SIPEC: the deep-learning Swiss knife for behavioral data analysis

Markus Marks[1,3], Jin Qiuhan[1,3], Oliver Sturman[2,3,], Lukas von Ziegler[2,3,], Sepp Kollmorgen[1,3], Wolfger von der Behrens[1,3], Valerio Mante[1,3], Johannes Bohacek[2,3,], Mehmet Fatih Yanik[1,3,*]

[1]Institute of Neuroinformatics ETH Zürich and University of Zürich, Switzerland
[2]Institute for Neuroscience, Department of Health Sciences and Technology, Lab of Molecular and Behavioral Neuroscience, ETH Zürich, Switzerland
[3]Neuroscience Center Zurich, ETH Zürich and University of Zürich, Switzerland
[*]correspondence to: yanik@ethz.ch

ABSTRACT

Analysing the behavior of individuals or groups of animals in complex environments is an important, yet difficult computer vision task. Here we present a novel deep learning architecture for classifying animal behavior and demonstrate how this end-to-end approach can significantly outperform pose estimation-based approaches, whilst requiring no intervention after minimal training. Our behavioral classifier is embedded in a first-of-its-kind pipeline (SIPEC) which performs segmentation, identification, pose-estimation and classification of behavior all automatically. SIPEC successfully recognizes multiple behaviors of freely moving mice as well as socially interacting non-human primates in 3D, using data only from simple mono-vision cameras in home-cage setups.

Introduction

While the analysis of animal behavior is crucial for systems neuroscience and preclinical assessment of therapies, it remains a highly laborious and error-prone process. Over the last few years, there has been a surge in machine learning tools for behavioral analysis, including segmentation, identification and pose estimation[1–8]. Although this has been an impressive feat for the field, a key element, the direct recognition of behavior itself has been rarely addressed. Unsupervised analysis of behavior[9,10] can be a powerful tool to capture the diversity of the underlying behavioral patterns, but the results of these methods do not align with human annotations and therefore require subsequent inspection. Here we demonstrate a complementary approach for researchers who seek to automatically identify particular behaviors of interest. Our approach relies on the initial annotation of exemplar behaviors, i.e. snippets of video footage, which are subsequently used to train a Deep Neural Network (DNN) to recognize these particular behaviors. Recently, there have been advances in the supervised

analysis of mouse behavior, using classifiers on top of pose-estimation generated features[11–14]. Sturman et. al.[13] demonstrated that the classification of mouse behaviors using features generated from pose-estimation algorithms can outperform the behavioral classification performance of commercial systems. Yet, such pose-estimation based classification of behavior remains a labor-intensive and error-prone process as we show below. Here, we designed a novel DNN-based architecture that is capable of classifying different behavioral states and even social interactions in an end-to-end fashion directly from video-data with minimal manual labeling while significantly outperforming the accuracy of pose-estimation based approaches. Our findings demonstrate that for the sole analysis of designated behaviors of interest, pose-estimation is not necessary.

We developed the first all-inclusive pipeline with modules for segmentation, identification, behavioral classification, and pose estimation of multiple and interacting animals in complex environments, called SIPEC, solely using DNNs and video data. We show SIPEC modules outperform current state-of-the-art approaches. SIPEC enables researchers to capture behavior from multiple animals in complex and changing environments over multiple days in 3D space, even from a single-camera with relatively little labeling in contrast to other approaches that use heavily equipped environments and large amounts of labeled data[7]. To rapidly train our modules, we use image augmentation[15] as well as transfer learning[16], optimized specifically for each module. To accelerate the reusability of SIPEC, we share the network weights among all four modules for mice and primates, which can be directly used for analyzing new animals in similar environments without further training, or serve as pre-trained networks to accelerate training of networks in different environments.

Results

Our algorithm performs segmentation (SIPEC:SegNet) followed by identification (SIPEC:IdNet), behavioral classification (SIPEC:BehaveNet) and finally pose estimation (SIPEC:PoseNet) from video frames (Figure 1). These four artificial neural networks, trained for different purposes, can also be used individually or combined in different ways (Figure 1a). To illustrate the utility of this feature, Figure 1b shows the output of pipelining SIPEC:SegNet and SIPEC:IdNet to track the identity and location of 4 primates housed together (Figure 1b). Similarly, Figure 1c shows the output of pipelining SIPEC:SegNet and SIPEC:PoseNet to do multi-animal pose estimation in a group of 4 mice.

Segmentation module SIPEC:SegNet. SIPEC:SegNet is based on the Mask-RCNN architecture[17], which we optimized for analyzing multiple animals and integrated into SIPEC. We further applied transfer learning[16] onto the weights of the Mask-RCNN ResNet-backbone[18] pretrained on the Microsoft Common Objects in Context (COCO dataset)[19] (see Methods for SIPEC:SegNet architecture and training). Moreover, we applied image augmentation[15] to increase generalizability and invariances, i.e. rotational invariance. For a given image, if we assume that N individuals are in the field of view (FOV), the output of SIPEC:SegNet are N segmentations or masks of the image. If the analysis is for multiple animals in a group, this step is mandatory, since subsequent parts of the pipeline are applied to the individual animals. Based on the masks, the center of masses (COMs) of the individual animals is calculated and serves

as a proxy for the animals' 2D spatial positions. Next, we crop the original image around the COMs of each animal, thus reducing the original frame to *N* COMs and *N* square-masked cutouts of the individuals. This output can then be passed onto other modules.

Segmentation performance on individual mice. We first examined the performance of SIPEC:SegNet on top-view video recordings of individual mice, behaving in an open-field test (OFT). 8 mice were freely behaving for 10 minutes in the TSE Multi Conditioning System's OFT arena, previously described in Sturman et al.[13]. We labeled the outlines of mice in a total of 23 frames using the VGG image annotator[20] from videos of randomly selected mice. To evaluate the performance, we used 5-fold cross-validation (CV). We assessed the segmentation performance on images of individual mice, where SIPEC:SegNet achieved a mean-Average Precision (mAP) of 1.0 ± 0 (mean ± s.e.m., see Methods for metric details). We performed a video-frame ablation study to find out how many labeled frames (outline of the animal, see Supplementary Figure 1) are needed for SIPEC:SegNet to reach peak performance (Figure 2b). While randomly selecting an increasing amount of training frames, we measured performance using CV. For single-mouse videos, we find that our model achieves 95% of mean peak performance (mAP of 0.95 ± 0.05) using as few as a total of 3 labeled frames for training.

Segmentation performance of groups of primates. To test SIPEC:SegNet for detecting instances of primates within a group, we annotated 191 frames from videos on different days (Day 1, Day 9, Day 16, Day 18). As exemplified in Figure 2a, the network handles even difficult scenarios very well: representative illustrations include ground-truth as well as predictions of moments in which multiple primates are moving rapidly while strongly occluded at varying distances from the camera. SIPEC:SegNet achieved a mAP of 0.91 ± 0.03 (mean ± s.e.m.) using 5-fold CV. When we performed the previously described ablation study, SIPEC:SegNet achieved 95% of mean peak performance (mAP of 0.87 ± 0.03) with only 30 labeled frames (Figure 2b).

Identification module SIPEC:IdNet. The identification network (SIPEC:IdNet) allows the determination of the identity of individual animals. Given SIPEC:IdNet receives input as a series (*T* time points) of cropped images of *N* individuals from SIPEC:SegNet, the output of SIPEC:IdNet are *N* identities. The input images from SIPEC:SegNet are scaled to the same average size (see Methods) before being fed into SIPEC:IdNet. We designed a feedforward classification neural network, which utilizes a DenseNet[21]-backbone pretrained on ImageNet[22]. This network serves as a feature-recognition network on single frames. We then utilize past and future frames by dilating the mask around the animal with each timestep. The outputs of the feature-recognition network on these frames are then integrated over *T* timesteps using a gated-recurrent-unit network (GRU[23,24]) (see Methods for architecture and training details). Based on the accuracy and speed requirements of a particular application, SIPEC:IdNet can integrate information from none to many temporally-neighboring frames. We developed an annotation tool for a human to assign identities of individual animals, in a multi-animal context, to segmentation masks in videoframes, which capture primates from different perspectives (Supplementary Figure 3). This tool was used for annotating identification data in the following sections. Below we compared SIPEC:IdNet's performance to that of the current state-of-the-art i.e. idTracker.ai[3]. idTracker.ai[3] requires tracking (sufficient overlap between segments of subsequent frames are used as a heuristic for being the same individual) to train for the

identification of individual animals and evaluated performance only within a single session. Particularly in complex or enriched home-cage environments, where animals are frequently obstructed as they move underneath/behind objects, this tracking becomes impossible, which causes failure of identification of the animals for the rest of the session. We evaluated the identification performance of SIPEC:IdNet without any tracking and even across sessions. Nonetheless, smoothing the outputs of SIPEC:IdNet as a secondary step can boost performance for continuous video sequences, but was not used for the following evaluation.

Identification of mice in an open-field test. We first evaluated the performance of SIPEC:IdNet in identifying 8 individual mice. We acquired 10 minute long videos of these mice behaving in the previously mentioned OFT (see Methods for details). While for the human observer, these mice are difficult to distinguish (Supplementary Figure 4), our network copes rather well. We used 5-fold CV to evaluate the performance, i.e. splitting the 10-minute videos into 2-minute long ones. Since this data is balanced, we use the accuracy metric for evaluation. We find that SIPEC:IdNet achieves 99 ± 0.5 % (mean and s.e.m.) accuracy, while the current state of the art idTracker.ai[3] only achieves 87 ± 0.2 % accuracy (Figure 2c). The ablation study shows that only 650 labeled frames (frame and identity of the animal) are sufficient for the SIPEC:IdNet to achieve 95% of its mean peak performance (Figure 2d). We tested how this performance translates to identifying the same animals during the subsequent days (Supplementary Figure 5). We find that identification performance is similarly high on the second day 86 ± 2 %, using the network trained on day 1. Subsequently, we tested identification robustness with respect to the interventions on day 3. Following a forced swim test, the identification performance of SIPEC:IdNet, trained on data of day 1, dropped dramatically to 4 ± 2 %, indicating that features utilized by the network to identify the mice are not robust to this type of intervention.

