ASBAR: an Animal Skeleton-Based Action Recognition framework. Recognizing great ape behaviors in the wild using pose estimation with domain adaptation

Michael Fuchs[1*], Emilie Genty[2], Klaus Zuberbühler[2], Paul Cotofrei[1]

1 Information Management Institute, University of Neuchâtel, Neuchâtel, Switzerland
2 Institute of Biology, University of Neuchâtel, Neuchâtel, Switzerland

* corresponding author: michael.fuchs@unine.ch (MF)

Abstract

To date, the investigation and classification of animal behaviors have mostly relied on direct human observations or video recordings with posthoc analysis, which can be labor-intensive, time-consuming, and prone to human bias. Recent advances in machine learning for computer vision tasks, such as pose estimation and action recognition, thus have the potential to significantly improve and deepen our understanding of animal behavior. However, despite the increased availability of open-source toolboxes and large-scale datasets for animal pose estimation, their practical relevance for behavior recognition remains under-explored. In this paper, we propose an innovative framework, *ASBAR*, for *Animal Skeleton-Based Action Recognition*, which fully integrates animal pose estimation and behavior recognition. We demonstrate the use of this framework in a particularly challenging task: the classification of great ape natural behaviors in the wild. First, we built a robust pose estimator model leveraging OpenMonkeyChallenge, one of the largest available open-source primate pose datasets, through a benchmark analysis on several CNN models from DeepLabCut, integrated into our framework. Second, we extracted the great ape's skeletal motion from the PanAf dataset, a large collection of in-the-wild videos of gorillas and chimpanzees annotated for natural behaviors, which we used to train and evaluate PoseConv3D from MMaction2, a second deep learning model fully integrated into our framework. We hereby classify behaviors into nine distinct categories and achieve a Top 1 accuracy of 74.98%, comparable to previous studies using video-based methods, while reducing the model's input size by a factor of around 20. Additionally, we provide an open-source terminal-based GUI that integrates our full pipeline and release a set of 5,440 keypoint annotations to facilitate the replication of our results on other species and/or behaviors. All models, code, and data can be accessed at: https://github.com/MitchFuchs/asbar.

Author summary

The study of animal behaviors has mostly relied on human observations and/or video analysis traditionally. In this paper, we introduce a new framework called *ASBAR* (for *Animal Skeleton-Based Action Recognition*) that integrates recent advances in machine learning to classify animal behaviors from videos. Compared to other methods that use the entire video information, our approach relies on the detection of the animal's pose (e.g., position of the head, eyes, limbs) from which the behavior can be recognized. We demonstrate its successful application in a challenging task for computers as it classifies

nine great ape behaviors in their natural habitat with high accuracy. To facilitate its use for other researchers, we provide a graphical user interface (GUI) and annotated data to replicate our results for other animal species and/or behaviors.

Introduction

Direct observation and manual annotation of animal behaviors are labor-intensive, time-consuming, prone to human error [67], and can present limitations such as information loss, especially in natural low-visibility settings, during complex, fast-paced social interactions involving multiple partners. The use of video recording and post-hoc annotation of video data has thus been considered methods of choice to study animal behavior. They allow the identification and interpretation of animal behaviors in fine-grained details, and facilitate the testing of reliability and replication of the coding. Animal behavior research would nevertheless greatly benefit from easing the burden of intensive and lengthy manual video annotation thanks to the help of automated animal behavior recognition systems. New machine learning tools could thus allow to identify relevant video sections containing social interactions and classify behaviour categories automatically. The development of such tools could furthermore significantly enlarge the scope and robustness of observational studies and therefore deepen our understanding of animal behaviors [1].

Recent advances in machine learning computer vision techniques can offer innovative ways to build such systems, as some of their models can be trained to recognize or detect behaviors recorded on videos. Specifically, models from the task known as *action recognition* can learn deep representations of the visual features in a video and classify the recognized action into the appropriate behavior category.

In the context of deep learning, two primary approaches have emerged within the action recognition task: video-based methods and skeleton-based methods.

Video-based action recognition involves analyzing raw RGB video data and identifying spatio-temporal patterns to recognize actions in video clips. This field of research typically relies on the use of Convolutional Neural Networks (CNNs) ([25]) and their adaptation to data extending over the temporal domain. Typical state-of-the-art video-based methods for human action recognition include models such as Two-Stream CNNs [50], C3D [57], I3D [7], (2+1)D ResNet [56], and SlowFast [13]. Similar methods have been subsequently adapted to classify animal behaviors from videos [52, 31, 48, 16, 4, 34] and extended to multimodal audio-visual data types [2].

On the other hand, skeleton-based methods predict the action class from the body's skeletal structure and its motion [15, 11]. They thus rely on an additional preprocessing step called *pose estimation*, in which the body parts, such as joints and bones, are detected and their coordinates extracted from each video frame [6]. Despite the additional workload that pose estimation represents, a skeleton-based approach offers numerous advantages for the advancement of behavioral analysis over other video-based methods for computational ethology [1, 67, 20]. Firstly, by focusing on the skeletal motion rather than the apparent animal's movements, such a model can recognize cross-subject behaviors within the same species (see e.g., [32] for humans, and [3, 53] for non-humans). Furthermore, as it is less subject to animal's physical traits, the model can be trained to recognize inter-species behaviors. Thirdly, video-based methods can be sensitive to visual setting changes [8] - such as different lighting conditions, changes in backgrounds, etc. - or even visual input changes imperceptible to the human eye ([54, 33]) which can lead to a drop in the network's ability to perform its task. In other words, a video-based model trained within a certain visual context (e.g., from cameras in a zoo) may completely fail to recognize behaviors from the same species in a different visual setting, for instance, in their natural habitat. Comparatively, skeleton-based

methods are less sensitive to these changes [18] and, given a robust pose estimator, are therefore likely to maintain a high action recognition accuracy. Furthermore, extracting the animal's pose coordinates from high-dimensional video segments drastically reduces the network's input space dimensionality, its computational complexity, and overall power consumption [15], which can represent game-changing advantages for field researchers with limited computational resources. Finally, identifying the pose offers a pre-computed geometrical quantification of the animal's body motion and behavioral changes [67, 45].

One of the major challenges for skeleton-based methods for animal behavior concerns the extraction process of the required pose-estimated data. Whereas for human pose estimation this process benefits from the availability of large-scale open source datasets and state-of-the-art keypoint detectors achieving high performance, trying to replicate the same performances for other animal species can easily become a daunting task. Thankfully, on one hand, the number of available large annotated datasets with animal pose is surging (e.g., Animal Kingdom [41], Animal Pose [5], AP-10K [65], OpenMonkeyChallenge [64], OpenApePose [10], MacaquePose [26], Horse-30 [36]). On the other hand, open-source frameworks for animal pose estimation are increasingly accessible (e.g., DeepLabCut [35, 37, 27], SLEAP [46, 44], or AniPose [23]). Despite these remarkable advances, studies leveraging pose estimation to classify animal behaviors remain under-explored, especially outside the laboratory settings. This could be primarily attributed to the scarcity of open-source datasets labeled with both keypoint coordinates and behavioral annotations. Moreover, an additional challenge lies in the tendency of researchers to investigate pose estimation and skeleton-based action recognition as two distinct subjects rather than two facets of a comprehensive approach to behavior analysis.

In practice, however, ethologists may not have the resources to manually label their behavioral data with the animals' poses and could therefore considerably benefit from robust, ready-to-use keypoint detectors that streamline the pose estimation process. In addition, some researchers may also be limited by resources to overcome the technical challenges of integrating both machine learning tasks of pose estimation and behavior recognition.

To alleviate these practical challenges, we introduce ASBAR, an innovative framework for animal skeleton-based action recognition. We demonstrate the successful application of our methodology by tackling a particularly challenging problem: classifying great ape behaviors in their natural habitat. Our contributions are fivefold:

- We developed a fully integrated data/model pipeline, comprising a first module for animal pose estimation based on the state-of-the-art DeepLabCut toolbox, and a second one for animal behavior recognition, based on the open-source toolbox MMaction2;
- We built a general and robust (to visual context shift) primate keypoint detector, by leveraging OpenMonkeyChallenge [64], one of the largest primate pose datasets with 26 primate species;
- We provide a methodology to extract the pose of great apes (chimpanzees and gorillas) from a dataset of videos representing large-scale footage from camera traps in the forest, and train a behavior classifier with 9 classes (such as walking, standing, climbing up), which achieves comparable results to video-based methods;
- We provide a dataset of 5,440 high-quality keypoint annotations from great apes in their natural habitat;
- We open-source a GUI that encapsulates the ASBAR framework to provide other researchers with easy access to our methods. This GUI is terminal-based and can

therefore be used to train DeepLabCut and MMaction2 on a remote machine (such as in cloud- or HPC-based settings) without any programming knowledge.

Materials and methods

Datasets

For the experiment concerning the classification of great ape behaviors in their natural habitat, three datasets are considered: OpenMonkeyChallenge, PanAf, and PanAf-Pose.

OpenMonkeyChallenge: OpenMonkeyChallenge (OMC) [64] is a collection of 111,529 images of 26 primate species, designed to provide a public benchmark dataset for a competitive non-human primate tracking challenge. The various sources for this dataset include images available on the web, photos from three US-National Primate Research Centers, and captures from multiview cameras at the Minnesota Zoo. All images were processed and annotated with species, bounding box, and pose information corresponding to 17 keypoints: nose, eyes, head, neck, shoulders, elbows, wrists, hip, tail, knees, and ankles. The commercial service commissioned to annotate the pose (Hive AI) was instructed to identify the best guess location in case of occluded keypoints and to specify their visibility. The dataset is split into a training (60%), validation (20%), and testing datasets (20%) (where the annotations of the latter have not been made public, but preserved for the competition). In our experiment, both training and validation datasets were merged to obtain a *pose* dataset including 89,223 images (see examples in Fig 1).

Fig 1. Image examples from OpenMonkeyChallenge (*pose* dataset). A large collection of primate images annotated with pose. The dataset was primarily designed for an open benchmarking competition and includes a total of more than 100,000 images of primates from 26 species.

PanAf: The Pan African Programme "The Cultured Chimpanzee" [12], whose mission is to enhance the comprehension of the evolutionary-ecological factors that shape chimpanzee behavior diversity, has accumulated thousands of hours of footage

from camera traps stationed in the forests of Central Africa. A collection of 500 videos of chimpanzees or gorillas, each with a duration of 15 seconds (180,000 frames @ 24 FPS), were annotated with bounding boxes coordinates for ape detection [63, 62] and behaviors for action recognition [48] (see examples in Fig 2). The nine labeled behaviors include: walking, standing, sitting, running, hanging, climbing up, climbing down, sitting on back, and camera interaction.

