Short description

Contains most of the configuration of BOVIDS subroutines. BOVIDS was originally created to be run in a local network on multiple computers and one server. Therefore, the server contained the “global” information to reduce redundancy.

General

1. BEHAVIOR_NAMES_BASE = List of names of the behaviors. Standard configuration is „Standing“, „LHU“, „LHD“. The names can be changed but make sure that the list indices correspond to the codes (0, 1, 2) used for action classification. If BOVIDS should be extended by a fourth behavior class, this needs adjustment in the program code which is straightforward but needs some programming knowledge.
2. BEHAVIOR_NAMES_SPECIAL = Names for „Out“ and „Truncation“. Those behaviors play a special role in BOVIDS and classes 3 and 4 are hard-coded. Adjustment needs changes in the program code.
3. COLOR_MAPPING = Dictionary whose keys are the entries of BEHAVIOR_NAMES_BASE and BEHAVIOR_NAMES_SPECIAL. The values signify the color used for plotting the timeline created per predicted night.
4. INTERVAL_LENGTH = Length (in seconds) of one time-interval. Standard is 7 seconds, larger values are unproblematic to use, shorter values require adjustment of the program code.
5. IMG_SIZE = Resolution of the images created during the object detection phase of the prediction pipeline. If EfficientnetB3 is used as an action classifier, this needs to be (300,300).
6. BATCH_SIZE_BEHAVIOR = Number of images that are predicted at once during action classification. Needs probably to be adjusted depending on the used GPU
7. GLOBAL_STARTING_TIME = Integer signifying when the global recording starts. If a video starts later, the first hours will be signified as “Out”.
8. GLOBAL_ENDING_TIME = Integer signifying when the global recording ends. If a video ends earlier, the last hours will be signified as “Out”.
9. GLOBAL_OBSERVATION_HOURS = Hours of recording (Difference between GLOBAL_ENDING_TIME and GLOBAL_STARTING_TIME).
10. PACKAGE_SIZE_DENSITY_STATISTICS = If set to 1.0, the object detection density (see readme.md) will be reported hourly. If set to 0.5, one block contains 30minutes and so on.

Post-Processing

1. POST_PROCESSING_RULES: Dictionary with strings as keys. Those keys are the name of a set of rules. Each value is a dictionary itself containing the following keys:
    a. ROLLING_AVERAGE_*:

1—3: Order of the rolling average used in time-intervals.

4. Weights of each behavior class during rolling average (should not be adjusted)

5—7: If not set to 1.0, the rolling average decreases / increases exponentially.

b. WEIGHTS_NETWORKS: Weights of the weighted average between the single frame's and the multiple frame's prediction.

c. MIN_LEN_*: We explain them at an example. If ASA = 3 then the sequence 0000200 will be post-processed to 0000000 because the sleeping behavior (2) lasted only one time-interval between to standing (0) phases. The 3 signifies that all sleeping sequences of less than three time-intervals between two standing phases are dismissed. The enumeration is historically grown, i.e. A = active (standing), L = lying (LHU), S = sleeping (LHD), O = Out.

d. Truncation: (See readme.md)

   A. MIN_LEN_TRUNCATION: Shorter sequences of "truncation" will be count as the previous behavior. Set to zero if truncation should be always reported.

   B. MIN_LEN_TRUNCATION_SWAP: Shorter sequences of truncation will be mapped to "TRUNCATION_REAL_BEHAVIOR_LONG"

   C. TRUNCATION_REAL_BEHAVIOR_LONG: If a sequence of truncation exceeds this value, the whole sequence will be mapped to this behavior. Depending on the enclosure, this might be set to 1 (LHU) or to 3 (Out). For testing purposes, it might be convenient to set it to 4 (truncated).

   D. TRUNCATION_INTERMEDIATE: Sequences of truncation exceeding MIN_LEN_TRUNCATION but being smaller than MIN_LEN_TRUNCATION_SWAP will be mapped to this behavior.

e. Out: Due to failure of the object detector or due to truncation effects, it might happen that sequences look like 0003300033001131133311111111111113111331111133331111. It is fairly safe to assume that the correct sequence would look like 0000000000001111111111111111111111111111111111111111. Therefore, fluctuations between out and a different behavior should be corrected if the percentage of out is not too large (i.e., there is a certain percentage of the real behavior visible) and the total duration of out is not too large.

