nature machine intelligence

ARTICLES

https://doi.org/10.1038/s42256-022-00477-5

Check for updates

Deep-learning-based identification, tracking, pose estimation and behaviour classification of interacting primates and mice in complex environments

Markus Marks[1,2], Qiuhan Jin[3], Oliver Sturman[4,2], Lukas von Ziegler[4,2], Sepp Kollmorgen[1,2], Wolfger von der Behrens[1,2], Valerio Mante[1,2], Johannes Bohacek[4,2] and Mehmet Fatih Yanik[1,2] ✉

Quantification of behaviours of interest from video data is commonly used to study brain function, the effects of pharmacological interventions, and genetic alterations. Existing approaches lack the capability to analyse the behaviour of groups of animals in complex environments. We present a novel deep learning architecture for classifying individual and social animal behaviour—even in complex environments directly from raw video frames—that requires no intervention after initial human supervision. Our behavioural classifier is embedded in a pipeline (SIPEC) that performs segmentation, identification, pose-estimation and classification of complex behaviour, outperforming the state of the art. SIPEC successfully recognizes multiple behaviours of freely moving individual mice as well as socially interacting non-human primates in three dimensions, using data only from simple mono-vision cameras in home-cage set-ups.

Although the analysis of animal behaviour is crucial for systems neuroscience[1] and preclinical assessment of therapies, it remains a highly labourious and error-prone process. Over the past few years there has been a surge in machine learning tools for behavioural analysis, including segmentation, identification and pose estimation[2–11]. Despite being an impressive feat for the field, a key element—the direct recognition of behaviour itself—has been rarely addressed. Unsupervised analysis of behaviour[12–17] can be a powerful tool to capture the diversity of the underlying behavioural patterns, but the results of these methods do not align with human annotations and therefore require inspection[15]. There have also been advances in the supervised analysis of mouse behaviour, using classifiers on top of pose-estimation-generated features[18–21], or manually defined features such as ellipses[22–25]. Sturman and colleagues[20] demonstrated that the classification of mouse behaviours using features generated from pose-estimation algorithms can outperform the behavioural classification performance of commercial systems. Yet, such pose-estimation-based behaviour classification remains a labour-intensive and error-prone process as we show below. Moreover, pose estimation in primates is difficult to achieve with current methods[26].

Here we demonstrate a complementary approach for researchers who automatically seek to identify behaviours of interest. Our approach relies on the initial annotation of example behaviours, that is, snippets of video footage. These snippets are subsequently used to train a deep neural network (DNN) to subsequently recognize such particular behaviours in arbitrarily long videos and complex environments. To achieve this, we designed a novel DNN architecture called SIPEC:BehaveNet, which uses raw video frames as input and substantially outperforms a pose-estimation-based approach tested on a well-annotated mouse dataset and reaches human-level performances for counting grouped behavioural events. In addition to this behavioural classification network, we developed an all-inclusive pipeline called SIPEC, with modules for segmentation (SIPEC:SegNet), identification (SIPEC:IdNet), behavioural classification (SIPEC:BehaveNet) and pose estimation (SIPEC:PoseNet) of multiple and interacting animals in complex environments. These four DNNs operate directly on videos and are developed and optimized for analysing animal behaviour and providing state-of-the-art performance. We use this pipeline to classify, for the first time, social interactions in home-caged primates from raw video frames and without needing to use any pose estimation.