Identification of individual primates in a group. To evaluate SIPEC:IdNet's performance on the identification of individual primates within a group, we used the SIPEC:SegNet-processed videos of the 4 macaques (see Section "Segmentation performance of groups of primates"). We annotated frames from 7 videos taken on different days, with each frame containing multiple individuals, yielding approximately 2200 labels. We used leave-one-out CV with respect to the videos in order to test SIPEC:IdNet generalization across days. Across sessions SIPEC:IdNet reaches an accuracy of 78 ± 3 % (mean ± s.e.m.) while idTracker.ai[3] achieves only 33 ± 3 % (Figure 2c), where the human expert (i.e. ground truth) had the advantage of seeing all the video frames and the entire cage (i.e. the rest of the primates). We did a separate evaluation of the identification performance on "typical frames" i.e. where the human expert can also correctly identify the primates using single frames. In this case, SIPEC:IdNet achieved a performance of 86 ± 3 (Supplementary Figure 6). The identification labels can then be further enhanced by greedy mask-match based tracking (see Methods for details). Supplementary Video 1 illustrates the resulting performance on a representative video snippet. We perform here an ablation study as well, which yields 95% of mean peak performance at 1504 annotated training samples (Figure 2d).

Behavioral classification module SIPEC:BehaveNet. SIPEC:BehaveNet offers researchers a powerful means to recognize specific animal behaviors in an end-to-end fashion within a

single neuronal net framework. SIPEC:BehaveNet uses video frames of *N* individuals over *T* time steps to classify what actions the animals are performing. We use a recognition network to extract features from the analysis of single frames, that we base on the Xception[25] network architecture. We initialize parts of the network with ImageNet[4] weights. These features are then integrated over time by a temporal convolution network[26,27] to classify the behavior of the animal in each frame (see Methods for architecture and training details).

*SIPEC end-to-end behavior recognition outperforms DLC-based approach***.** We compare our end-to-end approach to Sturman et al.[13], who recently demonstrated that they can classify behavior based on DLC[1] generated features. On top of a higher classification performance with fewer labels, SIPEC:BehaveNet does not require annotation and training for pose estimation, if the researcher is interested in behavioral classification alone. The increased performance with fewer labels comes at the cost of a higher computational demand since we increased the dimensionality of the input data by several orders of magnitude (12 pose estimates vs. 16384 pixels). To test our performance we used the data and labels from Sturman et al.[13] of 20 freely behaving mice in an OFT. The behavior of these mice was independently annotated by 3 different researchers on a frame-by-frame basis using the VGG video annotation tool[20]. Annotations included the following behaviors: supported rears, unsupported rears, grooming and none (unlabeled default class). While Sturman et al.[13] evaluated the performance of their behavioral event detection by averaging across chunks of time, evaluating the frame-by-frame performance is more suitable for testing the actual network performance since it was trained the same way. Doing such frame-by-frame analysis shows that SIPEC:BehaveNet has fewer false positives as well as false negatives with respect to the DLC-based approach of Sturman et al. We illustrate a representative example of the performance of both approaches for each of the behaviors with their respective ground truths (Figure 3a). We further resolved spatially the events that were misclassified by Sturman et al., that were correctly classified by SIPEC:BehaveNet and vice versa (Figure 3b). We calculated the percentage of mismatches, that occurred in the center or the surrounding area. For grooming events mismatches of Sturman et al.[13] and SIPEC:BehaveNet occurs similarly often in the center 41 ± 12 % (mean and s.e.m.) and 42 ± 12 % respectively. For supported and unsupported rearing events Sturman et al.[13] find more mismatches occurring in the center compared to SIPEC:BehaveNet (supported rears: 40 ± 4 % and 37 ± 6 %, unsupported rears: 12 ± 2 % and 7 ± 2 %). This indicates that the misclassifications of the pose estimation based approach are more biased towards the center than the ones of SIPEC:BehavNet. To quantify the behavioral classification over the whole timecourse of all videos of 20 mice, we used leave-one-out CV (Figure 3c). We used macro-averaged F1-score as a common metric to evaluate a multi-class classification task and Pearson correlation (see Methods for metrics) to indicate the linear relationship between the ground truth and the estimate over time. For the unsupported rears/grooming/supported rears behaviors SIPEC:BehaveNet achieves F1-Scores of 0.6 ± 0.16/0.49 ± 0.21/0.84 ± 0.04 (values reported as mean ± s.e.m.) respectively, while the performance of the manually intensive Sturman et al.[13]'s approach reaches only 0.49 ± 0.11/0.37 ± 0.2/0.84 ± 0.03, leading to a significantly higher performance of SIPEC:BehaveNet for the unsupported rearing (F1: $p=1.689\times10^{-7}$, Wilcoxon paired-test was used as recommended[28]) as well as the grooming (F1: $p=6.226\times10^{-4}$) behaviors. This improved performance is due to an increased precision as well as increased

recall for the different behaviors (Supplementary Figure 7a). As expected, more stereotyped behaviors with many labels like supported rears yield higher F1, while less stereotypical behaviors like grooming with fewer labels have lower F1 for both SIPEC:BehaveNet and DLC-based approach. Additionally, we computed the mentioned metrics on a dataset with shuffled labels to indicate chance performance for each metric as well as computed each metric when tested across human annotators to indicate an upper limit for frame-by-frame behavioral classification performance (Supplementary Figure 7b). While the overall human-to-human F1 is 0.79 ± 0.07 (mean ± s.e.m.), SIPEC:BehaveNet classifies with an F1 of 0.71 ± 0.07. As Sturman et al.[13] demonstrated for unsupported and supported rears, this performance is sufficient to reach human-like performance, when behavioral classifications are temporally grouped to single events rather than analyzed frame-by-frame. Lastly, we performed a frame ablation study and showed that SIPEC:BehaveNet needs only 114 minutes, less than 2 hours of labeled data, to reach peak performance in behavioral classification (Figure 3d).

Pose estimation module SIPEC:PoseNet. We also added an encoder-decoder architecture[29] module to SIPEC for performing pose estimation (SIPEC:PoseNet) (see Methods). SIPEC:PoseNet can be used to perform pose estimation on N animals, yielding K different coordinates for previously defined landmarks on the body of each animal. The main advantage of SIPEC:PoseNet in comparison to previous approaches is the inputs it receives from SIPEC:SegNet (top-down pose estimation). While bottom-up approaches such as DLC[1] require grouping of pose estimates to individuals, our top-down approach makes the assignment of pose estimates to individual animals trivial, as inference is performed on the masked image of an individual animal and pose estimates within that mask are assigned to that particular individual (Figure 1c). We labeled frames with 13 standardized body for tracking mice in OFT similarly to Sturman et. al.[13]. SIPEC:PoseNet achieves a Root-Mean-Squared-Error (RMSE) (see Methods) of 2.7 pixels in mice (Supplementary Figure 8) for a total of 950 labeled training frames, which is comparable to the 2.9 pixel RMSE reported for DLC[1]. In our top-down pose estimation framework previously published pose estimation methods, working on single animals, can also be easily substituted into our pipeline to perform multi-animal pose estimation in conjunction with SIPEC:SegNet.

Socially interacting primate behavior classification. We used the combined outputs of SIPEC:SegNet and SIPEC:IdNet, smoothed by greedy match based tracking, to generate videos of individual primates over time (see Methods for details). To detect social events, we used SIPEC:SegNet to generate additional video events covering "***pairs***" of primates. Whenever masks of individual primates came sufficiently close (see Methods), an interaction event was detected. We were able to rapidly annotate these videos again using the VGG video annotation tool[20] (overall 80 minutes of video are annotated from 3 videos, including the individual behaviors of object interaction, searching, **social grooming** and none (background class)). We then trained SIPEC:BehaveNet to classify frames of individuals as well as merged frames of pairs of primates socially interacting over time. We used grouped 5-fold stratified CV over all annotated video frames, with labeled videos being the groups. Overall SIPEC:BehaveNet achieved a macro-F1 of 0.72 ± 0.07 (mean ± s.e.m.) across all behaviors (Figure 4a). This performance is similar to the earlier mentioned mouse behavioral classification performance. The increased variance compared to the classification of mouse

behavior is expected as imaging conditions, as previously mentioned, are much more challenging and primate behaviors are much less stereotyped compared to mouse behaviors.

Tracking identity and position of individual primates among groups in 3D without stereovision. By performing SIPEC:SegNet and SIPEC:IdNet inference on a full one-hour video, we easily built a density map of positions of individuals within the husbandry (Figure 1a). With a single camera without stereovision, one cannot optically acquire depth information. Instead, we used the output masks of SIPEC:SegNet and annotated the position of the primates in 300 frames using a 3D model (Supplementary Figure 9). Subsequently, we generated 6 features using Isomap[30] and trained a multivariate linear regression model to predict the 3D positions of the primates (Figure 4b). Using 10-fold CV, our predicted positions using only single camera have an overall RMSE of only 0.43 ± 0.01 m (mean ± s.e.m.), that is of 0.27 ± 0.01 m in x-direction or 6% error w.r.t the room dimension in x-direction; 0.26 ± 0.01 m / 7% and 0.21 ± 0.01 m / 7% for the y and z coordinates respectively.

