cw-eval Documentation

Release 0.3

David Lindenbaum and Nick Weir

Dec 31, 2018

Contents:

Author CosmiQ Works

Version 0.3

CHAPTER 1

Core functionality

class cw_eval.baseeval.**EvalBase**(*ground_truth_vector_file*)

Object to test IoU for predictions and ground truth polygons.

Parameters **ground_truth_vector_file** (*str*) – Path to .geojson file for ground truth.

eval_iou(*miniou=0.5, iou_field_prefix='iou_score', ground_truth_class_field='', calculate_class_scores=True, class_list=['all']*)

Evaluate IoU between the ground truth and proposals.

Parameters

- **miniou** (*float, optional*) – Minimum intersection over union score to qualify as a successful object detection event. Defaults to `0.5`.
- **iou_field_prefix** (*str, optional*) – The name of the IoU score column in `self.proposal_GDF`. Defaults to `"iou_score"`.
- **ground_truth_class_field** (*str, optional*) – The column in `self.ground_truth_GDF` that indicates the class of each polygon. Required if using `calculate_class_scores`.
- **calculate_class_scores** (*bool, optional*) – Should class-by-class scores be calculated? Defaults to `True`.
- **class_list** (*list, optional*) – List of classes to be scored. Defaults to `['all']` (score all classes).

Returns

scoring_dict_list – list of score output dicts for each image in the ground truth and evaluated image datasets. The dicts contain the following keys:

```
('class_id', 'iou_field', 'TruePos', 'FalsePos', 'FalseNeg',
'Precision', 'Recall', 'F1Score')
```

Return type list

eval_iou_spacenet_csv (*miniou=0.5*, *iou_field_prefix='iou_score'*, *imageIDField='ImageId'*, *debug=False*, *minArea=0*)

Evaluate IoU between the ground truth and proposals in CSVs.

Parameters

- **miniou** (*float , optional*) – Minimum intersection over union score to qualify as a successful object detection event. Defaults to `0.5`.
- **iou_field_prefix** (*str , optional*) – The name of the IoU score column in `self.proposal_GDF`. Defaults to `"iou_score"` .
- **imageIDField** (*str , optional*) – The name of the column corresponding to the image IDs in the ground truth data. Defaults to `"ImageId"`.
- **debug** (*bool , optional*) – Argument for verbose execution during debugging. Defaults to `False` (silent execution).
- **minArea** (*float or int , optional*) – Minimum area of a ground truth polygon to be considered during evaluation. Often set to `20` in SpaceNet competitions. Defaults to `0` (consider all ground truth polygons).

Returns

scoring_dict_list – list of score output dicts for each image in the ground truth and evaluated image datasets. The dicts contain the following keys:

```
('imageID', 'iou_field', 'TruePos', 'FalsePos', 'FalseNeg',
'Precision', 'Recall', 'F1Score')
```

Return type list

load_proposal (*proposal_vector_file, conf_field_list=['conf'], proposalCSV=False, pred_row_geo_value='PolygonWKT_Pix', conf_field_mapping=[]*)

Load in a proposal geojson or CSV.

Parameters

- **proposal_vector_file** (*str*) – Path to the file containing proposal vector objects. This can be a .geojson or a .csv.
- **conf_field_list** (*list, optional*) – List of columns corresponding to confidence value(s) in the proposal vector file. Defaults to `['conf']`.
- **proposalCSV** (*bool, optional*) – Is the proposal file a CSV? Defaults to no (`False`), in which case it's assumed to be a .geojson.
- **pred_row_geo_value** (*str, optional*) – The name of the geometry-containing column in the proposal vector file. Defaults to `'PolygonWKT_Pix'`. Note: this method assumes the geometry is in WKT format.
- **conf_field_mapping** (*dict, optional*) – `'__max_conf_class'` column value:class ID mapping dict for multiclass use. Only required in multiclass cases.

Returns

Return type `0` upon successful completion.

