are saved into a vector $\mathbf{f}_0$. In the diffusion stage we applied the conjugate gradient to solve $\mathcal{L}_\alpha \mathbf{f}^* = \mathbf{f}_0$, where $\mathbf{f}^*$ will be a vector composed by the ranking scores after diffusion. Note that $\mathbf{f}^*$ contains ranking scores on feature level, we need to aggregate these scores by images. Finally, based on the ranking scores of the query over the gallery images on the original ranking list, we obtained a re-ranked list.

3.6 Evaluation

The competition was evaluated according to mean Average Precision @100 (mAP@100):

$$mAP@100 = \frac{1}{Q}\sum_{q=1}^{Q}\frac{1}{\min(m_q, 100)}\sum_{k=1}^{\min(n_q,100)} P_q(k)rel_q(k) \quad (2)$$

where:

- Q is the number of query images that depict landmarks from the index set
- m_q is the number of index images containing a landmark in common with the query image q (note that this is only for queries which depict landmarks from index set, so $m_q \neq 0$)
- n_q is the number of predictions made by the system for query q
- $P_q(k)$ is the precision at rank k for the q-th query
- $rel_q(k)$ denotes the relevance of prediction k for the q-th query (1 if correct, 0 otherwise)

There were several challenges when it came to evaluating our model's performance. First, the ground truth image clusters were not included in either the training or test datasets. This made it difficult to test the performance of our model without submitting our final results to the Kaggle website. In addition, Kaggle has a submission policy that only allows for up to 5 submissions a day. These factors made it difficult for us to tune the different hyperparameters for our model.

To overcome these challenges, we developed our own system for internal evaluation. In conjunction with the Landmark Retrieval Challenge, Google also ran the Landmark Recognition Challenge. The recognition challenge had the same test data, but had a different training dataset with no landmarks in common with the retrieval challenge. The training data however had a ground truth label for each image's respective landmark. To take advantage of these labels, we selected 10% of the full training data to use for our internal validation dataset that was further split into a training and test set. The created split was stratified so that it contained the same distribution of landmarks as the full training set. In this setting, we counted a correct retrieval as an image with the same landmark label as the query image. Using this, we were able to solve the aforementioned problems and have an efficient way to tune our hyperparameters.

4. Results

We performed the evaluation of some state-of-the-art approaches and combine both local (DELF) and global (DIR) feature-based methods. Note that there are two leader boards on Kaggle. The public leader board evaluates the scores based on 34% of your submission file to calculate the mAP score and show it on the public board.

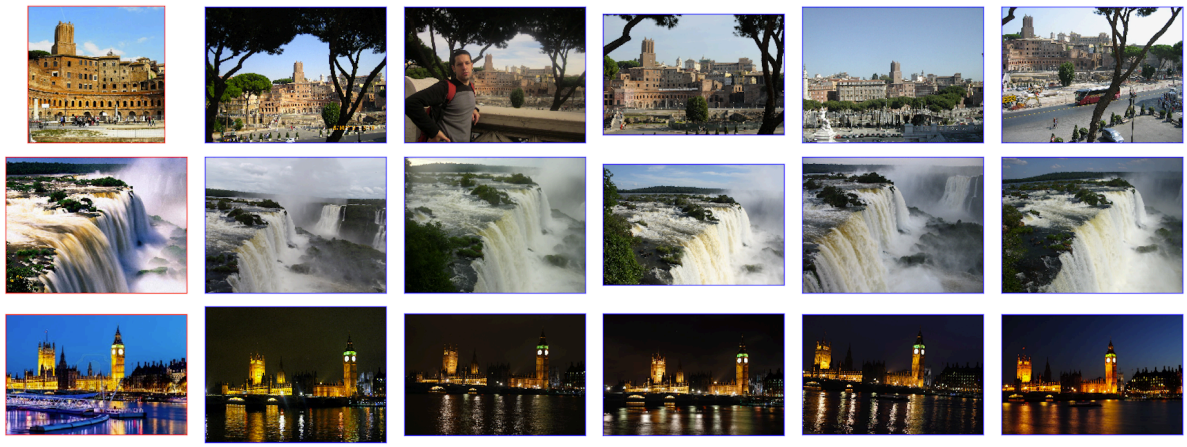

Fig. 3 We choose three non-distractor query images and show their top results. Images with red border indicate the selected queries and blue-bordered images are the corresponding results.