SIPEC:SegNet is a Mask R-CNN architecture[27], optimized to robustly segment animals despite occlusions, multiple scales and rapid movement, and enable tracking of animal identities within a session. SIPEC:IdNet has a DenseNet[28] backbone that yields visual features that are integrated over time through a gated-recurrent-unit network[29,30] to reidentify animals when temporal-continuity-based tracking does not work, for example when animals enter or exit a scene. This enables SIPEC to identify primates across weeks and to outperform the identification module of idtracker.ai[4] both within and across sessions (see the "Discussion" section), as well as PrimNet[31]. SIPEC:PoseNet performs top-down multi-animal pose estimation, which we compared with DeepLabCut (DLC)[2], another pose-estimation-based approach. SIPEC:BehaveNet uses an Xception[32] network in combination with a temporal convolution network (TCN)[33,34] to classify behavioural events directly from raw pixels. We use image augmentation[35] and transfer learning[36], optimized specifically for each task, to rapidly train our modules. SIPEC enables researchers to identify behaviours of multiple

[1]Institute of Neuroinformatics ETH Zürich and University of Zürich, Zurich, Switzerland. [2]Neuroscience Center Zurich, ETH Zürich and University of Zürich, Zurich, Switzerland. [3]Laboratory for Neuro- and Psychophysiology, Department of Neurosciences, KU Leuven, Leuven, Belgium. [4]Laboratory of Molecular and Behavioral Neuroscience, Institute for Neuroscience, Department of Health Sciences and Technology, ETH Zurich, Zurich, Switzerland.
✉e-mail: yanik@ethz.ch