Notes

Loads in a .geojson or .csv-formatted file of proposal polygons for comparison to the ground truth and stores it as part of the `EvalBase` instance. This method assumes the geometry contained in the proposal file is in WKT format.

load_truth(*ground_truth_vector_file*, *truthCSV=False*, *truth_geo_value='PolygonWKT_Pix'*)
Load in the ground truth geometry data.

Parameters

- **ground_truth_vector_file** (*str*) – Path to the ground truth vector file. Must be either .geojson or .csv format.
- **truthCSV** (*bool, optional*) – Is the ground truth a CSV? Defaults to `False`, in which case it's assumed to be a .geojson.
- **truth_geo_value** (*str, optional*) – Column of the ground truth vector file that corresponds to geometry.

Returns

Return type `0` if it completes successfully.

Notes

Loads the ground truth vector data into the `EvalBase` instance.

cw_eval.baseeval.**eval_base**(*ground_truth_vector_file*, *csvFile=False*, *truth_geo_value='PolygonWKT_Pix'*)
Deprecated API to EvalBase.

Deprecated since version 0.3: Use *EvalBase* instead.

cw_eval.evalfunctions.**calculate_iou**(*pred_poly*, *test_data_GDF*)
Get the best intersection over union for a predicted polygon.

Parameters

- **pred_poly** (`shapely.Polygon`) – Prediction polygon to test.
- **test_data_GDF** (`geopandas.GeoDataFrame`) – GeoDataFrame of ground truth polygons to test `pred_poly` against.

Returns **iou_GDF** – A subset of `test_data_GDF` that overlaps `pred_poly` with an added column `iou_score` which indicates the intersection over union value.

Return type `geopandas.GeoDataFrame`

cw_eval.evalfunctions.**process_iou**(*pred_poly*, *test_data_GDF*, *remove_matching_element=True*)
Get the maximum intersection over union score for a predicted polygon.

Parameters

- **pred_poly** (`shapely.geometry.Polygon`) – Prediction polygon to test.
- **test_data_GDF** (`geopandas.GeoDataFrame`) – GeoDataFrame of ground truth polygons to test `pred_poly` against.
- **remove_matching_element** (*bool, optional*) – Should the maximum IoU row be dropped from `test_data_GDF`? Defaults to `True`.

Returns

Return type *This function doesn't currently return anything.*

CHAPTER 2

SpaceNet Challenge eval code

cw_eval.challenge_eval.off_nadir_dataset.**eval_off_nadir**(*prop_csv*, *truth_csv*, *imageColumns={}*, *miniou=0.5*, *minArea=20*)

Evaluate an off-nadir competition proposal csv.

Uses `EvalBase` to evaluate off-nadir challenge proposals. See `imageColumns` in the source code for how collects are broken into Nadir, Off-Nadir, and Very-Off-Nadir bins.

Parameters

- **prop_csv** (*str*) – Path to the proposal polygon CSV file.
- **truth_csv** (*str*) – Path to the ground truth polygon CSV file.
- **imageColumns** (*dict, optional*) – dict of `(collect: nadir bin)` pairs used to separate collects into sets. Nadir bin values must be one of `["Nadir", "Off-Nadir", "Very-Off-Nadir"]` . See source code for collect name options.
- **miniou** (*float, optional*) – Minimum IoU score between a region proposal and ground truth to define as a successful identification. Defaults to 0.5.
- **minArea** (*float or int, optional*) – Minimum area of ground truth regions to include in scoring calculation. Defaults to `20`.

Returns

results_DF [`pd.DataFrame`] Summary `pd.DataFrame` of score outputs grouped by nadir angle bin, along with the overall score.

results_DF_Full [`pd.DataFrame`] `pd.DataFrame` of scores by individual image chip across the ground truth and proposal datasets.

Return type results_DF, results_DF_Full

CHAPTER 3

Indices and tables

- genindex
- modindex
- search

Python Module Index

C

Index