2nd Place Solution to Google Landmark Recognition Competition 2020

Shubin Dai
bestfitting.ai@gmail.com

ABSTRACT

In this paper, we describe the solution to the Google Landmark Recognition 2020 Challenge [11] held on Kaggle. We use deep convolutional neural networks with metric learning for feature extraction and matching to find candidate image queue first, then we get landmark id and score by using feature similarity and local feature matching. We suppress the influence of distractors by a heuristic approach and re-rank model. Our full pipeline, after ensembling 4 models, scores 0.6375 on the private leaderboard which help us to get the 2nd place in the competition.

1 INTRODUCTION

Google Landmark Recognition 2020 Competition [11] is the third landmark Recognition competition on Kaggle. The task of image Recognition is to build models that recognize the correct landmark (if any) in a dataset of challenging test images. This year, the competition is set as a code competition and collected a new set of test images, which emphasis building more efficient model and generalizing to unseen test set. Google Landmarks Dataset v2(GLDv2) [18] is the biggest landmark dataset, which contains approximately 5 million images, split into 3 sets of images: train, index and test. There are 4132914 images in train set, 761757 images in index set. The host provided a training data for this competition comes from a cleaned version of the GLDv2, including 1.5M training data and more than 80000 classes. Both GLDv2 train set and cleaned GLDv2 train set can be used for training in this competition.

2 CHALLENGE

In this section, we mainly describe the challenge evaluation metrics.

Evaluation metrics: Submissions are evaluated using Global Average Precision (GAP) at k, where k=1.

For each test image, we will predict one landmark label and a corresponding confidence score. The evaluation treats each prediction as an individual data point in a long list of predictions (sorted in descending order by confidence scores), and computes the Average Precision based on this list.

If a submission has N predictions (label/confidence pairs) sorted in descending order by their confidence scores, then the Global Average Precision is computed as:

$$GAP = \frac{1}{M}\sum_{i=1}^{N} P(i)rel(i) \tag{1}$$

where:

N is the total number of predictions returned by the system, across all queries

M is the total number of queries with at least one landmark from the training set visible in it (note that some queries may not depict landmarks)

P(i) is the precision at rank i

rel(i) denotes the relevance of prediction i: it's 1 if the i-th prediction is correct, and 0 otherwise

3 METHODS

3.1 Retrieval Models

To extract global descriptors of landmark images, convolutional neural networks are employed.

3.1.1 Model Design. Four models were used for final submission and each model includes a backbone model for feature extraction and head layers for classification. EfficientNet B5, B6, B7 [17] Resnet152 [7],are selected as the backbone model since their good performance on ImageNet and Google Landmark Retrieval Competition 2020 [12]. Head layers includes a pooling layer and two fully connected(fc) layers. We use generalized mean-pooling (GeM) [6] as pooling method since it has superior performance and p of GeM is set to 3.0 and fixed during the training. The first fc layer is often called embedding layer whose output size is 512, and we will extract the output of this layer as global description of an image. While the output size of the second fc layer is corresponding to the class number of training dataset(81313). Instead of using softmax loss for training, we train these models with arcmargin loss [2], the arcmargin-scale is set to 30, arcmargin-margin is set to 0.3. For validation set, sample 200 images from GLDv2 test set as val set and all the ground truth images of Google Landmark Retrieval Competition 2019 as index dataset and calculated the mAP@100 score. Despite the small sized validation set, the score correlated well with the leaderboard score.

