Deep Local and Global Image Features

This project presents code for deep local and global image feature methods, which are particularly useful for the computer vision tasks of instance-level recognition and retrieval. These were introduced in the DELF, Detect-to-Retrieve, DELG and Google Landmarks Dataset v2 papers.

We provide Tensorflow code for building and training models, and python code for image retrieval and local feature matching. Pre-trained models for the landmark recognition domain are also provided.

If you make use of this codebase, please consider citing the following papers:

DELF:

```
"Large-Scale Image Retrieval with Attentive Deep Local Features",
H. Noh, A. Araujo, J. Sim, T. Weyand and B. Han,
Proc. ICCV'17
```

Detect-to-Retrieve:

```
"Detect-to-Retrieve: Efficient Regional Aggregation for Image Search",
M. Teichmann*, A. Araujo*, M. Zhu and J. Sim,
Proc. CVPR'19
```

DELG:

```
"Unifying Deep Local and Global Features for Image Search",
B. Cao*, A. Araujo* and J. Sim,
Proc. ECCV'20
```

GLDv2:

```
"Google Landmarks Dataset v2 - A Large-Scale Benchmark for Instance-Level Recognition
and Retrieval",
T. Weyand*, A. Araujo*, B. Cao and J. Sim,
Proc. CVPR'20
```

News

- [Jul'20] Check out our ECCV'20 paper: "Unifying Deep Local and Global Features for Image Search"
- [Apr'20] Check out our CVPR'20 paper: "Google Landmarks Dataset v2 - A Large-Scale Benchmark for Instance-Level Recognition and Retrieval"

- [Jun'19] DELF achieved 2nd place in CVPR Visual Localization challenge (Local Features track). See our slides here.
- [Apr'19] Check out our CVPR'19 paper: "Detect-to-Retrieve: Efficient Regional Aggregation for Image Search"
- [Jun'18] DELF achieved state-of-the-art results in a CVPR'18 image retrieval paper: Radenovic et al., "Revisiting Oxford and Paris: Large-Scale Image Retrieval Benchmarking".
- [Apr'18] DELF was featured in ModelDepot
- [Mar'18] DELF is now available in TF-Hub

Datasets

We have two Google-Landmarks dataset versions:

- Initial version (v1) can be found here. In includes the Google Landmark Boxes which were described in the Detect-to-Retrieve paper.
- Second version (v2) has been released as part of two Kaggle challenges: Landmark Recognition and Landmark Retrieval. It can be downloaded from CVDF here. See also the CVPR'20 paper on this new dataset version.

If you make use of these datasets in your research, please consider citing the papers mentioned above.

Installation

To be able to use this code, please follow these instructions to properly install the DELF library.

Quick start

Pre-trained models

We release several pre-trained models. See instructions in the following sections for examples on how to use the models.

DELF pre-trained on the Google-Landmarks dataset v1 (link). Presented in the Detect-to-Retrieve paper. Boosts performance by ~4% mAP compared to ICCV'17 DELF model.

DELG pre-trained on the Google-Landmarks dataset v1 (R101-DELG, R50-DELG). Presented in the DELG paper.

DELG pre-trained on the Google-Landmarks dataset v2 (clean) (R101-DELG, R50-DELG). Presented in the DELG paper.

RN101-ArcFace pre-trained on the Google-Landmarks dataset v2 (train-clean) (link). Presented in the GLDv2 paper.

DELF pre-trained on Landmarks-Clean/Landmarks-Full dataset (link). Presented in the DELF paper, model was trained on the dataset released by the DIR paper.

Faster-RCNN detector pre-trained on Google Landmark Boxes (link). Presented in the Detect-to-Retrieve paper.

MobileNet-SSD detector pre-trained on Google Landmark Boxes (link). Presented in the Detect-to-Retrieve paper.

Besides these, we also release pre-trained codebooks for local feature aggregation. See the Detect-to-Retrieve instructions for details.

DELF extraction and matching

Please follow these instructions. At the end, you should obtain a nice figure showing local feature matches, as:

DELF training

Please follow these instructions.

DELG

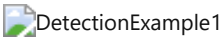
Please follow these instructions. At the end, you should obtain image retrieval results on the Revisited Oxford/Paris datasets.

GLDv2 baseline

Please follow these instructions. At the end, you should obtain image retrieval results on the Revisited Oxford/Paris datasets.

