Supported object detection evaluation protocols

The TensorFlow Object Detection API currently supports three evaluation protocols, that can be configured in `EvalConfig` by setting `metrics_set` to the corresponding value.

PASCAL VOC 2010 detection metric

`EvalConfig.metrics_set='pascal_voc_detection_metrics'`

The commonly used mAP metric for evaluating the quality of object detectors, computed according to the protocol of the PASCAL VOC Challenge 2010-2012. The protocol is available here.

Weighted PASCAL VOC detection metric

`EvalConfig.metrics_set='weighted_pascal_voc_detection_metrics'`

The weighted PASCAL metric computes the mean average precision as the average precision when treating all classes as a single class. In comparison, PASCAL metrics computes the mean average precision as the mean of the per-class average precisions.

For example, the test set consists of two classes, "cat" and "dog", and there are ten times more boxes of "cat" than those of "dog". According to PASCAL VOC 2010 metric, performance on each of the two classes would contribute equally towards the final mAP value, while for the Weighted PASCAL VOC metric the final mAP value will be influenced by frequency of each class.

PASCAL VOC 2010 instance segmentation metric

`EvalConfig.metrics_set='pascal_voc_instance_segmentation_metrics'`

Similar to Pascal VOC 2010 detection metric, but computes the intersection over union based on the object masks instead of object boxes.

Weighted PASCAL VOC instance segmentation metric

`EvalConfig.metrics_set='weighted_pascal_voc_instance_segmentation_metrics'`

Similar to the weighted pascal voc 2010 detection metric, but computes the intersection over union based on the object masks instead of object boxes.

COCO detection metrics

`EvalConfig.metrics_set='coco_detection_metrics'`

The COCO metrics are the official detection metrics used to score the COCO competition and are similar to Pascal VOC metrics but have a slightly different implementation and report additional statistics such as mAP at IOU thresholds of .5:.95, and precision/recall statistics for small, medium, and large objects. See the pycocotools repository for more details.

COCO mask metrics

`EvalConfig.metrics_set='coco_mask_metrics'`

Similar to the COCO detection metrics, but computes the intersection over union based on the object masks instead of object boxes.

Open Images V2 detection metric

`EvalConfig.metrics_set='oid_V2_detection_metrics'`

This metric is defined originally for evaluating detector performance on Open Images V2 dataset and is fairly similar to the PASCAL VOC 2010 metric mentioned above. It computes interpolated average precision (AP) for each class and averages it among all classes (mAP).

The difference to the PASCAL VOC 2010 metric is the following: Open Images annotations contain `group-of` ground-truth boxes (see Open Images data description), that are treated differently for the purpose of deciding whether detections are "true positives", "ignored", "false positives". Here we define these three cases:

A detection is a "true positive" if there is a non-group-of ground-truth box, such that:

- The detection box and the ground-truth box are of the same class, and intersection-over-union (IoU) between the detection box and the ground-truth box is greater than the IoU threshold (default value 0.5).
  Illustration of handling non-group-of boxes:
  alt groupof_case_eval
  - yellow box - ground-truth box;
  - green box - true positive;
  - red boxes - false positives.
- This is the highest scoring detection for this ground truth box that satisfies the criteria above.

A detection is "ignored" if it is not a true positive, and there is a `group-of` ground-truth box such that:

- The detection box and the ground-truth box are of the same class, and the area of intersection between the detection box and the ground-truth box divided by the area of the detection is greater than 0.5. This is intended to measure whether the detection box is approximately inside the group-of ground-truth box.

Illustration of handling `group-of` boxes:
alt groupof_case_eval

- yellow box - ground-truth box;
- grey boxes - two detections on cars, that are ignored;
- red box - false positive.

A detection is a “false positive” if it is neither a “true positive” nor “ignored”.

Precision and recall are defined as:

- Precision = number-of-true-positives/(number-of-true-positives + number-of-false-positives)
- Recall = number-of-true-positives/number-of-non-group-of-boxes

Note that detections ignored as firing on a `group-of` ground-truth box do not contribute to the number of true positives.

The labels in Open Images are organized in a hierarchy. Ground-truth bounding-boxes are annotated with the most specific class available in the hierarchy. For example, “car” has two children “limousine” and “van”. Any other kind of car is annotated as “car” (for example, a sedan). Given this convention, the evaluation software treats all classes independently, ignoring the hierarchy. To achieve high performance values, object detectors should output bounding-boxes labelled in the same manner.

The old metric name is DEPRECATED. `EvalConfig.metrics_set='open_images_V2_detection_metrics'`

OID Challenge Object Detection Metric

`EvalConfig.metrics_set='oid_challenge_detection_metrics'`

The metric for the OID Challenge Object Detection Metric 2018/2019 Object Detection track. The description is provided on the Open Images Challenge website.

The old metric name is DEPRECATED. `EvalConfig.metrics_set='oid_challenge_object_detection_metr`

OID Challenge Visual Relationship Detection Metric

The metric for the OID Challenge Visual Relationship Detection Metric 2018,2019 Visual Relationship Detection track. The description is provided on the Open Images Challenge website. Note: this is currently a stand-alone metric, that can be used only through the `metrics/oid_vrd_challenge_evaluation.py` util.

OID Challenge Instance Segmentation Metric

`EvalConfig.metrics_set='oid_challenge_segmentation_metrics'`

The metric for the OID Challenge Instance Segmentation Metric 2019, Instance Segmentation track. The description is provided on the Open Images Challenge website.