Skeleton Code for SegNet Implementation Assignment, Computer Vision Class, ETH Zurich

Overview

- `image_segmentation/`: includes training and validation scripts.
- `lib/`: contains core functions, data preparation, model definition, and utility functions.

Installation

0. If you already followed instructions of assignment 3, then you can skip to step 4.
1. Install either miniconda or anaconda.
2. Create a virtual environment and activate it:

   ```
   conda create -n CV22_Image_Segmentation python=3.8
   conda activate CV22_Image_Segmentation
   ```

3. Install PyTorch 1.9.1 from the official website. CPU-only version is sufficient for this assignment.
4. Install dependencies.

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

   NOTE: TensorBorad and tensorboardX may not be compatible on some platforms. If you encounter difficulties when installing them, just remove them from `requirements.txt`. They are used solely for visualizing your results, which is optional (but helpful) for building your model.

5. Add current project directory (which we will later denote as ${ROOT}) to PYTHONPATH environment variable. On Linux you can do the following:

   ```
   export PYTHONPATH=${PYTHONPATH}:${PWD}
   ```

Data Preparation for Multi-digit MNIST Dataset

1. Unzip the multi-digit-MNIST dataset (multi-digit-mnist-dataset.zip) to ${ROOT}. You should have the following directory structure after unzipping:

   ```
   ${ROOT}
    `-- data
        `-- multi-digit-mnist
            |-- batch00001.mat
            |...
            |-- testset001.mat
            |...
   ```

Implement Simplified SegNet Architecture

Complete the skeleton code in `lib/models/seg_net.py`. Please refer to the hints in comments, the original paper, as well as the slides from lab assignment session for details.

Training on Multi-digit MNIST Dataset

To train your model, run:

```
python image_segmentation/train_mnist.py
```

Validate the model

To validate the model after training, run:

```
python image_segmentation/validate_mnist.py
```

(Optional) If you installed TensorBoard and tensorboardX, then you should have TensorBoard logs saved to `out/logs`. You can monitor the logs (loss curves, validation visualization, etc.) on http://localhost:6006 via:

```
tensorboard --logdir out/logs --port 6006
```

References

1. The overall structure of the code (roughly) follows Simple Baselines for Human Pose Estimation and Tracking.
2. Multi-digit MNIST dataset was created using the script from Recurrent Pixel Embedding for Instance Grouping.

License