Fig 2. Image examples from PanAf (*behavior* dataset). Videos of gorillas and chimpanzees are captured in the forest using video camera traps. Notable visual challenges among others include the small size of certain individuals due to the camera distance, the abundant vegetation, nocturnal imaging, and changes in backgrounds.

PanAf-Pose: We annotated a set of 5,440 keypoints from 320 images extracted from the PanAf dataset. The chosen keypoints are similar to those annotated in OMC. The images were selected in a two-step process. Firstly, a shortlisting of about 4,000 images was extracted automatically from the PanAf dataset, for which two models (ResNet152 and EfficientNet-B6, see section Models benchmarking) showed a high overall prediction confidence. Secondly, from the shortlist, we manually selected 320 frames, from 32 different video clips, representing different scenes, lighting conditions, postures, sizes, and species - and avoiding consecutive frames as much as possible.

For each selected frame, we created an associated mini-clip of 34 frames (24 frames before and 10 frames after the frame to annotate) that captures the ape's motion and allows us to label occluded keypoints more precisely. We started our annotation process with a semi-automated labeling approach, i.e., using the best predictions of the ResNet152 model. Then, using DeepLabCut's napari plugin [51] for manual labeling, these predictions were refined, in a first phase, by a non-trained human (MF) and then finalized by a great ape behavior and signaling expert (EG) [17], to ensure high-quality annotations. Our annotations can be found at: https://github.com/MitchFuchs/asbar

Domain adaptation

The ASBAR framework requires two distinct datasets. The first dataset (denoted as *pose*) consists of images annotated with keypoint coordinates, used to build a pose

estimator model. The second dataset (denoted as *behavior*) comprises video clips labeled with specific behaviors. Poses are then extracted from these clips using the aforementioned pose estimator.

Under ideal conditions, the two datasets come from the same visual distribution, e.g., the images from the *pose* dataset represent a subset of the video frames from the *behavior* dataset. In practice, however, the time-consuming and high-cost process of annotating a video dataset, with both pose and behavior information, may be impractical. Ethologists may therefore opt to *shift domains* and combine pose and behavioral annotated data from two datasets coming from different visual distributions, e.g., with keypoint annotated images captured indoors and behavioral data labeled in videos recorded outdoors. We refer to the latter case as *domain adaptation*, where the *behavior* dataset is called 'out-of-domain', in opposition to 'within-domain' *pose* dataset. As our experiments suggest, large enough open-source pose datasets can be leveraged to create pose extractors robust to domain shifts and sufficient for behavior recognition.

Pose estimation

The task of estimating the coordinates of anatomical keypoints, denoted as *pose estimation*, is a crucial prerequisite for skeleton-based action recognition. In most cases, each RGB frame of a video clip of an action is preprocessed to extract the set of (x, y) coordinates in the image plane of each keypoint and its relative confidence c. In practice, this transformation is often performed via supervised CNN-based machine learning models trained for keypoint detection (Fig 3).

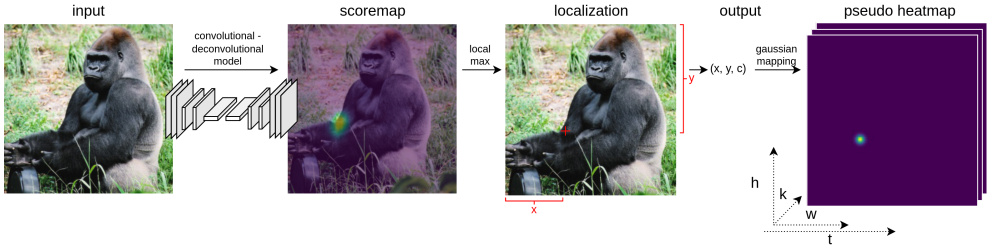

Fig 3. From RGB image to pseudo-heatmaps. The transformation of an RGB image into a 3D heatmap volume. An input image is passed through a Conv-Deconv architecture to output a probabilistic scoremap of the keypoint location (e.g., the right elbow). By finding a local maximum in the scoremap, the location coordinates and confidence can be extracted. Using a Gaussian transformation, a pseudo heatmap is generated for each keypoint and used as input of the subsequent behavior recognition model.

DeeperCut: In their paper [22], the authors developed DeeperCut, a model architecture for human pose estimation, by combining an adapted version of ResNet [21] (a state-of-the-art CNN-based image classifier trained on ImageNet) to create a deep representation of the visual features of an image, followed by a series of deconvolutional layers to recover the image's original size by upsampling. Such a method, inspired by semantic segmentation (see [59]), allows the network to output a keypoint probabilistic scoremap, i.e., a pixel-wise probability that a pixel represents the location of a specific keypoint (Fig 3). Using the *argmax* transformation, the coordinates (x, y) and the confidence c of the local maxima of a scoremap can be extracted, outputting the prediction of the specific keypoint. During the training step, a target scoremap for each keypoint is calculated, in which a probability of 1 is assigned to all pixels within a given distance from the ground-truth (x, y) coordinates (i.e., the correctly labeled location of

the keypoint) and a probability of 0 for all other remaining pixels. By computing the cross-entropy loss function between the target scoremap and the predicted one, the weights and biases of the convolutional-deconvolutional model can thus be learned during training by stochastic gradient descent. This original work was further adapted and extended to the detection of other animal species' body parts in DeepLabCut.

DeepLabCut: DeepLabCut (DLC) is an open-source multi-animal markerless pose estimation, tracking, and identification framework [35, 40, 27], and one of the first toolboxes leveraging the advances in human pose estimation for application to animals. For keypoint detection, the framework allows the choice from different CNN models, including ResNet [21] and EfficientNet [55] of various depths. Due to its high qualitative detection accuracy, the availability of a large collection of models with multiple animal species, and its active community of users and developers, DLC currently sets a standard for researchers studying animal behaviors from various fields such as neuroscience, ecology, etc. (see e.g., [60, 19, 66]).

In addition, one particular aspect of the framework is its user-friendly GUI, which facilitates adoption by users less inclined to programming. However, this GUI comes with a caveat: users whose local machine does not support GPU-accelerated CNNs training - as is often the case - will likely have to learn to use DLC's terminal commands to train their models on a remote platform (such as cloud computing or HPCs). To avoid such technical challenges, our ASBAR framework offers a terminal-based GUI, which fully integrates DLC's model training, thus allowing users to remotely train their models without any programming skills (See Fig S1 in Annex "Supporting Information" for UI element examples).

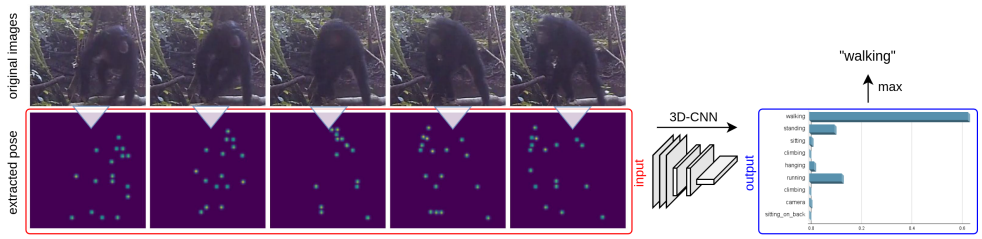

Fig 4. From extracted poses to behavior classification. From a set of consecutive RGB frames (e.g., 20 in our experiments), the animal pose is extracted, transformed into pseudo-heatmaps, and stacked as input of the behavior recognition model. A 3D-CNN is trained to classify the represented action into the correct behavior category (e.g., here 'walking')

Skeleton-based action recognition

This computer vision task involves the recognition of a specific action (performed by human or non-human individuals) from a sequence of skeletal joint data (i.e., coordinate lists), captured by sensors or extracted by markerless pose estimators.

GCN-based methods: Traditionally, skeleton-based action recognition studies [15] have focused on model architectures relying on Graph Convolutional Networks (GCNs) [24], in which the model's input consists of a graph $G = (V, E)$. For human action recognition [61], this graph consists of a set of vertices $V = \{v_{ti} | \ t = 1, \ldots, T, i = 1, \ldots, K\}$ corresponding to K keypoints over T frames, and a set of edges E partitioned into two subsets E_S and E_F, where $E_S = \{v_{ti}v_{tj}|(i,j) \in H\}$ with H being the set of anatomical body limbs and $E_F = \{v_{ti}v_{(t+1)i}|i \in H\}$ their temporal connection between frames. Each node's features vector includes its

coordinates (x, y) and the estimation confidence c. After defining the spatial temporal node's neighborhood, this data structure can be passed into GCNs to classify actions via supervised learning [61, 30, 29].

3DCNN-based methods: More recent model architectures, such as PoseConv3D in [11], have demonstrated superior performance when applying 3D-CNNs to pose estimated data rather than GCNs. Particularly in the context of animal behavior recognition, this approach is more suitable, as it significantly outperforms previous GCN-based architectures in differentiating between subtly different actions (such as in the case of FineGym [49]) and is more robust with noisy pose data. Furthermore, PoseConv3D can deal with multi-individual settings without additional computation expense (where GCN techniques see their number of trainable parameters and FLOPs increase linearly with each additional individual), generalizes better in cross-dataset setting, and can easily integrate dual-modality of pose and RGB data.

In comparison with GCN approaches, this type of architecture uses pose data to create 3D heatmap volumes instead of graphs. From a set of pose coordinates (x_k, y_k, c_k) corresponding to the (x, y) coordinates and c confidence of the k-th keypoint in a frame of size $H \times W$, a heatmap J can be generated by applying the following Gaussian transformation:

$$J_{kij} = e^{-\frac{(i-x_k)^2+(j-y_k)^2}{2\sigma^2}} \cdot c_k \quad (1)$$

where i, j refer to the pixel frame coordinates and σ is the variance of the Gaussian map (Fig 3). For each frame, a total of K heatmaps are produced. After transforming all T frames from the sample (i.e., video clip), all generated heatmaps are stacked in a 3D volume of size $K \times T \times H \times W$. This data can then be used to train an adapted video-based action recognition 3D-CNN model such as [57, 7, 13] in a supervised manner using stochastic gradient descent (Fig 4).

MMAction2: As a member of the OpenMMLab family [43], MMAction2 [38] is an open-source toolkit that supports different types of models for the task of action detection, localization, and recognition in video datasets. The toolkit implements state-of-the-art models for *action recognition, skeleton-based action recognition, spatio-temporal action detection*, and *temporal action localization*. MMAction2 has a modular design, allowing users to define requested modules, and provides several analysis tools, such as visualizers or model evaluators.

