

DEEP LEARNING BASED EVALUATION OF THE NIGHTLY BEHAVIOUR OF AFRICAN UNGULATES IN ZOOS

MAX HAHN-KLIMROTH^{†,*} TOBIAS KAPETANOPOULOS[†]

Goethe University,

Department of Computer Science and Mathematics, 10 Robert Mayer St, Frankfurt 60325, Germany.

JENNIFER GÜBERT PAUL WILHELM DIERKES

Goethe University, Faculty of Biological Sciences,

Bioscience Education and Zoo Biology, 13 Max-von-Laue St, Frankfurt 60438, Germany.

ABSTRACT. 1. The description and analysis of animal behaviour over long periods of time is one of the most important challenges in ecology. However, most of these studies are limited due to the time and cost required by human observers. The collection of data via video recordings allows observation periods to be extended. However, their evaluation by human observers is very time-consuming. Progress in automated evaluation, using suitable deep learning methods, seems to be a forward-looking approach to analyse even large amounts of video data in an adequate time frame.

2. In this study we present a multi-step convolutional neural network system for detecting animal behaviour states, which works with high accuracy. An important aspect of our approach is the introduction of model averaging and post-processing rules to make the system robust to outliers.

3. Our trained system achieves an in-domain classification accuracy of >0.92, which is improved to >0.96 by a post-processing step. In addition, the whole system performs even well in an out-of-domain classification task with two unknown types, achieving an average accuracy of 0.93. We provide our system at

https://github.com/Klimroth/Video-Action-Classifer-for-African-Ungulates-in-Zoos/tree/main/mrcnn_based

so that interested users can train their own models to classify images and conduct behavioural studies of wildlife.

4. The use of a multi-step convolutional neural network for fast and accurate classification of wildlife behaviour facilitates the evaluation of large amounts of image data in ecological studies and reduces the effort of manual analysis of images to a high degree. Our system also shows that post-processing rules are a suitable way to make species-specific adjustments and substantially increase the accuracy of the description of single behavioural phases (number, duration). The results in the out-of-domain classification strongly suggest that our system is robust and achieves a high degree of accuracy even for new species, so that other settings (e.g. field studies) can be considered.

E-mail addresses. {hahnklim, kapetano}@math.uni-frankfurt.de, {guebert, dierkes}@bio.uni-frankfurt.de.

Key words and phrases. convolutional neural networks, animal behaviour states, automated monitoring, deep learning tools, image classification, ecology of savannah animals.

[†] Contributed equally to this work as first authors. * Corresponding author.

9 **1.1. General.** Describing and analysing animal behaviour is a central element in ecology, ethology and neurosciences.
10 In order to characterise animal behaviour more closely and identify general behavioural patterns, it makes sense to in-
11 clude longer periods of time, different habitats and many individuals (Burger et al., 2020). While this is often a highly
12 demanding task in natural habitats, studies in zoos allow to develop, improve and evaluate methods helping to under-
13 stand behaviour patterns of various species (Kögler et al., 2020; Ryder & Feistner, 1995). Advances in digital infrastruc-
14 ture make it possible to collect and process observational data on a larger scale. However, the timely evaluation and
15 extraction of meaningful information from the mass of recorded behavioural data represents a major challenge that
16 can hardly be met by humans (Norouzzadeh et al., 2018). Consequently, to provide means of automatic evaluation of
17 animal behaviour, computer vision and deep learning techniques emerged during the last years in behavioural biology
18 and ecology (Chakravarty et al., 2020; Dell et al., 2014; Eikelboom et al., 2019; Valletta et al., 2017) .

19 Over the last decade, deep learning techniques in computer vision applications have become a crucial factor (Li
20 et al., 2019; Ng et al., 2015; Zha et al., 2015). Many state-of-the-art models that hold current benchmarks in computer
21 vision tasks like object detection or semantic segmentation use convolutional neural networks (CNNs) (Russakovsky et
22 al., 2015). Two deep learning approaches are common for inference tasks on video data. For video action classification,
23 neural networks can be trained on sequences of consecutive frames to leverage temporal features like motion that can
24 be strong cues to predict actions. These approaches work best with a medium to high frame-rate and high resolution.
25 Unfortunately, gathering such data over a longer period of time can be costly and may not be suitable for every research
26 application. Another common practice is to use a neural network for inference on single frames and inject temporal
27 higher logic to combine these predictions. This is the approach we are taking in our research presented here.

28 **1.2. Our contribution.** We present a deep learning approach to video action classification of four different behavioural
29 states of various African ungulates: *standing*, *lying – head up*, *lying – head down*, *being absent* (cf. Section 1.5). The goal
30 of our approach is to use a few manually annotated videos of individuals in a certain setting in order to subsequently
31 automatically evaluate a large video data set of this individual. This will be tackled by a three stage deep learning based
32 framework.

33 The first phase is an object recognition phase that serves three purposes. Firstly, it reduces background information
34 by localising the regions of interest that mostly consist of pixels filled by animals. It is thereby increasing the similarity
35 of sample images taken from different enclosures, which dramatically increases the power of transfer learning across
36 enclosures. Secondly, object detection can be used to distinguish between individuals within the same enclosure.
37 Lastly, it provides a clean way of detecting whether an animal is present or absent.

38 The second phase carries out a canonical classification task on the clean-cut images from phase 1. Our approach is
39 governed by an ensemble of two EfficientNetB3 (Tan & Le, 2019) image classifiers. One network predicts actions based
40 on single frame inputs and we accumulate the predictions to one prediction per time-interval (seven seconds). The
41 second classifier includes temporal dimension of the video by predicting the shown behaviour of this time-interval

directly. Therefore, the consecutive frames of this interval are concatenated to a single input (so-called *multi-frame encoding*, (Ji et al., 2013; Karpathy et al., 2014)). Subsequently, the final prediction per interval is based on an average over the predictions by the ensemble of classifiers. Finally, to further smooth predictions, we apply carefully chosen rolling averages during this process.

In the third phase, an application driven post-processing step takes place. After calculating predictions for each time-interval, we apply post-processing rules that, for instance, filter out very short activity phases of behaviours which are very unlikely to appear within the evaluated behavioural states or use information about the position of the animal in its enclosure.

1.3. Related work.

1.3.1. *Video action classification using CNNs.* Among the first appearances of CNNs for video action classification, Ji et al. (2013), Karpathy et al. (2014) discovered that encoding multiple frames performs marginally better than the frame-by-frame classification. The first milestone was reached by incorporating the temporal dimension of a video into the classification approach by different means of so-called optical flow calculations (Li et al., 2019; Ng et al., 2015; Simonyan & Zisserman, 2014; Zha et al., 2015). The current state of the art for video action classification is a two-stream approach (Feichtenhofer et al., 2016; Zhao et al., 2020) where each frame is fed into a CNN and gets predicted by a frame-by-frame classifier which gathers the spatial features of an image. In parallel, a sequence of consecutive frames is classified by a second CNN that captures the temporal dependencies of the video. The final prediction per frame is a fusion of the features given by this two streams.

1.3.2. *Deep learning approaches for action classification in behavioural studies.* In recent years, the use of computer vision and deep learning techniques has emerged in behavioural biology tasks (Christin et al., 2019; Dell et al., 2014; Valletta et al., 2017). Papers of this kind should be clustered by the nature of the data used. One class of experiments was performed under laboratory conditions: high frame-rate videos with a high contrast. A prominent example is the JAABA (Kabra et al., 2012) toolbox for video classification of behaviours of mice and *Drosophila* flies. Another example is *DeepBehaviour* (Graving et al., 2019) which is used to detect and track the trajectories of mice in a laboratory. Moreover, Stern et al. (2015) present a system for object detection and behaviour classification: they predict with great accuracy whether a *Drosophila*-fly is on some substrate or not.

Other projects need to process data recorded in the wild where the recorded image or video material poses a much greater challenge as variations in background, brightness, weather, camera specifics and recording angle etc. lead to highly complex datasets. For instance, Porto et al. (2013) presents a computer vision-based classifier using the Viola Jones detection algorithm to distinguish lying behaviour of dairy cows in free-stall stables. Norouzzadeh et al. (2018) use camera traps in the Serengeti to answer research questions on numbers, types and behaviour of recorded (larger) African mammals. Their behaviour classification task is to distinguish between the five activities standing, resting, moving, eating and interacting, for each detected individual. They apply a deep learning system harnessing 1.4 million images from the Snapshot Serengeti Dataset (Swanson et al., 2015) available to them.

1.4. **Our Objectives.** Understanding the behaviour of animals is a key element of ecology. For example, behavioural studies can improve our understanding of the habitat requirements or migration patterns of species, which in turn have important implications for nature conservation issues (Melzheimer et al., 2020; Teitelbaum et al., 2015). However, animal behaviour is complex, contextual and species-specific, so approach and analysis must differ depending on the thematic focus, the environmental variables or even the species themselves. In this context, videography is an inexpensive, non-invasive method for documenting animal behaviour. Although the manual methods of video evaluation allow for differentiated behavioural analysis, they are also very time-consuming, so that longer quantitative analyses are limited. Under controlled laboratory conditions valid solutions based on computer vision algorithms are available today, which allow to perform behavioural analyses routinely (cf. Section 1.3.2). On the other hand, for data recorded in setups where the environment variables are much more complex or the available image material is of lower quality, automatizing the evaluation process posed a major challenge for researchers so far.

A key objective of this work is to combine recent successes of deep learning with domain knowledge and expertise from behavioural biology. Our overall objective is to establish a pipeline that produces high-quality action classification with only little human labelling effort involved. In this study, the main objective is to build an accurate automatic pipeline to classify behaviours of animals recorded in zoo enclosures. We aim to achieve this using open-source software, low-budget technical equipment and make our code openly available on github, so that it may be easily reproducible by other research groups. We showcase a procedure that allows to significantly reduce manual labelling endeavours while maintaining high-quality labels in a controlled manner. The procedure goes as follows:

- Let a researcher manually label a small set of nights of an unknown individual.
- Split these into train and test set, i.e. reserve at least one night as hold-out test set.
- Fine-tune the object detection and the classification networks on the train data and evaluate the performance on the test set.
- Accuracy on the test set can further be improved by adding more labelled nights and tuning a post-processor.

