SONY

AITRIOS

Vision and Sensing Application SDK CVAT Image Annotation Functional Specifications

Version 0.2.0

2023 - 1 - 30

TOC

1. Change history

Date	What/Why
2023/01/30	Initial draft

2. Terms/Abbreviations

Terms/Abbreviations	Meaning
Tag	Label
Annotation	Label information, Labeling
Annotation information	Metadata for images
Dataset	A collection of image and annotation data for learning and model evaluation

3. Reference materials

- Reference/Related documents
  - Codespaces port forwarding
    - https://docs.github.com/en/codespaces/codespaces-reference/security-in-codespaces#port-forwarding
  - CVAT
    - https://github.com/opencv/cvat

4. Expected use case

- Create dataset
  - Image classification task
    - You want to output dataset from CVAT on Dev Container and create AI model of image classification seamlessly
  - Object detection task
    - You want to annotate your own image data
    - You want to customize dataset by adding data, editing tags, etc.

5. Functional overview/Algorithm

Functional overview

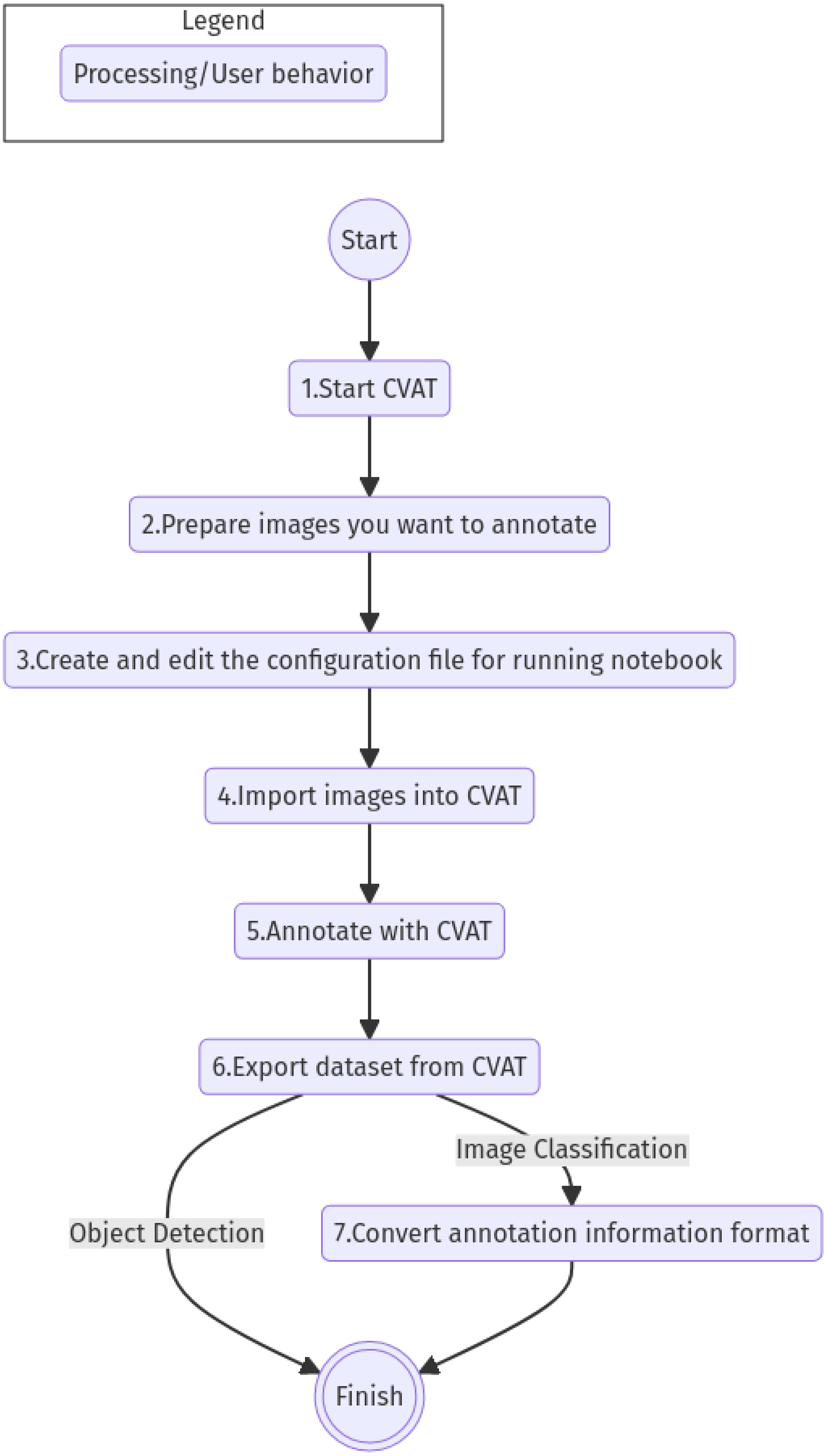
- Users can start CVAT in the SDK’s Dev Container (Local PC or Codespaces)
- Users can use CVAT to manually annotate data, and add and edit data annotations
- Users can import image data from local file system or Codespaces into CVAT
    - The storage locations where images can be imported are as follows:
        - SDK supports "Local PC" or "Codespaces"
        - Users import images from Codespaces using CVAT API in notebooks
- Users can export annotation information from CVAT to local file system or Codespaces
    - Users can select any export format with CVAT
        - For image classification, the SDK only supports "ImageNet 1.0"
        - For object detection, only formats with annotation information are supported

