

## Short description

Contains various small functions to extract images and annotation files from the BOVIDS style (see readme.md, annotation data) into a structure used for training deep neural networks (the object detector). Furthermore, it is possible to re-name classes inside one folder of annotation files (.xml files). This might be useful to correct typos or to join the data of various individuals of one species to a larger dataset (but keeping a one-class detection model).

## Requirements

- packages: xml
- images and annotations in the BOVIDS style

Images:

*ANNOTATION\_STORAGE/Bilder/SPECIESNAME/ZOONAME/ENCLOSURENUMBER/imagename.jpg*

Annotation:

*ANNOTATION\_STORAGE/Label/SPECIESNAME/ZOONAME/ENCLOSURENUMBER/imagename.xml*

### Step 1 – open spyder:

- Terminal / shell:
  - conda activate bovids
  - spyder

### Step 2 – modify parameters of the relevant task.

Task 1: extract images and annotation files

Task 1.1: extract images from a specific species: *extract\_od\_images\_enclosures()*

- ANNOTATION\_FOLDER\_IMAGES\_ES: Path to ANNOTATION\_STORAGE/Bilder/ [string]
- ANNOTATION\_FOLDER\_LABELS\_ES: Path to ANNOTATION\_STORAGE/Label/ [string]
- OUTPUT\_FOLDER\_ES: Destination (folder) of the output [string]
- SPECIES\_ES: List of strings containing the name of those species whose images / annotations are extracted. [list]
- EXCLUDE\_ENCLOSURES\_ES: List of strings containing specific enclosure codes that should be skipped/ignored. [list]
- EXCLUDE\_ZOOS\_ES: List of specific zoos (strings) that are excluded from extraction. [list]

Task 1.2: extract images from specific enclosures: *extract\_od\_images\_enclosures()*:

- ANNOTATION\_FOLDER\_IMAGES\_EE: Path to ANNOTATION\_STORAGE/Bilder/ [string]
- ANNOTATION\_FOLDER\_LABELS\_EE: Path to ANNOTATION\_STORAGE/Label/ [string]
- OUTPUT\_FOLDER\_EE: Destination (folder) of the output. [string]
- ENCLOSURES\_EE: List of enclosure codes that will be used. [list]

Task 1.3: extract images from the structure generated by evaluate\_bounding\_boxes.py:

*extract\_images\_evaluated()*:

- INPUT\_FOLDER\_BASE: Path (folder) to the output of evaluate\_bounding\_boxes, this folder contains subfolders “good”, “bad”, “swapped”. [string]
- ENCLOSURE\_CODE\_EI: Images and annotations of this enclosure code will be extracted. [string]
- WHICH\_ONES: List of strings, corresponding to the names of the subfolders (e.g. [“good”]) that should be used. Only use bad/swapped if you manually re-annotated those images! [list]
- OUTPUT\_FOLDER\_IMAGES\_EI: Destination (folder) of the output (images) [string]
- OUTPUT\_FOLDER\_LABELS\_EI: Destination (folder) of the output (annotation files) [string]

Task 2: Renaming of class labels

*rename\_annotation()*:

- LABEL\_RENAME\_FOLDER: Folder that contains annotation files (.xml) in which labels need to be renamed. [string]
- OLD\_LABEL: If empty string (“”), all labels are renamed to NEW\_LABEL. If it is not empty, only those labels with value OLD\_LABEL will be renamed. [string]
- NEW\_LABEL: New value (label) after renaming. [string]

Step 3 – run the script.

- extract images from a specific species: *extract\_od\_images\_species()*
- extract images from specific enclosures: *extract\_od\_images\_enclosures()*
- extract images from the structure generated by evaluate\_bounding\_boxes.py: *extract\_images\_evaluated()*
- renaming: *rename\_annotation()*