Given a pool of existing labelled data from 10 different species from the order of Cetartiodactyla, we aim to further predict unlabelled nights from the same or other individuals of that species. We therefore split nights into single-activity time-intervals (seven seconds long) and predict one out of four behavioural states: *standing*, *lying – head up*, *lying – head down* or *being absent* which are explained in Section 1.5. On the one hand, we are interested in the performance of neural networks on the task of inferring these states per interval. On the other hand, it is crucial that the entire system is also capable of predicting the behavioural phases of entire recording nights in such a way that typical biological parameters such as the number and duration of the phases are sufficiently accurate in order to use these predictions for behavioural research studies. Finally, we also investigate a slightly easier task: distinguishing *standing* from *lying* (independent from the head's position), which is of great interest for the identification of rhythmic activity patterns in nocturnal behaviour. We will refer to this as the task of *binary classification*.

1.5. **Background.** In order to keep studies comparable, behavioural research works with standardised ethograms that allow comparisons within a species or a related systematic group (Stanton et al., 2015). Therefore, the definition of annotated behavioural states is explained below. Our study focuses on the three basic behavioural categories: *standing*, *lying – head up* and *lying – head down*, which are defined in the following ethogram.

- **Standing:** The animal stands in an upright position on all four hooves. It does not matter what the animal is doing in this position, so for example it could be feeding, resting, walking or ruminating.
- **Lying – head up (LHU):** The animal's body lies on the ground and the head is lifted. We do not distinguish between being awake or being in the non-REM-sleep, furthermore, the animal could also be feeding, ruminating or resting.
- **Lying – head down (LHD):** The animal is lying with its head rested. The resting head lies down on the ground and is placed beside the body or sometimes in front of it.

A visualisation of each state can be found in Fig. 1. Additionally, if the animal cannot be seen in a frame, the desired label is *being absent*.



FIGURE 1. For three different species (top to bottom): Common Eland (*Taurotragus oryx*), Common Wildebeest (*Connochaetes taurinus*) and Waterbuk (*Kobus ellipsiprymnus*), the three behavioural states (left to right) standing, LHU and LHD are shown.

At this point, we shortly want to stress that LHD is a valid indicator for recognizing REM sleep. Indeed, identifying REM-sleep by a characteristic posture is a common practice in behavioural studies based on image and video material (Ternman et al., 2014). This is due to *postural atonia* being a characteristic of REM sleep (Lima et al., 2005; Zepelin et al., 2005), therefore, due to the lack of muscle tone, any body part (including the animal's head) needs to be laid down. Furthermore, at least for cows, it is well known that this kind of behavioural estimation for REM-sleep is highly sensitive (Ternman et al., 2014).

2. METHODS AND MATERIAL

2.1. The deep learning approach. Deep learning has three key drivers: algorithms, data and computational resources. For the first two stages of our prediction pipeline we apply deep learning algorithms from the last few years, whereby an ensemble of three neural networks has been established. However, we strongly believe that the specifics of their design are of less relevance and they could easily be exchanged with other neural networks deemed state-of-the-art for the respective tasks. In contrast, the data used for training the neural networks and evaluating their performance plays a crucial role for the experiments, hence we dedicated Subsection 2.2 to discuss it in great detail. Lastly, we were able to perform all experiments with just a single, mediocre GPU (RTX 2070). For all three models, the total training time amounted to 840 hours, and the entire pipeline now predicts behaviours for one hour of video material in 15 minutes.

2.2. Data. The data for this project spans 209 nights (2926 hours) of recordings of 65 individuals out of 10 different species, see Table 1. The videos were taken over the last three years with either a *Lupus LE139HD* or *Lupus LE338HD* camera stemming from zoo enclosures of one Dutch and ten German zoos. They have a frame-rate of 1 fps and a resolution of either 1080p or 720p. The recording time mostly ranges from 5 p.m. to 7 a.m., i.e. the time where the animal keepers are mostly absent, with night vision using the build-in infrared emitters of the cameras.

Compared to previous studies in behavioural biology (Graving et al., 2019; Kabra et al., 2012; Stern et al., 2015) recorded under *laboratory conditions*, our data is much more complex and noisy. Installing the cameras properly faces major issues as the enclosure structure and the husbandry is given by the zoos, i.e. the existing, limited installation options must be used if available and the animals should not be disturbed by the cameras. This leads to huge differences from enclosure to enclosure regarding the position and the angle in which the cameras can be installed. Furthermore, the angle of the camera might change due to external influences, visibility might worsen because of dirt sticking on the lens and the animals should not be able to reach the camera leading to a high degree of occlusions (sometimes the installation needs to be outside of the enclosure box) or truncation effects (blind spots in the enclosures). Some edge cases are illustrated and further elaborated on in the Appendix A.

For the task of object detection we have manually annotated bounding boxes for nearly 26k randomly sampled images – a detailed per species listing is provided in Table 2. A subset of 10% of these images is used as a test set, the remaining 90% build the training set of the object detector. For the main task of classifying behaviour we have complete labels for all 209 nights. For one common wildebeest, one bongo and three common elands we keep a hold-out

set of some nights for testing (these are the same nights containing the test images for object detection). Out of all other nights we randomly select a training set of about 95k images such that the three classes standing, LHU and LHD are almost balanced in number. For further evaluation of the single frame and single interval performance of the neural network predictors we proceed similarly with the test nights to obtain 6k images for the Common Elands and 4k images for each of the other two species, respectively (cf. Table 1). We refer to this subset of the test set as the validation set.

species	# zoos	# individuals	# labelled ~ 14 hour videos
Training data			
Common Wildebeest (<i>Connochaetes taurinus</i>)	3	11	22
Blesbok (<i>Damaliscus pygargus</i>)	2	4	8
Roan Antelope (<i>Hippotragus equinus</i>)	2	4	8
Sable Antelope (<i>Hippotragus niger</i>)	1	2	4
Waterbuck (<i>Kobus ellipsiprymnus</i>)	3	10	20
Bongo (<i>Tragelaphus eurycerus</i>)	2	9	18
Greater Kudu (<i>Tragelaphus strepsiceros</i>)	3	8	17
Common Eland (<i>Tragelaphus oryx</i>)	4	12	80
Sitatunga (<i>Tragelaphus speki</i>)	1	1	2
Okapi (<i>Okapia johnstoni</i>)	1	2	4
Testing data			
Common Wildebeest (<i>Connochaetes taurinus</i>)	1	1	2
Bongo (<i>Tragelaphus eurycerus</i>)	1	1	2
Common Eland (<i>Tragelaphus oryx</i>)	2	3	22

TABLE 1. The variety of data used for training and testing the deep learning classifier.

species	# zoos	# individuals	# training images
Common Wildebeest (<i>Connochaetes taurinus</i>)	3	9	3772
Blesbok (<i>Damaliscus pygargus</i>)	1	2	808
Roan Antelope (<i>Hippotragus equinus</i>)	2	4	1726
Sable Antelope (<i>Hippotragus niger</i>)	1	2	1058
Waterbuck (<i>Kobus ellipsiprymnus</i>)	4	11	4751
Bongo (<i>Tragelaphus eurycerus</i>)	2	10	4472
Greater Kudu (<i>Tragelaphus strepsiceros</i>)	3	4	1756
Common Eland (<i>Tragelaphus oryx</i>)	4	14	6913
Sitatunga (<i>Tragelaphus speki</i>)	1	1	542

TABLE 2. The training set for the object detector.

2.3. Phase 1: Object detection. The objective of phase 1 is to localise individuals by drawing a minimal rectangular bounding box around them, which can be cut-out and further classified into the action classes in phase 2. If no individual is detected, we can already predict the class as *being absent*. For object detection on single image frames we fine-tune a Mask R-CNN with ResNet-101 backbone that was pre-trained on the MS COCO database (Lin et al., 2014), which has animal object classes like zebras, elephants, dogs etc. and is hence a good base for transfer learning to our dataset. More precisely, we use the Matterport implementation (Waleed, 2017) of Mask R-CNN and fine-tune on the training data described above for 50 epochs to detect animals out of the listed 9 species. Due to some tough truncation

occurring in our data, we further run one round of offline hard example mining (Felzenszwalb et al., 2010): For each animal we run the trained model on 400 images from the nights used for training and inspected the obtained predictions. Then, the model failures, i.e. the poorly predicted bounding boxes, were re-annotated by hand and finally the network was re-trained for 15 epochs including these additional annotations.

After the per image prediction we apply the following post-processing steps that helped to make the overall predictions more robust to edge cases in the data and erroneous localization predictions. We only keep bounding box predictions of which the net’s confidence is at least 97%. We also allow a maximum of one box per image. At a first glance, this approach looks tailored to enclosures with one individual, but can, in fact, be easily extended to detect and distinguish multiple individuals within the same enclosure.

2.4. Phase 2: Action classification. In phase 2, we predict the action displayed in short sequences of cut-out frames. We follow a successful approach to video action classification (cf. Section 1.3.1) that is based on a two-stream system – the image frame is input in the first stream and motion cues from the temporal context are fed into the second stream. For this second input, optical flow is a common choice – which we tried as well, but found the performance to be inferior to the model we will describe below. A small ablation study and discussion on this can be found in the Appendix B and multi-frame encoding was chosen as an alternative way to input pixel motion information as a result thereof.