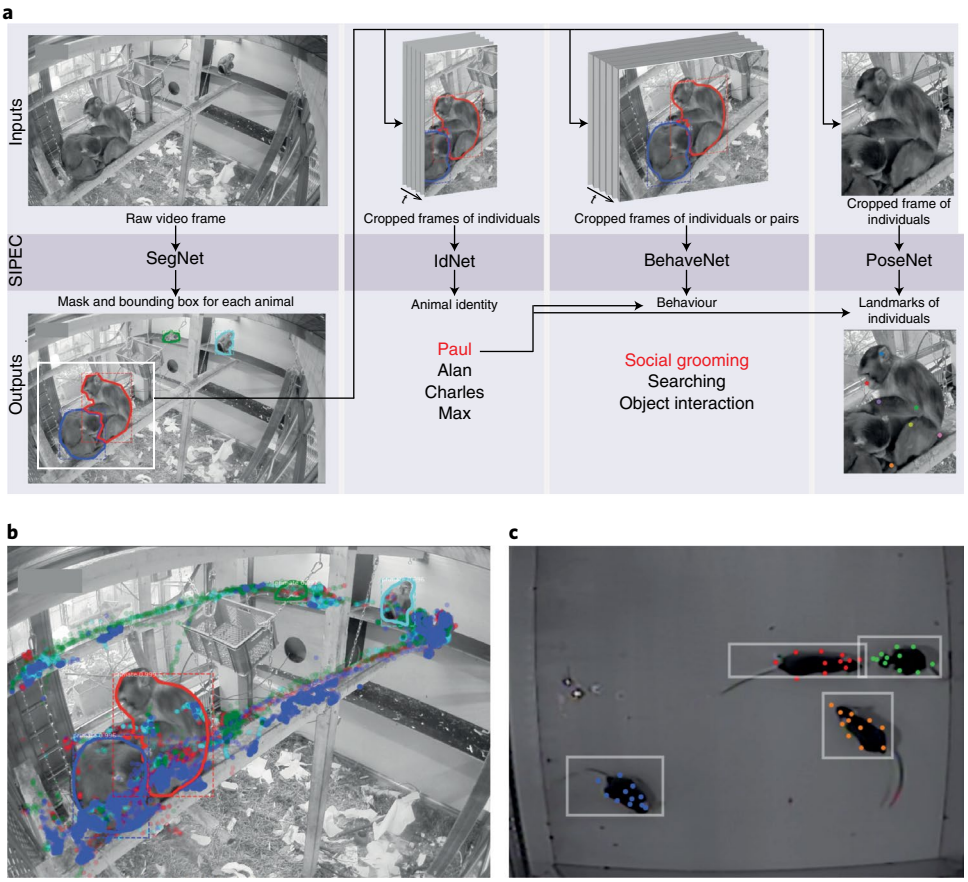


Fig. 1 | Overview of the SIPEC workflow and modules. a, From a given video, instances of animals are segmented with SIPEC:SegNet and indicated by masked outlines as well as bounding boxes. Individuals are then identified using SIPEC:IdNet. The pose and behaviour for each individual can be estimated/classified using SIPEC:PoseNet and SIPEC:BehaveNet, respectively. **b**, The outcome of SIPEC:SegNet is shown and SIPEC:IdNet modules are overlaid on a representative video frame. Time-lapsed positions of individual primates (centre of mass, COM) are plotted as circles with respective colours. **c**, Outputs of SIPEC:SegNet (boxes) and SIPEC:PoseNet (coloured dots) on a representative videoframe of mouse open-field data.

animals in complex and changing environments over multiple days or weeks in three-dimensional space, even from a single camera with relatively little labelling, by contrast to other approaches that use heavily equipped environments and large amounts of labelled data[8].

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 analysing 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), behavioural classification (SIPEC:BehaveNet) and, finally, pose estimation (SIPEC:PoseNet) from video frames (Fig. 1). These four artificial neural networks, trained for different purposes, could be used individually or combined in different ways (Fig. 1a). To illustrate the utility of this feature, Fig. 1b shows the output of pipelining SIPEC:SegNet and SIPEC:IdNet to track the identity and location of four primates housed together (Fig. 1b and Supplementary Video 1). Figure 1c shows the output of pipelining SIPEC:SegNet and SIPEC:PoseNet to do multi-animal pose estimation in a group of four mice.

Segmentation module SIPEC:SegNet. SIPEC:SegNet (see Methods and Supplementary Fig. 8) is based on the Mask R-CNN architecture[27], which we optimized for analysing multiple animals and integrated into SIPEC. We further applied transfer learning[36] onto the weights of the Mask R-CNN ResNet-backbone[37] pre-trained on the Microsoft Common Objects in Context (COCO) dataset[38] (see Methods for SIPEC:SegNet architecture and training). Moreover, we applied image augmentation[35] to increase network robustness against invariances (for example, rotational invariance) and therefore increase generalizability.

Segmentation performance on individual mice and groups of four. We first examined the performance of SIPEC:SegNet on top-view video recordings of individual mice, behaving in an open-field test (OFT). Although segmenting black mice on a blank background could be achieved by thresholding alone, we still included this task for completeness. Eight mice were freely behaving for 10 min in an OFT arena of TSE Systems' Multi Conditioning System, as previously described in a work by Sturman and co-workers[20]. We labelled the outlines of mice in a total of 23 frames using the VGG image annotator[39] from videos of randomly selected mice. We used fivefold cross-validation to evaluate the performance. 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 determine how many labelled frames (outlines of the animal, see Supplementary Fig. 1) are needed for SIPEC:SegNet to reach peak performance (Extended Data Fig. 1). We measured performance using cross-validation by randomly selecting an increasing amount of training frames. For single-mouse videos, we find that our model achieves 95% of its

mean peak performance (MAP of 0.95 ± 0.05) using as few as a total of three labelled frames for training. For segmentation in groups of four mice, we added 57 labelled 4-plex frames to the existing 23 labelled single-mouse frames, making a total of 80 labelled frames. Evaluated on a fivefold cross-validation, SIPEC:SegNet achieves an MAP of 0.97 ± 0.03 (Fig. 2b). We performed an ablation study as well and found that SIPEC:SegNet achieves better than 95% of its mean peak performance (MAP of 0.94 ± 0.05) using as few as only 16 labelled frames. We also report IOU and dice coefficient metrics to assess the overlap between prediction and ground truth (Fig. 2b).

