TensorFlow 1 Detection Model Zoo

We provide a collection of detection models pre-trained on the COCO dataset, the Kitti dataset, the Open Images dataset, the AVA v2.1 dataset the iNaturalist Species Detection Dataset and the Snapshot Serengeti Dataset. These models can be useful for out-of-the-box inference if you are interested in categories already in those datasets. They are also useful for initializing your models when training on novel datasets.

In the table below, we list each such pre-trained model including:

- a model name that corresponds to a config file that was used to train this model in the `samples/configs` directory,
- a download link to a tar.gz file containing the pre-trained model,
- model speed --- we report running time in ms per 600x600 image (including all pre and post-processing), but please be aware that these timings depend highly on one's specific hardware configuration (these timings were performed using an Nvidia GeForce GTX TITAN X card) and should be treated more as relative timings in many cases. Also note that desktop GPU timing does not always reflect mobile run time. For example Mobilenet V2 is faster on mobile devices than Mobilenet V1, but is slightly slower on desktop GPU.
- detector performance on subset of the COCO validation set, Open Images test split, iNaturalist test split, or Snapshot Serengeti LILA.science test split. as measured by the dataset-specific mAP measure. Here, higher is better, and we only report bounding box mAP rounded to the nearest integer.
- Output types (`Boxes` , and `Masks` if applicable)

You can un-tar each tar.gz file via, e.g.,:

```
tar -xzvf ssd_mobilenet_v1_coco.tar.gz
```

Inside the un-tar'ed directory, you will find:

- a graph proto (`graph.pbtxt`)
- a checkpoint (`model.ckpt.data-00000-of-00001` , `model.ckpt.index` , `model.ckpt.meta`)
- a frozen graph proto with weights baked into the graph as constants (`frozen_inference_graph.pb`) to be used for out of the box inference (try this out in the Jupyter notebook!)
- a config file (`pipeline.config`) which was used to generate the graph. These directly correspond to a config file in the samples/configs) directory but often with a modified score threshold. In the case of the heavier Faster R-CNN models, we also provide a version of the model that uses a highly reduced number of proposals for speed.
- Mobile model only: a TfLite file (`model.tflite`) that can be deployed on mobile devices.

Some remarks on frozen inference graphs:

- If you try to evaluate the frozen graph, you may find performance numbers for some of the models to be slightly lower than what we report in the below tables. This is because we discard detections with scores below a threshold (typically 0.3) when creating the frozen graph. This corresponds effectively to picking a point on the precision recall curve of a detector (and discarding the part past that point), which negatively impacts standard mAP metrics.
- Our frozen inference graphs are generated using the v1.12.0 release version of TensorFlow; this being said, each frozen inference graph can be regenerated using your current version of TensorFlow by re-running the exporter, pointing it at the model directory as well as the corresponding config file in samples/configs.

COCO-trained models

Model name	Speed (ms)	COCO mAP[^1]	Outputs
ssd_mobilenet_v1_coco	30	21	Boxes
ssd_mobilenet_v1_0.75_depth_coco ☆	26	18	Boxes
ssd_mobilenet_v1_quantized_coco ☆	29	18	Boxes
ssd_mobilenet_v1_0.75_depth_quantized_coco ☆	29	16	Boxes
ssd_mobilenet_v1_ppn_coco ☆	26	20	Boxes
ssd_mobilenet_v1_fpn_coco ☆	56	32	Boxes
ssd_resnet_50_fpn_coco ☆	76	35	Boxes
ssd_mobilenet_v2_coco	31	22	Boxes
ssd_mobilenet_v2_quantized_coco	29	22	Boxes
ssdlite_mobilenet_v2_coco	27	22	Boxes
ssd_inception_v2_coco	42	24	Boxes
faster_rcnn_inception_v2_coco	58	28	Boxes
faster_rcnn_resnet50_coco	89	30	Boxes
faster_rcnn_resnet50_lowproposals_coco	64		Boxes
rfcn_resnet101_coco	92	30	Boxes
faster_rcnn_resnet101_coco	106	32	Boxes
faster_rcnn_resnet101_lowproposals_coco	82		Boxes
faster_rcnn_inception_resnet_v2_atrous_coco	620	37	Boxes
faster_rcnn_inception_resnet_v2_atrous_lowproposals_coco	241		Boxes
faster_rcnn_nas	1833	43	Boxes
faster_rcnn_nas_lowproposals_coco	540		Boxes
mask_rcnn_inception_resnet_v2_atrous_coco	771	36	Masks
mask_rcnn_inception_v2_coco	79	25	Masks
mask_rcnn_resnet101_atrous_coco	470	33	Masks
mask_rcnn_resnet50_atrous_coco	343	29	Masks

Note: The asterisk (☆) at the end of model name indicates that this model supports TPU training.