The inputs to stream 1 are the cut-out boxes from phase 1, resized to a resolution of 300x300 pixels. A single input for the second stream consists of a four frame encoding of a 7s time-interval¹. The corresponding four cut-out boxes are resized to 150x150 pixels each and then combined to the same input size as stream 1. EfficientNet B3 (Tan & Le, 2019) was used for both classification tasks – a convolutional neural network which has proven itself to achieve state-of-the-art accuracy in vision classification tasks while being smaller and faster than comparable models. For both streams we use a network pre-trained on the ImageNet dataset (Xie et al., 2019; Yakubovskiy, 2019) with a customized classification head of three output units each. The networks were trained for 30 epochs with a batch size of 8, categorical cross-entropy loss and the Adam optimizer (Kingma & Ba, 2014) with initial learning rate 10^{-3} and exponential decay of 0.9. We further applied the following input augmentation steps during training: random center cropping by 0-16px, random horizontal flipping, random Gaussian blurring, brightness and contrast augmentation, and finally, random rotation by -25 to +25 degree.

2.5. Phase 3: Post-Processing. Finally, we apply a series of post-processing operations to make our prediction pipeline more robust and fitting to the task of predicting accurate time intervals of animal behaviour and leverage our knowledge on the temporal consistency of the data. To begin with, we average model predictions between the two streams of phase 2 and between consecutive intervals by applying a rolling average. An overview of the prediction pipeline up to this stage is illustrated in Fig. 2 and the details of the implementation can be found in Appendix C.

¹We take the first, third, fifth and seventh frame and arrange them as a square, cf. the image in Fig. 2

Next, we incorporate application driven rules to smooth predictions over time and include our domain knowledge of the animal's behaviours. As the steps before introduce only a weak temporal context we still observe flickering of the predictions due to small misclassifications or data edge cases. For example, in case an individual is heavily truncated or occluded, the predictions of consecutive intervals might jump between absent and other actions. Furthermore, we reject certain types of transitions that would lead to unrealistic short intervals of activity, such as a short sequence of standing between LHD events, and just keep the previous behaviour in such cases. To sum up, we obtain the final predictions by following the transition rules listed in Table 3.

previous behaviour	current behaviour	next behaviour	min. # time-intervals
standing / LHU	LHD	standing / LHU	3
LHD	LHU	LHD / standing	6
standing	LHU	LHD	6
standing	LHU	standing	25
LHD / LHU	standing	LHD / LHU	25
LHD / LHU / standing	being absent	LHD / LHU / standing	50

TABLE 3. Post-processing rules applied to the data of the elands and the bongo as well as the wildebeest. If the system detects a sequence of (previous behaviour, current behaviour, next behaviour) where the current behaviour is shorter than described, it will replace it by the previous behaviour.

2.6. Evaluation. Our objectives stated in Section 1.4 require to extend the usual testing ground for classification tasks: We are highly interested in the overall performance of the system on complete videos of known and unknown individuals. Therefore, we designed our test sets with respect to three levels of knowledge: Pure in-domain classification, weak in-domain classification and out-of-domain classification – testing for different levels of generalisation capabilities of our pipeline.

The easiest level – *pure in-domain* classification – describes the task of filling in missing behaviours for nights where some frames have already been labelled and used for training. This comes close to the usual test setup, only that the classes during training are balanced, while for this test set they are not.

For the remaining two levels, we tackle a prediction task more challenging than usually performed in statistical learning. In the classical setup, the entire dataset is split randomly, i.e. train and test set consist of independent and identically distributed samples from one and the same data-generating distribution. Thus, in our case this translates to train and test images being taken from the same set of nights. If instead test images are from new unseen nights, then they come from a shifted data distribution – in these nights the arrangement of the enclosure and the light conditions may be quite different from those nights of the training set. We define the *weak in-domain* classification task as classifying videos of an individual present in the training data but on nights which were not used for training. Lastly, we take this one step further in the *out-of-domain* classification task, where the system is evaluated on videos from individuals that did not appear in the training set – a far more severe distribution shift. Deep Learning systems are known to be brittle to distribution shifts (Quionero-Candela et al., 2009; Recht et al., 2019), hence the latter is a quite intricate challenge.

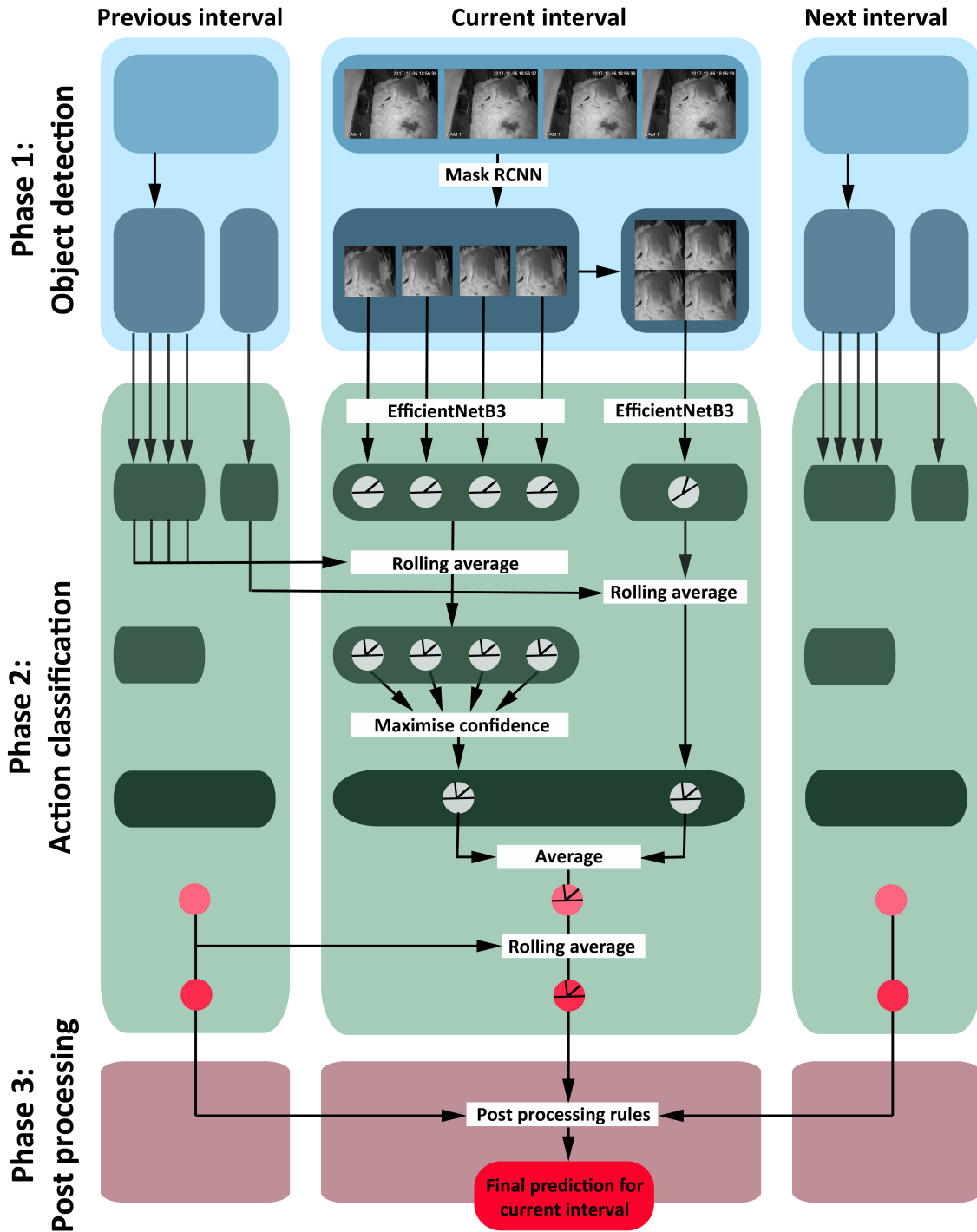


FIGURE 2. A visualisation of the prediction pipeline applied to each time-interval starting from the original video input of consecutive single frames. The circles represent predictions, thus probability measures on either {standing, LHU, LHD} or on {standing, LHU, LHD, being absent}.

230 To evaluate our action classifiers on single frames and single intervals, we use four commonly used measures for
 231 predictive performance: accuracy, recall, precision and f-score, which are defined below for completeness. To this

regard let

$$\mathbf{1}_{\{\mathcal{E}\}} = \begin{cases} 1, & \text{if } \mathcal{E} \text{ is true} \\ 0, & \text{otherwise} \end{cases}$$

denote the indicator function that takes the value 1 if (and only if) the expression \mathcal{E} is true. For n test intervals denote by $y = (y_1, \dots, y_n)$ their ground-truth label classes and by $\hat{y} = (\hat{y}_1, \dots, \hat{y}_n)$ the corresponding predictions by a model. The *accuracy* of the predictions \hat{y} is the proportion of correctly predicted labels, thus

$$\text{accuracy}(\hat{y}) = \frac{\sum_{i=1}^n \mathbf{1}_{\{\hat{y}_i = y_i\}}}{n}.$$

Despite this general performance measure, we introduce the following metrics to further illuminate performance per classification class c

$$\text{recall}_c(\hat{y}) = \frac{\sum_{i=1}^n \mathbf{1}_{\{\hat{y}_i = c, y_i = c\}}}{\sum_{i=1}^n \mathbf{1}_{\{y_i = c\}}} \quad \text{and} \quad \text{precision}_c(\hat{y}) = \frac{\sum_{i=1}^n \mathbf{1}_{\{\hat{y}_i = c, y_i = c\}}}{\sum_{i=1}^n \mathbf{1}_{\{\hat{y}_i = c\}}}.$$

The *recall* (or sensitivity) for a class c is the proportion of correct predictions of that class among all occurrences of the class label in the ground-truth, i.e. how many of this target intervals are predicted correctly. Naturally, recall can be increased by predicting this class more often, but in this case the potential for false predictions of this class rises. Hence, the *precision* (or positive predictive value) describes the proportion of the correct predictions among all predictions of this class. As these two values stress complementary performance properties, the *f-score* is defined as the harmonic mean of these two

$$\text{f-score}(x) = 2 \frac{\text{recall}(x) \times \text{precision}(x)}{\text{recall}(x) + \text{precision}(x)}$$

and gives a good measure for the overall prediction performance per class.