The ASBAR framework

ASBAR is a framework designed as an integrated data/model pipeline (marked in red in Fig 5), composed of two sequential modules corresponding to the two machine learning tasks of *pose estimation* and *action recognition*. The first module for animal pose estimation (marked in green in Fig 5) is developed based on the DeepLabCut toolbox. Specifically, it integrates DeepLabCut's methods for the creation of projects and multi-animal training datasets, model selection, training and evaluation, configuration editing, and video analysis. The second module for behavior recognition (marked in blue in Fig 5) is developed using MMAction2 and integrates its APIs for distributed training and testing.

Module Pose estimation

The goal of the first module is to extract pose information from the *behavior* dataset (see section **Domain Adaptation**) using a robust pose estimator model. This objective can be achieved by training and evaluating a keypoint detection model on the *pose* dataset and using it to predict the joint coordinates of the *behavior* dataset. The

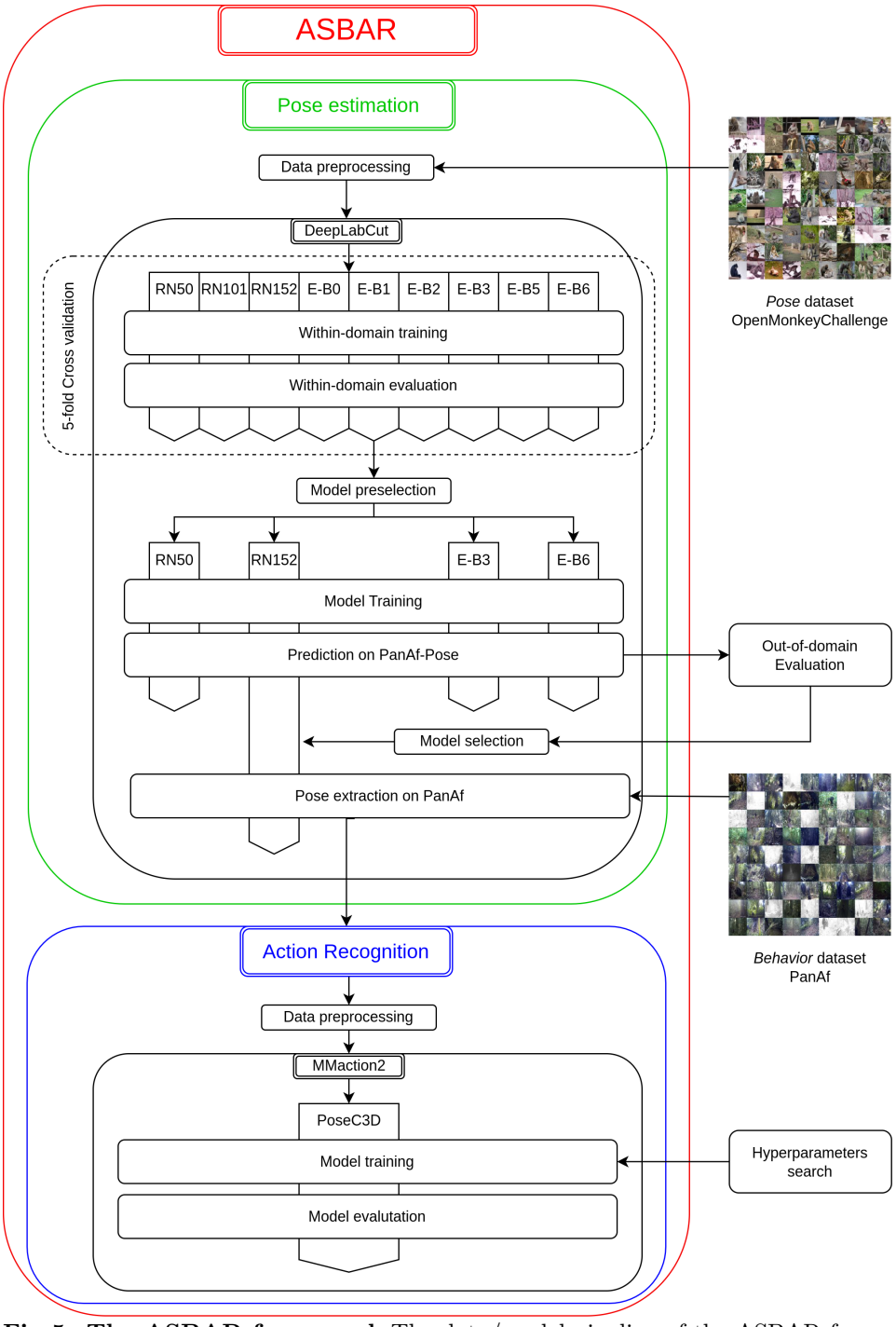

Fig 5. The ASBAR framework The data/model pipeline of the ASBAR framework (red). The framework includes two modules - the first *pose estimation* (green), which is based on the DeepLabCut toolbox, and the second *action recognition* (blue), integrating APIs from MMAction2.

functionalities of this module include: *data preprocessing, model benchmarking, model selection*, and *pose extraction*.

Data preprocessing: The framework offers four preprocessing steps: data formatting, data selection, dataset splitting for cross-validation, and configuration setup. Data formatting ensures that the information from the pose dataset meets the structural requirements of DLC. The data selection functionality allows users to select or deselect any of the species present in the pose dataset, any of the keypoints annotated, and to include/exclude invisible keypoints from their selection. For example, one can decide to create a dataset with only two species (e.g., chimpanzees and bonobos) containing only the annotations of three visible keypoints: the eyes and the nose. The third preprocessing step defines the data split for cross-validation and lets users choose between no cross-validation, 5-fold cross-validation, and 10-fold cross-validation for model assessment. Finally, users can choose to modify DLC configuration files, including the training hyperparameters.

Model benchmarking: Higher pose prediction performance is likely to positively affect the accuracy of the behavior recognition phase. Therefore, one may want to benchmark different pose estimation models by assessing their relative performance. Users can thus train and evaluate several model variations with different backbones (ResNet or EfficientNet) and various layer depths (ResNet50, ResNet101, EfficientNet-B0, etc.). To statistically validate the model's accuracy, the framework integrates an optional k-fold cross-validation procedure.

Model selection: In the case of a *behavior* dataset being 'within-domain' (see section **Domain Adaptation**), this assessment is sufficient for selecting the best model during the model benchmarking phase and proceeding with pose extraction. Conversely, if the *behavior* dataset is 'out-of-domain', as is the case in our experiments, an additional model selection step may be required, since a good 'within-domain' model may not be robust against visual domain shift. In the literature, models with EfficientNet backbones have shown higher generalization capacity in domain adaptation than those with ResNet [36]. To assess a model 'out-of-domain' performance, one can, for instance, compare its predictions to a set of manually labeled video frames from the behavior datasets. For more details, see our Results section.

Pose extraction: The pose information can be extracted from the *behavior* dataset by the selected model. Users can choose a particular snapshot of the trained model or let the framework pick the one with the lowest test set error.

Module Action recognition

The goal of the second module is to classify the behaviors from the *behavior* dataset, using as input the extracted pose generated at the end of the first module. The functionalities of this module include *data preprocessing, model training*, and *model evaluation*.

Data preprocessing: To allow behavior recognition from pose predictions, the framework implements four preprocessing steps: predictions filtering, data sampling, data formatting, and configuration setup. The prediction filtering step ensures that at most one prediction provided by the pose estimation model, for each frame and for each keypoint, is kept and the rest are filtered out. More specifically, from the set of all predicted coordinates that fall within the labeled bounding box, the pair of coordinates with the highest prediction confidence for each keypoint is kept, while all others are discarded. The data sampling step extracts the set of items (where an item represents a sequence of consecutive frames satisfying particular constraints of time and behavior label) from the behavior dataset. In the next step, the skeleton data is formatted to ensure compatibility with PoseConv3D's input data structure. Finally, users can choose to modify the PoseConv3D configuration file, including its hyperparameter values.

Model training: Users have the possibility to train several architectural variations of PoseC3D and RGBPose-C3D available in the MMAction2 toolbox [11, 38]. These can

be based on different 3D-CNN backbones (SlowOnly [14], C3D [57], or X3D [13]) with an I3D classification head on top [7]. Model training can be distributed across multi-GPU settings to reduce computation time. The optimal values of hyperparameters may be obtained using random or grid search with a hold-out approach.

Model evaluation: The model is a probabilistic classification model, which returns, for each sample, a list of potential behavior candidates, listed in descending order by the confidence probability. The final predicted behavior corresponds to the first candidate and is used to calculate the Top1 accuracy measure, i.e., the percentage of samples whose predicted behavior matches the ground-truth label. Meanwhile, the list of candidates can be used to calculate alternative accuracy measures, such as Top-k accuracy, which is the ratio of ground-truth behaviors ranked within the first k generated candidates. Mean class accuracy is another available metric, which computes the average Top1 accuracy per behavior class.

Artificial Intelligence Tools and Technologies

GPT-4 [42] was used for data visualization as well as sporadic text and code editing.

Results

In order to fully demonstrate the capability of the ASBAR framework to fulfill the task of animal behavior recognition from pose estimation, we designed a particularly challenging experiment: the classification of great ape natural behaviors in the wild. For this, we used PanAf as the *behavior* dataset and OpenMonkeyChallenge (OMC) as the *pose* dataset. The complete methodology of this experiment and the full results are presented following the data/model pipeline of the framework.

Pose estimation

For the pose estimation module, the OMC dataset represents the 'within-domain' dataset, whereas the PanAf dataset, constructed from a different visual distribution (the natural habitat of great apes), represents the 'out-of-domain' dataset.

Data preprocessing For our experiment, we selected all data from OMC, i.e., all 26 species and all 17 keypoints including those not visible. Given the large size of OMC (approx. 90k images), we chose a 5-fold cross-validation for the model benchmarking step. Most training hyperparameters are left as default, except for batch size and number of iterations, those values were chosen constrained by the hardware resources, as described below.