Landmark detection

Please follow these instructions. At the end, you should obtain a nice figure showing a detection, as:



Detect-to-Retrieve

Please follow these instructions. At the end, you should obtain image retrieval results on the Revisited Oxford/Paris datasets.

Code overview

DELF/D2R/DELG/GLD code is located under the `delf` directory. There are two directories therein, `protos` and `python`.

`delf/protos`

This directory contains protobufs for local feature aggregation (`aggregation_config.proto`), serializing detected boxes (`box.proto`), serializing float tensors (`datum.proto`), configuring DELF/DELG extraction (`delf_config.proto`), serializing local features (`feature.proto`).

`delf/python`

This directory contains files for several different purposes, such as: reading/writing tensors/features (`box_io.py` , `datum_io.py` , `feature_io.py`), local feature aggregation extraction and similarity computation (`feature_aggregation_extractor.py` , `feature_aggregation_similarity.py`) and helper functions for image/feature loading/processing (`utils.py` , `feature_extractor.py`).

The subdirectory `delf/python/examples` contains sample scripts to run DELF/DELG feature extraction/matching (`extractor.py` , `extract_features.py` , `match_images.py`) and object detection (`detector.py` , `extract_boxes.py`). `delf_config_example.pbtxt` shows an example instantiation of the DelfConfig proto, used for DELF feature extraction.

The subdirectory `delf/python/delg` contains sample scripts/configs related to the DELG paper: `extract_features.py` for local+global feature extraction (with and example `delg_gld_config.pbtxt`) and `perform_retrieval.py` for performing retrieval/scoring.

The subdirectory `delf/python/detect_to_retrieve` contains sample scripts/configs related to the Detect-to-Retrieve paper, for feature/box extraction/aggregation/clustering (`aggregation_extraction.py`, `boxes_and_features_extraction.py`, `cluster_delf_features.py`, `extract_aggregation.py`, `extract_index_boxes_and_features.py`, `extract_query_features.py`), image retrieval/reranking (`perform_retrieval.py`, `image_reranking.py`), along with configs used for feature extraction/aggregation (`delf_gld_config.pbtxt`, `index_aggregation_config.pbtxt`, `query_aggregation_config.pbtxt`) and Revisited Oxford/Paris dataset parsing/evaluation (`dataset.py`).

The subdirectory `delf/python/google_landmarks_dataset` contains sample scripts/modules for computing GLD metrics (`metrics.py`, `compute_recognition_metrics.py`, `compute_retrieval_metrics.py`), GLD file IO (`dataset_file_io.py`) / reproducing results from the GLDv2 paper (`rn101_af_gldv2clean_config.pbtxt` and the instructions therein).

The subdirectory `delf/python/training` contains sample scripts/modules for performing model training (`train.py`) based on a ResNet50 DELF model (`model/resnet50.py`, `model/delf_model.py`), also presenting relevant model exporting scripts and associated utils (`model/export_model.py`, `model/export_global_model.py`, `model/export_model_utils.py`) and dataset downloading/preprocessing (`download_dataset.sh`, `build_image_dataset.py`, `datasets/googlelandmarks.py`).

Besides these, other files in the different subdirectories contain tests for the various modules.

Maintainers

André Araujo (@andrefaraujo)

Release history

Jul, 2020

- Full TF2 support. Only one minor `compat.v1` usage left. Updated instructions to require TF2.2
- Refactored / much improved training code, with very detailed, step-by-step instructions

Thanks to contributors: Dan Anghel, Barbara Fusinska and André Araujo.

May, 2020

- Codebase is now Python3-first
- DELG model/code released
- GLDv2 baseline model released

Thanks to contributors: Barbara Fusinska and André Araujo.

April, 2020 (version 2.0)

- Initial DELF training code released.
- Codebase is now fully compatible with TF 2.1.

Thanks to contributors: Arun Mukundan, Yuewei Na and André Araujo.

April, 2019

Detect-to-Retrieve code released.

Includes pre-trained models to detect landmark boxes, and DELF model pre-trained on Google Landmarks v1 dataset.

Thanks to contributors: André Araujo, Marvin Teichmann, Menglong Zhu, Jack Sim.

October, 2017

Initial release containing DELF-v1 code, including feature extraction and matching examples. Pre-trained DELF model from ICCV'17 paper is released.

Thanks to contributors: André Araujo, Hyeonwoo Noh, Youlong Cheng, Jack Sim.