Discussion

We have presented SIPEC, a novel pipeline, using specialized deep neural networks to perform segmentation, identification, behavioral classification and pose estimation on multiple animals. With SIPEC we address multiple key challenges in the domain of behavioral analysis. Our **SIPEC:SegNet** enables the segmentation of animals with minimal labeling. Subsequently, **SIPEC:BehaveNet** enables end-to-end animal behavior recognition directly from raw video data. End-to-end classification has the advantage of not requiring adjustment of pre-processing or feature engineering to specific video conditions. Our approach outperforms pose estimation approaches on a well-annotated mouse behavioral dataset. We thus propose to skip pose-estimation if researchers are solely interested in classifying behavior. We note that our end-to-end approach increases the input-dimensionality of the behavioral classification network and therefore uses more computational resources and is slower than pose estimation based approaches. **SIPEC:IdNet** identifies primates in complex environments across days with high accuracy, while also offering smoothing of labels. We showed that identification accuracy is significantly higher for typical frames with good visibility of the animal. Therefore, the better the coverage by the camera system the better is the overall identification performance. Finally, **SIPEC:PosNet** enables top-down pose estimation of multiple animals in complex environments, making it easy to assign pose estimates to individual animals. All approaches are optimized through augmentation and transfer learning, significantly speeding up learning and reducing labeling in comparison to the other approaches we tested on mouse as well as non-human primate datasets. We demonstrate how SIPEC can analyze social interactions of non-human primates such as mutual grooming over multiple days after annotating sequences of interest in videos. Finally, we show how SIPEC enables 3D vision from a single-camera view, yielding an off-the-shelf solution for home-cage monitoring of primates, without the need for setting stereo-vision setups.

SIPEC can be used to study the behavior of primates and their social interactions over longer periods of time in a naturalistic environment. After initial training of SIPEC modules, they could automatically output a specific behavioral profile of each individual in a group, potentially over days, weeks or months and therefore also be used to quantify the changes in behavioral and social dynamics over time.

It would be interesting to integrate complementary unsupervised approaches[9,10] into SIPEC to cover behavioral sequences that are not classified by SIPEC:BehaveNet and give researchers the chance to explore the data of these not-specified behaviors in an unsupervised fashion. The output of other modules (SIPEC:SegNet, SIPEC:IdNet and SIPEC:PoseNet) could be used as inputs for these unsupervised approaches to assign results to individual animals.

SIPEC is easy to use for practitioners from the neuroscience community as well as very modular and hackable to encourage future improvements by researchers from the machine learning community. SIPEC can aid neuroscientific research by increasing throughput, reproducibility and transferability of results of behavioral analysis. To facilitate this, we make our code public with pre-trained networks for both mice and non-human primates.

Acknowledgments
This project was funded by the Swiss Federal Institute of Technology (ETH) Zurich and the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program (grant agreement No 818179). We'd like to thank Petra Tornmalm and Victoria de La Rochefoucauld for the annotation of primate data and feedback on primate behavior. We'd like to thank Paul Johnson, Baran Yasar, Bifeng Wu and Aagam Shah for helpful discussions and feedback.

Author contributions
M.M. developed, implemented and evaluated the SIPEC modules and framework. J.Q. developed segmentation filtering, tracking and 3D-estimation. M.M., W.B. and M.F.Y. wrote the manuscript. M.M., O.S., LvZ., S.K., W.B., V.M., J.B. and M.F.Y. conceptualized the study. All authors gave feedback on the manuscript.

Competing interests
The authors declare no competing interests.

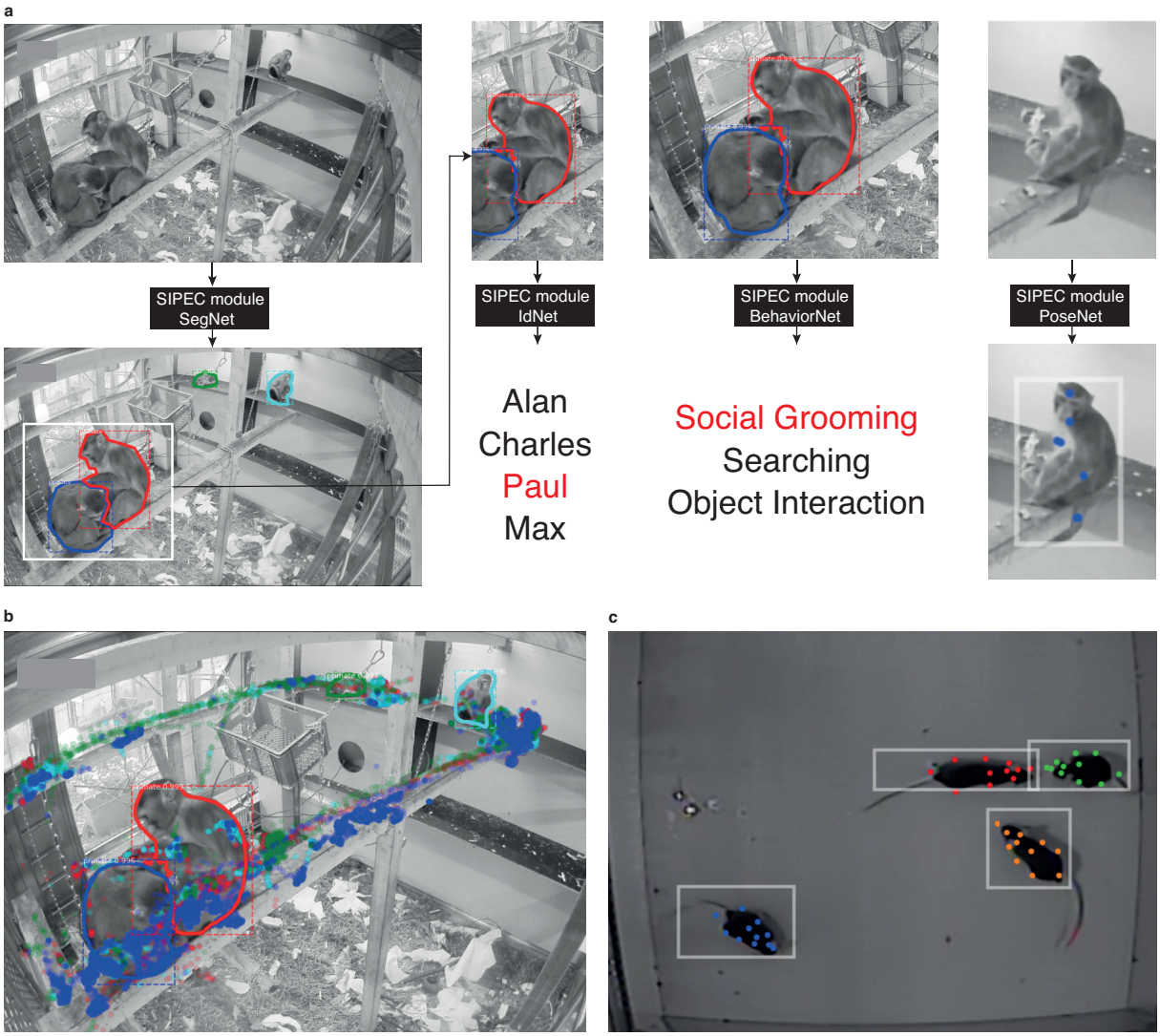


Fig. 1 | Overview of the SIPEC workflow and modules. a) From a given video, instances of animals are segmented with the segmentation network (SIPEC:SegNet), indicated by masked outline as well as bounding boxes. Subsequently, individuals are identified using the identification network (SIPEC:IdNet). For each individual, the pose and behavior can be estimated/classified using the pose estimation network (SIPEC:PoseNet) and the behavioral identification network (SIPEC:BehaveNet), respectively. b) Outcome of SIPEC:SegNet, and SIPEC:IdNet modules are overlaid on a representative video-frame. Time-lapsed positions of individual primates (center of mass) are plotted as circles with respective colors. c) Outputs of SIPEC:SegNet (boxes) and SIPEC:PoseNet (colored dots) on a representative video-frame of mouse open-field data.