Models benchmarking: We initially assess the 'within-domain' relative performance with 5-fold cross-validation of nine variations of pose estimation architectures, including three ResNet networks (ResNet-50, ResNet-101, and ResNet-152), and six EfficientNet networks (EfficientNet-B0, EfficientNet-B1, EfficientNet-B2, EfficientNet-B3, EfficientNet-B5, and EfficientNet-B6) [21, 55]. All models were pretrained on ImageNet [47] and subsequently trained on OMC data during 40,000 iterations. The number of iterations was set identical (40,000) for all models, as preliminary tests had shown to be enough for the loss convergence of the largest network (i.e., EfficientNet-B6). The batch size was set to 16, i.e., the largest number of images fitting into the GPU memory (NVIDIA A100 40GB) for EfficientNet-B6. The learning rate schedule is left as default: 1.0e-04 until iteration 7,500, then 5.0e-05 until iteration 12,000, and finally 1.0e-05 until the end. Cross-validation followed the standard 5-fold procedure where 80% of the data was used for training and the remaining 20% for testing, ensuring that all 89,223 images were part of the test set once.

Using our terminal-based GUI, all models were remotely trained on the Google Cloud platform either with a NVIDIA A100 40GB or a NVIDIA V100 16GB.

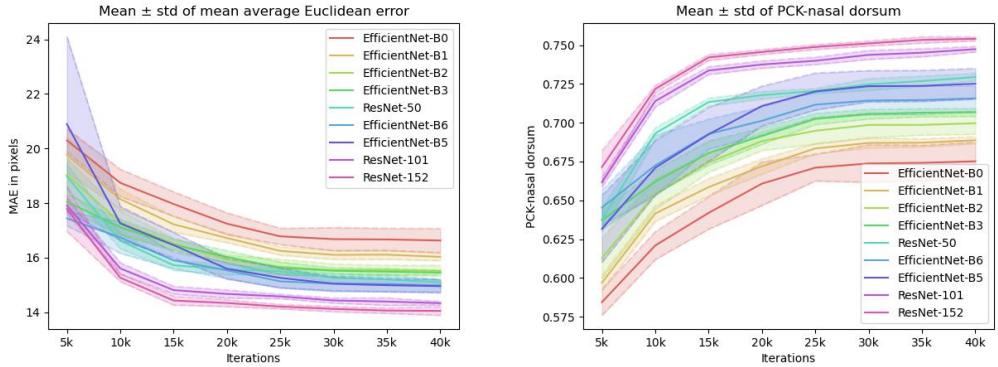

Fig 6. Model's relative performance throughout 'within-domain' training. The mean ± std of the Mean average Euclidean error (MAE) in pixels (*left*, lower is better) and percentage of correct keypoint (PCK nasal dorsum) (*right*, higher is better) for all nine model variations. Evaluation results of 5-fold cross-validation on test set data, at every 5,000 iterations.

During the training process, the snapshot model (the network's weights after i learning iterations) is saved at every 5,000 iterations and then evaluated on the test set data. Therefore, each of the eight snapshot models is evaluated five times, once for every fold of cross-validation. Due to the large size of OMC and the necessary number of evaluations for a given model ($8 * 5 = 40$) using a 5-fold cross-validation procedure, we customized the network evaluation method provided by DLC to speed up the processing time. For this, we implemented a batched data pipeline (compared to a single-image process in DLC) and restrict the evaluation only to the test set data (compared to both training set and test set evaluation, as implemented in DLC). These modifications, which altered the source code of DLC, were not integrated in the release of our framework. However, the default DLC network evaluation method is accessible within our provided GUI to allow the replication of our results and methodology. The performance metrics considered during the evaluation procedure are similar to DLC's evaluation method, i.e., the mean average Euclidean error (MAE), computed as the distance between the ground-truth labels from OMC ($\hat{x} \in \mathbb{R}^2$) and the ones predicted by the model ($x \in \mathbb{R}^2$), defined as the closest candidate to the ground-truth having a confidence of at least 0.1.

$$MAE = \frac{1}{J}\frac{1}{K}\sum_{k=1}^{K}\sum_{j=1}^{J}||\hat{x}_{jk} - x_{jk}|| \tag{2}$$

where J is the number of images (i.e., 89,223) and K is the number of keypoints (i.e., 17). See Fig S2 in Annex "Supporting Information" for prediction comparison. Additionally, we computed the percentage of correct keypoint (PCK) that falls within the distance of the nasal dorsum (PCK nasal dorsum, see Fig S3 in Annex "Supporting Information" for example), i.e., the length between the middle of the eyes and the tip of the nose, computed as

$$PCK_{\text{nasal dorsum}} = \frac{1}{J}\frac{1}{K}\sum_{k=1}^{K}\sum_{j=1}^{J}\delta(||\hat{x}_{jk} - x_{jk}|| < \epsilon) \tag{3}$$

where $\delta(\cdot)$ is a function that outputs 1 when its condition is met and 0 otherwise and ϵ is calculated in each frame as

$$\epsilon = \left|\left|\hat{x}_{\text{nose}} - \frac{1}{2}(||\hat{x}_{\text{left eye}} - \hat{x}_{\text{right eye}}||)\right|\right| \quad (4)$$

Within-domain evaluation: For each model, the deviation chart of the corresponding snapshot variants, plotting the mean and the standard deviation, are displayed in (Fig 6) (left chart for MAE metrics, right chart for PCK nasal dorsum metrics). To compare the performance of the models, we can define the 95% confidence interval for the mean of the performance metric MAE after 40,000 iterations. Due to the small number of evaluations (5) generated by the cross-validation procedure, the confidence interval is constructed using a t-distribution ($\alpha = 0.025, \nu = 4$). For the MAE metric, the corresponding confidence intervals for all nine models are displayed in (Fig 7). The results show that ResNet-152 performs statistically better (14.05 ± 0.199) than any other models, whereas ResNet-101 achieves second best results (14.334 ± 0.080). The performance of EfficientNet-B5 (14.958 ± 0.299), EfficientNet-B6 (14.981 ± 0.288) and ResNet-50 (15.098 ± 0.12) cannot be statistically distinguished; however all three perform better than EfficientNet-B3 (15.455 ± 0.097) and EfficientNet-B2 (15.519 ± 0.25). Finally, EfficientNet-B1 and EfficientNet-B0 both show higher error rate than all other models (16.031 ± 0.167 and 16.631 ± 0.546 respectively).

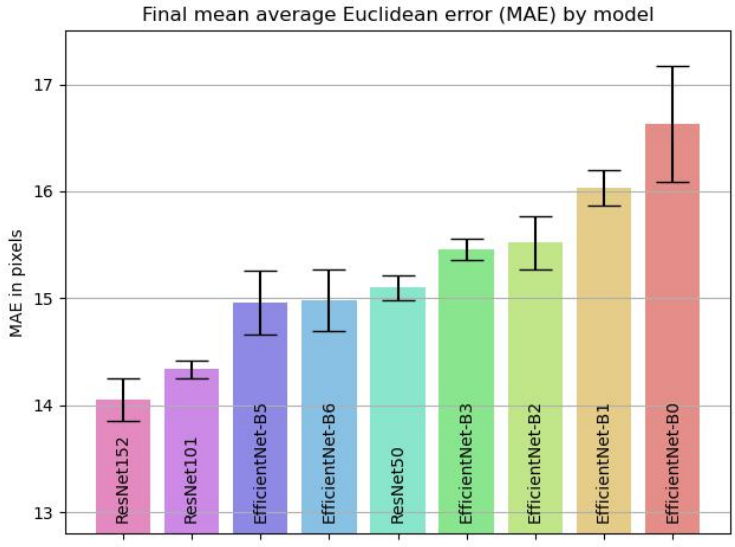

Fig 7. Final 'within-domain' model's relative performance. The mean and 95% confidence intervals of the MAE in pixels after 40,000 iterations (end of training). Disjoint confidence intervals represent significant statistical differences. ResNet152 in this task performs significantly better than any other model.

Model selection: While all models were previously trained and tested on images from the same 'in-domain' visual distribution (i. e. trained and tested on *pose* dataset OMC), their final goal is to predict keypoints on images coming from a different 'out-of-domain' dataset, the *behavior* dataset PanAf. A good 'within-domain' model does not necessarily guarantee robustness to visual domain shift. Therefore, to select the best model, we decided to pre-select four of the nine models as potential candidates for pose estimation in the context of domain adaptation. The set includes two ResNet models (ResNet-152, which performed best in the 'within-domain', and ResNet-50, due

to its wide use and popularity among researchers), and two EfficientNet models (B6 and B3). The model EfficientNet-B6 has shown best generalizing performance on 'out-of-domain' data in other experiments (see [36]), whereas EfficientNet-B3, although having a 'within-domain' accuracy lower than all three others, has a Top-1 accuracy on ImageNet [55] and the capacity to generalize on 'out-of-domain' data [36] higher than those of ResNet-152 and ResNet-50. Moreover, this model has a much lower computational cost (1.8G FLOPs compared to 4.1G FLOPs for ResNet-50 and 11G FLOPs for ResNet-152), which makes it much more portable for ethology in the field.

All four pre-selected models were trained on all 89,223 images from OMC for which annotations were made public (so no image from the OMC dataset is left out for testing purpose). During this second training phase, all models are retrained from scratch during 100,000 iterations with initial weights pre-trained on ImageNet. An increased number of training iterations reflects the larger number of images in the training dataset, while preventing underfitting. The batch size is set to 16 and learning rate schedule left again as default. During training, a model weight snapshot is saved at every 5,000 iterations for later evaluation, resulting in 20 snapshots for each of the four models.

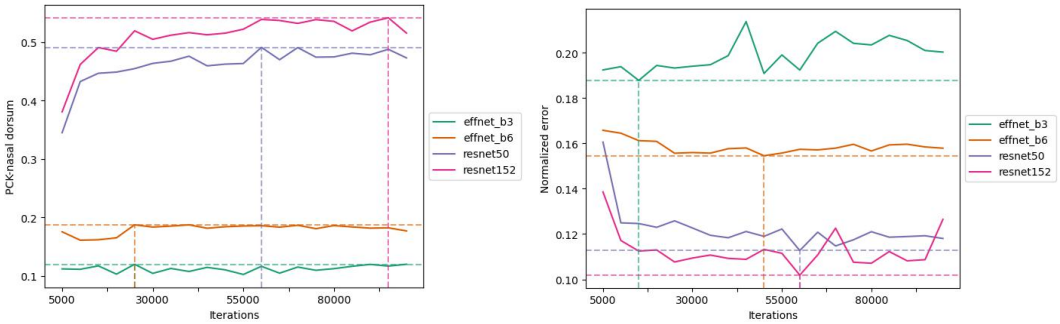

Fig 8. 'Out-of-domain' performance on PanAf-Pose. Models are compared with two metrics accounting for the animal's relative size and/or distance. PCK nasal dorsum (*left*, higher is better) and normalized error rate (*right*, lower is better) showcase the superiority of RN-152 to predict great ape poses in their natural habitat. Vertical/horizontal dashed lines represent max/min values and corresponding number of iterations. We select RN-152 at 60,000 iteration for pose extraction.

