

Panoptic Segmentation

Description

Panoptic Segmentation combines the two distinct vision tasks - semantic segmentation and instance segmentation. These tasks are unified such that, each pixel in the image is assigned the label of the class it belongs to, and also the instance identifier of the object it is a part of.

Environment setup

The code can be run on multiple GPUs or TPUs with different distribution strategies. See the TensorFlow distributed training [guide](#) for an overview of `tf.distribute`.

The code is compatible with TensorFlow 2.6+. See `requirements.txt` for all prerequisites.

```
$ git clone https://github.com/tensorflow/models.git
$ cd models
$ pip3 install -r official/requirements.txt
$ export PYTHONPATH=$(pwd)
```

Preparing Dataset

```
$ ./official/vision/beta/data/process_coco_panoptic.sh <path-to-data-directory>
```

Launch Training

```
$ export MODEL_DIR="gs://<path-to-model-directory>"
$ export TPU_NAME="<tpu-name>"
$ export ANNOTATION_FILE="gs://<path-to-coco-annotation-json>"
$ export TRAIN_DATA="gs://<path-to-train-data>"
$ export EVAL_DATA="gs://<path-to-eval-data>"
$ export OVERRIDES="task.validation_data.input_path=${EVAL_DATA},\
task.train_data.input_path=${TRAIN_DATA},\
task.annotation_file=${ANNOTATION_FILE},\
runtime.distribution_strategy=tpu"

$ python3 train.py \
  --experiment panoptic_fpn_coco \
  --config_file configs/experiments/r50fpn_1x_coco.yaml \
  --mode train \
  --model_dir $MODEL_DIR \
  --tpu $TPU_NAME \
  --params_override=$OVERRIDES
```

Launch Evaluation

```

$ export MODEL_DIR="gs://<path-to-model-directory>"
$ export NUM_GPUS="<number-of-gpus>"
$ export PRECISION="<floating-point-precision>"
$ export ANNOTATION_FILE="gs://<path-to-coco-annotation-json>"
$ export TRAIN_DATA="gs://<path-to-train-data>"
$ export EVAL_DATA="gs://<path-to-eval-data>"
$ export OVERRIDES="task.validation_data.input_path=${EVAL_DATA}, \
task.train_data.input_path=${TRAIN_DATA}, \
task.annotation_file=${ANNOTATION_FILE}, \
runtime.distribution_strategy=mirrored, \
runtime.mixed_precision_dtype=${PRECISION}, \
runtime.num_gpus=$NUM_GPUS"

$ python3 train.py \
  --experiment panoptic_fpn_coco \
  --config_file configs/experiments/r50fpn_1x_coco.yaml \
  --mode eval \
  --model_dir $MODEL_DIR \
  --params_override=$OVERRIDES

```

Note: The [PanopticSegmentationGenerator](#) layer uses dynamic shapes and hence generating panoptic masks is not supported on Cloud TPUs. Running evaluation on Cloud TPUs is not supported for the same reason. However, training is supported on both Cloud TPUs and GPUs.

Pretrained Models

Panoptic FPN

Backbone	Schedule	Experiment name	Box mAP	Mask mAP	Overall PQ	Things PQ	Stuff PQ	Checkp
ResNet-50	1x	panoptic_fpn_coco	38.19	34.25	39.14	45.42	29.65	
ResNet-50	3x	panoptic_fpn_coco	40.64	36.29	40.91	47.68	30.69	

Note: Here 1x schedule refers to ~12 epochs

Citation

```

@misc{kirillov2019panoptic,
  title={Panoptic Feature Pyramid Networks},
  author={Alexander Kirillov and Ross Girshick and Kaiming He and Piotr Dollár},
  year={2019},
  eprint={1901.02446},
  archivePrefix={arXiv},
  primaryClass={cs.CV}
}

```