Furthermore, we evaluate how application specific key figures are predicted. More precisely, the two key figures *amount of phases of a specific behaviour* and *total duration of a specific behaviour* are examined. Clearly, a good accuracy implies that the latter will be estimated quite well. The amount of phases, however, might not be estimated reliably if there is a lot of flickering in the predictions.

Finally, in order to evaluate the performance of the object detector, we apply the commonly used *average precision* (AP) metric with different *Intersection over Union* (IoU) thresholds. IoU is defined as the ratio of the area of intersection and area of union of the bounding box between the predicted and the ground-truth bounding box. As the object detection phase momentarily only distinguishes between the two classes *individual* and *background*, the AP@ t value equals the percentage of predicted bounding boxes that exhibit an IoU of at least $t\%$ with the ground-truth bounding box.

3. RESULTS

3.1. Evaluating the deep learning components. Before analysing and discussing the core target evaluation measures introduced in Section 2.6, let us first state the results for the single deep learning components. The results of the object detection component can be found in Fig. 3. It achieves an AP@75 of more than 0.95 on the whole testing set as well as on the class of elands in the testing set.

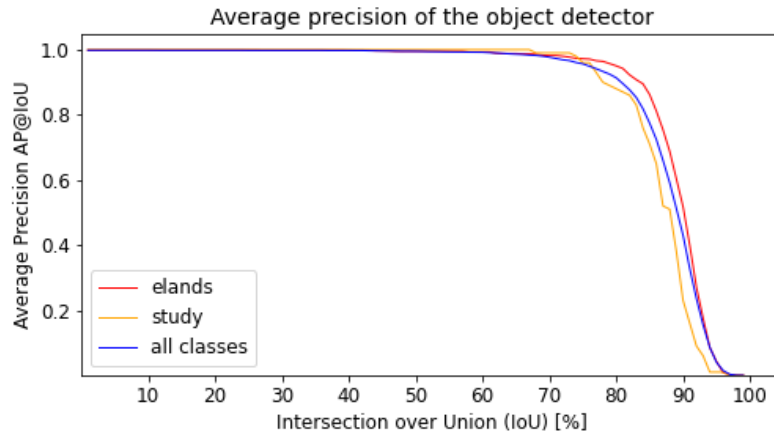


FIGURE 3. Average precision of the object detector (phase 1 of the deep learning pipeline). We report the mean AP values over all classes, the AP on all images of elands as well as the AP on the images presented during the human study.

For the action classification task, we first report performance on the balanced validation set, so that this leads to a testing environment compatible with common practices in deep learning. We achieve a testing accuracy of 0.881 for stream 1 and 0.954 for stream 2. Due to the specifics of the classification task and the data these numbers can hardly be set into comparison with typical benchmark classification tasks like the ImageNet Large Scale Visual Recognition Challenge. Consequently, to better assess the performance of our models, we conducted a human study where experts (E, $n = 11$) and novices (N, $n = 11$) received 100 randomly chosen single frames from the validation set. Both groups were given the same images, but once cut-out and later as the original entire frame. The result of the participants vs. stream 1 and 2 are listed in Table 4.

predictor	accuracy	f-score		
		standing	LHU	LHD
enclosure (E)	0.83 ± 0.03	0.93 ± 0.01	0.83 ± 0.03	0.75 ± 0.05
enclosure (N)	0.74 ± 0.03	0.87 ± 0.02	0.75 ± 0.03	0.57 ± 0.09
cut-out (E)	0.77 ± 0.03	0.87 ± 0.02	0.76 ± 0.02	0.70 ± 0.08
cut-out (N)	0.63 ± 0.04	0.77 ± 0.03	0.63 ± 0.04	0.56 ± 0.08
steam 1	0.84	0.95	0.81	0.68
steam 2	0.93	0.90	0.93	0.92

TABLE 4. Showing the results of the study comparing the accuracy and the f-score on all 100 images (mean \pm SEM), once presented as an image of the whole enclosure and once cut-out by the object detector. Those values are reported for the group of experts (E, $n = 11$) as well as the group of novices (N, $n = 11$) and for the two streams of the deep learning system.

The standalone performance of stream 1 can be easily compared with the cut-out performance of the human predictors. We see that it clearly outperforms the novices and slightly outperforms the expert, except for the LHD class, where some experts perform better. Comparing with the human predictors on enclosure level, stream 1 still outperforms the novices, while it performs on par with the experts. As the humans lose around 5-10% in performance through the cut-out process, stream 1 has to compensate for this imprecision of the object detection phase and still achieves human expert performance. Moreover, we add stream 2 in this table as well, knowing that its input spans 7s time-intervals which gives it a clear advantage over both humans and stream 1. Nevertheless, it is remarkably that this is enough to clearly outperform stream 1 and even the experts' predictions on the entire frame in all but the standing class. This underlines the benefits of including temporal information into the model. Still, stream 1 yields a useful addition as it has different strong points than stream 2, such as classifying standing, and hence, we see below that model averaging improves the overall prediction quality of the pipeline significantly. To conclude, the validation accuracy of both streams can be considered quite high and verify that the model generalises quite well, even more so, considering the data quality and possible label ambiguities cf. Appendix A.

3.2. Performance of the overall pipeline. In the following we present test results for time-interval predictions of stream 1, stream 2, the fusion step and after post-processing for three levels of generalisation performance as outlined in Section 2.6. The results are presented in Table 5 subdivided into the performance for individual animals. We furthermore report average recall, precision and f-score for the overall predictions and the accuracy for the binary classification task.

		Classifying standing, LHU and LHD				Binary classification
	# nights	avg. accuracy stream 1	avg. accuracy stream 2	avg. accuracy fused streams	avg. accuracy post-processed	avg. accuracy post-processed
Pure in-domain classification						
Eland 1	6	0.986 ± 0.004	0.969 ± 0.009	0.974 ± 0.007	0.978 ± 0.006	0.992 ± 0.003
Eland 2	2	0.989 ± 0.001	0.985 ± 0.000	0.989 ± 0.001	0.994 ± 0.001	0.998 ± 0.000
Eland 3	2	0.921 ± 0.046	0.887 ± 0.062	0.921 ± 0.040	0.963 ± 0.018	0.9973 ± 0.018
Weak in-domain classification						
Eland 1	8	0.936 ± 0.027	0.914 ± 0.030	0.924 ± 0.028	0.976 ± 0.007	0.982 ± 0.007
Eland 2	2	0.971 ± 0.012	0.960 ± 0.011	0.967 ± 0.013	0.977 ± 0.012	0.980 ± 0.013
Eland 3	2	0.960 ± 0.007	0.956 ± 0.006	0.970 ± 0.005	0.986 ± 0.001	0.997 ± 0.001
Out-of-domain classification						
Bongo	2	0.930 ± 0.023	0.945 ± 0.015	0.945 ± 0.015	0.944 ± 0.010	0.990 ± 0.002
Wildebeest	2	0.888 ± 0.020	0.867 ± 0.018	0.896 ± 0.011	0.913 ± 0.011	0.995 ± 0.002

TABLE 5. The accuracy reached by the different streams of the deep learning system. We report the accuracy as well as the SEM for both classification tasks.

The final results show an accuracy of at least 0.96 in all in-domain classification tasks for each of the three Elands tested. As for Eland 1 we have tested on the most data, these results should be considered the ones with highest statistical significance, where we achieve consistently above 0.97. In total, all components perform well above 0.90 with the single exception of stream 2 for Eland 3 in in-domain classification. However, this and other weak performing instances are well accounted for by the post-processor. The low performances could be due to longer phases of difficult

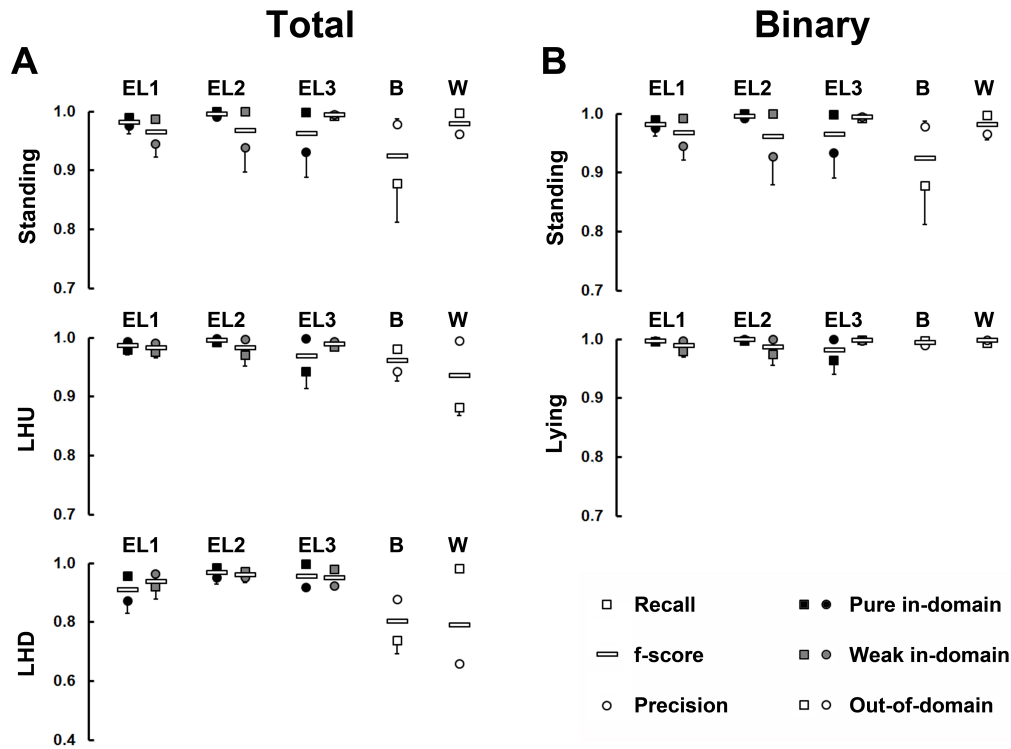


FIGURE 4. Overview on recall, precision and f-score for each behavioural class and each individual (EL–Eland, B–Bongo, W–Wilbeest) produced by the classification system on the testing dataset.

data, e.g. the animal spending much time in parts of the enclosure which are not fully visible (cf. Appendix A), or from a different distribution of actions, favouring actions like LHD where the model's performance is slightly worse as we see in Fig. 4, Column A. When comparing the f-scores for the three action classes we see for all three elands that performance is weakest for LHD. In contrast, the f-scores for the binary task are much higher (Fig. 4, Column B), hence most mistakes made by the model stem from confusing the classes LHD and LHD.

