Comp Viz

Release 1.0.0

Lucas Hirt

Nov 26, 2022

CONTENTS:

CHAPTER

ONE

COMP_VIZ

1.1 comp_viz package

1.1.1 Subpackages

comp_viz.object_detection package

Submodules

comp_viz.object_detection.model module

class comp_viz.object_detection.model.**Model**(*network_name*)

Bases: object

Computer vision object detection model.

Parameters

network_name (*string*) – A string representing the computer vision network that will be used for detection.

Variables

- **net_name** – Holds the string literal for the chosen computer vision network.
- **net** – Holds the crucial mxnet-gluoncv instantiated computer vision model for which our package aims to provides a layer of abstraction over.
- **inference_resolution** – Stores the resolution of images we are to perform inference on.
- **default_object_classes** – Stores the default object classes that come with chosen network:

get_classes()

Get list of the object classes that the computer vision model is detecting for in images.

Return type

List

get_image_prediction(*fname*, *nms=0.0*)

Get image with the bounding box detections and the prediction made by the computer vision model.

Parameters

- **fname** (*string*) – Path to an image file.

- **nms** (*float*) – Stands for non-maximal suppresion. If computer vision model detects and an object in the image with a confidence value less than the nms value, it will not include it in the returned results.

Returns
A pair of values, an image in the form of a numpy array, and the prediction dict.

Return type
numpy.array, dict

`get_prediction`(*fname*, *nms=0.0*) → dict

Get prediction made for an image by computer vision model.

Parameters

- **fname** (*string*) – Path to an image file.
- **nms** (*float*) – Stands for non-maximal suppresion. If computer vision model detects and an object in the image with a confidence value less than the nms value, it will not include it in the returned results.

Return type
dict

`list_classes`()

Print the object classes that the computer vision model is detecting for in images.

`reset_classes`()

Change the object classes that the computer vision model is detecting for in images back to defaults.

`set_classes`(*object_classes: list*)

Change the object classes that the computer vision model is detecting for in images. Ensures validity by referencing the original list of available object classes when model was first instantiated.

Parameters
object_classes (*List*) – List of new object classes to detect for. Ex. "person", "bicycle", "banana".

`show_image_prediction`(*fname*, *nms=0.0*)

Print image with the bounding box detections and the prediction made by the computer vision model.

Parameters

- **fname** (*string*) – Path to an image file.
- **nms** (*float*) – Stands for non-maximal suppresion. If computer vision model detects and an object in the image with a confidence value less than the nms value, it will not include it in the returned results.

Module contents

comp_viz.utils package

Submodules

comp_viz.utils.toolbox module

class comp_viz.utils.toolbox.**Models**

Bases: object

Utility class centered around conveying available functionality for the comp_viz package.

get_tasks()

Get available tasks for the comp_viz package.

Return type
list

list_tasks()

Show available tasks for the comp_viz package.

class comp_viz.utils.toolbox.**ObjectDetection**

Bases: object

Utility class centered around object detection tasks relevant but not limited to the comp_viz object detection package.

format_object_classes() → list

Given a list of object classes, format all the elements such that they are readable by the network.

Parameters
object_classes (*list*) – List of object classes.

Return type
list

get_network_resolution()

Get image inference resolution of the specified network.

Parameters
net_name (*string*) – A valid network name among the results in get_networks() or list_networks() method.

Return type
int

get_networks()

Get list of the available networks that can be used with the comp_viz object_detection sub-package.

Return type
List

get_pred_bboxes_image(*bboxes: list*, *labels=[]*, *class_names=[]*, *scores=[]*)

Given an image and detection bounding box features, plot the bounding box to the image and return it.

Parameters

- **img_fname** (*string*) – Path to image to plot bounding box to.
- **bboxes** (*List[List]*) – Bounding boxes of form [[x_min,y_min,x_max,y_max],...] to plot to the image/
- **labels** (*List[int]*) – Class id values to mape to each bounding box and class name.
- **class_names** (*List[string]*) – List of object classes:
- **scores** (*List[float]*) – List of confidence values for the bounding boxes.