3.1.2 Training Details. We trained our models by increasing image size step by step following the strategy of 1st place

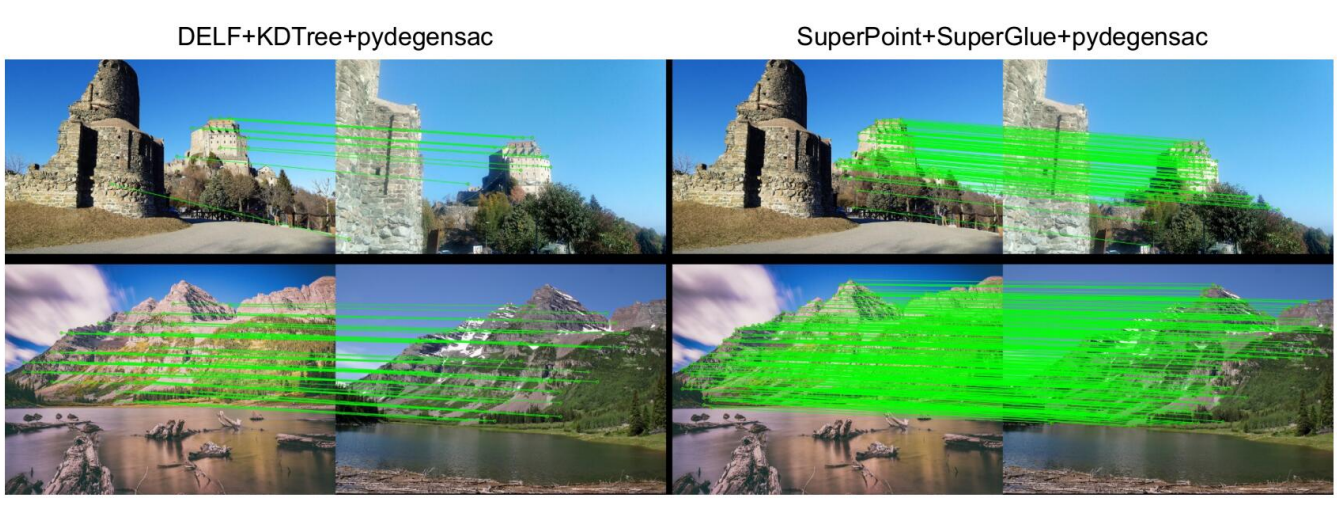

Figure 1: Visualization of feature correspondences between images.left is DELF+KDTree+pydegensac,right is SuperPoint+SuperGlue+pydegensac.

solution [8] to Google Landmark Retrieval 2020 with some modifications.

First, cleaned GLDv2 was used to train the model to classify 81313 landmark classes. EfficientNet B7 [17] backbone based model was trained 6 epochs with 448×448 image inputs at this step.

Second, in GLDv2, there are 3.2 million images belong to the 81313 classes in cleaned GLDv2. we defined these 3.2m images as GLDv2x. GLDv2x was used to finetune the model from step 1 for 4 epochs.

Third, model from step 2 was finetuned using 512×512 images from GLDv2x for 6 epochs.

Next, finetune with 640×640 images for 3 epochs and then 736×736 for 3 epochs

Stochastic gradient descent optimizer was used for training, where learning rate, momentum, weight decay are set to 1e-2, 0.9, 1e-5. learning rate was set to 0.001, 0.0001 for last 3-5 epochs. For image augmentation, left-right flip was used when image size is 448×448. When our models were finetuned on larger images, we used some complex augmentations, including RandomCrop, Brightness, Color, Cutout, Contrast, Shear, Translate, Rotate90.

After replacing the model in baseline kernel [9] from the host with trained EfficientNet B7 [17] model, the public and private score of B7 model are 0.5927/0.5582.

3.2 Validation Strategy

The val set part 1: the 1.3k landmark images from GLDv2 test set(exclude those not in 81k classes).
The val set part 2: sample 2.7k images from GLDv2x but not in cleaned GLDv2.

The index image set for val set: all the images of related landmarks from cleaned GLDv2 train set and sample some other images to get 200k images.

This strategy is quite stable during the whole competition. The reason we decreased one position from public leaderboard 1st to private leaderboard 2nd place is we didn't use full GLDv2x images as index image set for kNN search. There are many landmark images not in cleaned GLDv2.

3.3 Soft-Voting with spatial verification

Following 1st place solution to the Google Landmark Retrieval 2019 [15] and host-baseline-example [9] we can get the landmark id and score of a test image, after some parameter adjustment, we found SuperPoint [3] + SuperGlue [16] + pydegensac [4] was better than DELF local features [14] + KDTree + pydegensac combination. The threshold parameter t was set to 90 instead of 70.

We visualized the spatial verification results of image pairs, as shown in Figure 1, SuperPoint+SuperGlue+pydegensac is better.

The scored improved from 0.5927/0.5582 to 0.6146/0.5756, which can be top 10 on leaderboard.

3.4 Post-Processing

The competition metric is Global Average Precision (GAP), if non-landmark images (distractors) are predicted with higher confidence score than landmark images. Hence, it is essential to suppress the prediction confidence score of these distractors.

We tried rules from winner solutions of Google Landmark Recognition Competition 2019 [10] [15] [1] [5], the following are the effective rules:

1.Search top 3 non-landmark images from no-landmark image set for query an image, if the similarity of top3>0.3, then decrease the score of the query image [1].

