SONY

AITRIOS

Vision and Sensing Application SDK Image Download Functional Specifications

Copyright 2023 Sony Semiconductor Solutions Corporation

Version 0.2.0

2023 - 1 - 30

TOC

1. Change history

Date	What/Why
2022/12/12	Initial draft
2023/01/30	Directory structure change. Added that symbolic links must not be used. Following the deletion of the secret information, Initial was removed from the configuration file entry and added about creating a configuration file. Updated the PDF build environment.

2. Terms/Abbreviations

Terms/Abbreviations	Meaning
COCO	Common Objects in Context Large-scale object detection, segmentation, caption dataset

3. Reference materials

- Reference/Related documents

  - COCO Download

    - https://cocodataset.org/#download

  - COCO annotations file

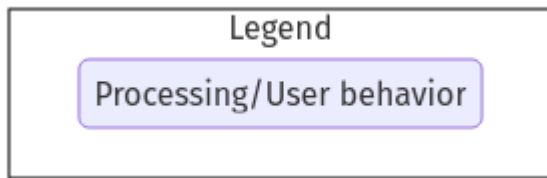   - http://images.cocodataset.org/annotations/annotations_trainval2017.zip

4. Expected use case

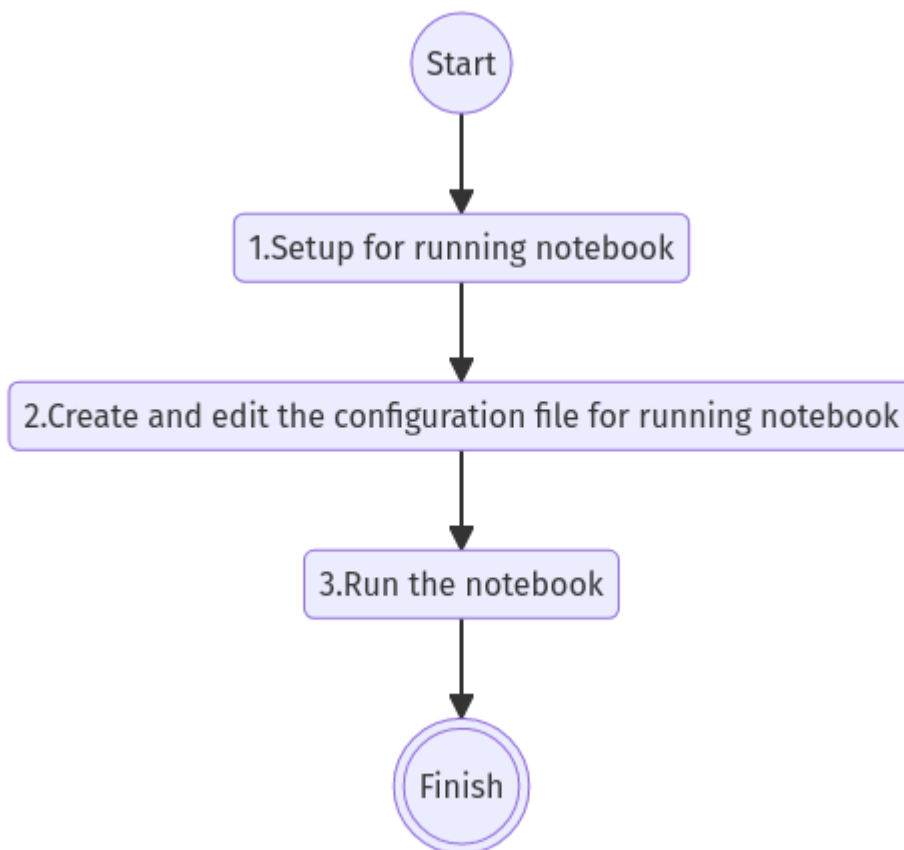- You want to download sample images from COCO for importing and annotating in Console for AITRIOS

5. Functional overview/Algorithm

Functional overview

The SDK downloads sample images from COCO in the following flow

Legend

Processing/User behavior

- Flow overview

Start

↓

1.Setup for running notebook

↓

2.Create and edit the configuration file for running notebook

↓

3.Run the notebook

↓

Finish

- Flow details

  1. Setup for running notebook

     - Get *COCO annotations file*

  2. Create and edit the configuration file for running notebook

     - Create and edit the configuration file *configuration.json* to configure notebook runtime settings

