

### Short description

Contains two sub-routines. First, converts the annotation files out of the structure induced by “prepare\_data\_od.py” into the yolo format. Second, given images and the yolo annotation, it can be used to train an object detection network.

### Requirements

- packages: tensorflow, xml
- BOVIDS specific packages: /global/yolo-v4-tf.keras-master/

#### Step 1 – open spyder:

- Terminal / shell:
  - conda activate bovids
  - spyder

#### Step 2 – adjust parameters:

##### General:

GPU\_TO\_USE = '0': Choose GPU that will be used. If only one GPU is present, set the value to '0'. [string]

YOLO\_LIBRARY: Path to /global/yolo-v4-tf.keras-master/ [string]

#### Task 1: convert annotation files from pascal VOC (.xml) to yolo (.txt)

- XML\_PATH = Path to the folder containing all .xml files used for training. [string]
  - E.g.: 'training\_folder/Labels/'
- CLASSES\_PATH = Path to a .txt file (will be created) that contains the class names (i.e. the individualnames) that are present in the xml files. We suggest saving this file next to the subfolders of images and labels. This class file is used later during prediction (and needs to appear in global\_configuration.py) and is used to identify the individual codes. [string]
  - E.g.: ,training\_folder/classes.txt'
- TXT\_PATH = Path to a .txt file (will be created) containing the annotation in yolo format. Is used during training. [string]
  - E.g.: ,training\_folder/annotation.txt'

*Task 2: training*

- INPUT\_MODEL = Leave the string empty to train from scratch on imagenet weights or input an existing model here for transfer learning. An existing model needs to have the same configuration (anchor points etc) and the same number of classes. [string]
  - E.g.: '../od\_networks/basenet\_antelope/'
- VAL\_SPLIT = Fraction of images that will be used as the validation set. Depending on the size of the dataset, this value might be chosen between 0.1 and 0.2. [float]
- IMAGE\_FOLDER = Path to the folder containing the images to train on. [string]
  - E.g.: 'training\_folder/Bilder/'
- TXT\_PATH (see above, this annotations will be used).
- NUM\_EPOCHS = number of epochs [integer]
  - Depending on the size of the dataset, normally one trains for at least 100 epochs, but anyone needs to check this parameter for best performance.
  - If validation loss stays relatively constant, the network is completely trained.
  - If validation loss was fairly small and begins to increase while training loss decreases further, this might be a sign of overfitting.
- OUTPUT\_PATH = Destination of fresh model (is a folder). [string]
  - E.g.: '../od\_networks/2021-05-02\_wildebeest\_retrained/'
- FROM\_EPOCH = First epoch of training process, is normally set to zero. [integer]
- CHECKPOINT\_PATH = Path in which the currently best weights are stored during training. [string]
  - E.g.: '../od\_training\_checkpoints/2021-05-02\_wildebeest\_retrained\_ckpt/'

*Step 3 – run the script:*

- Create annotation files (yolo): ***convert\_pascal\_annotations()***
- Train network: ***train\_network()***