Overall, the system performs well in the weak in-domain and the pure in-domain classification task, which shows that the deep learning models learn predictive features from characteristic postures of the behavioural states instead of memorising the training data. This is a good indication of robustness and generalisation capabilities of our model. This indistinguishability is found with respect to the accuracy score as well as with respect to recall, precision and f-score. In addition, the out-of-domain result for the classification of the behaviour of bongos and wildebeests with 0.94 and 0.91, respectively, also shows good testing accuracy. As expected, performance is slightly worse here, but given the system has not seen any data of these individuals, the values are quite high.

In the task of binary classification, our model succeeds with an accuracy of above 0.99 for pure in-domain classification, above 0.98 for weak in-domain classification and above 0.99 even for out-of-domain classification. This showcases the model's strong ability to reliably distinguish the features of standing and lying.

Finally, while the accuracy between the fused streams and the post-processed prediction does not vary that much, a suitable choice of post-processing rules effects the precision of the prediction of the average number of phases per video. Corresponding results will be stated in the next section.

3.3. On behavioural biological key figures. Finally, we turn to our last objective, namely predicting the number of activity phases per night and their total duration. Table 6 reports the average key figures over all predicted nights in comparison to the real quantities, again for three levels of generalisation.

	standing			LHU			LHD		
	real	w/o pp	pp	real	w/o pp	pp	real	w/o pp	pp
Pure in-domain classification									
Eland 1									
Avg. # phases	8.2 ± 0.3	26.5 ± 5.3	8.3 ± 0.4	17.2 ± 1.8	50.7 ± 4.4	17.5 ± 1.6	9.0 ± 1.6	26.2 ± 3.3	9.3 ± 1.4
Avg. duration [min]	195.2 ± 9.6	197.6 ± 11.0	197.8 ± 8.8	589.5 ± 10.0	575.9 ± 15.8	582.4 ± 12.4	51.5 ± 9.8	63.3 ± 15.0	58.7 ± 12.5
Eland 2									
Avg. # phases	6.0 ± 0.0	16.5 ± 1.5	6.0 ± 0.0	18.0 ± 1.4	32.5 ± 4.5	17.0 ± 1.4	11.0 ± 1.4	21.5 ± 4.5	11.5 ± 1.8
Avg. duration [min]	249.2 ± 0.5	247.7 ± 0.0	251.5 ± 0.4	543.3 ± 4.3	16.5 ± 1.5	544.1 ± 3.3	42.7 ± 1.8	46.2 ± 4.0	44.4 ± 2.9
Eland 3									
Avg. # phases	7.0 ± 0.0	66.0 ± 34.0	7.5 ± 0.4	22.0 ± 2.1	129.5 ± 70.5	24.0 ± 0.0	14.5 ± 1.8	37.5 ± 5.5	16.5 ± 0.4
Avg. duration [min]	222.5 ± 2.2	232.9 ± 1.1	239.2 ± 8.6	532.5 ± 1.7	468.1 ± 48.9	502.6 ± 16.8	85.1 ± 3.9	94.4 ± 5.4	92.4 ± 4.1
Weak in-domain classification									
Eland 1									
Avg. # phases	7.0 ± 0.3	88.2 ± 25.4	7.4 ± 0.2	14.5 ± 1.1	97.7 ± 24.3	15.1 ± 1.1	7.9 ± 1.0	24.8 ± 5.3	8.5 ± 1.2
Avg. duration [min]	211.9 ± 10.1	275.2 ± 24.6	220.0 ± 12.4	580.2 ± 7.6	516.8 ± 22.0	571.8 ± 9.9	38.4 ± 4.5	37.6 ± 5.2	37.0 ± 4.9
Eland 2									
Avg. # phases	6.5 ± 0.4	40.0 ± 11.0	8.5 ± 1.8	22.5 ± 0.4	53.5 ± 13.5	24.0 ± 2.8	15.5 ± 0.4	26.5 ± 4.5	16.0 ± 1.4
Avg. duration [min]	216.4 ± 10.8	230.9 ± 3.7	230.2 ± 1.6	563.6 ± 9.4	542.6 ± 5.0	548.7 ± 1.3	59.6 ± 1.7	63.1 ± 4.0	61.1 ± 2.9
Eland 3									
Avg. # phases	7.0 ± 0.7	23.5 ± 2.5	6.5 ± 0.4	21.0 ± 0.7	65.5 ± 8.5	22.0 ± 0.0	13.0 ± 1.4	35.0 ± 6.0	14.5 ± 0.4
Avg. duration [min]	234.3 ± 6.9	236.0 ± 9.4	233.8 ± 8.1	517.8 ± 11.5	498.6 ± 6.0	513.7 ± 10.5	87.4 ± 4.2	90.5 ± 1.4	92.5 ± 2.4
Out-of-domain classification									
Bongo									
Avg. # phases	10.5 ± 1.1	18.5 ± 2.5	7.0 ± 0.0	41.0 ± 2.8	150.5 ± 63.5	40.5 ± 4.6	29.5 ± 3.9	135.5 ± 63.5	33.0 ± 4.9
Avg. duration [min]	105.0 ± 42.7	102.2 ± 58.9	99.3 ± 44.5	631.0 ± 60.1	593.8 ± 100.5	655.2 ± 53.2	104.1 ± 17.4	140.8 ± 39.7	85.5 ± 8.7
Wildebeest									
Avg. # phases	8.5 ± 0.4	10.0 ± 0.0	9.0 ± 0.7	31.5 ± 3.9	32.0 ± 4	27.5 ± 1.8	23.0 ± 4.2	25.5 ± 3.5	22.0 ± 0.7
Avg. duration [min]	114.2 ± 14.7	118.8 ± 23.0	118.6 ± 16.1	597.2 ± 0.0	530.25 ± 10.6	529.1 ± 7.0	128.7 ± 14.6	191.0 ± 33.6	192.3 ± 23.1

TABLE 6. Overview on the accuracy of the deep learning system predicting the amount of phases and the average duration per night for the three behavioural states. The value in *w/o pp* is the output after fusing streams 1 and 2 while row *pp* lists the prediction after post-processing was applied. We report those quantities and their SEM for the pure in-domain, the weak in-domain and out-of-domain classification.

The results of the in-domain classification show that all predicted values generally agree well with the real values. This success had not been expected in advance, as deep learning pipelines like ours are generally prone to produce flickering. We were, however, able to smooth these out sufficiently by broadening the input data distribution through extensive augmentations and by the post-processing rules. For the average number of phases the errors made by our model are mostly below 10% and for average duration the overlap with the ground-truth is even higher. We stress at this point, that the decent performance of our deep learning pipeline is influenced by the post-processor heavily. For instance, even if the accuracy and the f-score between the fused streams and the post-processed prediction do not vary

much, the amount of phases per night is drastically overshoot for all individuals and over all classification tasks without post-processing.

Again, we point out the very reliant prediction of our pipeline for Eland 1. The quantities for out-of-domain application are mostly predicted with only small errors as well, even though they are of very different scale. The larger errors like LHU and LHD average duration for the Wildebeest are likely to be reduced through transfer-learning on a small amount of data containing videos from this individual. Furthermore, the post-processing rules might need to be adapted to the species. Other than this, the system only systematically overshoots the amount of LHU and LHD phases of Eland 3. Investigating this error we found the reason being short periods of LHU being misclassified as LHD. Remarkably, in almost all these falsely classified phases, the eland shows a grooming behaviour (at his hind leg) which cannot be distinguished from the LHD on a single image – nevertheless, identification of grooming phases given a video sequence is possible due to the head’s slight movement. Such kind of errors vanish, of course, in the binary classification task what can, for completeness, be seen in Table 7 of Appendix D.

4. DISCUSSION

The first part of our model pipeline succeeds strikingly in detecting individuals in their enclosures. As object detection is one showcase task for deep learning this was to be expected, but still our results are notably high for such a task. State-of-the-art performance on the COCO dataset (Lin et al., 2014) by very recent models like YOLOv4 (Bochkovskiy et al., 2020) or EfficientDet (Tan et al., 2020) achieve an AP@75 of less than 60, however this across many object classes and in very diverse scenes. Moreover, phase 2 of our deep-learning pipeline may still predict actions correctly even if phase 1 performs slightly erroneous localisation, i.e. failures with respect to the AP metric may still produce cut-out images with which actions can be predicted reliable, e.g. if the bounding box is slightly too big or part of the animal is truncated, which also occurs naturally due to truncation at the image borders. We conclude that our model performs the detection phase with great accuracy and robustness.