Export formats supported by CVAT	SDK support
CamVid 1.0	Yes
Cityscapes 1.0	Yes
COCO 1.0	Yes
COCO Keypoints 1.0	Yes
CVAT for images 1.1	Yes
CVAT for video 1.1	Yes
Datumaro 1.0	Yes
ICDAR Localization 1.0	Yes
ICDAR Recognition 1.0	No
ICDAR Segmentation 1.0	No
ImageNet 1.0	Yes
KITTI 1.0	Yes

Export formats supported by CVAT	SDK support
LabelMe 3.0	Yes
LFW 1.0	No
Market-1501 1.0	No
MOT 1.1	Yes
MOTS PNG 1.0	No
Open Images V6 1.0	Yes
PASCAL VOC 1.1	Yes
Segmentation mask 1.1	Yes
TFRecord 1.0	Yes
VGGFace2 1.0	Yes
WiderFace 1.0	Yes
YOLO 1.1	Yes

- For image classification, annotation information exported from CVAT can be converted into a format for use in the SDK for AI learning and quantization
- The image format supported by the SDK is JPEG

- Flow overview


Legend
Processing/User behavior
Start
1.Start CVAT
2.Prepare images you want to annotate
3.Create and edit the configuration file for running notebook
4.Import images into CVAT
5.Annotate with CVAT
6.Export dataset from CVAT
Image Classification
Object Detection
7.Convert annotation information format
Finish

- Flow details

  1. Start CVAT

     - Follow the procedures in the README to set up CVAT

  2. Prepare images you want to annotate

     - Prepare images to annotate

  3. Create and edit the configuration file for running notebook

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

       > ℹ Only when running the notebook

  4. Import images into CVAT

     - Import images using notebooks or CVAT Web UI

  5. Annotate with CVAT

     - Annotate imported images with CVAT Web UI

  6. Export dataset from CVAT

     - Export dataset using notebooks or CVAT Web UI

  7. Convert annotation information format (for image classification only)

     - Convert annotation information exported from CVAT into a format for use in the SDK for AI learning and quantization

6. User interface specifications

How to start each function

1. Jump to the `README.md` in the `tutorials` directory from the hyperlink in the SDK environment top directory
2. Jump to the `2_prepare_dataset` directory from the hyperlink in the `README.md` in the `tutorials` directory
3. Jump to the `annotate_images` directory from the hyperlink in the `README.md` in the `2_prepare_dataset` directory
4. Open the `README.md` in the `image_classification` directory or `object_detection` directory from the hyperlink in the `README.md` in the `annotate_images` directory
5. Run "Set up CVAT" and wait until the startup log stops
6. Open the 8080 port in your web browser in the [**Port Forwarding**] tab of VS Code
    - Wait until startup is complete and the CVAT login screen appears
    - (First-time only)Run a command in the [**Terminal**] tab of VS Code to create an account with superuser privileges for CVAT
      The commands are in the `README.md` in the `image_classification` directory or `object_detection` directory
    - Enter account information for CVAT superuser privileges at the CVAT login screen of your web browser
    - When authentication is successful, you are taken to the CVAT initial screen

Prepare images you want to annotate

1. Create the `images` directory under the `image_classification` directory or `object_detection` directory and store in it the images you want to import into CVAT and annotate

    > ℹ Directories can have any structure (If there is a child directory, images in the child directory will also be imported)

Create and edit the configuration file for running notebook

1. Create and edit the configuration file, `configuration.json`, of the execution directory in three cases:
   "When importing images from Dev Container local storage" or "When exporting annotation information to Dev Container local storage" or "When converting annotation information format"

ℹ All parameters are required, unless otherwise indicated.

