
epic-masks

Release 0.0.1

Dec 01, 2020

Contents:

| | | |
|----------|----------------------------|-----------|
| 1 | API Reference | 1 |
| 1.1 | epic_kitchens | 1 |
| 2 | Installation | 5 |
| 3 | Usage | 7 |
| 4 | Indices and tables | 9 |
| | Python Module Index | 11 |
| | Index | 13 |

This page contains auto-generated API reference documentation¹.

1.1 epic_kitchens

1.1.1 Subpackages

`epic_kitchens.masks`

Submodules

`epic_kitchens.masks.io`

Module Contents

Functions

`load_detections`(filepath: Union[Path, str]) → Iterator[FrameObjectDetections]

`save_detections`(filepath: Union[Path, str], detections: List[FrameObjectDetections]) → None

`epic_kitchens.masks.io.load_detections` (filepath: Union[Path, str]) → Iterator[FrameObjectDetections]

`epic_kitchens.masks.io.save_detections` (filepath: Union[Path, str], detections: List[FrameObjectDetections]) → None

¹ Created with `sphinx-autoapi`

`epic_kitchens.masks.types`

Module Contents

Classes

BBox

ObjectDetection

FrameObjectDetections

```
class epic_kitchens.masks.types.BBox
```

```
    left :float
```

```
    top  :float
```

```
    right :float
```

```
    bottom :float
```

```
    static from_protobuf (bbox: pb.BBox)
```

```
    to_protobuf (self)
```

```
class epic_kitchens.masks.types.ObjectDetection
```

```
    bbox :BBox
```

```
    score :float
```

```
    pred_class :int
```

```
    mask :np.ndarray
```

```
    static from_protobuf (detection: pb.ObjectDetection)
```

```
    to_protobuf (self)
```

```
class epic_kitchens.masks.types.FrameObjectDetections
```

```
    video_id :str
```

```
    frame_number :int
```

```
    objects :List[ObjectDetection]
```

```
    static from_protobuf (video_id: str, frame_detections: pb.FrameObjectDetections)
```

```
    static from_protobuf_str (video_id: str, pb_str: bytes)
```

```
    to_protobuf (self)
```

`epic_kitchens.masks.visualisation`

Module Contents

Classes

DetectionRenderer

Functions

resize_mask(mask: np.ndarray, height: int, width: int, smooth: bool = True) → np.ndarray

resize_bbox(bbox: BBox, height: int, width: int) → Tuple[int, int, int, int]

epic_kitchens.masks.visualisation.**resize_mask** (mask: np.ndarray, height: int, width: int, smooth: bool = True) → np.ndarray

epic_kitchens.masks.visualisation.**resize_bbox** (bbox: BBox, height: int, width: int) → Tuple[int, int, int, int]

class epic_kitchens.masks.visualisation.**DetectionRenderer** (*display_mask:*
 bool = True, display_bbox: bool = True, score_threshold:
 float = 0, smooth_mask: bool = True)

render_detections (self, img: PIL.Image.Image, detection: FrameObjectDetections)

CHAPTER 2

Installation

```
$ pip install git+https://github.com/epic-kitchens/epic-kitchens-100-object-masks.git
```


CHAPTER 3

Usage

Visualise the masks like so:

```
from typing import Union
from pathlib import Path
import PIL.Image
from epic_kitchens.masks.io import load_detections
from epic_kitchens.masks.visualisation import DetectionRenderer

class LazyFrameLoader:
    def __init__(self, path: Union[Path, str], frame_template: str = 'frame_{:010d}.
↳ jpg'):
        self.path = Path(path)
        self.frame_template = frame_template

    def __getitem__(self, idx: int) -> PIL.Image.Image:
        return PIL.Image.open(str(self.path / self.frame_template.format(idx + 1)))

detections = load_detections('detections/P01_101.pkl')
frames = LazyFrameLoader('frames/P01_101')
renderer = DetectionRenderer()

frame_idx = 100
renderer.render_detections(frames[frame_idx], detections[frame_idx])
```

A Jupyter notebook example is included that demonstrates how to load detections and visualise them.

CHAPTER 4

Indices and tables

- `genindex`
- `search`

e

- `epic_kitchens`, [1](#)
- `epic_kitchens.masks`, [1](#)
- `epic_kitchens.masks.io`, [1](#)
- `epic_kitchens.masks.types`, [2](#)
- `epic_kitchens.masks.visualisation`, [2](#)

B

BBox (class in *epic_kitchens.masks.types*), 2
bbox (*epic_kitchens.masks.types.ObjectDetection* attribute), 2
bottom (*epic_kitchens.masks.types.BBox* attribute), 2

D

DetectionRenderer (class in *epic_kitchens.masks.visualisation*), 3

E

epic_kitchens (module), 1
epic_kitchens.masks (module), 1
epic_kitchens.masks.io (module), 1
epic_kitchens.masks.types (module), 2
epic_kitchens.masks.visualisation (module), 2

F

frame_number (*epic_kitchens.masks.types.FrameObjectDetection* attribute), 2
FrameObjectDetections (class in *epic_kitchens.masks.types*), 2
from_protobuf() (*epic_kitchens.masks.types.BBox* static method), 2
from_protobuf() (*epic_kitchens.masks.types.FrameObjectDetections* static method), 2
from_protobuf() (*epic_kitchens.masks.types.ObjectDetection* static method), 2
from_protobuf_str() (*epic_kitchens.masks.types.FrameObjectDetection* static method), 2

L

left (*epic_kitchens.masks.types.BBox* attribute), 2
load_detections() (in module *epic_kitchens.masks.io*), 1

M

mask (*epic_kitchens.masks.types.ObjectDetection* attribute), 2

O

ObjectDetection (class in *epic_kitchens.masks.types*), 2
in objects (*epic_kitchens.masks.types.FrameObjectDetections* attribute), 2

P

pred_class (*epic_kitchens.masks.types.ObjectDetection* attribute), 2

R

render_detections() (*epic_kitchens.masks.visualisation.DetectionRenderer* method), 3
resize_bbox() (in module *epic_kitchens.masks.visualisation*), 3
resize_mask() (in module *epic_kitchens.masks.visualisation*), 3
right (*epic_kitchens.masks.types.BBox* attribute), 2

S

save_detections() (in module *epic_kitchens.masks.io*), 1
score (*epic_kitchens.masks.types.ObjectDetection* attribute), 2

T

to_protobuf() (*epic_kitchens.masks.types.BBox* method), 2
to_protobuf() (*epic_kitchens.masks.types.FrameObjectDetections* method), 2
to_protobuf() (*epic_kitchens.masks.types.ObjectDetection* method), 2
top (*epic_kitchens.masks.types.BBox* attribute), 2

V

`video_id` (*epic_kitchens.masks.types.FrameObjectDetections*
attribute), [2](#)