Efficient large-scale image retrieval with deep feature orthogonality and Hybrid-Swin-Transformers

Christof Henkel
NVIDIA
chenkel@nvidia.com

Abstract

We present an efficient end-to-end pipeline for large-scale landmark recognition and retrieval. We show how to combine and enhance concepts from recent research in image retrieval and introduce two architectures especially suited for large-scale landmark identification. A model with deep orthogonal fusion of local and global features (DOLG) using an EfficientNet backbone as well as a novel Hybrid-Swin-Transformer is discussed and details how to train both architectures efficiently using a step-wise approach and a sub-center arcface loss with dynamic margins are provided. Furthermore, we elaborate a novel discriminative re-ranking methodology for image retrieval. The superiority of our approach was demonstrated by winning the recognition and retrieval track of the Google Landmark Competition 2021.

1 Introduction

The Google Landmark Dataset v2 (GLDv2) has become a popular dataset for evaluating performance of architectures and methods used for solving large-scale instance-level recognition tasks [15]. The original dataset consists of over five million images with over 200,000 classes, originating from local experts who uploaded to Wikimedia Commons. Besides its size, the dataset poses several interesting challenges such as long-tailed distribution of classes, intra-class variability and noisy labels. Since 2019, GLDv2 has been used to asses and test state-of-the art instance-level recognition methods as part of the Google Landmark Competition hosted on kaggle.com. The winning solution of 2019 led to a cleaned version of GLDv2, which is a subset with 1.5 million images containing 81,313 classes and will be denoted by GLDv2c in the following. Furthermore, GLDv2x will be used to denote the subset of GLDv2 which is restricted to the 81,313 classes present in GLDv2c but was not cleaned. The yearly competition is divided into a recognition and retrieval task. The recognition track is about correctly classifying a landmark for a set of test images, where a significant amount of non-landmarks used as distractors are present. It is evaluated using Global Average Precision (GAP) [11, 15] as metric. For retrieval the task is to find similar images in a database with respect to a given query image and is evaluated using mean Average Precision@100 (mAP). In contrast to recent years the 2021 competition is evaluated with special focus on a more representative test set[1]. The competition follows a code competition setup, where participants are asked to upload their solution code rather than raw predictions. The submitted code will infer on a hidden test set offline, where resources available in the offsite environment are restricted to 12h runtime using a two-core CPU and a P100 GPU. During the competition participants are evaluated on the test set and a part of the predictions is used to calculate the above mentioned metrics and show the best score for each participant on a public leaderboard. After the competition the score with respect to the remaining part is released and used to determine and display the final scoring (private leaderboard). For training our models we used pytorch with mixed precision using 8xV100 NVIDIA GPUs with distributed data parallel (DDP). Moreover, we use several implementations and pretrained weights from timm [16]. Our training code will be available online[2].

[1] see [7] for details

[2] https://github.com/ChristofHenkel/kaggle-landmark-2021-1st-place

2 Methodology

2.1 Validation strategy

For recognition track we use the same local validation as in the last years winners solution [6], which leverages the 2019 test set and respective post-competition released test labels. The retrieval task is assessed in a similar way using the 2019 test query and index dataset together with the post-competition released retrieval solution. However, in our local validation we only considered index images that are matches of any query image to significantly reduce the computation time for evaluation. With this approach we achieved a very good correlation between local validation and leaderboard. For both tasks we tracked the respective competition metric during experiments at the end of every training epoch.

2.2 Modeling

For both tracks we developed deep learning based architectures, that learn an image descriptor, a high dimensional feature vector, which enables to differentiate between a larger number of classes yet allows for intra-class variability. Although historically global- and local landmark descriptors are trained separately and predictions are combined in a two-stage fashion, attempts are made to not only include the training of local descriptors in a single architecture (e.g. [9], [1]) but also omit spatial verification and fuse global and local descriptors within a single-stage model (see [17]). Given a tight competition timeline and restricted inference run-time we focused on single-stage models resulting in a single image descriptor. However, our modeling efforts put local features in the focus as they are especially important for landmark identification.