Return type
numpy.ndarray

list_networks()

Show list of the available networks that can be used with the comp_viz object_detection sub-package.

resize_bbox(*orig: tuple*, *dest: tuple*)

Given a bounding box of the format [x,min, y_min, x_max, y_max] and the original image resolution, return a new bounding box resized to the desired image size.

Parameters

- **bbox** (*list*) – Bounding box of the form [x_min, y_min, x_max, y_max].
- **orig** (*Tuple or ndarray.shape*) – Original image resolution of form (height, width, shape). Ex. (500,800,3)
- **dest** (*Tuple or ndarray.shape*) – Image to resize resolution of form (height, width, shape). Ex. (600,900,3)

Return type
list

show_pred_bboxes_image(*bboxes: list*, *labels=[]*, *class_names=[]*, *scores=[]*)

Given an image and detection bounding box features, plot the bounding box to the image and show it.

Parameters

- **img_fname** (*string*) – Path to image to plot bounding box to.
- **bboxes** (*List[List]*) – Bounding boxes of form [[x_min,y_min,x_max,y_max],...] to plot to the image/
- **labels** (*List[int]*) – Class id values to mape to each bounding box and class name.
- **class_names** (*List[string]*) – List of object classes:
- **scores** (*List[float]*) – List of confidence values for the bounding boxes.

Return type
void

class comp_viz.utils.toolbox.**Tools**

Bases: object

Utility class centered around images and filesnames.

exists() → bool

Boolean function to determine if path to filename exists.

Parameters
fname (*string*) – Path to file.

Return type
boolean

filename_show_image()

Given path to an image file, show the said image file to the screen.

Parameters
fname (*string*) – Path to image.

Return type
void

get_cv2_image()

Given path to an image file, return the said image in the form of an numpy ndarray using openCV.

Parameters
fname (*string*) – Path to file.

Return type
numpy.ndarray

get_mxnet_image()

Given path to an image file, return the said image in the form of an mxnet ndarray.

Parameters
fname (*string*) – Path to file.

Return type
mxnet.ndarray.ndarray.NDArray

save_image(*path: str*)

Given an image in the form of an ndarray, save it to the path specified.

Parameters

- **img** (*numpy.ndarray or mxnet.ndarray.ndarray.NDArray*) – Image in the form of an ndarray.
- **path** (*string*) – Path to save image to.

Return type
void

show_image()

Given an image in the form of an numpy ndarray, show the image to the screen.

Parameters
img (*numpy.ndarray or mxnet.ndarray.ndarray.NDArray*) – Image in the form of an ndarray.

Return type
void

verify_exists()

Module contents

1.1.2 Submodules

1.1.3 comp_viz.config module

class comp_viz.config.**CompViz**

Bases: object

Most parental configuration class for the comp_viz package.

Variables
version – Version number for the comp_viz package.

version = '1.0.0'

class comp_viz.config.**Models**

Bases: *CompViz*

Configuration class for available functionality for the comp_viz package.

Variables
tasks – List of supported tasks provided by comp_viz package.

tasks = ['Object Detection']

class comp_viz.config.**ObjectDetection**

Bases: *CompViz*

Configuration class for the object detection task for the comp_viz package.

Variables
networks – Dictionary of supported networks for object detection for the comp viz package. Each network has an associated inference resolution.

networks = {'center_net_resnet101_v1b_dcnv2_coco': {'resolution': 416}, 'faster_rcnn_fpn_resnet50_v1b_coco': {'resolution': 416}, 'faster_rcnn_fpn_syncbn_resnest269_coco': {'resolution': 416}, 'ssd_512_resnet50_v1_coco': {'resolution': 416}, 'yolo3_darknet53_coco': {'resolution': 416}, 'yolo3_mobilenet1.0_coco': {'resolution': 416}}

1.1.4 Module contents

PYTHON MODULE INDEX

C

INDEX

T

V