Out-of-domain evaluation: Each saved snapshot is evaluated using as a test set the PanAf-Pose dataset, a selection of 320 images from the *behavior* dataset PanAf, for which we separately labeled the ground-truth keypoint coordinates (see section PanAf-Pose). To filter out noisy model predictions, we maintained the minimum confidence threshold for pose prediction at the default value of 0.1.

To evaluate the performance, we computed two metrics, namely the normalized error rate (NMER) and PCK-nasal dorsum (see equation 3). Both metrics account for the ape's relative distance and/or size, which is particularly relevant in the case of images captured through fixed forest camera traps. The normalized error is computed as the mean raw pixel distance between the best prediction (with confidence threshold set to 0.1) and its corresponding ground truth, divided by the square root of the bounding box area [36].

$$NMER = \frac{1}{J}\frac{1}{K}\sum_{k=1}^{K}\sum_{j=1}^{J}\frac{||\hat{x}_{jk} - x_{jk}||}{\sqrt{w_j h_j}} \tag{5}$$

where w and h represent the width and height of the bounding box.

Our results suggest that ResNet-152 best generalized on 'out-of-domain' data

(Fig 8), evidenced by its highest overall PCK nasal dorsum of 54.17% across all keypoints (n=5,440) and the lowest normalized error rate standing at 10.19%.

Pose extraction: Given ResNet-152 high accuracy observed at 60,000 iterations — marked by a detection rate of 53.9% (very close to highest observed value) and the minimal normalized error (10.19%) — we decided to use this snapshot of ResNet-152 as the final keypoint detector with domain adaptation. This optimal pose estimation model is applied to predict the pose information on all 180,000 frames of PanAf. Due to the visual domain shift between OMC and PanAf, we drastically reduced the model's minimum confidence threshold for predictions from 10^{-1} to 10^{-6} in order to generate a sufficient number of keypoints candidates and avoid "no prediction" case.

Alternative performance evaluation

To reach a deeper understanding of the performance of the final pose estimation model related to the keypoint type and the primate species, we evaluated the model, using a 5-fold cross-validation, on all 89,223 images from the OMC dataset ('within-domain') and all 320 images from PanAf-Pose dataset ('out-of-domain'). The minimum confidence threshold was set to its default value.

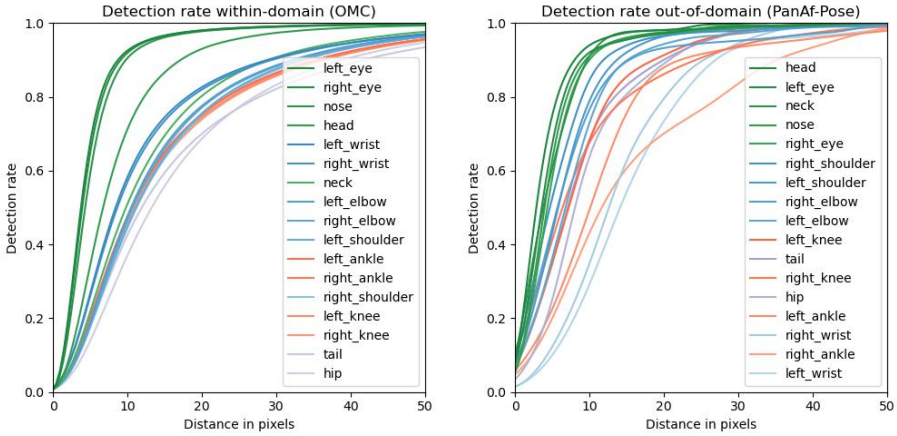

Fig 9. Keypoint detection rate on 'within-domain' vs. 'out-of-domain' test data. The keypoint detection rate at a pixel distance, i. e. the percentage of keypoints detected within a distance, is visualized for OMC (*left*) and PanAf-Pose (*right*). For instance, within a distance of 10 pixels or less, the nose is detected in around 95% of the 89,223 images of OMC. In comparison, within the same distance, the tail is only detected in around 38% of the cases.

Keypoint detection rate

Our experiments showed that, for non-human primates, not all keypoints are equal, i.e., some are easier to predict than others. For each keypoint, the detection rate at a given distance is computed, i.e., the cumulative distribution of the predicted distance in pixels (Fig 9). On the *left* chart from Fig 9, the test predictions of ResNet-152 during cross-validation on OMC are gathered ($n = 89,223 * 17 = 1,516,791$). From this, we may conclude that: (1) keypoints of facial features (nose, left eyes, right eyes) are more easily detectable than others; (2) the head's position can be predicted more accurately than any other keypoints below the neck; (3) generally upper body limbs (wrists, elbows, shoulders) are recognized more effectively than lower body limbs (ankles and knees); (4) the location of keypoints from limb extremities is predicted more accurately

than those closer to the body (wrists vs. elbows/shoulder and ankles vs. knees); (5) the position of tail and hip are the hardest to predict accurately.

These observations could partially be explained by the fact that (a) facial features can more easily be distinguished visually, that (b) the individual's head and limb extremities tend to stick out from the rest of the body, and that (c) lower limbs, hips, and tail can often be partially occluded and therefore more ambiguous to locate.

Compared to the detection rate on PanAf-Pose (*right* chart from Fig 9), we observe a similar *s*-shape distribution, demonstrating the model's robustness to domain shift. Note here that all ground truth annotations were precisely labeled by a great ape behavior and signaling expert (EG), which may result in a lower detection rate for specific keypoints as ground-truth annotations in OMC may be less accurate (e. g., in OMC, fingers are sometimes labeled instead of wrists and toes instead of ankles). Also, consider that the dataset is much smaller (320 images, i. e. 5,440 keypoints), includes only two species (gorillas and chimpanzees), and that the x-axis is the absolute distance in pixels for both charts (i. e. the image size was not taken into account).

Per species accuracy

To evaluate the performance of the model related to both dimensions, keypoints, and species, we restricted the analysis to the OMC dataset. In Fig 10, the normalized error rate (NMER, see equation 5) and 95% confidence interval are visualized for Chimpanzee (n=6,190) and Gorilla (n=1,777). We can conclude that the model's performance is statistically dependent on the species, as the error rate confidence intervals for all Gorilla keypoints are disjoint from the ones of Chimpanzees. The fact that the ranges of the confidence interval are always lower for Gorillas indicates that the model can detect the keypoint location more accurately for this species. See Fig S4 in Annex "Supporting Information" for other species grouped by family.

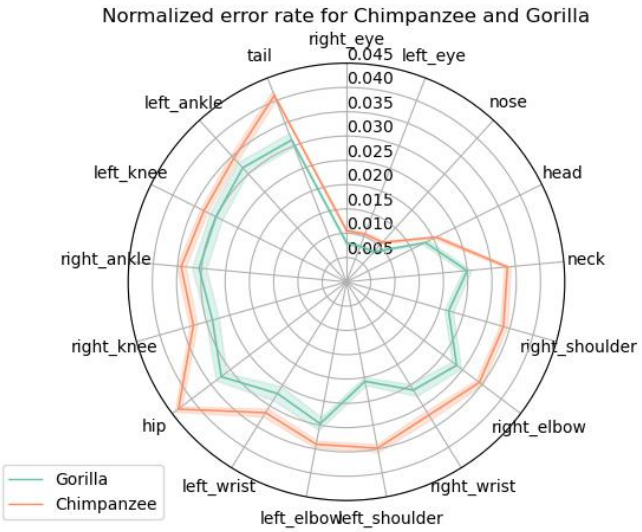

Fig 10. Normalized error rate for Chimpanzee and Gorilla The mean and 95% confidence interval for NMER. Disjoint confidence intervals suggest statistical significance. Here the model's error rate is lower for all Gorilla keypoints, i. e. those keypoints can be predicted more accurately by the model.

Behavior Recognition

Data preprocessing: For the data sampling step, we adopted the methodology proposed by [48], which involves setting a minimum threshold of 72 consecutive frames (corresponding time: 3 seconds) that exhibit the same behavior. This approach ensures that only prolonged and significant behavioral patterns are considered. The chosen video clips are then divided into samples, each one comprising 20 consecutive frames, with no gaps or overlaps between them. However, since the validation set suggested in [48] included only four out of the nine classes of behavior, we reshuffled the dataset split using a random 70-15-15 distribution of the 500 video clips. Given the low minimum confidence threshold for pose prediction (10^{-6}) used during the pose extraction step, data filtering ensures that at most 17 keypoint predictions per frame are selected, i.e. those with the highest confidence and within the given bounding box. The resulting data were formatted and stored as triplets of (x, y, confidence). The model was configured to use only joints (not limbs) with a sigma value of 0.6 (see examples in Fig 4). We did not account for the probabilistic score of the predictions (the parameter c_k in equation 1) as our minimum confidence threshold was extremely low, and detection probabilities often close to 0.

Model training: A PoseConv3D with a ResNet3dSlowOnly backbone and an I3D classification head on top was trained and tested. We selected this architecture due to its high performance on NTU60-XSub [32], a benchmark dataset for human action recognition, as reported by [11]. The search for optimal hyperparameters on the validation set was conducted on the HPC cluster of the University of Neuchâtel, on 4x NVIDIA RTX 2080 Ti (4x 11GB). Specifically, we used random search and grid search to define the following training parameters: number of videos per GPU (32), number of epochs (100), initial learning rate (0.512), momentum (0.858), weight decay (0.0005). Other parameters remained similar to [11]. To best demonstrate an unadulterated skeleton-based approach, the model was trained on pose estimated data only and did not make use of **RGB+Pose** multimodal capability (i. e., no additional data stream was fed with RGB information).

Model evaluation: We defined our key metrics as Top1 accuracy, Top3 accuracy, and mean class accuracy similar to [48] for comparison. After 80 epochs, we reached our highest Top1 accuracy score on the validation set at 62.95%, which was used for our final test results described below.

Final results on PanAf

Our results (Table 1) demonstrate the successful application of our skeleton-based action recognition for animals. In particular, in the context of the highly relevant yet challenging task of automating the recognition of great ape behaviors in the wild, its accuracy is comparable to other video-based techniques (as in [48]). On the PanAf dataset, we achieved 74.98% Top1 accuracy in classifying nine different classes of behaviors. To the best of our knowledge, this also represents the first use of a skeleton-based approach to classify great ape behavior. Note here that the full size of the PanAf dataset after pose extraction, i. e. the input features of the behavior classifier, can be stored in text format on less than 60 MB of memory, around 20 times less than the storage space needed for the same dataset following a video-based approach. For ethologists working in the field, with scarce and often unreliable computational, storage, and transfer resources, this can represent a significant improvement without any performance loss in behavior recognition.