In the following we present two architectures especially suited for large-scale image recognition/ retrieval with noisy data and high intra-class variability. Both conceptually share a large part of an EfficientNet [14] based convolutional neural network (CNN) encoder and a sub-center arcface classification head with dynamic margins [3], which was shown to be superior to classical arcface as demonstrated by the 3rd place 2020 recognition solution [5]. For training we resize all image to a square size and apply shift, scale, rotate and cutout augmentation. We use Adam optimizer with weight decay and learning rate and batch size varying per model. We follow a cosine annealing learning rate schedule with one warm-up epoch.

2.2.1 DOLG-EfficientNet with sub-center arcface

We implemented DOLG [17], but with some adjustments to improve the performance. Firstly, we used an EfficientNet encoder, which was pretrained on ImageNet. We added the local branch after the third EfficientNet block and extract 1024-dimensional local features using three different dilated convolutions, where dilation parameters differ per model. The local features of the three dilated convolutions are concatenated in feature dimension and self-attended using spatial-2d-attention. The local features are then fused orthogonally with the global feature vector, which resulted from a GeM pooling [12] of the fourth EfficientNet block output projected to 1024 dimensions.[3] The fused features are aggregated using average pooling before they are fed into a neck consisting of a fully connected layer followed by batch-norm and parameterized ReLU activation (FC-BN-PReLU)[4] resulting in a 512-dimensional single descriptor. For training, this single descriptor is fed into a sub-center (k=3) arcface head with dynamic margins predicting 81,313 classes.

Our DOLG-EfficientNet models are trained following a 3-step procedure. Firstly, the models are trained for ten epochs on GLDv2c using a small image size. Then training is continued for 30-40 epochs on the more noisy GLDv2x using a medium image size. Finally, the models are finetuned for a few epochs on a large image size also using GLDv2x.

2.2.2 Hybrid-Swin-Transformer with sub-center arcface

The second architecture leverages recent advances in using transformers for computer vision problems. We appended a vision transformer to a CNN-encoder resulting in a Hybrid-CNN-Transformer model. As such the CNN-part can be interpreted as a local feature extractor, while the transformer-part acts as a graph neural net on those local features aggregating them to a single vector. More precisely we used a Swin-Transformer[8] as it especially flexible at various scales. As input for the transformer

[3] see [17] for details of local branch and orthogonal fusion module

[4] see neck Option-D from [4] for detailed description and rational

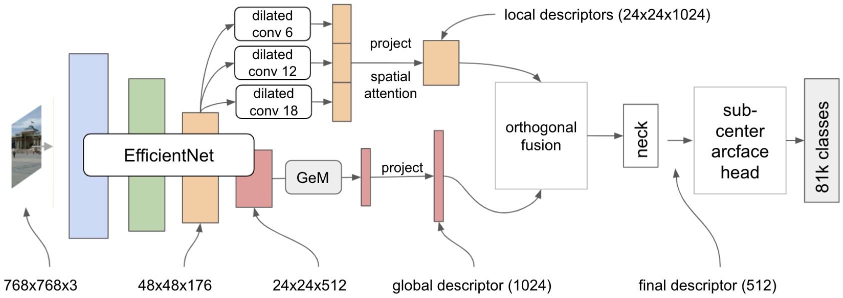

Figure 1: Model architecture DOLG-EfficientNet-B5

we use the output of a few blocks of EfficientNet which is flatten, position embedded and projected to match the transformer dimensions and extended with a virtual token representing the aggregated information. After passing through the Swin-Transformer we fed the features of the virtual token into a FC-BN-PReLU neck to derive the final descriptor. The hybrid model is trained using an sub-center arcface head with 81,313 classes and dynamic margins.

When combining an EfficientNet encoder and a Swin-Transformer which both have been pre-trained on ImageNet individually, a careful training recipe is important to avoid overflow and other training issues resulting in NaNs especially when using mixed precision. We used the following four-step approach. Firstly, initialize the transformer-neck-head part by training on a small image size using GLDv2c for 10 epochs. Next, exchange the transformers orignal patch embedding with a 2-block EfficientNet encoder, freeze the previously transformer-neck-head part and train the newly added CNN part for one epoch on medium size images. thirdly, unfreeze and train the whole model for 30-40 epochs using GLDv2x. Finally, insert a further pretrained EfficientNet block between the CNN and Swin-Transformer and finetune the whole model for a few epochs using large images and GLDv2x.