Segmentation performance of groups of primates. We annotated 191 frames from videos on different days (days 1, 9, 16 and 18) to test SIPEC:SegNet's ability to detect instances of primates within a group. As exemplified in Fig. 2a, the network even handles difficult scenarios very well: representative illustrations include the ground truth, as well as predictions of moments where multiple primates are moving rapidly while strongly occluded at varying distances from the camera. SIPEC:SegNet achieved a MAP of 0.91 ± 0.03 using fivefold cross-validation. When we performed the previously described ablation study, SIPEC:SegNet achieved a 95% of MAP of 0.87 ± 0.03 with only 30 labelled frames (Fig. 2b). We also report the intersection over union (IOU) and dice coefficient metrics to assess the overlap between prediction and ground truth (Fig. 2c).

Pose-estimation module SIPEC:PoseNet. We also added a pose-estimation network—built on an encoder–decoder architecture[40] with an EfficientNet[41] backbone—to SIPEC (SIPEC:PoseNet; see Methods and Supplementary Fig. 7). SIPEC:PoseNet can be used to perform pose estimation on N animals (where N is the total number of animals or less), yielding K different coordinates for previously defined landmarks on each animal's body. The main advantage of SIPEC:PoseNet over past approaches is that it receives its inputs from SIPEC:SegNet (top-down pose estimation). While bottom-up approaches such as DLC[2] 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 (Fig. 1c). Similarly to Sturman and colleagues[20], we labelled frames with 13 standardized body parts of individual mice in an OFT to train and test the performance of SIPEC:PoseNet against that of DLC[2]. SIPEC:PoseNet achieves a root-mean-square error (RMSE) (see Methods) of 2.9 pixels in mice (Fig. 2d) for a total of 96 labelled training frames, whereas DLC[2] achieves a 3.9-pixel RMSE. Past published post-estimation methods for single animals can easily be substituted into our pipeline to perform multi-animal pose estimation in conjunction with SIPEC:SegNet.

Identification module SIPEC:IdNet. SIPEC:IdNet (see Methods and Supplementary Fig. 6) allows the identity of individual animals to be determined. Given that SIPEC:IdNet receives input as a series (T time steps) 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[28]-backbone pre-trained on ImageNet[42]. 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 time step. The outputs of the feature-recognition network on these frames are then integrated over T time steps using a gated-recurrent-unit network (see Methods for architectural and training details). SIPEC:IdNet can integrate information from zero to many temporally neighbouring frames based on a particular application's accuracy and speed requirements. We used spatial area dropout augmentations to increase robustness against occlusions[43]. We developed an annotation tool for a human to assign identities of individual animals—in a multi-animal context—to segmentation masks in video frames capturing primates from different perspectives (Supplementary Fig. 2). This tool was used for annotating identification data, as described in the following sections. Below we compare the performance of SIPEC:IdNet with that of the current state of the art; those being PrimNet and the identification module of idTracker.ai for primate reidentification. The former relies on faces of individuals being clearly visible for reidentification, which in our case is not possible for most of the video frames, whereas the latter is a self-supervised algorithm for tracking the identity of individual animals within a single session, particularly in complex or enriched home-cage environments in which animals are frequently obstructed as they move underneath/behind objects or enter/exit the scene, and where background or lighting conditions change constantly—temporally based tracking and identification as idtracker.ai performs it becomes impossible. We evaluated the identification performance of SIPEC:IdNet across sessions with the identification module of idTracker.ai, providing each network with identical training and testing data. Although idtracker.ai behaves in a self-supervised manner, the identification module it uses to distinguish animals is trained with the labels generated by idTracker.ai's cascade algorithm in a supervised fashion. Apart from reidentifying animals across sessions using SIPEC:IdNet, SIPEC:SegNet segmentation masks can be used via greedy mask matching (see Methods) to track the identities of animals temporally as well (Supplementary Videos 2–4) or to smooth the outputs of SIPEC:IdNet as a secondary step, which can boost performance for continuous video sequences, but this advantage was not used in the following evaluations for mice and primates.