A limitation of our methodology can, however, be observed as the mean class accuracy, an important metric when dealing with unbalanced datasets, is lower than in

Table 1. Performance comparison with previous studies. Comparison of Top1 accuracy, Top3 accuracy, and mean class accuracy with previous video-based methods. Our framework improves both TopK accuracies while shrinking by a factor of around 20 times the volume of the behavior recognition model.

	Approach	Top1 acc	Top3 acc	Mean class acc
Two-Stream CNNs [48]	video-based	73.52%	94.07%	**42.33%**
ASBAR	skeleton-based	**74.98%**	**98.58%**	33.58%

previous video-based experiments (Table 1). The final model's confusion matrix is depicted in Fig 11, where one can observe that the model predominantly predicts behavior classes for which a larger number of samples are available in the dataset. Moreover, we note that three classes of behaviors, namely *climbing down*, *running* and *camera interaction*, have none of their samples correctly classified. This indicates the need for future work in animal pose estimation with domain adaptation and skeleton-based approaches for behavior classification.

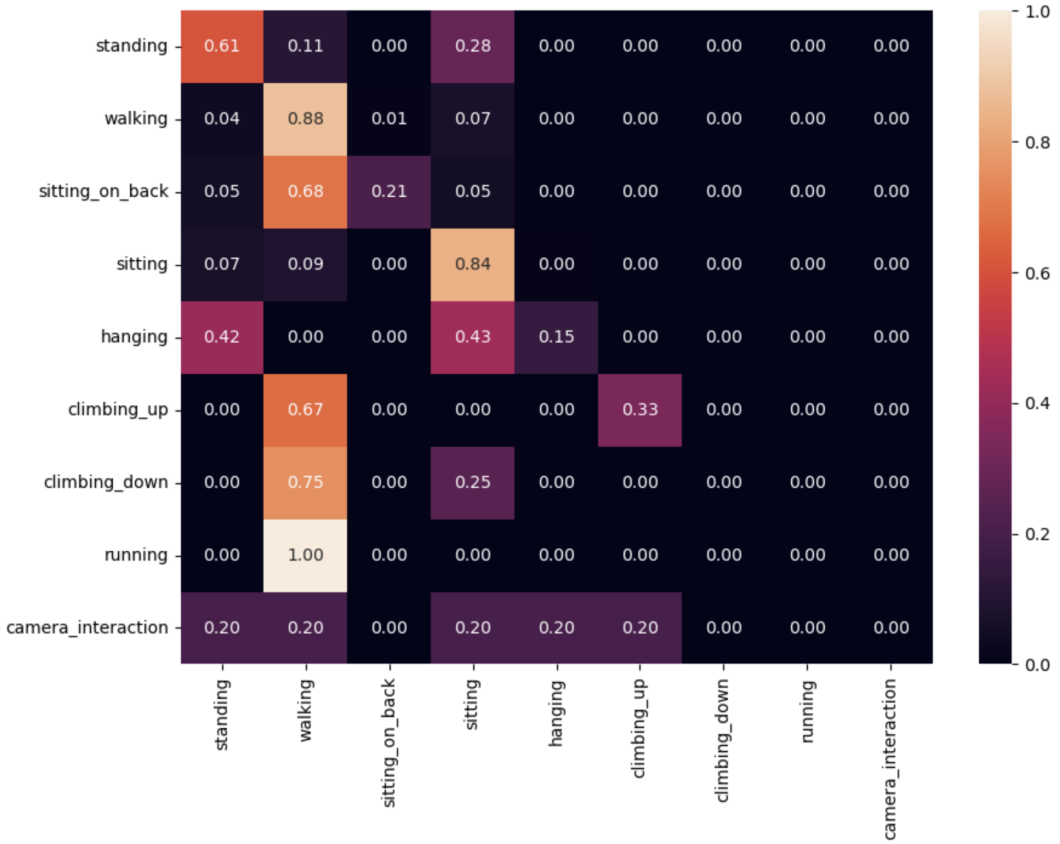

Fig 11. Final confusion matrix on PanAf For each true behavior label (vertically), the percentage of prediction is reported across all predicted behaviors (horizontally). The cells on the diagonal represent the percentage of correct predictions per class. For example, 61% of all the samples labeled 'standing' were correctly classified, while the remaining ones were wrongly predicted as 'sitting' (28%) and 'walking' (11%).

Discussion

Despite a rapid increase in open-source resources, such as large-scale animal pose datasets and machine learning toolboxes for animal pose estimation and human skeleton-based action recognition, their integration has mostly remained unexplored in the context of animal behavior recognition, especially outside of captive settings. By introducing ASBAR, a framework that combines both machine learning tasks of animal pose estimation and skeleton-based action recognition, we offer a new data/model pipeline, methodology, and GUI to help researchers automatically classify animal behaviors via pose estimation. In our experiments, we demonstrate the use and successful implementation of the framework by applying it to a complex and challenging task for machine learning models: the classification of great ape behaviors in their natural habitat. Compared to previous video-based approaches, we achieve comparable classification performance results, while reducing the action recognition model input size by a factor of around 20, leading to less required computational power, storage space, and data transfer necessity. Additionally, our skeleton-based method is known to be less sensitive to changes in visual settings and therefore less context-dependent. The pose estimator used in our experiments can extract poses from many other primate species without any further training (as it was trained with 26 species) and is robust to visual domain shift, which can speed up the pose estimation process for primatologists in diverse fields.

Regarding our experiment's results, we denote the need for future work, as our method achieves lower mean class accuracy than previous video-based methods. On one hand, this can potentially be explained by the noisy nature of the pose extracted data due to domain adaptation and, on the other hand, the action recognition model's failure to correctly classify behaviors for which only a small number of samples are available, due to the unbalanced nature of the behavior dataset.

We believe that our final model's performance can be increased either algorithmically and/or by enhancing data quality. In terms of algorithms, [11] reports higher accuracies when the data pipeline uses the keypoint detection as pose scoremaps rather than compressing them as (x, y, c) triplets, particularly in the context of lower-quality pose predictions, which may be relevant with domain adaptation. Additionally, making use of the RGB-Pose dual-modality may reflect positively on the model's performance as it fuses the predictions of both RGB and pose information for final behavior classification. Finally, we note that the performance assessment of 'within-domain' and 'out-of-domain' pose estimation models relying on EfficientNet's architectures did not achieve higher results than the ResNet one, as others have suggested [36]. This might have been caused by the initial learning rate being similar for all models for comparison, whereas EfficientNet models may have required hyperparameter tuning.

In terms of data quality, techniques to reduce the domain adaptation gap between *pose* and *behavior* datasets may result in more accurate pose extraction [58]. Similarly, the use of the extracted pose as pseudo-labels for semi-supervised learning could lead to significant performance gains for 'out-of-domain' pose estimation [5, 28, 39]. More specifically for pose prediction on PanAf, training a pose estimator on OpenApePose [10] may improve the final behavior classification.

In the future, our framework can also be extended to other animal pose datasets such as Animal Kingdom [41], MammalNet [9], or Animal Pose [5]. Beyond action recognition, the task of spatio-temporal action detection, a task highly relevant for ethologists, could be straightforwardly integrated into the framework, as it is accessible on MMaction2's platform already.

In conclusion, we demonstrated the practical use and relevance of skeleton-based action recognition approaches in computational ethology as well as future research directions for further improvement. Our framework and provided GUI offer other

researchers the tools and methodology to apply ASBAR in their own research to study other behaviors and/or animal species.

ANNEX Supporting information

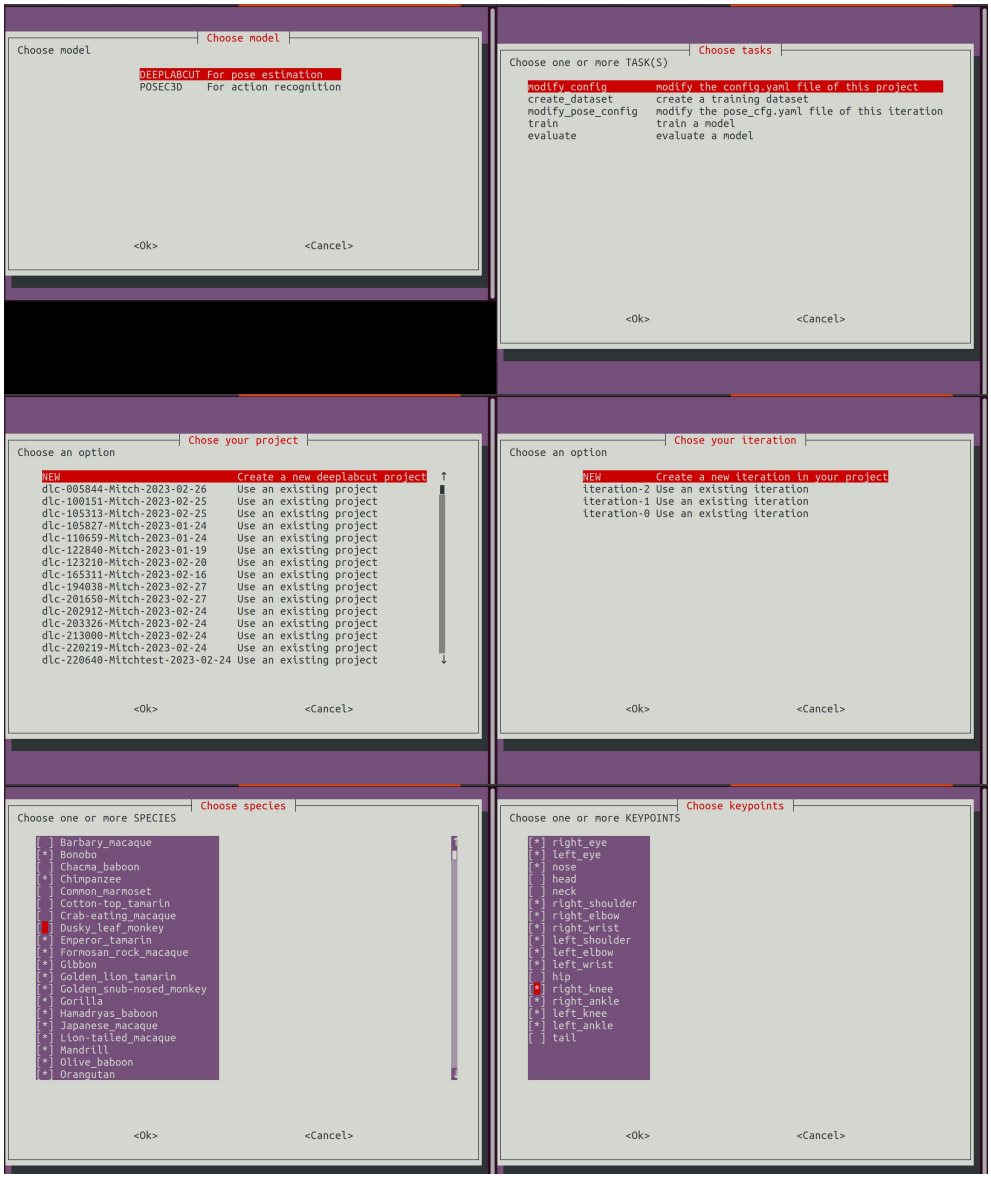

Fig S1. Examples of UI elements of the ASBAR graphical user interface The GUI is terminal-based and therefore can be rendered even when accessed on a distant machine, such as a cloud-based platform or a remote high-performance computer. Researchers may thus train and evaluate remotely the different models of DeepLabCut and MMaction2 without the need to write any programming code or terminal commands. See more details at https://github.com/MitchFuchs/asbar

