Mask R-CNN with deep mask heads

This project brings insights from the DeepMAC model into the Mask-RCNN architecture. Please see the paper The surprising impact of mask-head architecture on novel class segmentation for more details.

Code structure

- This folder contains forks of a few Mask R-CNN files and repurposes them to support deep mask heads.
- To see the benefits of using deep mask heads, it is important to train the mask head with only groundtruth boxes. This is configured via the `task.model.use_gt_boxes_for_masks` flag.
- Architecture of the mask head can be changed via the config value `task.model.mask_head.convnet_variant`. Supported values are `"default"`, `"hourglass20"`, `"hourglass52"`, and `"hourglass100"`.
- The flag `task.model.mask_head.class_agnostic` trains the model in class agnostic mode and `task.allowed_mask_class_ids` controls which classes are allowed to have masks during training.
- Majority of experiments and ablations from the paper are perfomed with the DeepMAC model in the Object Detection API code base.

Prerequisites

Prepare dataset

Use create_coco_tf_record.py to create the COCO dataset. The data needs to be store in a Google cloud storage bucket so that it can be accessed by the TPU.

Start a TPU v3-32 instance

See TPU Quickstart for instructions. An example command would look like:

```
ctpu up --name <tpu-name> --zone <zone> --tpu-size=v3-32 --tf-version nightly
```

This model requires TF version `>= 2.5`. Currently, that is only available via a `nightly` build on Cloud.

Install requirements

SSH into the TPU host with `gcloud compute ssh <tpu-name>` and execute the following.

```
$ git clone https://github.com/tensorflow/models.git
$ cd models
$ pip3 install -r official/requirements.txt
```

Training Models

The configurations can be found in the `configs/experiments` directory. You can launch a training job by executing.

```
$ export
CONFIG=./official/projects/deepmac_maskrcnn/configs/experiments/deep_mask_head_rcnn_vo

$ export MODEL_DIR="gs://<path-for-checkpoints>"
```

```
$ export ANNOTAION_FILE="gs://<path-to-coco-annotation-json>"
$ export TRAIN_DATA="gs://<path-to-train-data>"
$ export EVAL_DATA="gs://<path-to-eval-data>"
# Overrides to access data. These can also be changed in the config file.
$ export OVERRIDES="task.validation_data.input_path=${EVAL_DATA},\
task.train_data.input_path=${TRAIN_DATA},\
task.annotation_file=${ANNOTAION_FILE},\
runtime.distribution_strategy=tpu"

$ python3 -m official.projects.deepmac_maskrcnn.train \
  --logtostderr \
  --mode=train_and_eval \
  --experiment=deep_mask_head_rcnn_resnetfpn_coco \
  --model_dir=$MODEL_DIR \
  --config_file=$CONFIG \
  --params_override=$OVERRIDES\
  --tpu=<tpu-name>
```

`CONFIG_FILE` can be any file in the `configs/experiments` directory. When using SpineNet models, please specify `--experiment=deep_mask_head_rcnn_spinenet_coco`

Note: The default eval batch size of 32 discards some samples during validation. For accurate vaidation statistics, launch a dedicated eval job on TPU `v3-8` and set batch size to 8.

Configurations

In the following table, we report the Mask mAP of our models on the non-VOC classes when only training with masks for the VOC calsses. Performance is measured on the `coco-val2017` set.

Backbone	Mask head	Config name	Mask mAP
ResNet-50	Default	`deep_mask_head_rcnn_voc_r50.yaml`	25.9
ResNet-50	Hourglass-52	`deep_mask_head_rcnn_voc_r50_hg52.yaml`	33.1
ResNet-101	Hourglass-52	`deep_mask_head_rcnn_voc_r101_hg52.yaml`	34.4
SpienNet-143	Hourglass-52	`deep_mask_head_rcnn_voc_spinenet143_hg52.yaml`	38.7

See also

- DeepMAC model in the Object Detection API code base.
- Project website - git.io/deepmac

Citation

```
@misc{birodkar2021surprising,
      title={The surprising impact of mask-head architecture on novel class segmentation},
      author={Vighnesh Birodkar and Zhichao Lu and Siyang Li and Vivek Rathod and Jonathan Huang},
      year={2021},
```

```
      eprint={2104.00613},
      archivePrefix={arXiv},
      primaryClass={cs.CV}
}
```