2.3 Submission ensemble

The winning submission for recognition track is an ensemble of eight models including three DOLG and three Hybrid-Swin-Transformer models with varying EfficientNet backbones and input image sizes. We additionally trained two pure EfficientNet models following code and instructions provided by the 3rd place team of google landmark recognition competition 2020 [5] where models are trained on the full GLDv2. For the winning submission of the retrieval track we used nearly the same ensemble but exchanged one of the pure EfficientNet models with an EfficientNet trained following the procedure of 2nd place team of google landmark recognition competition 2020 [2]. Table 1 gives an overview of backbones, image sizes, data used and resulting scores. Instead of increasing the image size for the last training step, we reduced the stride in the first convolutional layer from (2,2) to (1,1) for some models, being especially beneficial for small original images.

model	image size	stride	data	private score recognition	public score recognition	private score retrieval	public score retrieval
DOLG-EfficientNet-B5	768	2	GLDv2x	0.476	0.497	0.478	0.464
DOLG-EfficientNet-B6	768	2	GLDv2x	0.476	0.479	0.474	0.454
DOLG-EfficientNet-B7	448	1	GLDv2x	0.465	0.484	0.470	0.458
EfficientNet-B3-Swin-Base-224	896	2	GLDv2x	0.462	0.487	0.481	0.454
EfficientNet-B5-Swin-Base-224	448	1	GLDv2x	0.462	0.482	0.476	0.443
EfficientNet-B6-Swin-Base-384	384	1	GLDv2x	0.467	0.492	0.487	0.462
EfficientNet-B3	768	2	GLDv2	0.463	0.487		
EfficientNet-B6	512	2	GLDv2	0.470	0.484	0.454	0.441
EfficientNet-B5	704	2	GLDv2x			0.459	0.428
Ensemble Recognition				0.513	0.534		
Ensemble Retrieval						0.537	0.518

Table 1: Overview of model ensemble

2.4 Inference

For both tracks we used an retrieval approach for prediction, where for each query image most similar reference images are searched in database of index images using cosine similarity between L2-normalized image descriptors. For the recognition task the train set is used as index im-

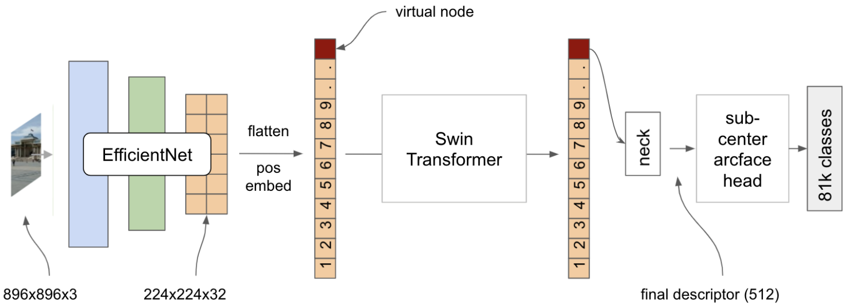

Figure 2: Hybrid-Swin-Transformer, exemplary shown for EfficientNet-B5-Swin-Base224

age database and the landmark label of the most similar train images are used as prediction for a given test image. In contrast, for retrieval an additional database of index images to retrieve most similar images from is predefined. For recognition track, to add more intra-class variety, we expand the offsite train set with full GLDv2 and landmark images from WIT [13], both filtered to contain only landmarks of the offline train set. For retrieval we further extend with the index set of the 2019 retrieval competition.

2.4.1 Retrieval post-processing

In order to retrieve most similar index images given a query image using our ensemble we rank all index images using cosine similarity of the 512-dimensional descriptor for each model individually, resulting in query-index pair scores. We then re-rank the index images by a discriminative re-ranking procedure derived from the one introduced in the retrieval task of the 2019 Google Landmark competition by the winning team [10], where in a first step we label the query and index images using their top3 cosine similarity to a given train set. However, in contrast to the hard re-ranking procedure illustrated in [10], we used a soft up-rank procedure by adding the top3 index-train cosine similarity to the query-index scores if labels of query and index image match. We saw further benefit when additionally performing a soft down-ranking procedure. We implemented the down-ranking by substracting 0.1 times the top3 index-train cosine similarity if labels of query and index image do not match. For each model in our ensemble we extract the top750 index image ids and related scores for each query image using the above method and aggregate the resulting 6000 scores by summing per each image id before we take the top100 as a final prediction.