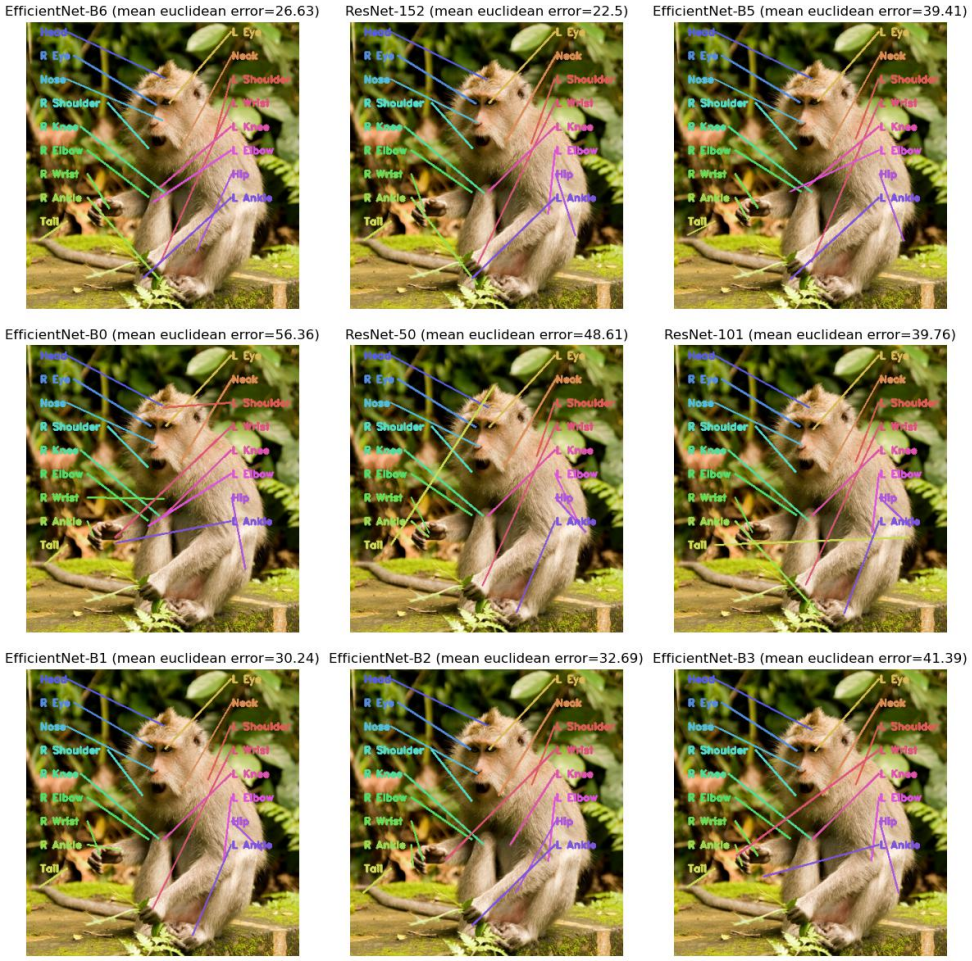

Fig S2. Prediction comparison of the nine models at test time. After 40,000 training iterations, the models' test predictions are visually compared on one example of the test set. Note for example that i) ResNet-50 (*center*) wrongly predicts the top of the head as the tail's position, ii) only three models can predict the left ankle's position accurately (ResNet-50 (*center*), ResNet-101 (*center right*), and EfficientNet-B1 (*bottom left*)) and iii) no model correctly detects the left knee's location.

Fig S3. PCK nasal dorsum The turquoise segment represents the length between the center of the eyes and the tip of the nose, i.e., the nasal dorsum. Any model prediction (represented in green) that falls within this distance of the ground-truth location (indicated in red) is considered as detected. In this case, all keypoints are detected except for the shoulders, neck, left wrist, and the hip (circled in purple). Hence, for this image, the detection rate would be $12/17 = 0.706 = 70.56\%$.

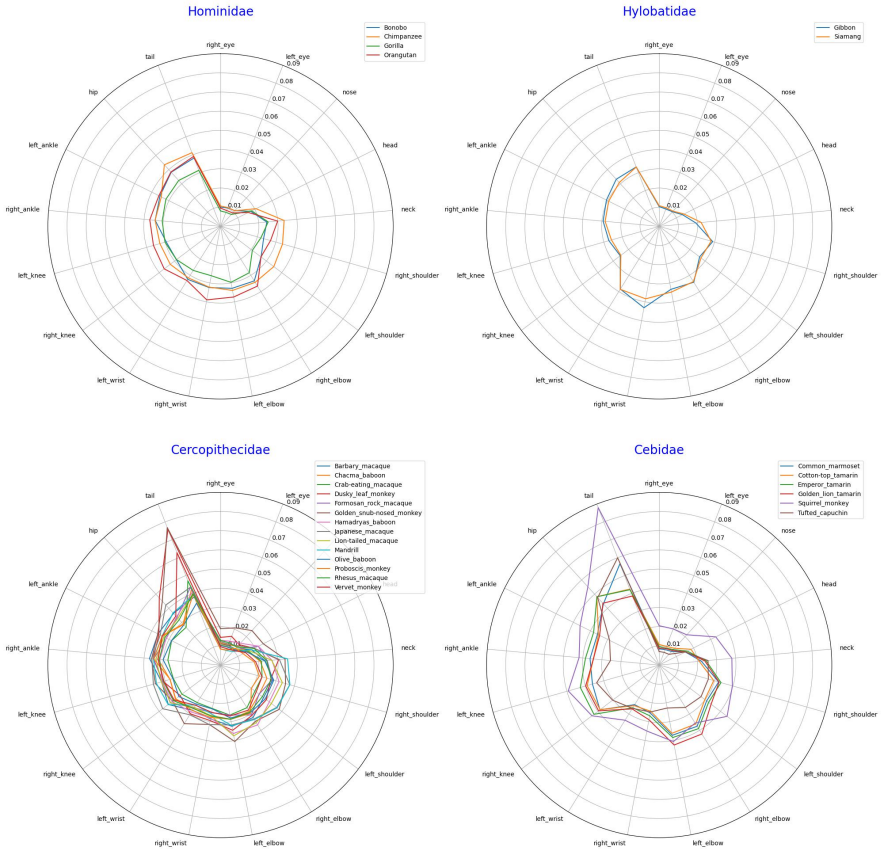

Fig S4. Normalized error rate by families, species and keypoints. For all OMC images at test time, we visualize the normalized error rate (NMER) for each species.

Acknowledgments

We extend our sincere gratitude to the team behind the Pan African Programme: 'The Cultured Chimpanzee', along with their partners, for granting us permission to use their data for this study. For access to the videos from the dataset, please reach out directly with the copyright holder Pan African Programme at http://panafrican.eva.mpg.de. In particular, we would like to thank H Kuehl, C Boesch, M Arandjelovic, and P Dieguez. Further acknowledgments go to: K Corogenes, E Normand, V Vergnes, A Meier, J Lapuente, D Dowd, S Jones, V Leinert, EWessling, H Eshuis, K Langergraber, S Angedakin, S Marrocoli, K Dierks, T C Hicks, J Hart, K Lee, M Murai and the team at Chimp&See.

The work that allowed for the collection of the PanAf dataset was made possible due to the generous support from the Max Planck Society, Max Planck Society Innovation Fund, and Heinz L. Krekeler. By extension, we also wish to thank: Foundation Ministre de la Recherche Scientifique, and Ministre des Eaux et Forêts in Cote d'Ivoire; Institut Congolais pour la Conservation de la Nature and Ministre de la Recherche Scientifique in DR Congo; Forestry Development Authority in Liberia; Direction des Eaux, Forêts Chasses et de la Conservation des Sols in Senegal; and Uganda National Council for Science and Technology, Uganda Wildlife Authority, and National Forestry Authority in

Uganda.

In addition, we would like to thank the team at NCCR Evolving Language and in particular Guanghao You, for allowing us to use their computational platform.

References

[1] David J. Anderson and Pietro Perona. "Toward a Science of Computational Ethology". In: *Neuron* 84.1 (Oct. 2014), pp. 18–31.

[2] Max Bain et al. "Automated audiovisual behavior recognition in wild primates". In: *Science Advances* 7.46 (2021), eabi4883.

[3] Praneet C Bala et al. "Automated markerless pose estimation in freely moving macaques with OpenMonkeyStudio". In: *Nature communications* 11.1 (2020), pp. 1–12.

[4] James P Bohnslav et al. "DeepEthogram, a machine learning pipeline for supervised behavior classification from raw pixels". In: *eLife* 10 (Sept. 2021). Ed. by Mackenzie W Mathis et al., e63377.

[5] Jinkun Cao et al. "Cross-domain adaptation for animal pose estimation". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019, pp. 9498–9507.

[6] Zhe Cao et al. "Realtime multi-person 2d pose estimation using part affinity fields". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 7291–7299.

[7] Joao Carreira and Andrew Zisserman. "Quo vadis, action recognition? a new model and the kinetics dataset". In: *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017, pp. 6299–6308.

[8] Chen Chen, Roozbeh Jafari, and Nasser Kehtarnavaz. "A survey of depth and inertial sensor fusion for human action recognition". In: *Multimedia Tools and Applications* 76 (2017), pp. 4405–4425.

