Apply Deep Metric Learning to Few-Shot Image Segmentation

Individual Project

Explore the docs »

View Demo · Report Bug · Request Feature

▾ Table of Contents

About The Project

This project is code implementation of my final project report in KCL. In this repo, deep metric learning is applied to few shot segmentation. According to the information provided by the mask, the distance between feature vectors of the same category in the high-dimensional space is as small as possible, and the distance between feature vectors of different categories is close or far away.

Built With

- pytorch
- pytorch-metric-learning
- segmentation_models

Folder Structure

- dataloaders/
    - `coco.py`, `common.py`, `customized.py` are the class of dataset and dataloaders.
    - `transforms.py` contains different transform functions.
- models/

    - `fewshot.py` is the class definition of train model, including model initialization and forward function.
    - `few_proto.py` is the class definition of test prototype model.
    - `vgg.py` is the class definition of VGG feature extractor.

- util/

    - `metric.py` is the metric functions to evaluate models.
    - `utils.py` contains some other functions used in the model training and testing.

- experiments/ contains the training, testing and visualize scripts.

- `config.py` is the configuration file to set the training and testing parameters.

- `train_metric.py` is the main function to start training the model.

- `test_proto.py` and `test_metric_knn.py` are the main functions to start testing model.

- `visualization.py` is the script to visualize testing result.

Getting Started

To get a local copy up and running follow these simple steps.

Prerequisites

- Python 3.6+
- PyTorch 1.9.0
- pytorch-metric-learning
- segmentation_models
- torchvision 0.2.1+
- pycocotools
- sacred 0.7.5
- tqdm 4.32.2

Installation

1. Download the source code
2. Install requirements

```
pip install -r requirements.txt
```

Preparation

1. Prepare Pascal-5i dataset

    - Download VOC2012 from officail website and put them under ./data/Pascal/

    - Download SegmentationClassAug, SegmentationObjectAug, ScribbleAugAuto from here and put them under ./data/Pascal/VOCdevkit/VOC2012/

   - Download Segmentation from here and use it to replace VOCdevkit/VOC2012/ImageSets/Segmentation.

2. Prepare MSCOCO dataset
   - Download COCO 2014 form officail website and put them under ./data/COCO/
3. Prepare pretrain model
   - Download pretrained model here and put them under ./

Usage

1. Train the model

```
sh experiments/train.sh
```

2. Test the model

```
sh experiments/test.sh
```

3. Visualize

```
sh experiments/vis.sh
```

Roadmap

See the open issues for a list of proposed features (and known issues).

Contributing

Contributions are what make the open source community such an amazing place to learn, inspire, and create. Any contributions you make are **greatly appreciated**.

1. Fork the Project
2. Create your Feature Branch (`git checkout -b feature/AmazingFeature`)
3. Commit your Changes (`git commit -m 'Add some AmazingFeature'`)
4. Push to the Branch (`git push origin feature/AmazingFeature`)
5. Open a Pull Request

License

Distributed under the MIT License. See `LICENSE` for more information.

Contact

Project Link: https://github.com/xumengen/FSGM

References

- PANet
- pytorch-metric-learning
- segmentation_models