2.4.2 Recognition post-processing

We use our ensemble to extract eight 512-dimensional vectors for each train and test image. Vectors are then averaged per model type (DOLG-EfficientNet, Hybrid-Swin-Transformer, pure EfficientNet) resulting in three 512-dimensional vectors which are then concatenated leading to a 1536-dimensional image descriptor. We use cosine similarity to find the closest training images for each test image and apply the post-processing from [6] for re-ranking and non-landmark identification, which results in the final predictions.

3 Conclusion

We presented several improvements to previous approaches for large-scale landmark identification leading to winning both tracks of the 2021 Google landmark competition. We showed how deep orthogonal features or vision transformers help to efficiently leverage local information extracted with a CNN backbone and stressed the superiority of sub-center arcface when confronted with long-tailed class distributions and intra-class variability. We confirmed the applicability of the re-ranking and non-landmark identification of [6] to the more balanced 2021 test set and explained a novel soft discriminative up- and down-ranking procedure for the retrieval task.

References

[1] B. Cao, A. Araujo, and J. Sim. Unifying deep local and global features for efficient image search. *arXiv preprint arXiv:2001.05027*, 2020.

[2] S. Dai. Instance level recognition. `https://github.com/bestfitting/instance_level_recognition`, 2020.

[3] J. Deng, J. Guo, T. Liu, M. Gong, and S. Zafeiriou. Sub-center arcface: Boosting face recognition by large-scale noisy web faces. In *European Conference on Computer Vision*, pages 741–757. Springer, 2020.

[4] J. Deng, J. Guo, N. Xue, and S. Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4690–4699, 2019.

[5] Q. Ha, B. Liu, F. Liu, and P. Liao. Google landmark recognition 2020 competition third place solution, 2020.

[6] C. Henkel and P. Singer. Supporting large-scale image recognition with out-of-domain samples. *arXiv preprint arXiv:2010.01650*, 2020.

[7] Z. Kim, A. Araujo, B. Cao, C. Askew, J. Sim, M. Green, N. Yilla, and T. Weyand. Towards a fairer landmark recognition dataset. *arXiv preprint arXiv:2108.08874*, 2021.

[8] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo. Swin transformer: Hierarchical vision transformer using shifted windows, 2021.

[9] H. Noh, A. Araujo, J. Sim, T. Weyand, and B. Han. Large-scale image retrieval with attentive deep local features. In *Proceedings of the IEEE international conference on computer vision*, pages 3456–3465, 2017.

[10] K. Ozaki and S. Yokoo. Large-scale landmark retrieval/recognition under a noisy and diverse dataset, 2019.

[11] F. Perronnin, Y. Liu, and J.-M. Renders. A family of contextual measures of similarity between distributions with application to image retrieval. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2358–2365. IEEE, 2009.

[12] F. Radenović, G. Tolias, and O. Chum. Fine-tuning cnn image retrieval with no human annotation. *IEEE transactions on pattern analysis and machine intelligence*, 41(7):1655–1668, 2018.

[13] K. Srinivasan, K. Raman, J. Chen, M. Bendersky, and M. Najork. Wit: Wikipedia-based image text dataset for multimodal multilingual machine learning. *arXiv preprint arXiv:2103.01913*, 2021.

[14] M. Tan and Q. Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning*, pages 6105–6114. PMLR, 2019.

[15] T. Weyand, A. Araujo, B. Cao, and J. Sim. Google landmarks dataset v2-a large-scale benchmark for instance-level recognition and retrieval. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2575–2584, 2020.

[16] R. Wightman. Pytorch image models. `https://github.com/rwightman/pytorch-image-models`, 2019.

[17] M. Yang, D. He, M. Fan, B. Shi, X. Xue, F. Li, E. Ding, and J. Huang. Dolg: Single-stage image retrieval with deep orthogonal fusion of local and global features. *arXiv preprint arXiv:2108.02927*, 2021.