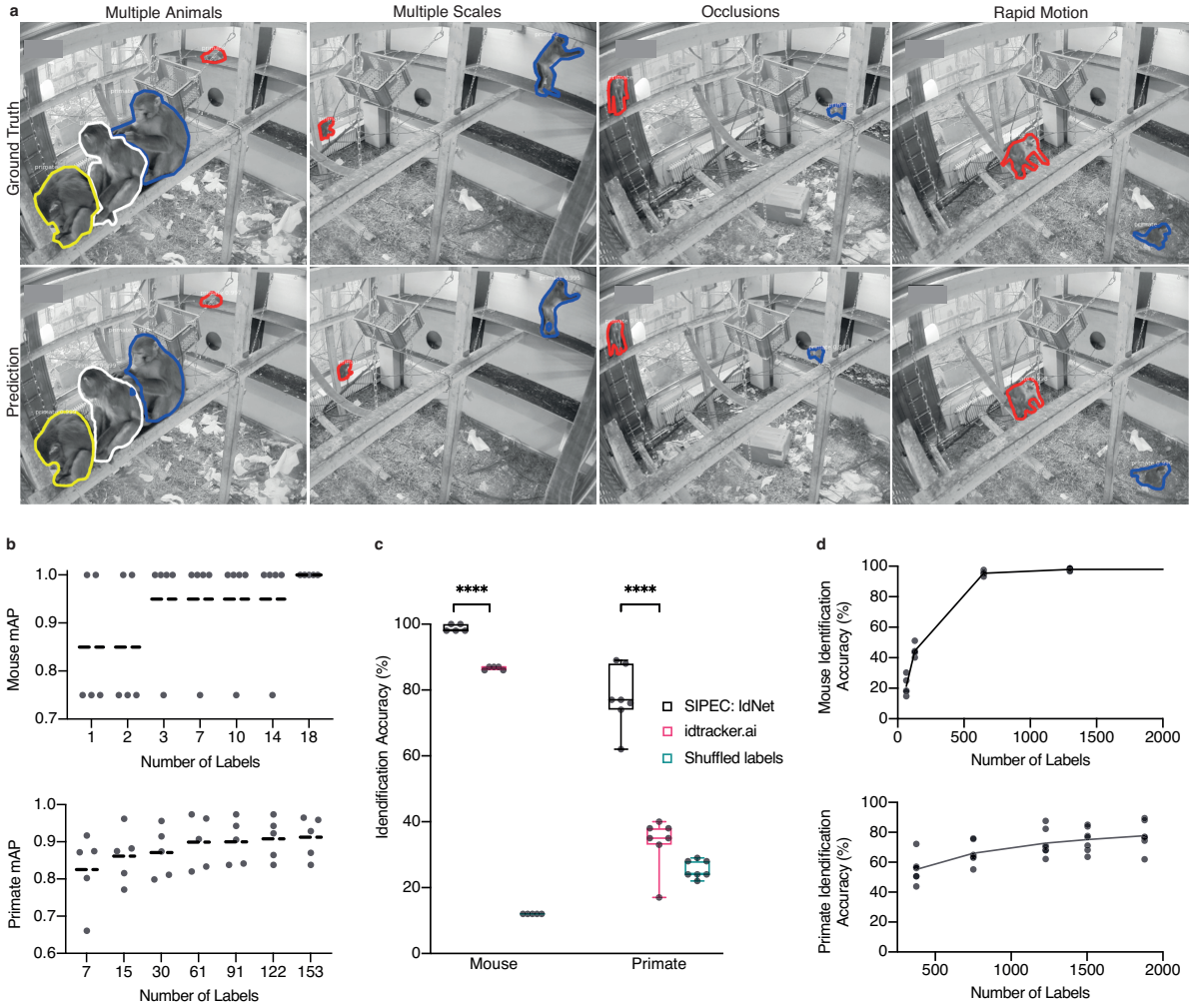


Fig. 2 | Performance of the segmentation (SIPEC:SegNet) and identification (SIPEC:IdNet) modules under demanding video conditions and using few labels. a) Qualitative comparison of ground truth (top row) versus predicted segmentation masks (bottom row) under challenging conditions; multiple animals, at varying distances from the camera, under strong visual occlusions, and in rapid motions. b) SIPEC:SegNet performance in mAP (mean average precision) for primates and mice as a function of the number of labels. The black lines indicate the mean for 5-fold CV while black circles indicate mAP for individual folds. c) Comparison of identification accuracy for SIPEC:IdNet module, idtracker.ai[3], and randomly shuffled labels (chance performance). 8 videos from 8 individual mice and 7 videos from 4 group-housed primates are used. All data is represented by a minimum to maximum box-and-whisker plot, showing all points. d) The accuracy of SIPEC:IdNet (for primates and mice) as a function of the number of training labels used. The black lines indicate the mean for 5-fold CV with individual folds displayed.

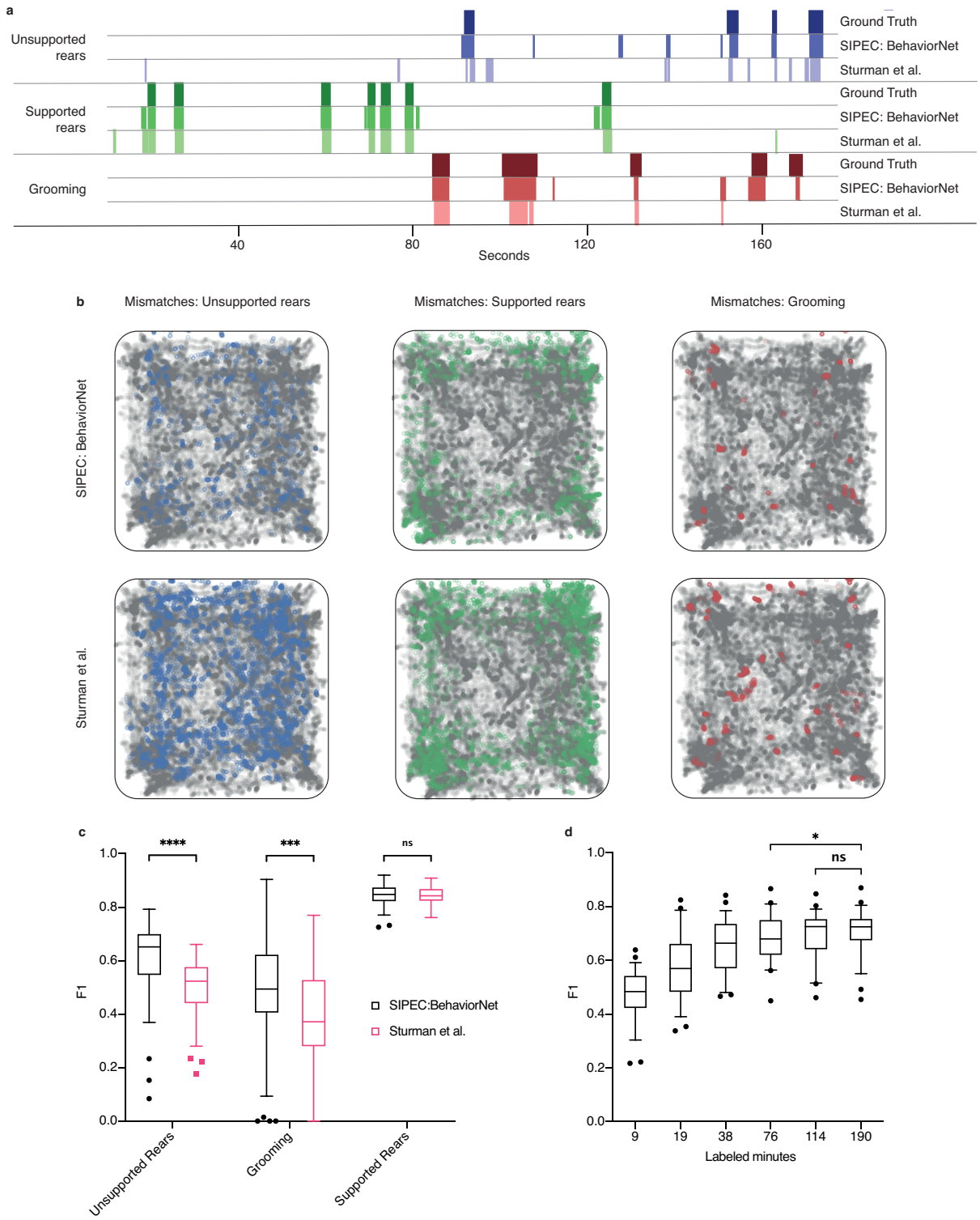


Fig. 3 | SIPEC:BehaveNet outperforms DeepLabCut-based approach (Sturman et al.[13]). a) Comparison of behavioral classification by human annotator (ground truth), SIPEC:BehaveNet, and Sturman et al.[13] b) Errors in the classification of mouse behavior in open arena for SIPEC:BehaveNet versus Sturman et al. Each colored dot represents a behavioral event that is incorrectly classified by that method (while correctly classified by the other) with respect to the ground truth. none-classified (background class) positions of mice are indicated as grey dots. c) Frame-by-frame classification performance per video (n=20 mice) compared to ground truth. Wilcoxon paired test:* $p <= 0.05$; *** $p <= 0.001$; **** $p <= 0.0001$.

d) SIPEC:BehaveNet classification performance as a function of labeled minutes. All data is represented by a Tukey box-and-whisker plot, showing all points.

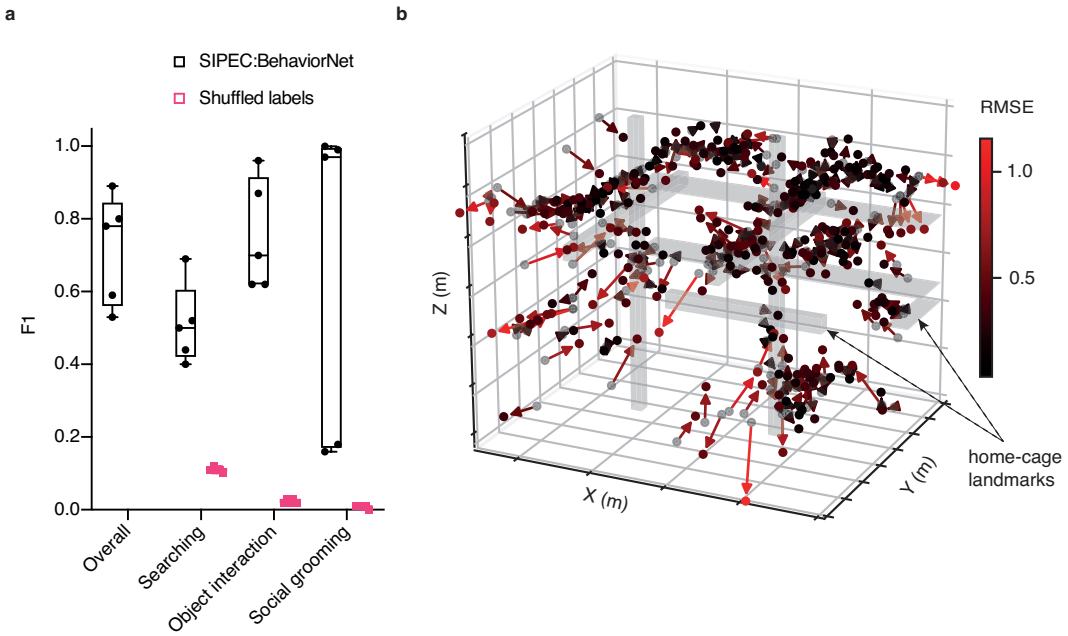


Fig. 4 | SIPEC can recognize social interactions of multiple primates and infer their 3D position using a single camera. a) Performance of SIPEC:BehaveNet for individual and social behaviors with respect to ground truth evaluated using grouped 5-fold CV. Behaviors include searching, object interaction, and social grooming; while the performance is measured using F1. F1 on shuffled labels is included for comparison. All data is represented by a minimum to maximum box-and-whisker plot, showing all points. b) Evaluation of 3D position estimates of primates in home-cage. Annotated positions (n=300) are marked by black spots while predicted positions are marked as red-hued spots at the end of the solid arrows (color-coded using a red gradient with brighter red indicating higher RMSE of predicted to true position).