Identification of mice in an open-field test. We first evaluated the performance of SIPEC:IdNet in identifying eight individual mice. We acquired 10-min-long videos of these mice behaving in the previously mentioned OFT (see Methods for details). Although these mice are difficult to distinguish by human observers (Supplementary Fig. 3), our network copes well. We used fivefold cross-validation to evaluate the performance, that is, splitting the 10 min videos into 2-min-long ones, using one fold for testing and the rest to train the network. As these data are balanced, we use the accuracy metric for evaluation. We find that SIPEC:IdNet achieves an accuracy of 99 ± 0.5%, whereas idTracker.ai only achieves 87 ± 0.2% (Fig. 2e). The ablation study shows that only 650 labelled frames (the frame and the identity of the animal) are sufficient for the SIPEC:IdNet to achieve 95% of its mean peak performance (Fig. 2f). We tested how this performance translates into identifying the same animals during the following days (Extended Data Fig. 2) and found that identification performance is similarly high on day 2 (86 ± 2%) when using the network trained on day 1. We subsequently tested identification robustness with respect to the interventions on day 3. Following a forced-swimming test, the identification performance of SIPEC:IdNet trained on data from day 1 dropped dramatically to 4 ± 2%. This indicates that features utilized by the network to identify the mice are not robust to this type of intervention, that is, their behaviour and outlook is greatly altered by the stress and residual water on the fur.

Identification of individual primates in a group. We used the SIPEC:SegNet-processed videos of the four macaques to evaluate the performance of SIPEC:IdNet in identifying individual primates within a group (see the 'Segmentation performance of groups of primates' section). We annotated frames from seven videos taken on different days (with each frame containing multiple individuals), yielding approximately 2,200 labels for cut-outs of individual primates. We used leave-one-out cross-validation with respect to