   A. OUT_FLUCTUATION_REMOVAL_MAX: Threshold (time-intervals) for the duration of Out that might not be exceeded in order to remove the "out" sequences.

B. OUT_FLUCTUATION_REMOVAL_MIN_BEHAV: Percentage (number between 0.0 and 1.0) of how often the real behavior (e.g. standing, lying) needs to be detected in such a fluctuating sequence.

Extraction of images and object detection

1. DETECTION_MIN_CONFIDENCE = Standard value used as a threshold for the score required to accept a bounding box during object detection. Can be much larger in enclosures containing exactly one individual than in enclosures with multiple individuals.
2. MIN_DETECTION_SCORE = Dictionary with keys enclosure-individual-code or enclosure-code with corresponding value between 0.0 and 1.0. This threshold will be used during object detection instead of DETECTION_MIN_CONFIDENCE, if the matching enclosure-individual code is found, this value will be taken, otherwise, the value of the enclosure code. If both keys are not present, the standard value (DETECTION_MIN_COFIDENCE) is used.
3. IOU_THRESHOLD = Dictionary with keys enclosure-individual-code or enclosure-code with corresponding value between 0.0 and 1.0. This IoU threshold will be used during object detection if multiple individuals are present instead of 0.5 (standard value in yolo library). Smaller IoU threshold allows bounding boxes to be more similar (this is useful if a mother and her newborn are filmed as the bounding box for the newborn might be almost contained in the one of the adult animal).
4. VIDEO_ORDER_PLACEMENT = Dictionary such that keys are the enclosure_video_code or the enclosure_code and the values are a list of the video numbers of the corresponding streams ordering them by the desired occurrence. (example: 'oryx_fantasyzoo_2': [3, 1] would place videostream 3 first and videostream 1 second for this enclosure). More details can be found in the instructions *object_detection - create_annotation_images*.
5. VIDEO_BLACK_REGIONS = Dictionary such that keys are the enclosure_video_code or the enclosure_code and the values are a list of numpy arrays signifying the polygons that are drawn in black on the video streams (after merging streams). More details can be found in the instructions *object_detection - create_annotation_images*.
6. BASE_OD_NETWORK_GLOBAL = Dictionary whose keys are either
    a. Enclosure individual code
    b. Enclosure code
    c. Species_zoo
    d. Species.

The value is the name of the object detection network that should be used. BOVIDS chooses the network based on the specificity of the keys in this dictionary (if there is a matching enclosure individual code, this network is chosen, otherwise, it will look for an enclosure code, the species zoo combination or the species). Make sure that you only write the name of the network (a folder) here and not the path (e.g. '2020-01-01_basenet_eland/'), the path will be put into "prediction/configuration.py"!

7. OD_NETWORK_LABELS_GLOBAL = Dictionary whose keys are either
    a. Enclosure individual code
    b. Enclosure code
    c. Species_zoo
    d. Species.

   The value is the name of the .txt file containing the class names corresponding to the object detection network that should be used. This file is the same as used for training the object detector. BOVIDS chooses the file based on the specificity of the keys in this dictionary. Make sure that you only write the name of the file here and not the path (e.g.: '2020-01-01_basenet_eland/classes.txt'), the path will be put into "prediction/configuration.py"!

Action classification

This section consists of four variables (dictionaries), namely BEHAVIOR_NETWORK_JOINT_GLOBAL, BEHAVIOR_NETWORK_SINGLE_FRAME_GLOBAL, BEHAVIOR_NETWORK_JOINT_GLOBAL_BINARY, BEHAVIOR_NETWORK_SINGLE_FRAME_GLOBAL_BINARY. The first one corresponds to multiple frame action classification on three classes (total), the second on the total classification with single frames. Analogously third and fourth. Therefore, only BEHAVIOR_NETWORK_JOINT_GLOBAL will be explained in detail.

1. BEHAVIOR_NETWORK_JOINT_GLOBAL = Dictionary whose keys are either
    a. Individual code
    b. Species_zoo.

   The value is the name of the action classifier used for those individuals (.h5 file). BOVIDS chooses the file based on the specificity of the keys in this dictionary. Make sure that you only write the name of the network here and not the path (e.g.: '2021-02-03_eland_actionclassification_multipleframe_total.h5'), the path will be put into "prediction/configuration.py"!

If standard networks for a broader range of individuals should be used, the user can easily adapt get_behaviornetwork() and get_object_detection_network().