Methods

Animals. C57BL/6J (C57BL/6JRj) mice (male, 2.5 months of age) were obtained from Janvier (France). Mice were maintained in a temperature- and humidity-controlled facility on a 12-h reversed light-dark cycle (lights on at 08:15 am) with food and water ad libitum. Mice were housed in groups of 5 per cage and used for experiments when 2.5–4 months old. For each experiment, mice of the same age were used in all experimental groups to rule out confounding effects of age. All tests were conducted during the animals' active (dark) phase from 12–5 pm. Mice were single housed 24 h before behavioral testing in order to standardize their environment and avoid disturbing cage mates during testing. The animal procedures of these studies were approved by the local veterinary authorities of the Canton Zurich, Switzerland, and carried out in accordance with the guidelines published in the European Communities Council Directive of November 24, 1986 (86/609/EEC).

Acquisition of mouse data. For mouse behavioral data and annotation we refer to Sturman et al.[13]. For each day we randomized the recording chamber of mice used. On day 1,2 we recorded animals 1-8 individually. On day 3, for measuring the effect of interventions on performance, were forced swim tested in water for 5 minutes immediately before to the recording sessions.

Acquisition of primate data. 4 male rhesus macaques (Primates were recorded with a 1080p camera withing their home-cage. The large indoor room measures about 15m^2. Videos were acquired using a Bosch Autodome IP starlight 7000 HD camera with 1080p resolution at 50 Hz.

Annotation of segmentation data. To generate training data for segmentation training, we randomly extracted frames of mouse and primate videos using a standard video player. Next, we used the VIA video annotator[20] to draw outlines around the animals.

Generation and annotation of primate behavioral videos. For creating the dataset, 3 primate videos of 20-30 minutes were annotated using the VIA video annotator[20]. These videos were generated by previous outputs of SIPEC:SegNet and SIPEC:IdNet. Frames of primates, that were identified as the same over consecutive frames, were stitched together in order to generate individualized videos. To generate videos of social interactions, we dilated the frames of each primate in each frame and checked if their overlap crossed a threshold, in which case we recalculated the COM of those two masks and center-cropped the frame around it. Labeled behaviors included 'searching', 'object interacting', 'social grooming' and 'none' (background class).

Tracking. Based on the outputs of the segmentation masks, we implemented greedy-match based tracking. For a given frame the bounding box of a given animal is assigned to the bounding box in the previous frame with the largest spatial overlap. We used the resulting track-identities to smooth the labels that were output by SIPEC:IdNet.

Identification labeling with the SIPEC toolbox. As part of SIPEC we release a GUI that allows to label for identification when multiple animals are present (Supplementary Figure 4). For that SIPEC:SegNet has to be trained and inference has to be performed on videos to be id-labeled. SIPEC:SegNet results can then be loaded from the GUI and overlaid with the original videos. Each box then marks an instance of the species that is to be labeled in green. For each of the animals, a number on the keyboard can be defined, which corresponds to the permanent id of the animal. This number has then to be pressed and the mask-focus jumps to the next mask until all masks in that frame are annotated. Subsequently, the GUI jumps to the next frame in either regular intervals or randomly throughout the video, as predefined by the user. Once a predefined number of masks is reached, results are saved and the GUI is closed.

SIPEC:SegNet Network Architecture and training. SIPEC:SegNet was designed by optimizing the Mask R-CNN architecture. We utilized a ResNet101 and feature pyramid network (FPN)[31] as the basis of a convolutional backbone architecture. These features were fed to the region proposal network (RPN), which applies convolutions onto these feature maps and proposes regions of interest (ROIs). Subsequently, these are passed to a ROIAlign layer, which performs feature pooling, while preserving the pixel-correspondence in the original image. Per level of the pyramidal ROIAlign layer we assign a ROI feature map from the different layers of the FPN feature maps. Now multiple outputs are generated from the FPN, one is classifying if an animal is identified. The regressor head of the FPN returns bounding-box regression offsets per ROI. Another fully convolutional performs the mask prediction, returning a binary mask for each animal ROI. The network is trained using stochastic gradient descent, minimizing a multi-task loss for each ROI:

$$\mathrm{L} = L_{mask} + L_{regression} + L_{class}$$

where L_{mask} is the average binary cross-entropy, applied to each ROI. $L_{regression}$ is a regression loss function, modified to be outlier robust as in the original Fast R-CNN paper[32]. L_{class} is calculated for each of the anchors as a logarithmic loss of non-object vs object. The learning rate was 0.0025 and training was done by first training the output layers for some epochs and then incrementally training previous blocks.

SIPEC:IdNet Network Architecture and training. SIPEC:IdNet was based on the DenseNet architecture[21] for frame-by-frame identification. It consists of 4 dense blocks, which consist of multiple sequences of a batch normalization layer, a ReLU activation and a convolution. The resulting feature maps are concatenated to the outputs of the following sequences of layers (skip-connections). The resulting blocks are connected through transitions, that are convolutional followed by pooling layers. After the last dense block, there is an average pooling layer that we connect to a Dropout[33] layer with a dropout rate of 0.5 followed by the softmax classification layer. For the recurrent SIPEC:IdNet we remove the softmax layer and feed the output of the average pooling layers for each timepoint into a batch normalization layer[34] followed by 3 layers of bidirectional gated recurrent units[23,24] with leaky ReLU activation[35,36] (alpha=0.3) followed by a 0.2 Dropout[33] followed by the softmax layer. The input for SIPEC:IdNet is the output cutouts of individuals, generated by SIPEC:SegNet (for the single-

animal case a background-subtracted thresholding and centered-cropping would also work). For the recurrent case, the masks of past or future frames are dilated with a factor that increases with distance in time in order to increase the field of view. We pre-trained first the not-recurrent version of SIPEC:IdNet using Adam[37] with an lr=0.00025, a batch size of 16 and using a weighted cross-entropy loss. We used a learning rate scheduler in the following form:

$$L_{E+1} = \frac{L_E}{k^E} \quad (2)$$

Where E stands for epoch, using a k=1.5. Subsequently we removed the softmax layer and fixed the weights of the network. We then trained the recurrent SIPEC:IdNet again using Adam[37] and an lr=0.00005, k=1.25 and a batch size of 6.

SIPEC:BehaveNet Network Architecture and training. SIPEC:BehaveNet was constructed as an end-to-end action recognition network. It consists of a feature recognition network that performs on a single frame basis and a network, which integrates these features over time (Supplementary Figure 2). The feature recognition network (FRN) is based on the Xception[25] architecture, which consists of an entry flow, a middle flow and an exit flow. The entry flow initially processes the input with convolution and ReLU blocks. Subsequently, we pass the feature maps through 3 blocks of separable convolution layers, followed by ReLU , separable convolution and a max pooling layer. The outputs of these 3 blocks are convolved and concatenated. And passed to the middle flow. The Middle flow consists of 8 blocks of a ReLU layer followed by a separable convolution layer. The Exit receives the feature maps from the middle flow and passes it one more entry-flow like block, followed by 2 times of separable convolution and ReLU units. Finally, these features are integrated by a global average pooling layer and then the softmax output. This FRN was first pre-trained on frame-by-frame basis using an lr=0.00035, gradient clipping norm of 0.5 and batch size=36 using the Adam[37] optimizer. For mouse data we reduced the original Xception architecture by the first 17 layers, in order to speed up computation and reduce overfitting. After training the FRN the outputting dense and softmax layers were removed and all weights were fixed for further training. The FRN-features were integrated over time by a non-cause Temporal Convolution Network[26]. It is non-causal, because for classification of behavior at timepoint t it integrates features from $[t-n,t+n]$ with n being the number of timesteps, therefore looking not only backward in time but also forward. In this study, we used an n of 10. The FRN features are transformed by multiple TCN blocks of the following form: 1D-Convolution followed by batch normalization, a ReLU activation and spatial dropout. The optimization was performed using Adam[37] as well with a learning rate of 0.0001 and a gradient clipping norm of 0.5, trained with a batch size of 16.

Loss adaptation. To overcome the problem of strong data imbalance (most frames are annotated as 'none', i.e. no labeled behavior), we used a multi-class adaptation of the in object detection often used Focal loss[38] for action recognition, to discount the contribution of the background class to the overall loss:

$$L_{focal} = -\alpha(1 - p_t)^{\gamma} \log p_t$$

We used a gamma = 3.0 and an alpha = 0.5. For evaluation, we used the commonly used *F1* metric to assess multi-class classification performance, while using *Pearson Correlation* to assess temporal correlation.

SIPEC:PoseNet Network Architecture and training. In combination with SIPEC:SegNet we can perform top-down pose estimation with SIPEC:PoseNet. That means, instead of pose estimation network outputting for one landmark multiple possible outputs, corresponding to different animals, we can first segment different animals and then run SIPEC:PoseNet per animal on its cropped frame. In principle, every architecture can now be run on the cropped animal frame, including DLC[1]. We ship SIPEC with a SIPEC:PoseNet architecture that is based on a simple encoder-decoder design[29]. For processing target images for pose-regression, we convolved pose landmark locations in the image with a 2D Gaussian kernel. Since there were many frames with an incomplete number of labels, we defined a custom cross-entropy-based loss function, which was 0 for non-existing labels.

$$L_{incomplete} = \begin{cases} CrossEntropy \\ 0, if\ labels\ does\ not\ exist \end{cases}$$

Implementation and Hardware. For all neural network implementations, we used Tensorflow[39] and Keras[40]. Computations were done on either NVIDIA RTX 2080 Ti or V100 GPUs.

