Short description

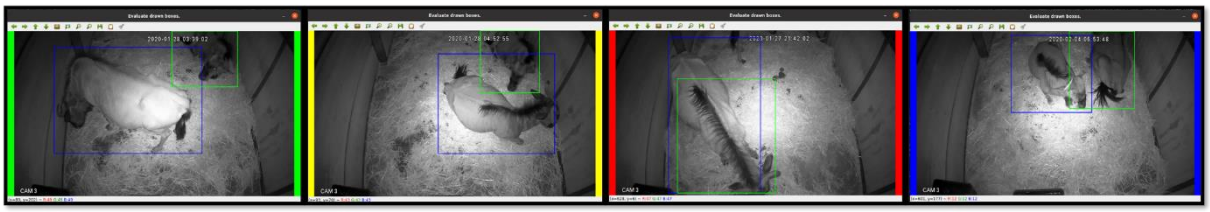
Script containing a GUI used for evaluating how good automatically generated bounding boxes are drawn. Can furthermore be used to store the images and the corresponding annotation files accordingly to their evaluation status. The evaluation value is stored in a .npy file and can be re-read by the script later.

Good images: Bounding box is drawn so well that the image and the annotation file can be used to train a network. (green)

Medium images: Bounding box is okay (good to use for action classification) but not perfect, maybe a bit too large, a missing hoof or a part of the nose is truncated. (yellow)

Bad images: Bounding box is drawn poorly. (red)

Swapped images: Only in a multi-class detection case (multiple individuals). The bounding boxes themselves are okay, but the label is wrong. (blue)



Requirements

- packages: XML, opencv, numpy
- images and corresponding annotation files in the structure generated by generate_annotation_files.py

Step 1 – open spyder:

- Terminal / Shell:
  - conda activate bovids
  - spyder

Step 2 – adjust parameters:

- BASEPATH_INPUT = Path to the folder containing two subfolders „Bilder“ (images) and „Label“ (annotation files) in the structure of generate_annotation_files.py. [string]

- ENCLOSURE_CODE = enclosure code of the enclosure that should be evaluated. [string]
    - E.g.: 'Wildebeest_FancyZoo_3'
- TMP_LABEL_FILE = Destination of corresponding .npy file to store evaluation status. [string]
    - E.g.: 'U:/ evaluation/od_network_one/Wildebeest_FancyZoo_3.npy'
- MOVE_BASEPATH = Destination (folder) in which the evaluated images and their corresponding annotation files will be moved. Script will create subfolders for each evaluation status. [string]
- FLAT_IMAGE = Decides the resolution of the images. Use False if the image is generated from 1,3 or 4 streams. [boolean]
- COLOR_FOR_INDIVIDUALS = Given a class name in a multi-class detection case (multiple individuals), for each class the color of the bounding box can be chosen arbitrarily (BGR code). [dictionary]
- E.g.: { ‚Wildebeest_FancyZoo_5‘: (255, 0, 0), ‚Wildebeest_FancyZoo_6‘: (0, 187, 130) }

Control keys to navigate on GUI, can be adjusted:

- MOVE_LEFT = '4'
- MOVE_RIGHT = '5'
- END_EVALUATION = 'p'

- SET_GOOD = 'a'
- SET_MEDIUM = 's'
- SET_BAD = 'd'

Step 3 - run the script:

- Run the script, Folgende Befehle in Konsole eintippen, je nachdem bei welchem Schritt man ist:
    - ***evaluate_folder():*** starts evaluation of the bounding boxes.
    - ***get_statistics()***: shows important statistics stored in the .npy file.
    - ***move_data_by_evaluation_value()***: copies the images and labels corresponding to the evaluation value.