ℹ All values are case sensitive, unless otherwise indicated.

ℹ Do not use symbolic links to files and directories.

Configuration	Meaning	Range	Remarks
cvat_username	Username to log in to CVAT		Specify when importing or exporting
cvat_password	Password of the user logging in to CVAT		Specify when importing or exporting
cvat_project_id	Project ID to import images into CVAT or export dataset from CVAT		Specify when importing or exporting
import_dir	Path to store images to import into CVAT and annotate	Absolute path or relative to notebook (*.ipynb)	Specify when importing
import_image_extension	Image extension to import into CVAT and annotate		Specify when importing
import_task_name	Task name created when importing into CVAT		Specify when importing

Configuration	Meaning	Range	Remarks
`export_format`	Format for exporting annotation information from CVAT		Specify when exporting
`export_dir`	Path to store annotation information to export from CVAT	Absolute path or relative to notebook (*.ipynb)	Specify when exporting or format converting
`dataset_conversion_base_file`	Path of file to convert format	Absolute path or relative to notebook (*.ipynb)	Specify when converting format (image classification only)
`dataset_conversion_dir`	Path to store annotation information to export from CVAT and convert for use in AI model learning and quantization of the SDK	Absolute path or relative to notebook (*.ipynb)	Specify when converting format (image classification only) If the directory contains an existing dataset, the error message is displayed and running is interrupted.
`dataset_conversion_validation_split`	Percentage of images in the dataset that are not used for training but are used for validation, when converting it's format	Greater than 0.0 and less than 1.0	Specify when converting format (image classification only)
`dataset_conversion_seed`	Random seed value for shuffling images in the dataset when converting it's format	0 - 4294967295	Specify when converting format (image classification only)

Import images into CVAT

- Import images from Dev Container local storage

  1. (Only if you have not created a project) Create a project in CVAT Web UI by selecting the [**Create a new project**] from the [**+**] in the menu [**Project**]

  2. Add a label by selecting the [**Add label**] in the [**Constructor**] from the project you created

  3. Import images in the directory specified by `import_dir` by running the `import_api.ipynb` in the `image_classification` directory or `object_detection` directory.
     (At this time, a task is created with the name specified by the `import_task_name` and associated with the project. If you import multiple times with the same name specified, a task with the same task name is created with a different task ID.)

     - The script does the following:

       - Checks that *configuration file* exists in the execution directory

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

         - Pressing the stop button on a cell while the cell is running interrupts processing

       - Checks that *configuration file* includes each parameter

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

         - Pressing the stop button on a cell while the cell is running interrupts processing

       - Reads the value of each parameter from *configuration file* to prepare the information needed for API client authentication

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

         - Pressing the stop button on a cell while the cell is running interrupts processing

       - Reads the value of each parameter from *configuration file* and load the images

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

         - Pressing the stop button on a cell while the cell is running interrupts processing

       - Successful authentication and displays image in the project

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

         - Pressing the stop button on a cell while the cell is running interrupts processing

       - CVAT Web UI can verify that images have been imported into project tasks

- Import images from a local environment with a web browser running
    1. (Only if you have not created a project) Create a project in CVAT Web UI by selecting the [**Create a new project**] from the [**+**] in the menu [**Project**]
    2. Create a task by selecting the [**Create a new task**] from the [**+**] at the bottom of the project you created
    3. Open the [**Click or drag files to this area**] on the [**My computer**] tab in the [**Select files**] item of the task and select an image file
    4. Press the [**Submit & Open**] button to import

> See documentation for import procedures

Annotate with CVAT

1. If necessary, select the [**Add label**] in the [**Constructor**] in the CVAT project to add labels
2. Select the [**Jobs**] in a task in the project to go to the job screen
3. Select the tag you want to associate from the [**Setup tag**] and click it to annotate the image
4. To move to the next image, click the [**>**] button at the top of the image, then press the key on the next image as preceding to associate the tag
5. After annotating up to the last image, display the menu from the [**≡(menu)**] button and click the [**Finish the job**] to complete