  3. Run the notebook

     - Run the notebook to download sample images from COCO
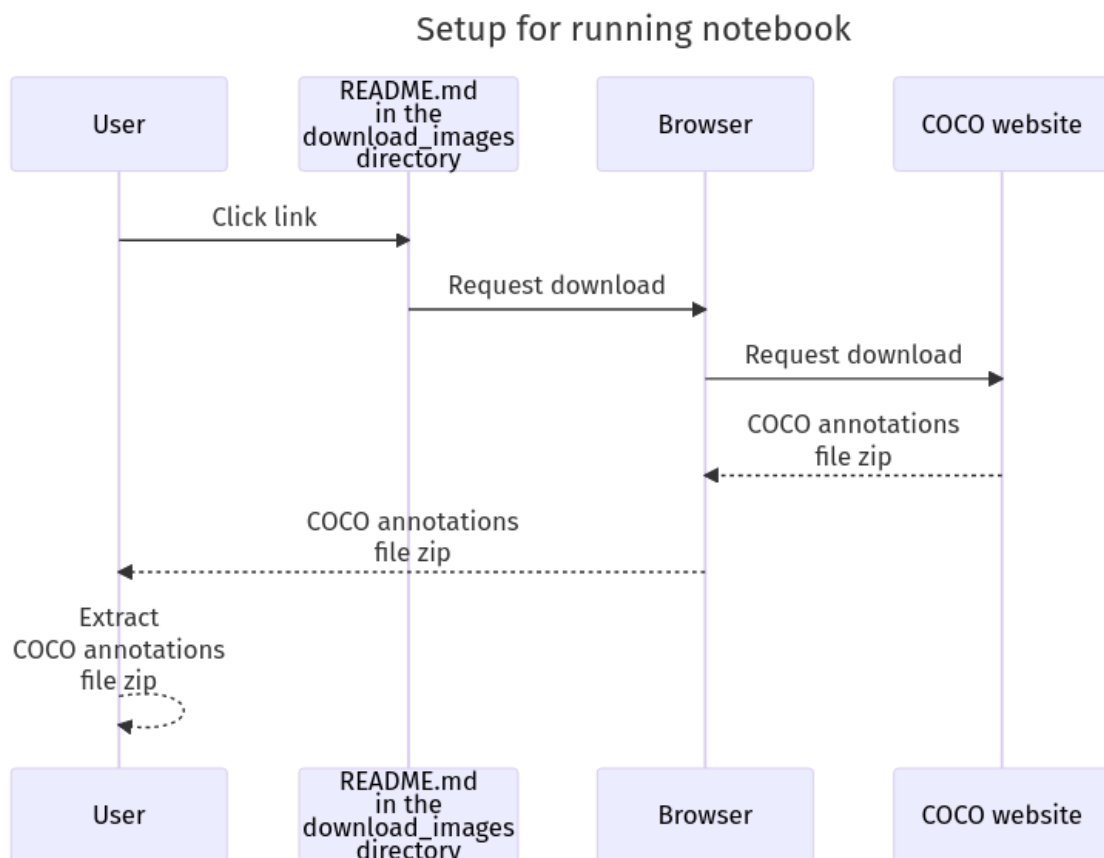
6. User interface specifications

How to start each function

1. Launch the SDK environment and preview the `README.md` in the top directory

2. Jump to the `README.md` in the `tutorials` directory from the hyperlink in the SDK environment top directory

3. Jump to the `2_prepare_dataset` directory from the hyperlink in the `README.md` in the `tutorials` directory

4. Jump to the `download_images` directory from the hyperlink in the `README.md` in the `2_prepare_dataset` directory

5. Jump to each feature from each file in the `download_images` directory

Setup for running notebook

1. Download *COCO annotations file* in your browser from the procedure and link provided in `README.md` in the `download_images` directory

2. Extract the downloaded *COCO annotations file* zip to the path specified in the configuration file

Setup for running notebook

Create and edit the configuration file for running notebook

1. Create and edit the configuration file, `configuration.json`, in the `download_images` directory

Do not use symbolic links to files and directories.

Configuration	Meaning	Range	Remarks
`annotation_file`	COCO annotations file path	Absolute path or relative to configuration.json/Notebook (*.ipynb)	・Don't abbreviate
`category_names`	Image category names Downloads only images matching this category	["Category1", "Category2", ・・・] See *category names list* for categories	・Optional ・If omitted or no category is specified (an empty list is specified), all categories will be downloaded
`max_download_count`	Maximum number of images downloaded per category	Downloads the following number of images per category 0: Downloads all images 1: Downloads a single image 2: Downloads two images ・・・ All images: Downloads all images All images+1: Downloads all images	・Optional ・If omitted or 0 is specified, all images matching the conditions will be downloaded ・If category_names is omitted or category is not specified (an empty list is specified), this parameter is the total number of downloads, not the number of downloads per category
`licenses`	Image license Downloads only images applicable to this license	[License1 ID, License2 ID, ・・・] See *license list* for license	・Optional ・If omitted or no license is specified (an empty list is specified), all licenses are applicable for download

Configuration	Meaning	Range	Remarks
`remove_categories`	Category names to exclude Images matching this category will not be downloaded	["Category1", "Category2", ・・・] See _category names list_ for categories	・Optional ・If omitted or no category is specified (an empty list is specified), no exclusion will occur
`output_dir`	Directory to store download images	Absolute path or relative to configuration.json/Notebook (*.ipynb)	・Don't abbreviate