Note: If you download the tar.gz file of quantized models and un-tar, you will get different set of files - a checkpoint, a config file and tflite frozen graphs (txt/binary).

Mobile models

Model name	Pixel 1 Latency (ms)	COCO mAP	Outputs
ssd_mobiledet_cpu_coco	113	24.0	Boxes
ssd_mobilenet_v2_mnasfpn_coco	183	26.6	Boxes
ssd_mobilenet_v3_large_coco	119	22.6	Boxes
ssd_mobilenet_v3_small_coco	43	15.4	Boxes

Pixel4 Edge TPU models

Model name	Pixel 4 Edge TPU Latency (ms)	COCO mAP (fp32/uint8)	Outputs
ssd_mobiledet_edgetpu_coco	6.9	25.9/25.6	Boxes
ssd_mobilenet_edgetpu_coco	6.6	-/24.3	Boxes

Pixel4 DSP models

Model name	Pixel 4 DSP Latency (ms)	COCO mAP (fp32/uint8)	Outputs
ssd_mobiledet_dsp_coco	12.3	28.9/28.8	Boxes

Kitti-trained models

Model name	Speed (ms)	Pascal mAP@0.5	Outputs
faster_rcnn_resnet101_kitti	79	87	Boxes

Open Images-trained models

Model name	Speed (ms)	Open Images mAP@0.5[^2]	Outputs
faster_rcnn_inception_resnet_v2_atrous_oidv2	727	37	Boxes
faster_rcnn_inception_resnet_v2_atrous_lowproposals_oidv2	347		Boxes
facessd_mobilenet_v2_quantized_open_image_v4 [^3]	20	73 (faces)	Boxes

Model name	Speed (ms)	Open Images mAP@0.5[^4]	Outputs
faster_rcnn_inception_resnet_v2_atrous_oidv4	425	54	Boxes
ssd_mobilenetv2_oidv4	89	36	Boxes
ssd_resnet_101_fpn_oidv4	237	38	Boxes

iNaturalist Species-trained models

Model name	Speed (ms)	Pascal mAP@0.5	Outputs
faster_rcnn_resnet101_fgvc	395	58	Boxes
faster_rcnn_resnet50_fgvc	366	55	Boxes

AVA v2.1 trained models

Model name	Speed (ms)	Pascal mAP@0.5	Outputs
faster_rcnn_resnet101_ava_v2.1	93	11	Boxes

Snapshot Serengeti Camera Trap trained models

Model name	COCO mAP@0.5	Outputs
faster_rcnn_resnet101_snapshot_serengeti	38	Boxes
context_rcnn_resnet101_snapshot_serengeti	56	Boxes

Pixel 6 Edge TPU models

Model name	Pixel 6 Edge TPU Speed (ms)	Pixel 6 Speed with Post-processing on CPU (ms)	COCO 2017 mAP (uint8)	Outputs
spaghettinet_edgetpu_s	1.3	1.8	26.3	Boxes
spaghettinet_edgetpu_m	1.4	1.9	27.4	Boxes
spaghettinet_edgetpu_l	1.7	2.1	28.0	Boxes

[^1]: See MSCOCO evaluation protocol. The COCO mAP numbers, with the exception of the Pixel 6 Edge TPU models, are evaluated on COCO 14 minival set (note that our split is different from COCO 17 Val). A full list of image ids used in our split could be found here. [^2]: This is PASCAL mAP with a slightly different way of true positives computation: see Open Images evaluation protocols, oid_V2_detection_metrics. [^3]: Non-face boxes are dropped during training and non-face groundtruth boxes are ignored when evaluating. [^4]: This is Open Images Challenge metric: see Open Images evaluation protocols, oid_challenge_detection_metrics.