> See documentation for annotation procedures

Export dataset from CVAT

- Export dataset to Dev Container local storage

  1. Export dataset from the project specified by `cvat_project_name` by running the `export_api.ipynb` in the `image_classification` directory or `object_detection` directory

     - The script does the following:

       - Checks that *configuration file* exists in the execution directory

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

         - Pressing the stop button on a cell while the cell is running interrupts processing

       - Checks that *configuration file* includes each parameter

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

         - Pressing the stop button on a cell while the cell is running interrupts processing

       - Reads the value of each parameter from *configuration file* to prepare the information needed for API client authentication

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

         - Pressing the stop button on a cell while the cell is running interrupts processing

       - After successful authentication, download a zip file of the dataset to the directory specified by `export_dir`

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

         - Pressing the stop button on a cell while the cell is running interrupts processing

         - If the directory specified by `export_dir ` does not already exist, it is created at the same time.

- Export dataset to a local environment running a web browser
    1. In the CVAT Web UI, click the project’s [⋮] and then click the [**Export dataset**] from the menu that appears
    2. Select and click the [**ImageNet 1.0**] from the [**Export format**] in the [**Export project ～ as a dataset**] dialog
    3. Enter the name of the file to download in the [**Custom name**]
    4. Check the [**Save images**] to include image files in the export file
    5. Use your browser’s download function to specify the download destination and download a zip file.
- In the case of image classification, the directory structure in the exported zip file is as follows:
  There are directories with annotation names, and each directory contains image files associated with the annotation

  For object detection, directory structure varies by format

  ```
  Exported zip file
    ├ Tag A/
    │   ├ Image file
    │   ├ Image file
    │   ├ ・・・・
    ├ Tag B/
    │   ├ Image file
    │   ├ Image file
    │   ├ ・・・・
    ├ ・・・・
  ```

Convert annotation information format (for image classification only)

1. Convert the format of a zip file of the dataset specified by **`dataset_conversion_base_file`** by running the **`convert_dataset.ipynb`** in the **`image_classification`** directory

   - If the directory specified by **`dataset_conversion_dir`** is **`tutorials/_common/dataset`**, annotation information is stored in the **`tutorials/_common/dataset`** directory as follows:

     ```
     tutorials/
       ├ 2_prepare_dataset/
       │  └ annotate_images/
       │     └ image_classification/
       │        ├ configuration.json
       │        └ images/
       │            ├  Image file
       │            ├  Image file
       │            ├ · · · ·
       └ _common
         └ dataset
           ├ **.zip (1)
           ├ cvat_exported/ (2)
           │  ├ Image class name/
           │  │   └ Image file
           │  ├ Image class name/
           │  │   └ Image file
           │  ├ · · · ·
           ├ labels.json (3)
           ├ training/  (4)
           │  ├ Image class name/
           │  │   └ Image file
           │  ├ Image class name/
           │  │   └ Image file
           │  ├ · · · ·
           └ validation/ (5)
               ├ Image class name/
               │   └ Image file
               ├ Image class name/
               │   └ Image file
               ├ · · · ·
     ```

     (1) Data to be converted. Zip file exported from CVAT

     (2) Intermediate output data during conversion. The contents of the zip file exported from CVAT are extracted into this directory

     (3) Intermediate output data during conversion. Labels information file created from the **`cvat_exported`** directory

(4) Conversion output data. Extracted for training from the `cvat_exported` directory

(5) Conversion output data. Extracted for validation from the `cvat_exported` directory

- The format of label information files is a json file with the label name and its id value as follows:

  ```
  {"daisy": 0, "dandelion": 1, "roses": 2, "sunflowers": 3, "tulips": 4}
  ```

- If the directory specified by `dataset_conversion_dir` does not already exist, it is created at the same time.

7. Target performances/Impact on performances

- Usability
  - When the SDK environment is built, CVAT can be run without any additional installation steps
  - UI response time of 1.2 seconds or less

8. Assumption/Restriction

- CVAT fails to start if Codespaces Machine Type is minimally configured (2-core), so you must select a Machine Type greater than 4-core
- If you cancel and restart an import or export process, start each process from the beginning instead of resuming in the middle

9. Remarks

- No error codes and messages defined in the SDK
- About putting the password in the document
  - No security issues because *port forward* is set to private by default, which means only the creator of the Codespaces can access the port
- How to check the version of CVAT
  - After logging in to CVAT with Web UI, the version number is listed in the dialog that appears when you click your username and click [**About**]

10. Unconfirmed items

None