Image Semantics Documentation

Release 0.0.0

Justin Brooks

Jan 21, 2019

Contents

Warning: Currently a work in progress!

Image understanding is widely used in many areas like satellite imaging, robotic technologies, sensory networks, medical and biomedical imaging, intelligent transportation systems, etc. Recently semantic analysis has become an active research topic aimed at resolving the gap between low level image features and high level semantics which is a promoting approach in image understanding.

With many image annotation semantics existing in the field of computer vision, it can become daunting to manage. This package provides the ability to convert and visualize many different types of annotation formats for object dectection and localization.

CHAPTER 1

API Reference

If you are looking for information on a specific function, class or method, this part of the documentation is for you.

1.1 API

This part of the documentation covers all the interfaces of Image Segmantic.

1.1.1 Annotation Object

class imantics.**Annotation**(*image*, *category*, *bbox=None*, *mask=None*, *polygons=None*, *id=0*, *color=None*, *metadata={}*)

Annotation is a marking on an image.

This class acts as a level ontop of *BBox*, *Mask* and *Polygons* to manage and generate other annotations or export formats.

area
Qantity that expresses the extent of a two-dimensional figure

array
Numpy array boolean mask repsentation of the annotations

bbox
BBox repsentation of the annotations

export(*style='coco'*)
Exports object into specified style

classmethod from_bbox(*image*, *category*, *bbox*)
Creates annotation from bounding box

Parameters

- **image** (*Image*) – image assoicated with annotation

- **category** (Category) – category to label annotation
- **polygons** (*BBox*, list, tuple) – bbox to create annotation from

classmethod from_mask(*image*, *category*, *mask*)
Creates annotation class from a mask

Parameters

- **image** (*Image*) – image assoicated with annotation
- **category** (Category) – category to label annotation
- **mask** (*Mask*, numpy.ndarray, list) – mask to create annotation from

classmethod from_polygons(*image*, *category*, *polygons*)
Creates annotation from polygons

Accepts following format for lists:

```
# Segmentation Format
[
    [x1, y1, x2, y2, x3, y3,...],
    [x1, y1, x2, y2, x3, y3,...],
    ...
]
```

or

```
# Point Format
[
    [[x1, y1], [x2, y2], [x3, y3],...],
    [[x1, y1], [x2, y2], [x3, y3],...],
    ...
]
```

No sepcificaiton is reqiured between which format is used

Parameters

- **image** (*Image*) – image assoicated with annotation
- **category** (Category) – category to label annotation
- **polygons** (*Polygons*, list) – polygons to create annotation from

mask

Returns annotation's *Mask* object

polygons
Polygons repsentation of the annotations

size
Tuple of width and height

1.1.2 Category Object

class imantics.**Dataset**(*name*, *images*, *id=0*, *metadata={}*)

add(*image*)
Adds image(s) to the current dataset

Parameters **image** – image, list of images, or path to image(s)

export(*style='coco'*)
Exports object into specified style

1.1.3 Bounding Box Object

class imantics.**BBox**(*bbox*, *style=None*)
Bounding Box is an enclosing retangular box for a image marking

INSTANCE_TYPES = (<class 'numpy.ndarray'>, <class 'list'>, <class 'tuple'>)
Value types of *BBox*

MIN_MAX = 'minmax'
Bounding box format style [x1, y1, x2, y2]

WIDTH_HEIGHT = 'widthheight'
Bounding box format style [x1, y1, width, height]

1.1.4 Dataset Object

class imantics.**Dataset**(*name*, *images*, *id=0*, *metadata={}*)

add(*image*)
Adds image(s) to the current dataset

Parameters **image** – image, list of images, or path to image(s)

export(*style='coco'*)
Exports object into specified style

1.1.5 Image Object

class imantics.**Image**(*image_array*, *annotations=[]*, *path=''*, *id=0*, *metadata={}*)

add(*annotation*, *category=None*)
Adds a annotaiton, list of annotaitons, mask, polygon or bbox to current image. If annotation is not a Annotation a category is required List of non-Annotaiton objects will have the same category

Parameters

- **annotation** – annotaiton to add to current image
- **category** – required if annotation is not an Annotation object

export(*style='coco'*)
Exports object into specified style

classmethod from_path(*path*)
Returns an array of images if path is a directory Returns an image if path is a file

1.1.6 Mask Object

class imantics.**Mask**(*array*)
Mask class

contains(*item*)
Checks whether a point (tuple), array or mask is within current mask. Note: Masks and arrays must be fully contained to return True

Parameters **item** – object to check

Returns boolean if item is contained

intersect(*other*)
Intersects the array of the specified mask with this masks's array and returns the result as a new mask.

Parameters **other** – mask (or numpy array) to intersect with

Returns resulting mask

iou(*other*)
Intersect over union value of the specified masks

Parameters **other** – mask (or numpy array) to compute value with

Returns resulting float value

subtract(*other*)
Subtracts the array of the specified mask from this masks's array and returns the result as a new mask.

Parameters **other** – mask (or numpy array) to subtract

Retrn resulting mask

union(*other*)
Unites the array of the specified mask with this mask's array and returns the result as a new mask. :param other: mask (or numpy array) to unite with :return: resulting mask

1.1.7 Polygons Object

class imantics.**Polygons**(*polygons*)

Python Module Index

i

Index