To put our action classification results into context, it is crucial to compare data variety and complexity. The data for our deep learning system consists of low frame-rate videos recorded under challenging conditions: various enclosures, zoos, species and individuals are to be dealt with. Furthermore, the installation of the cameras was subject to restrictive conditions – the videos were recorded at night with the use of infrared emitters, from different camera angles and sometimes with parts of the enclosure missing or obscured. To sum up, our data-distribution is much more intricate than the laboratory conditions of most previous approaches (Graving et al., 2019; Kabra et al., 2012; Stern et al., 2015; Weygandt & Mathis, 2020) and is in its complexity on par with Porto et al. (2013).

Porto et al. (2013) create a system that works for a specific enclosure (in-domain classification) and they achieve a classification accuracy of 0.92 on a binary classification task distinguishing between standing and lying behaviours. Data-wise Porto et al. (2013) were able to capture enclosures from a bird’s-eye view without occlusions or truncation, arguably leading to a better starting point for classification than our data. Despite this, our models perform notably

361 better by achieving an average weak in-domain accuracy of 0.979 for classifying three action classes and performance
362 even improves to 0.986 accuracy when reducing to only standing and lying behaviour classes.

363 Our data is less complex than the data used by Norouzzadeh et al. (2018) who have single images taken as snapshots
364 in the wild under various light and weather conditions. They achieve an in-domain accuracy of 0.762 with respect to
365 classifying six (non-exclusive) behaviour classes of over 26 species. This situation could rather be compared to the
366 out-of-domain performance of stream 1 (so without leveraging temporal context), where we achieved an accuracy of
367 0.93 for the Bongo and 0.888 for the Wildebeest.

368
369 Finally, for the biological key figures our model recognises most sequences correctly. More precisely, the few errors
370 occurring during prediction seem to average out very well over multiple videos, also in the weak in-domain classifi-
371 cation task. On this basis, our model can be used to automatically label raw data recordings from Elands 1-3 without
372 further human supervision. With the application presented here, the bottleneck of many behavioural biological stud-
373 ies could be overcome – manually evaluating a huge stock of recorded raw data. We are confident that our methods
374 transfer well to different studies, our high out-of-domain accuracy is a good indication for this. Hence, our approach
375 may be used in black box fashion by only adapting the post-processing rules to the specialties of the animal's be-
376 haviour. Even more though, when having already established a well-performing system, as usual in transfer learning,
377 the amount of labelled data needed for fine-tuning is likely to decrease significantly.

378 Machine learning applications have the potential to greatly expand the scope of ecological behavioural studies in
379 this area (Christin et al., 2019), as large amounts of data can be analysed in a reasonable time frame and the effort for
380 manual analysis is drastically reduced (Tabak et al., 2018). For the investigation of complex behaviour and movement
381 patterns in the wild, for example, sensors are used that record acceleration data in addition to GPS data. For instance,
382 Rast et al. (2020) present a framework for recording the behaviour of wild red foxes based on an artificial neural net-
383 work (ANN) trained on captive red foxes. The latter aspect shows that captive animals can play an important role in
384 methodological developments. However, this method is limited in its breadth by the number of individuals equipped
385 with transmitters and, although within a moderate range, is an invasive research method. A non-invasive approach to
386 behavioural research in the wild is the analysis of image or video material using convolutional neural networks (Fer-
387 reira et al., 2020; Tabak et al., 2018; Weinstein, 2018). This approach is still at the beginning of its development. Current
388 studies discuss factors that influence the accuracy of the analyses. Apart from purely methodological aspects, such as
389 the size of image classes needed for model training, the variability of the image material (e.g. diversity of backgrounds,
390 lighting conditions) or the object detection methods make the analyses difficult. Therefore, the targeted adaptation of
391 a system is particularly necessary for the evaluation of behaviour. Our study shows that the developed system achieves
392 a very high accuracy with a manageable amount of training data, both in the pure in-domain and the weak in-domain
393 classification. Furthermore, the results of the out-of-domain classification show that the network can be reliably ap-
394 plied to other species, which are similar in their behaviour to the species from the training.

We thereby conclude that with this line of research we have opened the door to scale up studies of behavioural biology by reducing human resources needed for manual and repetitive labeling tasks, and this way researchers have the opportunity to focus instead on the core tasks of setting up interesting experiments and interpreting distilled information. To further extend the scope of applications, a next step would be to include enclosures with multiple individuals. This sets a stronger focus on the object detection phase, where a distinction of the individuals needs to be performed. This can be a challenging task if resemblance between individuals is strong, and possibly requires a tracker post-processing single detections. On the other hand, our system could be applied to other ungulates like Perissodactyla, who have a different REM sleep posture (Pedersen et al., 2004) which requires to adjust the definition of LHD with modified post-processing rules or fine-tuned networks.

ACKNOWLEDGEMENT

The study was supported by Opel-Zoo Foundation Professorship in Zoo Biology from the *von Opel Hessische Zoostiftung*. The authors gratefully acknowledge support from the participating zoos and especially from their staff that allowed and supported the data collection (in alphabetical order): Allwetterzoo Münster, Kölner Zoo, Königlicher Burger's Zoo Arnheim, Opel-Zoo Kronberg, Zoo Dortmund, Zoo Frankfurt, Zoo Heidelberg, Zoo Krefeld, Zoo Neuwied, Zoo Osnabrück, Zoom Erlebniswelt Gelsenkirchen. We especially thank Opel-Zoo Kronberg and Zoo Neuwied for the permission to publish image material. Finally, we thank all participants of the human accuracy study.

AUTHORS' CONTRIBUTIONS

MHK and TK conceived the ideas and designed methodology; JG collected the data; MHK, TK and JG analysed the data; PD, MHK and TK led the writing of the manuscript. All authors contributed critically to the drafts and gave final approval for publication.

DATA ACCESSIBILITY

The python code is available at https://github.com/Klimroth/Video-Action-Classifier-for-African-Ungulates-in-Zoos/tree/main/mrcnn_based and is also stored at figshare 10.6084/m9.figshare.13526171.

REFERENCES

- Bochkovskiy, A., Wang, C., & Liao, H. M. (2020). YOLOv4: Optimal speed and accuracy of object detection.
- Bradski, G. (2000). The OpenCV Library. *Dr. Dobbs' Journal of Software Tools*.
- Burger, A. L., Hartig, J., & Dierkes, P. W. (2020). Shedding light into the dark: Age and light shape nocturnal activity and sleep behaviour of giraffe. *Applied Animal Behaviour Science*, 229, 105012. <https://doi.org/10.1016/j.applanim.2020.105012>

- Chakravarty, P., Cozzi, G., Dejnabadi, H., Léziart, P.-A., Manser, M., Ozgul, A., & Aminian, K. (2020). Seek and learn: Automated identification of microevents in animal behaviour using envelopes of acceleration data and machine learning. *Methods in Ecology and Evolution*. <https://doi.org/10.1111/2041-210X.13491>
- Christin, S., Hervet, É., & Lecomte, N. (2019). Applications for deep learning in ecology. *Methods in Ecology and Evolution*, 10(10), 1632–1644. <https://doi.org/10.1111/2041-210X.13256>
- Dell, A. I., Bender, J. A., Branson, K., Couzin, I. D., de Polavieja, G. G., Noldus, L., Pérez-Escudero, A., Perona, P., Straw, A. D., Wikelski, M., & Brose, U. (2014). Automated image-based tracking and its application in ecology. *Trends in Ecology & Evolution*, 29(7), 417–428. <https://doi.org/10.1016/j.tree.2014.05.004>
- Eikelboom, J. A. J., Wind, J., van de Ven, E., Kenana, L. M., Schroder, B., de Knegt, H. J., van Langevelde, F., & Prins, H. H. T. (2019). Improving the precision and accuracy of animal population estimates with aerial image object detection. *Methods in Ecology and Evolution*, 10(11), 1875–1887. <https://doi.org/10.1111/2041-210X.13277>
- Farnebäck, G. (2003). Two-frame motion estimation based on polynomial expansion, 363–370. https://doi.org/10.1007/3-540-45103-X_50
- Feichtenhofer, C., Pinz, A., & Zisserman, A. (2016). Convolutional two-stream network fusion for video action recognition, 1933–1941. <https://doi.org/10.1109/CVPR.2016.213>
- Felzenszwalb, P. F., Girshick, R. B., McAllester, D., & Ramanan, D. (2010). Object detection with discriminatively trained part-based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(9), 1627–1645. <https://doi.org/10.1109/TPAMI.2009.167>
- Ferreira, A. C., Silva, L. R., Renna, F., Brandl, H. B., Renoult, J. P., Farine, D. R., Covas, R., & Doutrelant, C. (2020). Deep learning-based methods for individual recognition in small birds (E. Codling, Ed.). *Methods in Ecology and Evolution*, 11(9), 1072–1085. <https://doi.org/10.1111/2041-210x.13436>
- Graving, J. M., Chae, D., Naik, H., Li, L., Koger, B., Costelloe, B. R., & Couzin, I. (2019). Deepposekit, a software toolkit for fast and robust animal pose estimation using deep learning. *eLife*, 8, e47994. <https://doi.org/10.7554/eLife.47994>
- Ji, S., Xu, W., Yang, M., & Yu, K. (2013). 3d convolutional neural networks for human action recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(1), 221–231. <https://doi.org/10.1109/TPAMI.2012.59>
- Kabra, M., Robie, A., Rivera-Alba, M., Branson, S., & Branson, K. (2012). Jaaba: Interactive machine learning for automatic annotation of animal behavior. *Nature methods*, 10. <https://doi.org/10.1038/nmeth.2281>
- Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R., & Fei-Fei, L. (2014). Large-scale video classification with convolutional neural networks. *2014 IEEE Conference on Computer Vision and Pattern Recognition*, 1725–1732. <https://doi.org/10.1109/CVPR.2014.223>
- Kingma, D., & Ba, J. (2014). Adam: A method for stochastic optimization. *International Conference on Learning Representations*.