[9] Jun Chen et al. "MammalNet: A Large-Scale Video Benchmark for Mammal Recognition and Behavior Understanding". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2023, pp. 13052–13061.

[10] Nisarg Desai et al. "OpenApePose: a database of annotated ape photographs for pose estimation". In: *arXiv preprint arXiv:2212.00741* (2022).

[11] Haodong Duan et al. "Revisiting skeleton-based action recognition". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 2969–2978.

[12] Max Planck Institute for Evolutionary Anthropology. *Pan African programme: The Cultured Chimpanzee*. URL: `http://panafrican.eva.mpg.de/index.php`.

[13] Christoph Feichtenhofer. "X3d: Expanding architectures for efficient video recognition". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020, pp. 203–213.

[14] Christoph Feichtenhofer et al. "SlowFast Networks for Video Recognition". In: *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. 2019, pp. 6201–6210.

[15] Liqi Feng et al. "A comparative review of graph convolutional networks for human skeleton-based action recognition". In: *Artificial Intelligence Review* (2022), pp. 1–31.

[16] Liqi Feng et al. "Action Recognition Using a Spatial-Temporal Network for Wild Felines". In: *Animals* 11.2 (Feb. 2021), p. 485.

[17] Emilie Genty and Michael Fuchs. *GApS: A Coding Scheme for Great Apes Signals in ELAN*. Jan. 2023. DOI: 10.5281/zenodo.7371604. URL: https://greatapesgestures.github.io.

[18] Fei Han et al. "Space-time representation of people based on 3D skeletal data: A review". In: *Computer Vision and Image Understanding* 158 (2017), pp. 85–105.

[19] Abigail Hardin and Ingo Schlupp. "Using machine learning and DeepLabCut in animal behavior". In: *acta ethologica* 25.3 (2022), pp. 125–133.

[20] Benjamin Y Hayden, Hyun Soo Park, and Jan Zimmermann. "Automated pose estimation in primates". In: *American journal of primatology* 84.10 (2022), e23348.

[21] Kaiming He et al. "Deep residual learning for image recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.

[22] Eldar Insafutdinov et al. "Deepercut: A deeper, stronger, and faster multi-person pose estimation model". In: *European conference on computer vision*. Springer. 2016, pp. 34–50.

[23] Pierre Karashchuk et al. "Anipose: a toolkit for robust markerless 3D pose estimation". In: *Cell reports* 36.13 (2021).

[24] Thomas N. Kipf and Max Welling. "Semi-Supervised Classification with Graph Convolutional Networks". In: *Proceedings of the 5th International Conference on Learning Representations* (ICLR). ICLR '17. Palais des Congrès Neptune, Toulon, France, 2017.

[25] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. "ImageNet classification with deep convolutional neural networks". In: *Communications of the ACM* 60.6 (May 2017), pp. 84–90.

[26] Rollyn Labuguen et al. "MacaquePose: A novel "in the wild" macaque monkey pose dataset for markerless motion capture". In: *Frontiers in behavioral neuroscience* 14 (2021), p. 581154.

[27] Jessy Lauer et al. "Multi-animal pose estimation, identification and tracking with DeepLabCut". In: *Nature Methods* 19 (2022), pp. 496–504.

[28] Chen Li and Gim Hee Lee. "From synthetic to real: Unsupervised domain adaptation for animal pose estimation". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2021, pp. 1482–1491.

[29] Fanjia Li et al. "Enhanced Spatial and Extended Temporal Graph Convolutional Network for Skeleton-Based Action Recognition". In: *Sensors* 20.18 (2020), p. 5260.

[30] Maosen Li et al. "Actional-structural graph convolutional networks for skeleton-based action recognition". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019, pp. 3595–3603.

[31] Weining Li, Sirnam Swetha, and Mubarak Shah. *Wildlife action recognition using deep learning*. 2020. URL: https://www.crcv.ucf.edu/wp-content/uploads/2018/11/Weining_L_Report.pdf.

[32] Jun Liu et al. "NTU RGB+D 120: A Large-Scale Benchmark for 3D Human Activity Understanding". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 42.10 (2020), pp. 2684–2701.

[33] Aleksander Madry et al. "Towards deep learning models resistant to adversarial attacks". In: *arXiv preprint arXiv:1706.06083* (2017).

[34] Markus Marks et al. "Deep-learning-based identification, tracking, pose estimation and behaviour classification of interacting primates and mice in complex environments". In: *Nature machine intelligence* 4.4 (2022), pp. 331–340.

[35] Alexander Mathis et al. "DeepLabCut: markerless pose estimation of user-defined body parts with deep learning". In: *Nature neuroscience* 21.9 (2018), pp. 1281–1289.

[36] Alexander Mathis et al. "Pretraining boosts out-of-domain robustness for pose estimation". In: *Proceedings of the IEEE/CVF winter conference on applications of computer vision*. 2021, pp. 1859–1868.

[37] Mackenzie Weygandt Mathis and Alexander Mathis. "Deep learning tools for the measurement of animal behavior in neuroscience". In: *Current opinion in neurobiology* 60 (2020), pp. 1–11.

[38] MMAction2 Contributors. *OpenMMLab's Next Generation Video Understanding Toolbox and Benchmark*. Version 1.0.0. July 21, 2020. URL: `https://github.com/open-mmlab/mmaction2`.

[39] Jiteng Mu et al. "Learning from synthetic animals". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 12386–12395.

[40] Tanmay Nath et al. "Using DeepLabCut for 3D markerless pose estimation across species and behaviors". In: *Nature protocols* 14.7 (2019), pp. 2152–2176.

[41] Xun Long Ng et al. "Animal kingdom: A large and diverse dataset for animal behavior understanding". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 19023–19034.

[42] OpenAI. *GPT-4 Technical Report*. 2023. arXiv: `2303.08774 [cs.CL]`.

[43] OpenMMLab Contributors. *Open-source Computer Vision Deep Learning Algorithm System*. Jan. 11, 2018. URL: `https://github.com/open-mmlab`.

[44] Talmo D Pereira et al. "SLEAP: A deep learning system for multi-animal pose tracking". In: *Nature methods* 19.4 (2022), pp. 486–495.

[45] Talmo D. Pereira, Joshua W. Shaevitz, and Mala Murthy. "Quantifying behavior to understand the brain". In: *Nature Neuroscience* 23.12 (Nov. 2020), pp. 1537–1549.

[46] Talmo D. Pereira et al. "Fast animal pose estimation using deep neural networks". In: *Nature Methods* 16.1 (Dec. 2018), pp. 117–125.

[47] Olga Russakovsky et al. "ImageNet Large Scale Visual Recognition Challenge". In: *International Journal of Computer Vision (IJCV)* 115.3 (2015), pp. 211–252.

[48] Faizaan Sakib and Tilo Burghardt. "Visual Recognition of Great Ape Behaviours in the Wild". English. In: International Conference on Pattern Recognition (ICPR) Workshop on Visual Observation and Analysis of Vertebrate And Insect Behavior , VAIB ; Conference date: 10-01-2021 Through 15-01-2021. Jan. 2021.

[49] Dian Shao et al. "Finegym: A hierarchical video dataset for fine-grained action understanding". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020, pp. 2616–2625.

[50] Karen Simonyan and Andrew Zisserman. "Two-Stream Convolutional Networks for Action Recognition in Videos". In: *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 1*. NIPS'14. Montreal, Canada: MIT Press, 2014, pp. 568–576.

[51] Nicholas Sofroniew et al. *napari: a multi-dimensional image viewer for Python*. DOI: `10.5281/zenodo.3555620`. URL: `https://github.com/napari/napari`.

[52] Ulrich Stern, Ruo He, and Chung-Hui Yang. "Analyzing animal behavior via classifying each video frame using convolutional neural networks". In: *Scientific reports* 5.1 (2015), p. 14351.

[53] Oliver Sturman et al. "Deep learning-based behavioral analysis reaches human accuracy and is capable of outperforming commercial solutions". In: *Neuropsychopharmacology* 45.11 (2020), pp. 1942–1952.

[54] Christian Szegedy et al. "Intriguing properties of neural networks". In: *arXiv preprint arXiv:1312.6199* (2013).

[55] Mingxing Tan and Quoc Le. "Efficientnet: Rethinking model scaling for convolutional neural networks". In: *International conference on machine learning*. PMLR. 2019, pp. 6105–6114.

[56] Du Tran et al. "A closer look at spatiotemporal convolutions for action recognition". In: *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*. 2018, pp. 6450–6459.

[57] Du Tran et al. "Learning spatiotemporal features with 3d convolutional networks". In: *Proceedings of the IEEE international conference on computer vision*. 2015, pp. 4489–4497.

[58] Mei Wang and Weihong Deng. "Deep visual domain adaptation: A survey". In: *Neurocomputing* 312 (2018), pp. 135–153.

[59] Panqu Wang et al. "Understanding convolution for semantic segmentation". In: *2018 IEEE winter conference on applications of computer vision (WACV)*. Ieee. 2018, pp. 1451–1460.

[60] Charlotte Wiltshire et al. "DeepWild: Application of the pose estimation tool DeepLabCut for behaviour tracking in wild chimpanzees and bonobos". In: *Journal of Animal Ecology* (2023).

[61] Sijie Yan, Yuanjun Xiong, and Dahua Lin. "Spatial temporal graph convolutional networks for skeleton-based action recognition". In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 32. 1. 2018.

[62] Xinyu Yang, Tilo Burghardt, and Majid Mirmehdi. "Dynamic curriculum learning for great ape detection in the wild". In: *International Journal of Computer Vision* (2023), pp. 1–19.

[63] Xinyu Yang, Majid Mirmehdi, and Tilo Burghardt. "Great ape detection in challenging jungle camera trap footage via attention-based spatial and temporal feature blending". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*. 2019.

[64] Yuan Yao et al. "OpenMonkeyChallenge: Dataset and Benchmark Challenges for Pose Estimation of Non-human Primates". In: *International Journal of Computer Vision* 131.1 (2023), pp. 243–258.

[65] Hang Yu et al. "Ap-10k: A benchmark for animal pose estimation in the wild". In: *arXiv preprint arXiv:2108.12617* (2021).

[66] Wei Zhan et al. "Key points tracking and grooming behavior recognition of Bactrocera minax (Diptera: Trypetidae) via DeepLabCut". In: *Mathematical problems in engineering* 2021 (2021), pp. 1–15.

[67] Lukas von Ziegler, Oliver Sturman, and Johannes Bohacek. "Big behavior: challenges and opportunities in a new era of deep behavior profiling". In: *Neuropsychopharmacology* 46.1 (2021), pp. 33–44.