2.If a landmark is predicted>20 times in the test set, then treat all the images of that landmark as non-landmarks [15].

As many features(ransac inliers, similarity to index images, similarity to non-landmark images etc.) can be used for determining whether an image is non-landmark or not, we developed a model which can be called re-rank model following the re-rank strategy in tweet sentiment extraction [13], the difference is that we use tree model instead of NN model.

After post-processing, the score of EfficientNet B7 model improved from 0.6146/0.5756 to 0.6797/0.6301 which can be top-3 on leaderboard.

4 RESULTS

models/methods	public	private
EfficientNet B7+1 scale Racsac	0.5890	0.5521
EfficientNet B7+3 scale Ransac	0.5927	0.5582
EfficientNet B7+SSD	0.6146	0.5756
EfficientNet B7+SSD+PP	0.6797	0.6301
Ensemble B7+B6+B5+Resnet152	0.6838	0.6375

Table 1: Leaderboard performance of methods. SSD denotes SuperPoint+SuperGlue+pydegensac. PP denotes Post-processing.

5 CONCLUSION

In this paper, We presented a detailed solution for the Google Landmark Recognition 2020. The solution used metric learning models which trained step by step on bigger and larger images to find candidate images from index set for a query image, then get landmark id and score by soft-voting based on similarity and spatial-verification. To suppress distractors, the post-processing rules and models played a important role in our final pipeline.

REFERENCES

[1] Kaibing Chen, Cheng Cui, Yuning Du, X. Meng, and H. Ren. 2019. 2nd Place and 2nd Place Solution to Kaggle Landmark Recognition andRetrieval Competition 2019. *ArXiv* abs/1906.03990 (2019).

[2] Jiankang Deng, J. Guo, and S. Zafeiriou. 2019. ArcFace: Additive Angular Margin Loss for Deep Face Recognition. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2019), 4685–4694.

[3] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. 2018. SuperPoint: Self-Supervised Interest Point Detection and Description. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)* (2018), 337–33712.

[4] M. Fischler and R. Bolles. 1981. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM* 24 (1981), 381–395.

[5] Yinzheng Gu and Chuanpeng Li. 2019. Team JL Solution to Google Landmark Recognition 2019. *ArXiv* abs/1906.11874 (2019).

[6] Yinzheng Gu, Chuanpeng Li, and J. Xie. 2018. Attention-aware Generalized Mean Pooling for Image Retrieval. *ArXiv* abs/1811.00202 (2018).

[7] Kaiming He, X. Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep Residual Learning for Image Recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2016), 770–778.

[8] SeungKee Jeon. 2020. 1st Place Solution to Google Landmark Retrieval 2020. *ArXiv* abs/2009.05132 (2020).

[9] kaggle. 2020. *Host-baseline.* https://www.kaggle.com/camaskew/host-baseline-example

[10] kaggle. 2020. *landmark-recognition-2019.* https://www.kaggle.com/c/landmark-recognition-2019/overview

[11] kaggle. 2020. *landmark-recognition-2020.* https://www.kaggle.com/c/landmark-recognition-2020/overview

[12] kaggle. 2020. *landmark-retrieval-2020.* https://www.kaggle.com/c/landmark-retrieval-2020/overview

[13] kaggle m.y. 2020. *re-ranking candidates.* https://www.kaggle.com/c/tweet-sentiment-extraction/discussion/159315

[14] Hyeonwoo Noh, André Araujo, Jack Sim, Tobias Weyand, and B. Han. 2017. Large-Scale Image Retrieval with Attentive Deep Local Features. *2017 IEEE International Conference on Computer Vision (ICCV)* (2017), 3476–3485.

[15] K. Ozaki and Shuhei Yokoo. 2019. Large-scale Landmark Retrieval/Recognition under a Noisy and Diverse Dataset. *ArXiv* abs/1906.04087 (2019).

[16] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. 2020. SuperGlue: Learning Feature Matching With Graph Neural Networks. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2020), 4937–4946.

[17] M. Tan and Quoc V. Le. 2019. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. *ArXiv* abs/1905.11946 (2019).

[18] Tobias Weyand, A. Araujo, Bingyi Cao, and Jack Sim. 2020. Google Landmarks Dataset v2 – A Large-Scale Benchmark for Instance-Level Recognition and Retrieval. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2020), 2572–2581.