Open Images Challenge Evaluation

The Object Detection API is currently supporting several evaluation metrics used in the Open Images Challenge 2018 and Open Images Challenge 2019. In addition, several data processing tools are available. Detailed instructions on using the tools for each track are available below.

NOTE: all data links are updated to the Open Images Challenge 2019.

Object Detection Track

The Object Detection metric protocol requires a pre-processing of the released data to ensure correct evaluation. The released data contains only leaf-most bounding box annotations and image-level labels. The evaluation metric implementation is available in the class `OpenImagesChallengeEvaluator`.

1. Download class hierarchy of Open Images Detection Challenge 2019 in JSON format.
2. Download ground-truth boundling boxes and image-level labels.
3. Run the following command to create hierarchical expansion of the bounding boxes and image-level label annotations:

```
HIERARCHY_FILE=/path/to/challenge-2019-label500-hierarchy.json
BOUNDING_BOXES=/path/to/challenge-2019-validation-detection-bbox
IMAGE_LABELS=/path/to/challenge-2019-validation-detection-human-imagelabels

python object_detection/dataset_tools/oid_hierarchical_labels_expansion.py \
    --json_hierarchy_file=${HIERARCHY_FILE} \
    --input_annotations=${BOUNDING_BOXES}.csv \
    --output_annotations=${BOUNDING_BOXES}_expanded.csv \
    --annotation_type=1

python object_detection/dataset_tools/oid_hierarchical_labels_expansion.py \
    --json_hierarchy_file=${HIERARCHY_FILE} \
    --input_annotations=${IMAGE_LABELS}.csv \
    --output_annotations=${IMAGE_LABELS}_expanded.csv \
    --annotation_type=2
```

1. If you are not using TensorFlow, you can run evaluation directly using your algorithm's output and generated ground-truth files. {value=4}

After step 3 you produced the ground-truth files suitable for running 'OID Challenge Object Detection Metric 2019' evaluation. To run the evaluation, use the following command:

```
INPUT_PREDICTIONS=/path/to/detection_predictions.csv
OUTPUT_METRICS=/path/to/output/metrics/file

python models/research/object_detection/metrics/oid_challenge_evaluation.py \
    --input_annotations_boxes=${BOUNDING_BOXES}_expanded.csv \
    --input_annotations_labels=${IMAGE_LABELS}_expanded.csv \
    --
input_class_labelmap=object_detection/data/oid_object_detection_challenge_500_label_map.pbtxt
 \
    --input_predictions=${INPUT_PREDICTIONS} \
    --output_metrics=${OUTPUT_METRICS} \
```

Note that predictions file must contain the following keys: ImageID,LabelName,Score,XMin,XMax,YMin,YMax

For the Object Detection Track, the participants will be ranked on:

- "OpenImagesDetectionChallenge_Precision/mAP@0.5IOU"

To use evaluation within TensorFlow training, use metric name `oid_challenge_detection_metrics` in the evaluation config.

Instance Segmentation Track

The Instance Segmentation metric can be directly evaluated using the ground-truth data and model predictions. The evaluation metric implementation is available in the class `OpenImagesChallengeEvaluator`.

1. Download class hierarchy of Open Images Instance Segmentation Challenge 2019 in JSON format.

2. Download ground-truth bounding boxes and image-level labels.

3. Download instance segmentation files for the validation set (see Open Images Challenge Downloads page). The download consists of a set of .zip archives containing binary .png masks. Those should be transformed into a single CSV file in the format:

   ImageID,LabelName,ImageWidth,ImageHeight,XMin,YMin,XMax,YMax,IsGroupOf,Mask where Mask is MS COCO RLE encoding, compressed with zip, and re-coded with base64 encoding of a binary mask stored in .png file. See an example implementation of the encoding function here.

4. Run the following command to create hierarchical expansion of the instance segmentation, bounding boxes and image-level label annotations: {value=4}

```
HIERARCHY_FILE=/path/to/challenge-2019-label300-hierarchy.json
BOUNDING_BOXES=/path/to/challenge-2019-validation-detection-bbox
IMAGE_LABELS=/path/to/challenge-2019-validation-detection-human-imagelabels

python object_detection/dataset_tools/oid_hierarchical_labels_expansion.py \
    --json_hierarchy_file=${HIERARCHY_FILE} \
    --input_annotations=${BOUNDING_BOXES}.csv \
    --output_annotations=${BOUNDING_BOXES}_expanded.csv \
    --annotation_type=1

python object_detection/dataset_tools/oid_hierarchical_labels_expansion.py \
    --json_hierarchy_file=${HIERARCHY_FILE} \
    --input_annotations=${IMAGE_LABELS}.csv \
    --output_annotations=${IMAGE_LABELS}_expanded.csv \
    --annotation_type=2

python object_detection/dataset_tools/oid_hierarchical_labels_expansion.py \
    --json_hierarchy_file=${HIERARCHY_FILE} \
    --input_annotations=${INSTANCE_SEGMENTATIONS}.csv \
    --output_annotations=${INSTANCE_SEGMENTATIONS}_expanded.csv \
    --annotation_type=1
```