Table. 1 Scoreboard for the internal validation set, the public set, and the private set. mAP@100 was calculated for each proposed method.

	mAP@100		
method	internal	public	private
1. DELF	0.393	0.288	0.292
2. DIR	0.436	0.419	0.396
3. DIR + AQE	0.452	0.502	0.483
4. DIR + DBA + AQE	0.476	0.524	0.512
5. DIR + DBA + DELF + RANSAC + AQE	0.515	0.580	0.545
6. DIR + Diffusion	0.527	0.526	0.523
7. DIR + Diffusion + AQE	-	0.531	0.532
8. Fusion of 5 and 7	-	0.586	0.550
9. Diffusion re-ranking of 8	-	**0.600**	**0.561**

The private leader board, represents the final scores which competitors are judged on, and evaluates the scores based on the held-out 66% at the end of the competition. The results for the internal, public, and private datasets are displayed in Table 1. Some results on our internal evaluation dataset are missing because of time restrictions. We observe that commonly used techniques, such as QE and DBA, provide significant boost to the score. Using a simple late fusion with max normalization of our proposed pipeline with DIR + Diffusion + AQE, we were able to increase our mAP to 0.550. A final re-ranking with diffusion resulted in our best score of 0.561 mAP.

We manually selected some good query images that contain clear landmark and show their top 5 retried results from our proposed approaches. In figure 3, we observe that top results are all correctly retrieved from the given queries.

5. Conclusion

At the end of the competition, we finished 7th place with a score of 0.561 mAP. This was a significant drop from the 0.600 mAP score we achieved on the public leader board. This result could be attributed to some overfitting by our model, but our best submissions on the public dataset also ended up being the best on our private dataset as well. It's more likely that our model was just not able to generalize well to the held-out landmarks. In the future, we should consider creating a stronger internal evaluation system and potentially test different type of descriptors to be more resistant to this problem.

Acknowledgments: This work was supported by JST ACT-I Grant Number JPMJPR16UO, Japan.

References

[1] Arandjelović, R. and Zisserman, A.: Three Things Everyone Should Know to Improve Object Retrieval, *CVPR*, IEEE (2012).

[2] Gordo, A., Almazan, J., Revaud, J. and Larlus, D.: End-to-end Learning of Deep Visual Representations for Image Retrieval, *IJCV*, Vol. 124, No. 2, pp. 237–254 (2017).

[3] Iscen, A., Tolias, G., Avrithis, Y., Furon, T. and Chum, O.: Efficient Diffusion on Region Manifolds: Recovering Small Objects with Compact CNN Representations, *CVPR*, IEEE (2017).

[4] Jégou, H., Douze, M. and Schmid, C.: Product Quantization for Nearest Neighbor Search, *IEEE TPAMI*, Vol. 33, No. 1, pp. 117–128 (2011).

[5] Malkov, Y. A. and Yashunin, D. A.: Efficient and Robust Approximate Nearest Neighbor Search using Hierarchical Navigable Small World Graphs, *CoRR*, Vol. abs/1603.09320 (2016).

[6] Noh, H., Araujo, A., Sim, J., Weyand, T. and Han, B.: Large-Scale Image Retrieval With Attentive Deep Local Features, *ICCV*, IEEE (2017).

[7] Tolias, G., Sicre, R. and Jégou, H.: Particular Object Retrieval with Integral Max-pooling of CNN Activations, *CoRR*, Vol. abs/1511.05879 (2015).

[8] Zhou, D., Bousquet, O., Lal, T. N., Weston, J. and Schölkopf, B.: Learning with Local and Global Consistency, *NIPS*.