- Kögler, J., Barbosa Pacheco, I., & Dierkes, P. W. (2020). Evaluating the quantitative and qualitative contribution of zoos and aquaria to peer-reviewed science. *Journal of Zoo and Aquarium Research*, 8(2), 124–132. <https://doi.org/10.19227/jzar.v8i2.471>
- Li, C., Zhong, Q., Xie, D., & Pu, S. (2019). Collaborative spatiotemporal feature learning for video action recognition. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. <https://doi.org/10.1109/cvpr.2019.00806>
- Lima, S. L., Rattenborg, N. C., Lesku, J. A., & Amlaner, C. J. (2005). Sleeping under the risk of predation. *Animal Behaviour*, 70(4), 723–736. <https://doi.org/10.1016/j.anbehav.2005.01.008>
- Lin, T., Maire, M., Belongie, S., Bourdev, L., Girshick, R., Hays, J., Perona, P., Ramanan, D., Zitnick, C., & Dollár, P. (2014). Microsoft coco: Common objects in context. <http://arxiv.org/abs/1405.0312>
- Melzheimer, J., Heinrich, S. K., Wasiolka, B., Mueller, R., Thalwitzer, S., Palmegiani, I., Weigold, A., Portas, R., Roeder, R., Krofel, M., Hofer, H., & Wachter, B. (2020). Communication hubs of an asocial cat are the source of a human–carnivore conflict and key to its solution. *Proceedings of the National Academy of Sciences*, 202002487. <https://doi.org/10.1073/pnas.2002487117>
- Ng, J. Y., Hausknecht, M., Vijayanarasimhan, S., Vinyals, O., Monga, R., & Toderici, G. (2015). Beyond short snippets: Deep networks for video classification. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 4694–4702. <https://doi.org/10.1109/CVPR.2015.7299101>
- Norouzzadeh, M. S., Nguyen, A., Kosmala, M., Swanson, A., Palmer, M. S., Packer, C., & Clune, J. (2018). Automatically identifying, counting, and describing wild animals in camera-trap images with deep learning. *Proceedings of the National Academy of Sciences*, 115(25), E5716–E5725. <https://doi.org/10.1073/pnas.1719367115>
- Pedersen, G. R., Søndergaard, E., & Ladewig, J. (2004). The influence of bedding on the time horses spend recumbent. *Journal of Equine Veterinary Science*, 24(4), 153–158. <https://doi.org/10.1016/j.jevs.2004.03.013>
- Porto, S., Arcidiacono, C., Anguzza, U., & Cascone, G. (2013). A computer vision-based system for the automatic detection of lying behaviour of dairy cows in free-stall barns. *Biosystems Engineering*, 115, 184–194. <https://doi.org/10.1016/j.biosystemseng.2013.03.002>
- Quionero-Candela, J., Sugiyama, M., Schwaighofer, A., & Lawrence, N. D. (2009). *Dataset shift in machine learning*. The MIT Press. <https://doi.org/10.5555/1462129>
- Rast, W., Kimmig, S. E., Giese, L., & Berger, A. (2020). Machine learning goes wild: Using data from captive individuals to infer wildlife behaviours (C. M. Danforth, Ed.). *PLOS ONE*, 15(5), e0227317. <https://doi.org/10.1371/journal.pone.0227317>
- Recht, B., Roelofs, R., Schmidt, L., & Shankar, V. (2019). Do ImageNet classifiers generalize to ImageNet? (K. Chaudhuri & R. Salakhutdinov, Eds.), 97, 5389–5400. <http://proceedings.mlr.press/v97/recht19a.html>
- Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., & Fei-Fei, L. (2015). ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3), 211–252. <https://doi.org/10.1007/s11263-015-0816-y>

- 493 Ryder, O. A., & Feistner, A. T. C. (1995). Research in zoos: A growth area in conservation. *Biodiversity and Conservation*,
494 4(6), 671–677. <https://doi.org/10.1007/bf00222522>
- 495 Simonyan, K., & Zisserman, A. (2014). Two-stream convolutional networks for action recognition in videos. *Proceedings*
496 *of the 27th International Conference on Neural Information Processing Systems - Volume 1*, 568–576. <https://doi.org/10.5555/2968826.2968890>
- 497
- 498 Stanton, L. A., Sullivan, M. S., & Fazio, J. M. (2015). A standardized ethogram for the felidae: A tool for behavioral re-
499 searchers. *Applied Animal Behaviour Science*, 173, 3–16. <https://doi.org/10.1016/j.applanim.2015.04.001>
- 500 Stern, U., He, R., & Yang, C. (2015). Analyzing animal behavior via classifying each video frame using convolutional
501 neural networks. *Scientific Reports*, 5(1). <https://doi.org/10.1038/srep14351>
- 502 Swanson, A., Kosmala, M., Lintott, C., Simpson, R., Smith, A., & Packer, C. (2015). Snapshot serengeti, high-frequency
503 annotated camera trap images of 40 mammalian species in an african savanna. Dryad Digital Repository.
504 <https://doi.org/10.5061/dryad.5pt92>
- 505 Tabak, M. A., Norouzzadeh, M. S., Wolfson, D. W., Sweeney, S. J., Vercauteren, K. C., Snow, N. P., Halseth, J. M., Salvo,
506 P. A. D., Lewis, J. S., White, M. D., Teton, B., Beasley, J. C., Schlichting, P. E., Boughton, R. K., Wight, B., Newkirk,
507 E. S., Ivan, J. S., Odell, E. A., Brook, R. K., ... Miller, R. S. (2018). Machine learning to classify animal species in
508 camera trap images: Applications in ecology (T. Photopoulou, Ed.). *Methods in Ecology and Evolution*, 10(4),
509 585–590. <https://doi.org/10.1111/2041-210x.13120>
- 510 Tan, M., & Le, Q. (2019). EfficientNet: Rethinking model scaling for convolutional neural networks. *Proceedings of Ma-*
511 *chine Learning Research*, 97, 6105–6114.
- 512 Tan, M., Pang, R., & Le, Q. V. (2020). Efficientdet: Scalable and efficient object detection, 10778–10787. <https://doi.org/10.1109/CVPR42600.2020.01079>
- 513
- 514 Teitelbaum, C. S., Fagan, W. F., Fleming, C. H., Dressler, G., Calabrese, J. M., Leimgruber, P., & Mueller, T. (2015). How far
515 to go? determinants of migration distance in land mammals. *Ecology Letters*, 18(6), [https://onlinelibrary.wiley.com/doi/pdf/](https://onlinelibrary.wiley.com/doi/pdf/10.1111/ele.12435)
516 545–552. <https://doi.org/10.1111/ele.12435>
- 517 Ternman, E., Pastell, M., Agenäs, S., Strasser, C., Winckler, C., Nielsen, P., & Hänninen, L. (2014). Agreement between
518 different sleep states and behaviour indicators in dairy cows. *Applied Animal Behaviour Science*, 160. <https://doi.org/10.1016/j.applanim.2014.08.014>
- 519
- 520 Valletta, J. J., Torney, C., Kings, M., Thornton, A., & Madden, J. (2017). Applications of machine learning in animal
521 behaviour studies. *Animal Behaviour*, 124, 203–220. <https://doi.org/10.1016/j.anbehav.2016.12.005>
- 522 Waleed, A. (2017). Mask r-cnn for object detection and instance segmentation on keras and tensorflow. Github.
- 523 Weinstein, B. G. (2018). Scene-specific convolutional neural networks for video-based biodiversity detection (W. Pearse,
524 Ed.). *Methods in Ecology and Evolution*, 9(6), 1435–1441. <https://doi.org/10.1111/2041-210x.13011>
- 525 Weygandt, M., & Mathis, A. (2020). Deep learning tools for the measurement of animal behavior in neuroscience. *Cur-*
526 *rent Opinion in Neurobiology*, 60, 1–11. <https://doi.org/10.1016/j.conb.2019.10.008>
- 527 Xie, Q., Hovy, E., Luong, M., & Le, Q. (2019). Self-training with noisy student improves imagenet classification.

- 528 Yakubovskiy, P. (2019). Efficientnet. Github.
- 529 Zepelin, H., Siegel, J., & Tobler, I. (2005). Mammalian sleep. *Principles and Practice of Sleep Medicine*, 91–100. <https://doi.org/10.1007/s00114-005-0618-0>
- 530
- 531 Zha, S., Luisier, F., Andrews, W., Srivastava, N., & Salakhutdinov, R. (2015). Exploiting image-trained CNN architectures
532 for unconstrained video classification. *Proceedings of the British Machine Vision Conference 2015*. <https://doi.org/10.5244/c.29.60>
- 533
- 534 Zhao, Y., Man, K. L., Smith, J., Siddique, K., & Guan, S. (2020). Improved two-stream model for human action recogni-
535 tion. *EURASIP Journal on Image and Video Processing*, 2020(1). <https://doi.org/10.1186/s13640-020-00501-x>

APPENDIX A. DATA QUALITY

This section contains examples of images frames that are challenging to the deep learning pipeline. I.e., Fig. 5 shows a blind spot in the enclosure of Eland 1 which is compensated by the post-processor – if the Eland is staying below the red line for at least 70 seconds, the behavioural state is assumed to be lying.

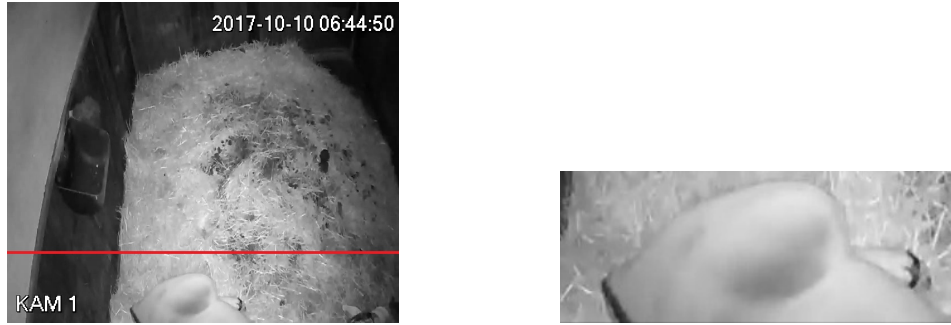


FIGURE 5. Example of an event of high truncation. The image is likely to be misclassified as standing without post-processing. The l.h.s. shows the recorded image and on the r.h.s. the result of the object detection phase is shown.