1. If you are not using TensorFlow, you can run evaluation directly using your algorithm's output and generated ground-truth files. {value=4}

```
INPUT_PREDICTIONS=/path/to/instance_segmentation_predictions.csv
OUTPUT_METRICS=/path/to/output/metrics/file

python models/research/object_detection/metrics/oid_challenge_evaluation.py \
    --input_annotations_boxes=${BOUNDING_BOXES}_expanded.csv \
    --input_annotations_labels=${IMAGE_LABELS}_expanded.csv \
    --input_class_labelmap=object_detection/data/oid_object_detection_challenge_500_label_map.pbtxt \
    --input_predictions=${INPUT_PREDICTIONS} \
    --input_annotations_segm=${INSTANCE_SEGMENTATIONS}_expanded.csv
    --output_metrics=${OUTPUT_METRICS} \
```

Note that predictions file must contain the following keys: ImageID,ImageWidth,ImageHeight,LabelName,Score,Mask

Mask must be encoded the same way as groundtruth masks.

For the Instance Segmentation Track, the participants will be ranked on:

- "OpenImagesInstanceSegmentationChallenge_Precision/mAP@0.5IOU"

Visual Relationships Detection Track

The Visual Relationships Detection metrics can be directly evaluated using the ground-truth data and model predictions. The evaluation metric implementation is available in the class `VRDRelationDetectionEvaluator`, `VRDPhraseDetectionEvaluator`.

1. Download the ground-truth visual relationships annotations and image-level labels.
2. Run the follwing command to produce final metrics:

```
INPUT_ANNOTATIONS_BOXES=/path/to/challenge-2018-train-vrd.csv
INPUT_ANNOTATIONS_LABELS=/path/to/challenge-2018-train-vrd-labels.csv
INPUT_PREDICTIONS=/path/to/predictions.csv
INPUT_CLASS_LABELMAP=/path/to/oid_object_detection_challenge_500_label_map.pbtxt
INPUT_RELATIONSHIP_LABELMAP=/path/to/relationships_labelmap.pbtxt
OUTPUT_METRICS=/path/to/output/metrics/file

echo "item { name: '/m/02gy9n' id: 602 display_name: 'Transparent' }
item { name: '/m/05z87' id: 603 display_name: 'Plastic' }
item { name: '/m/0dnr7' id: 604 display_name: '(made of)Textile' }
item { name: '/m/04lbp' id: 605 display_name: '(made of)Leather' }
item { name: '/m/083vt' id: 606 display_name: 'Wooden'}
">>${INPUT_CLASS_LABELMAP}

echo "item { name: 'at' id: 1 display_name: 'at' }
item { name: 'on' id: 2 display_name: 'on (top of)' }
item { name: 'holds' id: 3 display_name: 'holds' }
item { name: 'plays' id: 4 display_name: 'plays' }
item { name: 'interacts_with' id: 5 display_name: 'interacts with' }
item { name: 'wears' id: 6 display_name: 'wears' }
item { name: 'is' id: 7 display_name: 'is' }
item { name: 'inside_of' id: 8 display_name: 'inside of' }
item { name: 'under' id: 9 display_name: 'under' }
item { name: 'hits' id: 10 display_name: 'hits' }
"> ${INPUT_RELATIONSHIP_LABELMAP}

python object_detection/metrics/oid_vrd_challenge_evaluation.py \
    --input_annotations_boxes=${INPUT_ANNOTATIONS_BOXES} \
    --input_annotations_labels=${INPUT_ANNOTATIONS_LABELS} \
    --input_predictions=${INPUT_PREDICTIONS} \
    --input_class_labelmap=${INPUT_CLASS_LABELMAP} \
    --input_relationship_labelmap=${INPUT_RELATIONSHIP_LABELMAP} \
    --output_metrics=${OUTPUT_METRICS}
```

Note that predictions file must contain the following keys:
ImageID,LabelName1,LabelName2,RelationshipLabel,Score,XMin1,XMax1,YMin1,YMax1,XMin2,XMax2,YMin2,YMax2

The participants of the challenge will be evaluated by weighted average of the following three metrics:

- "VRDMetric_Relationships_mAP@0.5IOU"
- "VRDMetric_Relationships_Recall@50@0.5IOU"
- "VRDMetric_Phrases_mAP@0.5IOU"