Running DeepLab on PASCAL VOC 2012 Semantic Segmentation Dataset

This page walks through the steps required to run DeepLab on PASCAL VOC 2012 on a local machine.

Download dataset and convert to TFRecord

We have prepared the script (under the folder `datasets`) to download and convert PASCAL VOC 2012 semantic segmentation dataset to TFRecord.

```
# From the tensorflow/models/research/deeplab/datasets directory.
sh download_and_convert_voc2012.sh
```

The converted dataset will be saved at ./deeplab/datasets/pascal_voc_seg/tfrecord

Recommended Directory Structure for Training and Evaluation

```
+ datasets
  + pascal_voc_seg
    + VOCdevkit
      + VOC2012
        + JPEGImages
        + SegmentationClass
    + tfrecord
    + exp
      + train_on_train_set
        + train
        + eval
        + vis
```

where the folder `train_on_train_set` stores the train/eval/vis events and results (when training DeepLab on the PASCAL VOC 2012 train set).

Running the train/eval/vis jobs

A local training job using `xception_65` can be run with the following command:

```
# From tensorflow/models/research/
python deeplab/train.py \
    --logtostderr \
    --training_number_of_steps=30000 \
    --train_split="train" \
    --model_variant="xception_65" \
    --atrous_rates=6 \
    --atrous_rates=12 \
    --atrous_rates=18 \
    --output_stride=16 \
    --decoder_output_stride=4 \
    --train_crop_size="513,513" \
    --train_batch_size=1 \
    --dataset="pascal_voc_seg" \
    --tf_initial_checkpoint=${PATH_TO_INITIAL_CHECKPOINT} \
    --train_logdir=${PATH_TO_TRAIN_DIR} \
    --dataset_dir=${PATH_TO_DATASET}
```

where ${PATH_TO_INITIAL_CHECKPOINT} is the path to the initial checkpoint (usually an ImageNet pretrained checkpoint), ${PATH_TO_TRAIN_DIR} is the directory in which training checkpoints and events will be written to, and ${PATH_TO_DATASET} is the directory in which the PASCAL VOC 2012 dataset resides.

Note that for {train,eval,vis}.py:

1. In order to reproduce our results, one needs to use large batch size ($>$ 12), and set fine_tune_batch_norm = True. Here, we simply use small batch size during training for the purpose of demonstration. If the users have limited GPU memory at hand, please fine-tune from our provided checkpoints whose batch norm parameters have been trained, and use smaller learning rate with fine_tune_batch_norm = False.
2. The users should change atrous_rates from [6, 12, 18] to [12, 24, 36] if setting output_stride=8.
3. The users could skip the flag, `decoder_output_stride`, if you do not want to use the decoder structure.

A local evaluation job using `xception_65` can be run with the following command:

```
# From tensorflow/models/research/
python deeplab/eval.py \
    --logtostderr \
    --eval_split="val" \
    --model_variant="xception_65" \
    --atrous_rates=6 \
    --atrous_rates=12 \
    --atrous_rates=18 \
    --output_stride=16 \
    --decoder_output_stride=4 \
    --eval_crop_size="513,513" \
    --dataset="pascal_voc_seg" \
    --checkpoint_dir=${PATH_TO_CHECKPOINT} \
    --eval_logdir=${PATH_TO_EVAL_DIR} \
    --dataset_dir=${PATH_TO_DATASET}
```

where ${PATH_TO_CHECKPOINT} is the path to the trained checkpoint (i.e., the path to train_logdir), ${PATH_TO_EVAL_DIR} is the directory in which evaluation events will be written to, and ${PATH_TO_DATASET} is the directory in which the PASCAL VOC 2012 dataset resides.

A local visualization job using `xception_65` can be run with the following command:

```
# From tensorflow/models/research/
python deeplab/vis.py \
    --logtostderr \
    --vis_split="val" \
    --model_variant="xception_65" \
    --atrous_rates=6 \
    --atrous_rates=12 \
    --atrous_rates=18 \
    --output_stride=16 \
    --decoder_output_stride=4 \
    --vis_crop_size="513,513" \
    --dataset="pascal_voc_seg" \
    --checkpoint_dir=${PATH_TO_CHECKPOINT} \
    --vis_logdir=${PATH_TO_VIS_DIR} \
    --dataset_dir=${PATH_TO_DATASET}
```

where ${PATH_TO_CHECKPOINT} is the path to the trained checkpoint (i.e., the path to train_logdir), ${PATH_TO_VIS_DIR} is the directory in which evaluation events will be written to, and ${PATH_TO_DATASET} is the directory in which the PASCAL VOC 2012 dataset resides. Note that if the users would like to save the segmentation results for evaluation server, set also_save_raw_predictions = True.

Running Tensorboard

Progress for training and evaluation jobs can be inspected using Tensorboard. If using the recommended directory structure, Tensorboard can be run using the following command:

```
tensorboard --logdir=${PATH_TO_LOG_DIRECTORY}
```

where `${PATH_TO_LOG_DIRECTORY}` points to the directory that contains the train, eval, and vis directories (e.g., the folder `train_on_train_set` in the above example). Please note it may take Tensorboard a couple minutes to populate with data.

Example

We provide a script to run the {train,eval,vis,export_model}.py on the PASCAL VOC 2012 dataset as an example. See the code in local_test.sh for details.

```
# From tensorflow/models/research/deeplab
sh local_test.sh
```