Category names list

type	category name			
Person	person			
Vehicle	bicycle	car	motorcycle	airplane
	bus	train	truck	boat
Outdoor	traffic light	fire hydrant	stop sign	parking meter
	bench			
Animal	bird	cat	dog	horse
	sheep	cow	elephant	bear
	zebra	giraffe		
Accessory	backpack	umbrella	handbag	tie
	suitcase			
Sports	frisbee	skis	snowboard	sports ball
	kite	baseball bat	baseball glove	skateboard
	surfboard	tennis racket		

type	category name			
Kitchen	bottle	wine glass	cup	fork
	knife	spoon	bowl	
Food	banana	apple	sandwich	orange
	broccoli	carrot	hot dog	pizza
	donut	cake		
Furniture	chair	couch	potted plant	bed
	dining table	toilet		
Electronic	tv	laptop	mouse	remote
	keyboard	cell phone		
Appliance	microwave	oven	toaster	sink
	refrigerator			
Indoor	book	clock	vase	scissors
	teddy bear	hair drier	toothbrush	

License list

License	Commercial use	Redistribution	ID
Attribution-NonCommercial-ShareAlike License (CC BY-NC-SA 2.0)	No	Yes	1
Attribution-NonCommercial License (CC BY-NC 2.0)	No	Yes	2
Attribution-NonCommercial-NoDerivs License (CC BY-NC-ND 2.0)	No	Yes	3

License	Commercial use	Redistribution	ID
Attribution License (CC BY 2.0)	Yes	Yes	4
Attribution-ShareAlike License (CC BY-SA 2.0)	Yes	Yes	5
Attribution-NoDerivs License (CC BY-ND 2.0)	Yes	Yes	6
No known copyright restrictions	See *NOTE*	See *NOTE*	7
United States Government Work	Yes (with exceptions)	Yes (with exceptions)	8

No known copyright restrictions

This means that the copyright of each work is not cleared. For details, you have to check the Rights Statement of each participating institution. There are various cases as follows:

- The copyright is in the public domain because it has expired

- The copyright was put into the public domain for other reasons, such as failure to comply with necessary procedures and conditions

- The institution owns the copyright but is not interested in exercising control or has sufficient legal rights to allow others to use the work without restriction

Run the notebook

1. Open the notebook, *.ipynb*, in the `download_images` directory, and run the python scripts in it

    ○ The script does the following:

        ▪ Checks that *configuration.json* exists in the `download_images` directory

            ▪ If an error occurs, the error description is displayed and running is interrupted.

        ▪ Checks the contents of *configuration.json*

            ▪ If an error occurs, the error description is displayed and running is interrupted.

        ▪ Checks that *COCO annotations file* exists in the directory specified in *configuration.json* for `annotation_file`

            ▪ If an error occurs, the error description is displayed and running is interrupted.

        ▪ Reads the COCO annotations file from the directory specified in *configuration.json* for `annotation_file` makes the necessary settings in Pycocotools:

        ▪ If an error occurs in external software, for example, Pycocotools, the error output by the external software is displayed and running is interrupted

- Downloads images determined from following

  - Images matching the category specified in *configuration.json* for `category_names`

    - if only values not present in the *category names list* are set to the `category_names`, error description is displayed and running is interrupted

    - If both present and not present values in the *category names list* are set to the `category_names`, warns about values not present in the list and continues using only values present

  - Number of images downloaded specified in *configuration.json* for `max_download_count`

    - If negative value is set to the `max_download_count` error description is displayed and running is interrupted

  - Images applicable to the license specified in *configuration.json* for `licenses`

    - if only values not present in the *license list* are set to the `licenses`, error description is displayed and running is interrupted

    - If both present and not present values in the *license list* are set to the `licenses`, warns about values not present in the list and continues using only values present

  - Excludes images matching the category specified in *configuration.json* for `remove_categories`

    - if only values not present in the *category names list* are set to the `remove_categories`, error description is displayed and running is interrupted

    - If both present and not present values in the *category names list* are set to the `remove_categories` list, warns about values not present in the list and continues using only values present

  - If the number of images matching the preceding conditions reaches zero, a warning is displayed

- Outputs the downloaded images to the directory specified in *configuration.json* for `output_dir`. if the directory does not already exist, it is created at the same time

- Displays the following while downloading:

```
downloading id: 416256
downloaded 1/30 images (t=1.0s)

downloading id: 269314
downloaded 2/30 images (t=1.0s)

downloading id: 17029
downloaded 3/30 images (t=1.4s)


...
downloading id: 117374
downloaded 30/30 images (t=1.3s)
```

- While downloading, you can interrupt with the Stop Cell Execution of notebook cell function

7. Target performances/Impact on performances

- None

8. Assumption/Restriction

- None

9. Remarks

- None

10. Unconfirmed items

- None