
gpod Documentation

Release 0.1.2

Evgeny Nekrasov

December 18, 2016

1	Overview	3
2	Table of contents	5
2.1	API Reference	5
3	Indices and tables	11

Source code [GitHub project](#)

Bug tracker [GitHub issues](#)

License [BSD 3-Clause](#)

Generated December 18, 2016

Overview

GPOD is Python framework to build object detectors. It was developed with a focus on enabling fast experimentation.

Table of contents

2.1 API Reference

2.1.1 Detectors

class `gpod.Detector2D` (*loader*, *descriptor*, *classifier*, *frame*, *frame_step*=0.25, *scale_step*=0.75, *max_scale_steps*=100)

2D object detector. Gaussian pyramid image is used to perform multi-scale object detection.

Parameters **loader** : function

A user defined function to load images. It should accept string (path) and return ndarray with 2 or 3 dimensions.

descriptor : function

A user defined function to preprocess image patches before feeding it to a classifier. It should accept ndarray and return ndarray.

classifier : object

This is assumed to implement the scikit-learn classifier interface. Either classifier needs to provide a 'fit' and 'predict_proba' methods.

frame : (int, int)

The size of image patches to sample.

frame_step : float or (int, int), default=0.25

The step of frame moving. If float, frame_step is frame*frame_step.

scale_step : float, default=0.75

Downscale step in gaussian pyramid image.

max_scale_steps : int, default=100

Maximum number of downscale steps.

Attributes

loader	(function)
descriptor	(function)
classifier	(object)
X	(ndarray)
y	(ndarray)
classes_	(list of strings)

Methods

batch_detect (*targets*, *classes=None*, *threshold=0.5*, *post_processing='NMS'*, *overlap_threshold=0.3*)

Detect objects on images using Detector2D.classifier.

Parameters *targets* : list of strings or list of ndarray

List of images to detect objects. If string in list, path to the image. If ndarray in list, images itself.

classes : string or list of strings or set of strings, default=None

Which object classes need to detect. If None, all classes would be detected.

threshold : float, default=0.5

The minimum probability returned by classifier to recognize object.

post_processing : float, default='NMS'

Which post processing apply to the recognized objects. Options: 'None', 'NMS'.

overlap_threshold : float, default=0.3

A parameter for post processing. 0.0 - frames have no overlapping area, 1.0 frames are identical.

Returns *list of Mark2D* : list of Mark2D objects with frames of recognized objects

batch_detect_crop (*targets*, *classes*, *threshold=0.5*, *post_processing='NMS'*, *overlap_threshold=0.3*, *flatten=False*)

Detect objects on images using Detector2D.classifier and crop recognized object images.

Parameters *targets* : list of strings or list of ndarray

List of images to detect objects. If string in list, path to the image. If ndarray in list, images itself.

classes : string or list of strings or set of strings

Which object classes need to detect.

threshold : float, default=0.5

The minimum probability returned by classifier to recognize object.

post_processing : float, default='NMS'

Which post processing apply to the recognized objects. Options: 'None', 'NMS'.

overlap_threshold : float, default=0.3

A parameter for post processing. 0.0 - frames have no overlapping area, 1.0 frames are identical.

flatten : bool, default=False

Whether to flatten list of images.

Returns list : list of lists of ndarray or list of ndarray

batch_detect_mark (*targets, classes, threshold=0.5, post_processing='NMS', overlap_threshold=0.3, color=0*)

Detect objects on images using Detector2D.classifier and paint frames around objects.

Parameters targets : list of strings or list of ndarray

List of images to detect objects. If string in list, path to the image. If ndarray in list, images itself.

classes : string or list of strings or set of strings

Which object classes need to detect.

threshold : float, default=0.5

The minimum probability returned by classifier to recognize object.

post_processing : float, default='NMS'

Which post processing apply to the recognized objects. Options: 'None', 'NMS'.

overlap_threshold : float, default=0.3

A parameter for post processing. 0.0 - frames have no overlapping area, 1.0 frames are identical.

color : float or int or array like , default=0

A color to paint frames on image. If array like, len(color) must be equal to the number of image channels.

Returns list of ndarray : original images with painted frames

detect (*target, classes=None, threshold=0.5, post_processing='NMS', overlap_threshold=0.3*)

Detect objects on an image using Detector2D.classifier.

Parameters target : string or ndarray

Image to detect objects. If string, path to the image. If ndarray, image itself.

classes : string or list of strings or set of strings, default=None

Which object classes need to detect. If None, all classes would be detected.

threshold : float, default=0.5

The minimum probability returned by classifier to recognize object.

post_processing : float, default='NMS'

Which post processing apply to the recognized objects. Options: 'None', 'NMS'.

overlap_threshold : float, default=0.3

A parameter for post processing. 0.0 - frames have no overlapping area, 1.0 frames are identical.