3D location labeling. To annotate the 3D location of a primate, we firstly create a precise model of the physical room (Supplementary Figure 9). For a given mask-cutout of a primate, we place an artificial primate at an approximate location in the 3D-model. We can then directly readout the 3D-position of the primate. 300 samples are annotated which altogether cover the most frequent parts of primate positions.

3D location estimation. To regress the animal position in 3D, we trained a manifold embedding using Isomap[30] using the mask size (normalized sum of positively classified pixels), the x and y pixel positions and their pairwise multiplications as features. We used the resulting 6 Isomap features, together with the inverse square root of the mask size, mask size and x-y-position in pixel space to train an ordinary least squares regression model to predict the 3D position of the animal.

Metrics used.

$$Pearson_{xy} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}}$$

$$RMSE = \sqrt{\frac{\sum_{n=1}^{N}(\hat{y}_n - y_n)^2}{N}}$$

$$precision = \frac{TP}{TP + FP}$$

$$recall = \frac{TP}{TP + FN}$$

Where TP denote True Positives, FP False Positives, TN True Negatives and FN False Negatives.

$$F1 = 2 \cdot \frac{precsion \cdot recall}{precision + recall}$$

$$IoU(M_{GT}, M_P) = \frac{M_{GT} \cap M_P}{M_{GT} \cup M_P}$$

Where M_{GT} denotes the ground truth mask and M_P the predicted one. We now calculate the mAP for detections with an IoU > 0.5 as follows:

$$mAP = \sum_{n=0} (\boldsymbol{r_{n+1}} - \boldsymbol{r_n})\, \boldsymbol{\rho_{interp}}(\boldsymbol{r_{n+1}})$$

With

$$\rho_{interp}(r_{n+1}) = \max_{\tilde{r}:\tilde{r} \geq r_{n+1}} \rho(\tilde{r})$$

Where $\rho(r)$ denotes precision measure at a given recall value.

Data Availability

Data available upon reasonable request.

Code Availability

We provide the code for SIPEC at: https://github.com/damaggu/SIPEC and the GUI for the identification of animals https://github.com/damaggu/idtracking_gui.

References

1. Mathis, A. *et al. DeepLabCut: markerless pose estimation of user-defined body parts with deep learning*. (2018).
2. Geuther, B. Q. *et al.* Robust mouse tracking in complex environments using neural networks. *Communications biology* **2**, 124 (2019).
3. Romero-Ferrero, F., Bergomi, M. G., Hinz, R. C., Heras, F. J. & de Polavieja, G. G. idtracker. ai: tracking all individuals in small or large collectives of unmarked animals. *Nature methods* **16**, 179 (2019).
4. Forys, B., Xiao, D., Gupta, P., Boyd, J. D. & Murphy, T. H. Real-time markerless video tracking of body parts in mice using deep neural networks. *bioRxiv* 482349 (2018).
5. Pereira, T. D. *et al.* Fast animal pose estimation using deep neural networks. *Nature methods* **16**, 117 (2019).
6. Graving, J. M. *et al.* DeepPoseKit, a software toolkit for fast and robust animal pose estimation using deep learning. *eLife* **8**, e47994 (2019).
7. Bala, P. C. *et al.* Automated markerless pose estimation in freely moving macaques with OpenMonkeyStudio. *Nature Communications* **11**, 4560 (2020).
8. Günel, S. *et al.* DeepFly3D, a deep learning-based approach for 3D limb and appendage tracking in tethered, adult Drosophila. *eLife* **8**, e48571 (2019).
9. Wiltschko, A. B. *et al.* Mapping sub-second structure in mouse behavior. *Neuron* **88**, 1121–1135 (2015).
10. Hsu, A. I. & Yttri, E. A. *B-SOiD: An Open Source Unsupervised Algorithm for Discovery of Spontaneous Behaviors*. http://biorxiv.org/lookup/doi/10.1101/770271 (2019) doi:10.1101/770271.
11. Nilsson, S. R. *et al. Simple Behavioral Analysis (SimBA) – an open source toolkit for computer classification of complex social behaviors in experimental animals*.

http://biorxiv.org/lookup/doi/10.1101/2020.04.19.049452 (2020) doi:10.1101/2020.04.19.049452.

12. Segalin, C. *et al. The Mouse Action Recognition System (MARS): a software pipeline for automated analysis of social behaviors in mice*. http://biorxiv.org/lookup/doi/10.1101/2020.07.26.222299 (2020) doi:10.1101/2020.07.26.222299.

13. Sturman, O. *et al.* Deep learning-based behavioral analysis reaches human accuracy and is capable of outperforming commercial solutions. *Neuropsychopharmacology* 1–13 (2020) doi:10.1038/s41386-020-0776-y.

14. Nourizonoz, A. *et al.* EthoLoop: automated closed-loop neuroethology in naturalistic environments. *Nature Methods* **17**, 1052–1059 (2020).

15. Jung, A. B. *et al. imgaug*. (2020).

16. Yosinski, J., Clune, J., Bengio, Y. & Lipson, H. How transferable are features in deep neural networks? in *Advances in neural information processing systems* 3320–3328 (2014).

17. He, K., Gkioxari, G., Dollár, P. & Girshick, R. Mask r-cnn. in *Proceedings of the IEEE international conference on computer vision* 2961–2969 (2017).

18. He, K., Zhang, X., Ren, S. & Sun, J. Deep residual learning for image recognition. in *Proceedings of the IEEE conference on computer vision and pattern recognition* 770–778 (2016).

19. Lin, T.-Y. *et al.* Microsoft coco: Common objects in context. in *European conference on computer vision* 740–755 (Springer, 2014).

20. Dutta, A. & Zisserman, A. The VIA Annotation Software for Images, Audio and Video. in *Proceedings of the 27th ACM International Conference on Multimedia* (ACM, 2019). doi:10.1145/3343031.3350535.

21. Huang, G., Liu, Z., Van Der Maaten, L. & Weinberger, K. Q. Densely connected convolutional networks. in *Proceedings of the IEEE conference on computer vision and pattern recognition* 4700–4708 (2017).

22. Krizhevsky, A., Sutskever, I. & Hinton, G. E. Imagenet classification with deep convolutional neural networks. in *Advances in neural information processing systems* 1097–1105 (2012).

23. Cho, K. *et al.* Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. *arXiv:1406.1078 [cs, stat]* (2014).

24. Chung, J., Gulcehre, C., Cho, K. & Bengio, Y. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. *arXiv:1412.3555 [cs]* (2014).

25. Chollet, F. Xception: Deep learning with depthwise separable convolutions. in *Proceedings of the IEEE conference on computer vision and pattern recognition* 1251–1258 (2017).

26. Oord, A. van den *et al.* Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499* (2016).

27. Bai, S., Kolter, J. Z. & Koltun, V. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv preprint arXiv:1803.01271* (2018).

28. Demšar, J. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine learning research* **7**, 1–30 (2006).

29. Xiao, B., Wu, H. & Wei, Y. Simple Baselines for Human Pose Estimation and Tracking. *arXiv:1804.06208 [cs]* (2018).

30. Tenenbaum, J. B. A Global Geometric Framework for Nonlinear Dimensionality Reduction. *Science* **290**, 2319–2323 (2000).

31. Lin, T.-Y. *et al.* Feature Pyramid Networks for Object Detection. in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* 936–944 (IEEE, 2017). doi:10.1109/CVPR.2017.106.

32. Girshick, R. Fast R-CNN. in *2015 IEEE International Conference on Computer Vision (ICCV)* 1440–1448 (2015). doi:10.1109/ICCV.2015.169.

33. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. & Salakhutdinov, R. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. 30.

34. Ioffe, S. & Szegedy, C. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. *arXiv:1502.03167 [cs]* (2015).

35. Maas, A. L., Hannun, A. Y. & Ng, A. Y. Rectifier Nonlinearities Improve Neural Network Acoustic Models. 6.

36. Xu, B., Wang, N., Chen, T. & Li, M. Empirical Evaluation of Rectified Activations in Convolutional Network. *arXiv:1505.00853 [cs, stat]* (2015).

