Short description

Given a training csv file as input such that all contained nights are already annotated manually (and boris_to_csv was conducted), the script will sample images from those nights and apply the object detector to them to generate a training almost balanced classes for training the action classifier per individual. The handling is fairly like generate_annotation_files.py (object detection).

Requirements

- packages: numpy, opencv, tensorflow
- BOVIDS specific packages: global_configuration.py, /global/yolo-v4-tf.keras-master/
- For step 1 (sampling images), a comma-separated training-csv file is required. It contains the necessary information which nights (thus, enclosures with given video streams and individuals on a specific date) should be used for sampling. For each of those nights a "boris-csv-file" containing the manual annotation is required. An example is stored in the example folder.

Date	**Species**	**Zoo**	**Enclosure**	**Video**	**Individuals**
18.02.2020	Wildebeest	FancyZoo	1	1	1
31.12.2018	Eland	NiceZoo	2	3;4	2
10.01.2020	Kudu	GreatZoo	3	4	2;6

Step 1 – open spyder:

- Terminal / shell:
  - conda activate bovids
  - spyder

Step 2 – adjust parameters:

General

- GLOBAL_CONFIGURATION_PATH: Path to global_configuration.py [string]
  - E.g.: '…/global/'
- YOLO_LIBRARY = Path to yolo library [string]
  - E.g.: '…/gloabal/yolo-v4-tf.keras-master/'
- VIDEO_LENGTH = 14, length (in hours) oft he video files, standard value. [integer]

- WEIGHT_FACTOR = Can be set to a number larger 1.0 if not every image should be sampled. If it is set to 3.0, every third image will be taken. [float]

Task 1: Sample images from video files (balanced classes)

- CREATE_IMAGES: (True / False). If True, the task „create images“ is carried out. [boolean]
- CSV_OVERVIEW_FILE = Path to the training csv file (see above) [string]
- BASE_PATH_TO_DATA = Path (folder) which is the starting point of data navigation as explained in readme.md. I.e., contains a subfolder for each species containing subfolders of the zoos etc. [string]
- BASE_PATH_TO_LABELS = Path (folder) which is the starting point of data navigation as explained in readme.md. (“action classification storage”) I.e., contains a subfolder for each species containing subfolders of the zoos etc.
    - Therefore, for each night, we find the following file: DATA_STORAGE/Auswertung/SPECIESNAME/ZOONAME/Auswertung/BORIS_KI/csv-Dateien/YYYY-MM-DD_ENCLOSURECODE_SUM-7s_pred.csv
- OUTPUT_PATH_MULTIPLE_FRAME = Destination (folder) in which the produced images will be stored. [string]
- VIDEO_LEN_SPECIAL = Dictionary with zoonames as keys. Can be used if the video length varies between different zoos. [dictionary]
    - E.g.: {'FancyZoo': 12}

Task 2 – object detection and individual detection:

- Input folder is OUTPUT_PATH_MULTIPLE_FRAME of task 1. The content of the training csv file is ignored, the labels will be created for all images in this folder. I.e., the script decides which object detection network is used by the information from global_configuration.py based on the enclosure code.
- CREATE_YOLO_CUTOUT = *True*, if step is conducted; *False otherwise* [boolean]
- YOLO_OUTPUT_FOLDER = Destination (folder) for the balanced images (individuals are cut out). Created subfolders for MulipleFrame and SingleFrame automatically. [string]
- Folder (path) in which the object detection networks can be found as the global_configuration does only contain the filename. [string]

- MAX_DETECTIONS = Number of (different) individuals that might be detected. [integer]
- MIN_CONFIDENCY = Number between 0 and 1. Bounding boxes with smaller certainty will be dismissed. [float]

Step 3 – run the script

- Processing will start immediately.