Returns Mark2D : Mark2D object with frames of recognized objects

detect_crop (*target, classes, threshold=0.5, post_processing='NMS', overlap_threshold=0.3*)

Detect objects on an image using Detector2D.classifier and crop recognized object images.

Parameters target : string or ndarray

Image to detect objects. If string, path to the image. If ndarray, image itself.

classes : string or list of strings or set of strings

Which object classes need to detect.

threshold : float, default=0.5

The minimum probability returned by classifier to recognize object.

post_processing : float, default='NMS'

Which post processing apply to the recognized objects. Options: 'None', 'NMS'.

overlap_threshold : float, default=0.3

A parameter for post processing. 0.0 - frames have no overlapping area, 1.0 frames are identical.

Returns list of ndarray : list of cropped images

detect_mark (*target*, *classes*, *threshold*=0.5, *post_processing*='NMS', *overlap_threshold*=0.3, *color*=0)

Detect objects on an image using Detector2D.classifier and paint frames around objects.

Parameters target : string or ndarray

Image to detect objects. If string, path to the image. If ndarray, image itself.

classes : string or list of strings or set of strings

Which object classes need to detect. If None, all classes would be detected.

threshold : float, default=0.5

The minimum probability returned by classifier to recognize object.

post_processing : float, default='NMS'

Which post processing apply to the recognized objects. Options: 'None', 'NMS'.

overlap_threshold : float, default=0.3

A parameter for post processing. 0.0 - frames have no overlapping area, 1.0 frames are identical.

color : float or int or array like , default=0

A color to paint frames on image. If array like, len(color) must be equal to the number of image channels.

Returns ndarray : original image with painted frames

fit (*path*, *store_data*=True, *augmentation*=False, *augmentation_factor*=1, *crop*=False, *horizontal_flip*=False, *vertical_flip*=False, *custom_augmentation_func*=None)

Fit Detector2D.classifier with passed data. If crop is False images should be the same size equal to frame size, otherwise images could be equal or greater than frame size.

Parameters path : string, list of lists of strings or dict of lists of strings

If string, a path to folder with folders with images. Otherwise, strings are paths to individual images. If dict, key is a class name.

store_data : bool, default=True

Whether to store the data used to train classifier.

augmentation : bool, default=False

Whether to use augmentation.

augmentation_factor : int, default=1

How much samples extract from one image. Useful only when augmentation is True.

crop : bool, default=False

Whether to use crop. Useful only when augmentation is True.

horizontal_flip : bool, default=False

Whether to perform horizontal flip with probability 50%. Useful only when augmentation is True.

vertical_flip : bool, default=False

Whether to perform vertical flip with probability 50%. Useful only when augmentation is True.

custom_augmentation_func : function, default=None

A user defined function to modify images. It should accept ndarray and return ndarray. Useful only when augmentation is True.

Returns **Detector2D** : an instance of self

2.1.2 Mark Objects

class `gpod.Mark2D` (*list_of_frames=None*)

Mark2D object is a container for object frames on image. It supports slicing by string or set of strings to extract frames of some class or classes. It supports slicing by float range to extract frames of probabilities in specified range.

Parameters **list_of_frames** : list

A list of tuples: (x: int, y: int, size_x: int, size_y: int, probability : float, class: string)

Methods

apply_nms (*overlap_threshold*)

Apply non-maximum suppression (NMS).

Parameters **overlap_threshold** : float

A parameter for NMS. 0.0 - frames have no overlapping area, 1.0 frames are identical.

Returns **Mark2D** : Mark2D with filtered frames

get_list_of_frames ()

Returns **list** : list of frames

Indices and tables

- `genindex`

A

`apply_nms()` (`gpod.Mark2D` method), 9

B

`batch_detect()` (`gpod.Detector2D` method), 6

`batch_detect_crop()` (`gpod.Detector2D` method), 6

`batch_detect_mark()` (`gpod.Detector2D` method), 7

D

`detect()` (`gpod.Detector2D` method), 7

`detect_crop()` (`gpod.Detector2D` method), 7

`detect_mark()` (`gpod.Detector2D` method), 8

`Detector2D` (class in `gpod`), 5

F

`fit()` (`gpod.Detector2D` method), 8

G

`get_list_of_frames()` (`gpod.Mark2D` method), 9

M

`Mark2D` (class in `gpod`), 9