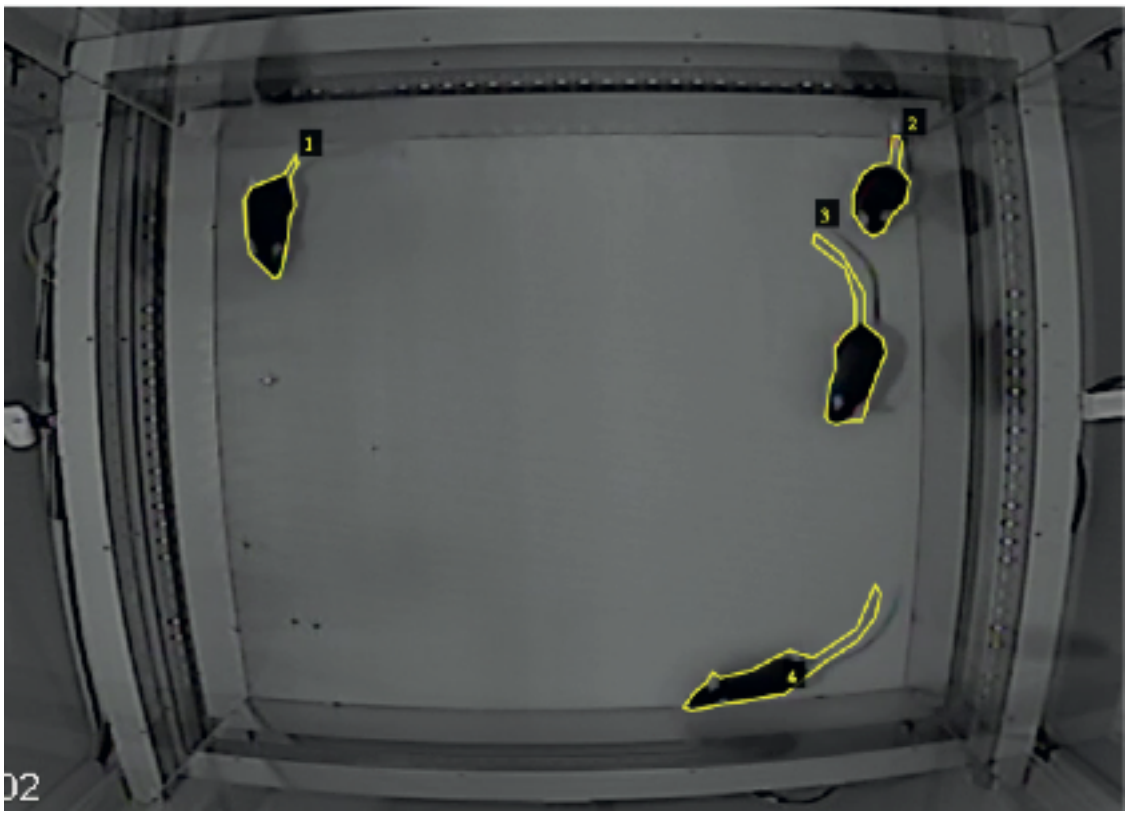
37. Kingma, D. P. & Ba, J. Adam: A Method for Stochastic Optimization. *arXiv:1412.6980 [cs]* (2017).

38. Lin, T.-Y., Goyal, P., Girshick, R., He, K. & Dollár, P. Focal Loss for Dense Object Detection. *arXiv:1708.02002 [cs]* (2018).

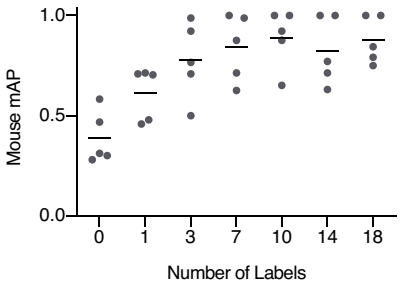
39. Abadi, M. *et al.* TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems. 19.

40. Chollet, F. *Keras*. (2015).

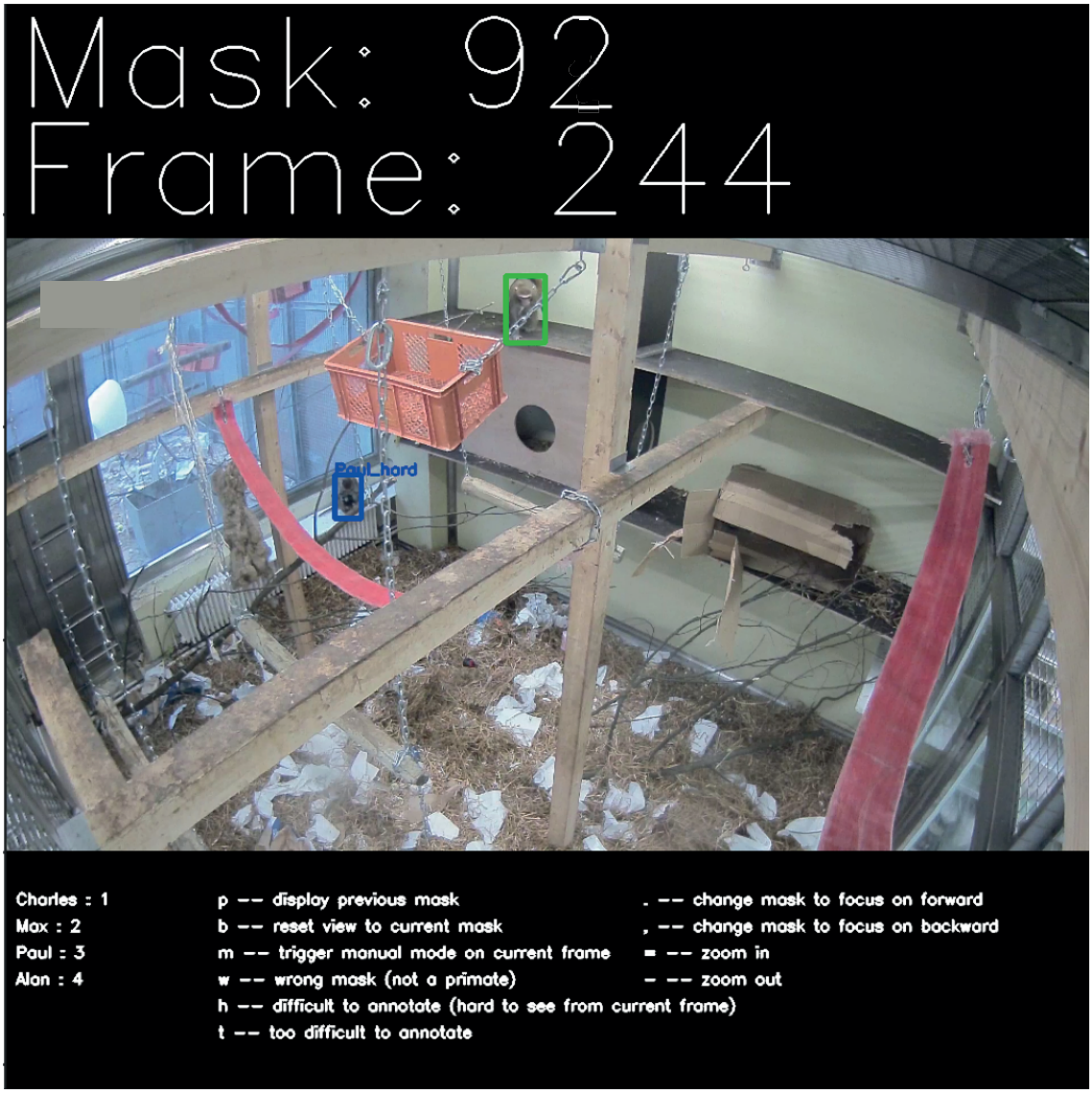
Supplementary



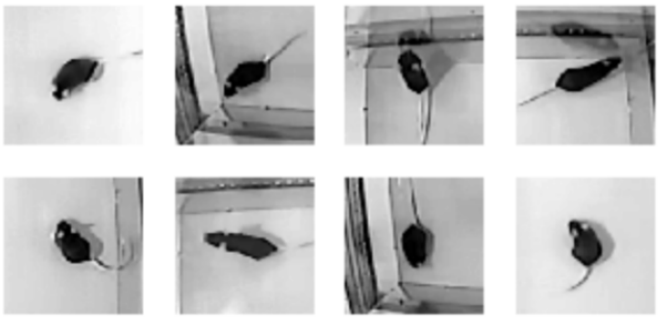
Supplementary Figure 1 | Segmentation annotation illustration. Examplary frame of mice in OFT with manually annotated outlines.



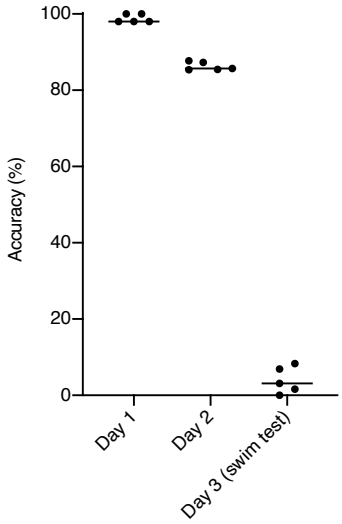
Supplementary Figure 2 | Mouse 4plex segmentation. Number of 4plex labels needed to retrain single-mouse model for recovering mAP. 0 labels stands for the model trained on a single animal. All data is represented mean, showing all points.