Different hard examples are given in Fig. 6. Even if the object detector predicted the bounding box accurately, the high amount of truncation resulting from a poor installment of the camera (due to a lack of better installment options) makes the images challenging to classify.



FIGURE 6. Example of hard to classify images due to the camera's position.

APPENDIX B. ON THE USE OF OPTICAL FLOW AS SECOND STREAM INPUT

The current state-of-the-art approach towards video action classification would use optical flow calculations in stream 2 of the system to explicitly input motion cues to the classifier. In the following we report our results when applying this approach to the setting at hand. For calculating the optical flow we used OpenCV's implementation of the Farneback-algorithm for dense optical flow (Bradski, 2000; Farneback, 2003) with different types of parameter settings, i.e. using various window-sizes (blurring vs. robustness) and Gaussian filters. The classification task was

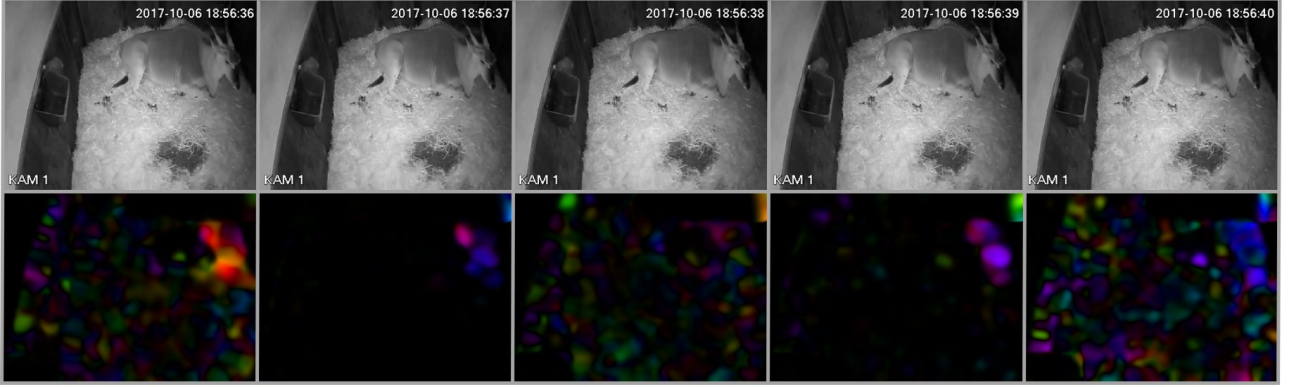


FIGURE 7. Optical flow of a lying eland of five consecutive frames. While the spatial dimension does not change notably, the optical flow is very sensitive to the high amount of background noise.

governed by a ResNet-101 CNN trained on the same training set as the system at hand. The validation accuracy only reached 0.57 and even the training accuracy did not pass 0.84.

At first glance, it is surprising that the optical flow stream was outperformed significantly by the multi-frame encoded setting as the different postures we try to classify clearly deviate in the amount of motion the individual is showing. However, we found that in the large temporal difference of one second between two consecutive frames, background motion such as floating dust, hay or straw, crossing insects and brightness changes due to infrared-emitters, lead to plenty of spurious motion cues. See for example Fig. 7 which shows the optical flow of five consecutive frames. As a result, the training signal stemming from the optical flow tended to be very brittle, which we think is the reason for the bad performance, especially the bad validation performance. One might be able to improve on this by outlier rejection and other pre-processing steps, but such a tuning likely leads to a strong bias towards specifics of environmental variables like the enclosure and the camera etc., hence might generalize poorly to nights of new individuals. Therefore, we choose to continue with the multi-frame encoding as second stream instead which proved to be more flexible and robust.

APPENDIX C. MODEL AVERAGING

This section gives the implementation details of the model averaging described in Fig. 2.

C.1. Single frame classification. After the object detection phase, we are left with up to four images per 7s time-interval. Each of those images p is predicted by the first EfficientNet B3 yielding a distribution $x_p = (x_{p,0}, x_{p,1}, x_{p,2})$, where $x_{p,0}$ is the probability that the animal is standing in the image, $x_{p,1}$ that it shows LHU and $x_{p,2}$ that it shows LHD. Let $x'_p = (x_{p,0}, x_{p,2}, x_{p,3}, 0)$ be the adjusted distribution such that the probability of being absent is set to zero. If in frame i the animal is detected in phase 1, we generate x'_i as described, otherwise we set $x'_i = (0, 0, 0, 1)$.

Next, a rolling average of order 16 is applied to x' which covers local temporal dependencies. Formally, the rolling average of order k generates a sequence of distributions \tilde{x} such that $\tilde{x}_i = (\tilde{x}_{i,0}, \tilde{x}_{i,1}, \tilde{x}_{i,2}, \tilde{x}_{i,3})$ is given by

$$\tilde{x}_{i,j} \propto \sum_{\ell=\max\{0,i-k\}}^{i-1} \tilde{x}_{\ell,j} + x'_{i,j},$$

where \propto stands for being proportional up to normalising \tilde{x}_i back to a probability distribution. Finally, the prediction for time interval j is given as the average over the predictions on its contained images. Thus, if interval j consists of frames $i, i + 2, i + 4, i + 6$, we set

$$y_j = \frac{\tilde{x}_i + \tilde{x}_{i+2} + \tilde{x}_{i+4} + \tilde{x}_{i+6}}{4}.$$

Now, stream 1 outputs a sequence y_1, \dots, y_m such that y_j describes predicted probabilities for each behaviour in time-interval j .

C.2. Four frame classification. For the second stream a second EfficientNet B3 produces a distribution

$$\omega'_j = (\omega_{j,0}, \omega_{j,1}, \omega_{j,2}, 0)$$

per time-interval j by predicting behaviours on four-frame encoded input images, and $\omega'_j = (0, 0, 0, 1)$ if and only if the animal is not detected during phase 1 on any of the four images. As above, we then apply a rolling average, but now of order 4 such that in total it accounts for a similar time-period as the rolling average in stream 1, processing $\omega' = \omega'_1, \dots, \omega'_m$ to the stream's outcome $y' = y'_1, \dots, y'_m$.

C.3. Post-processing details. Beside the post-processing rules listed in Table 3, enclosure specific settings were incorporated into the post-processor of Eland 1 (cf. Section 3.1). As the installed camera left a blind spot, the animal can be highly truncated (see Fig. 5). As one would expect, the corresponding images were prone to misclassification. As the object detection phase gives access to the coordinates of the drawn bounding box, it is possible to mark any frame in which the bounding box starts below a certain line (sketched as a red line in Fig. 5), as truncated. Now, if a sequence of truncation is shorter than 10 time-intervals, the sequence was labelled as the previously shown behaviour. Otherwise, the assigned label was set to be LHU, as it is very unlikely due to the enclosure's design that the animal was standing in the blind spot for a longer period of time.

APPENDIX D. FURTHER EVALUATION RESULTS

Finally, we report the statistical key quantities of interest in the binary classification task in Table 7.

	standing		lying	
	real	prediction	real	prediction
Pure in-domain classification				
Eland 1				
Avg. # phases	8.2 ± 0.3	8.3 ± 0.4	8.2 ± 0.3	8.3 ± 0.4
Avg. duration [min]	195.2 ± 9.6	197.8 ± 8.8	640.9 ± 8.9	641.1 ± 8.6
Eland 2				
Avg. # phases	6.0 ± 0.0	6.0 ± 0.0	7.0 ± 0.0	6.5 ± 0.4
Avg. duration [min]	249.2 ± 0.5	251.1 ± 0.1	590.9 ± 0.5	588.9 ± 0.1
Eland 3				
Avg. # phases	7.0 ± 0.0	7.5 ± 0.4	7.5 ± 0.4	8.5 ± 0.4
Avg. duration [min]	222.5 ± 2.2	238.6 ± 8.6	617.6 ± 2.2	595.6 ± 12.7
Weak in-domain classification				
Eland 1				
Avg. # phases	7.0 ± 0.3	7.5 ± 0.3	6.9 ± 0.3	7.5 ± 0.3
Avg. duration [min]	211.9 ± 10.1	220.8 ± 12.0	622.1 ± 11.6	614.3 ± 13.0
Eland 2				
Avg. # phases	6.5 ± 0.4	8.5 ± 1.8	7.0 ± 0.0	9.0 ± 1.4
Avg. duration [min]	216.4 ± 10.8	233.0 ± 0.4	623.2 ± 11.2	607.0 ± 0.4
Eland 3				
Avg. # phases	7.0 ± 0.7	6.5 ± 0.4	8.0 ± 0.7	7.5 ± 0.4
Avg. duration [min]	234.3 ± 6.9	233.8 ± 8.1	605.2 ± 7.3	606.2 ± 8.1
Out-of-domain classification				
Bongo				
Avg. # phases	10.5 ± 1.1	7.0 ± 0.0	11.5 ± 1.1	8.0 ± 0.0
Avg. duration [min]	105.0 ± 42.7	99.3 ± 44.5	735.1 ± 42.7	740.7 ± 44.5
Wildebeest				
Avg. # phases	8.5 ± 0.4	9.0 ± 0.7	9.0 ± 0.0	9.5 ± 0.4
Avg. duration [min]	114.2 ± 14.7	118.2 ± 16.4	725.9 ± 14.7	721.8 ± 16.4

TABLE 7. Overview on the accuracy of the deep learning system predicting the amount of phases as well as the average duration of each behavioural state in the binary classification task.