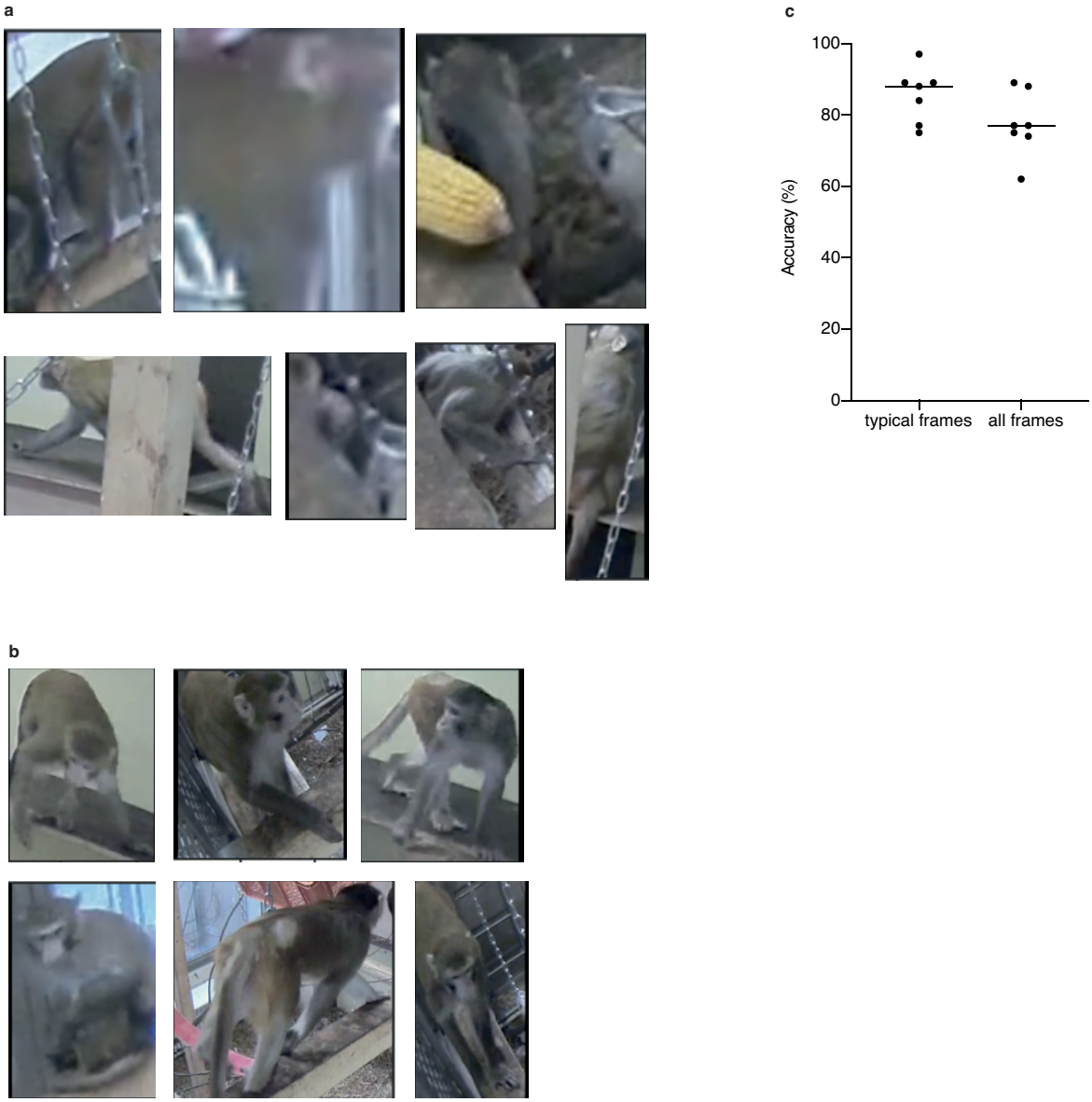
Supplementary Figure 3 | Identification Graphical User Interface. Mask-box results from SIPEC:SegNet is overlaid over frames in blue and can be labeled one by one. The current box to be labeled is in green. A simple keyboard input scheme is provided within the GUI. Names of individuals and the number of masks to be labeled can be set by the user.



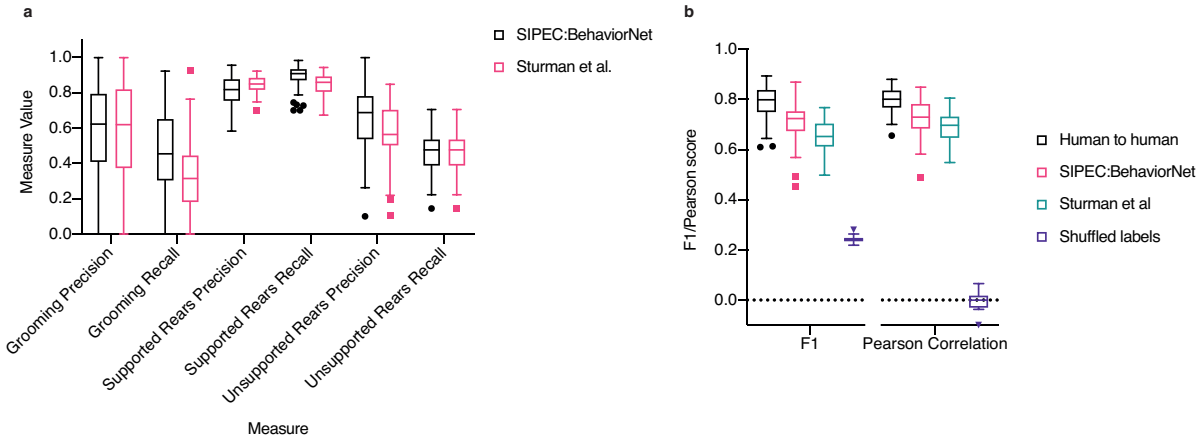
Supplementary Figure 4 | Example frames of the 8 distinct mice.



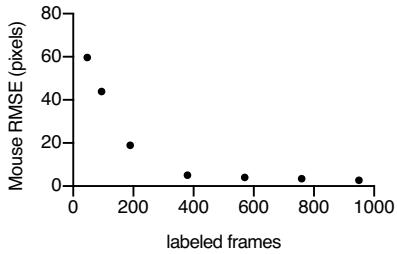
Supplementary Fig. 5 | Identification performance of mice across days and interventions. Identification accuracy across days for models trained on day 1. While the performance for the day the model is trained on is very high it drops when tested on day 2, but is still significantly above chance level. When tested on day 3, after a forced swim test intervention, the performance drops significantly. All data is represented mean, showing all points.



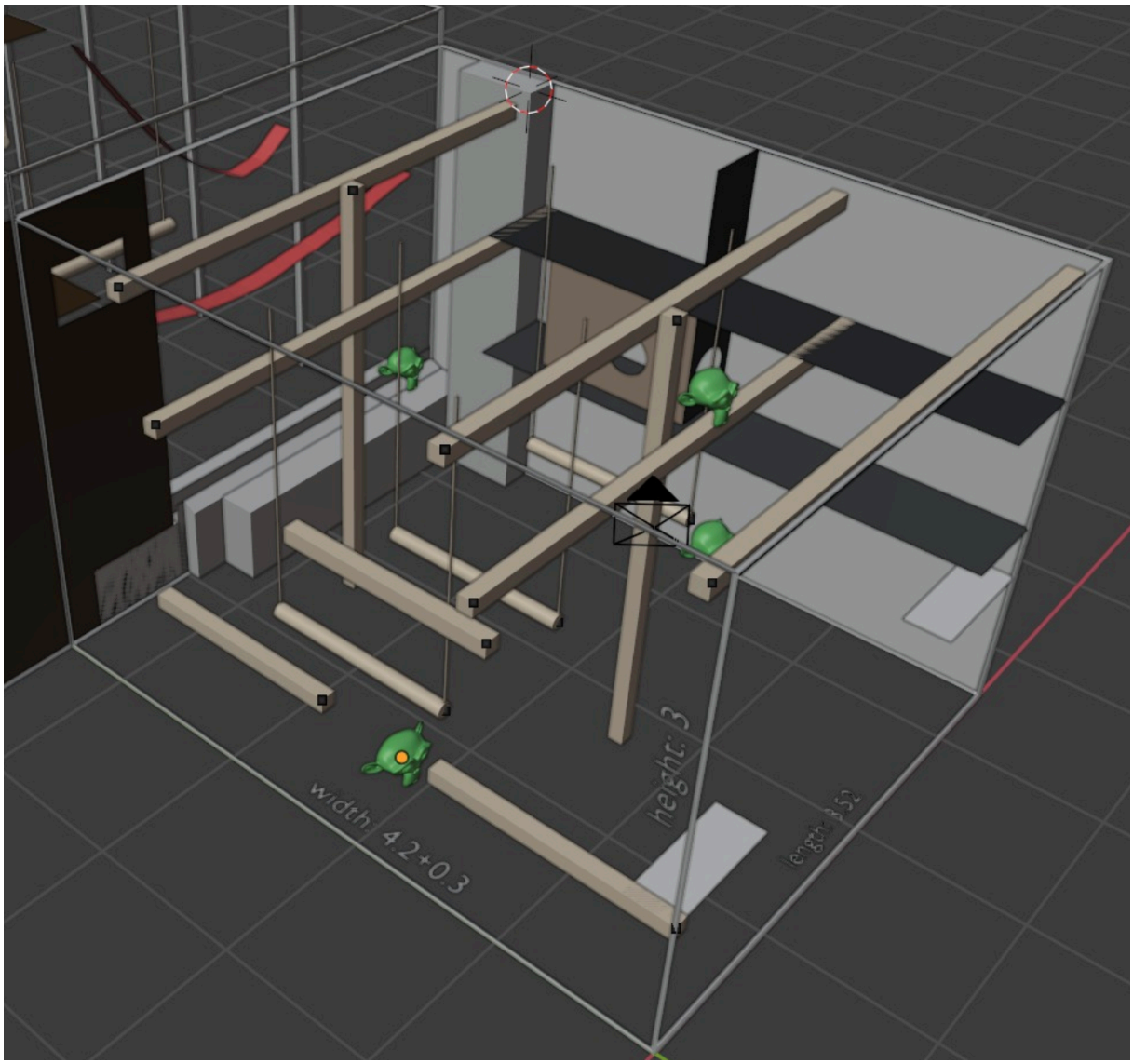
Supplementary Figure 6 | Identification of typical vs difficult frames. a) Displayed are very difficult exemplary frames, which are also beyond human single-frame recognition, that are excluded for the 'typical' frame evaluation. b) Exemplary frames are shown, used for the 'typical' frames analysis. c) Identification performance is significantly higher on 'typical' frames than on all frames. All data is represented mean, showing all points.



Supplementary Figure 7 | Additional behavioral evaluation. a) Overall increased F1 score is caused by an increased recall in case of grooming events and precision for unsupported rearing events. b) Comparison of F1 values as well as Pearson Correlation of SIPEC:BehaveNet to human-to-human performance. All data is represented by a Tukey box-and-whisker plot, showing all points.



Supplementary Figure 8 | Pose estimation. Pose estimation performance of SIPEC:PoseNet as a function of labeled frames for estimating the location of 13 standardized body parts on a video frame containing a single mouse in OFT.



Supplementary Figure 9 | 3D model used for annotation of primate 3D-location data.