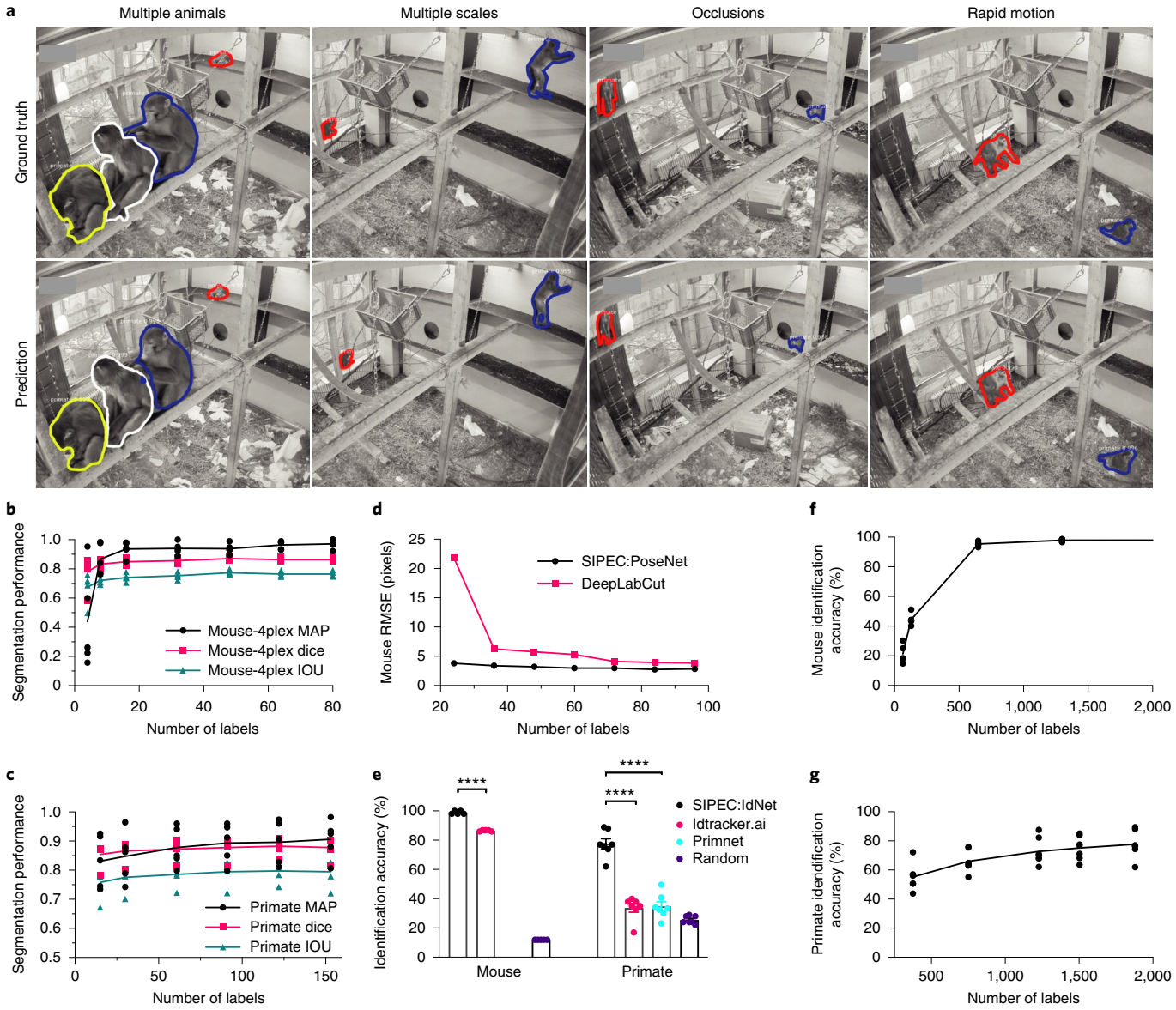


Fig. 2 | Performance of SIPEC:SegNet, SIPEC:PoseNet and SIPEC:IdNet under demanding video conditions while using few labels. a, Qualitative comparison of ground truth versus predicted segmentation masks under various challenging conditions. **b**, SIPEC:SegNet performance in MAP, dice and IOU is shown for mice as a function of the number of labels. The lines indicate the means for fivefold cross-validation. **c**, SIPEC:SegNet performance in MAP, dice and IOU is shown for primates as a function of the number of labels. **d**, The performance of SIPEC:PoseNet in comparison with DLC measured as the RMSE in pixels on single mouse pose-estimation data. **e**, Comparison of identification accuracies for the SIPEC:IdNet module, idtracker.ai, PrimNet and randomly shuffled labels (chance performance). Eight videos from eight individual mice and seven videos across four different days from four group-housed primates are used. **f**, The accuracy of SIPEC:IdNet for mice as a function of the number of training labels used. **g**, The accuracy of SIPEC:IdNet for primates as a function of the number of training labels used. The black lines in **f** and **g** indicate the mean for fivefold cross-validation with individual folds displayed. All data are represented by the mean, showing all points. Wilcoxon paired test: ****$P \leq 0.0001$.

the videos to test SIPEC:IdNet generalization across days. Across sessions, where the human expert (the ground truth) has the advantage of seeing all of the video frames and the entire cage (that is, the rest of the primates), SIPEC:IdNet has an accuracy of up to 78 ± 3%, whereas idTracker.ai and PrimNet only achieve 33 ± 3% and 34 ± 3%, respectively (Fig. 2e). We performed a separate evaluation of the identification performance on typical frames, that is, the human expert can correctly identify the primates using single frames. In this case, SIPEC:IdNet achieved a performance of 86 ± 3 (Extended Data Fig. 3). The identification labels can then be further enhanced by greedy mask-matching-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 its mean peak performance at 1,504 annotated training samples (Fig. 2g).

Behavioural classification module SIPEC:BehaveNet. SIPEC:BehaveNet (see Methods and Supplementary Fig. 9) offers researchers a powerful means to recognize specific animal behaviours directly from raw pixels using a single neuronal net framework. SIPEC:BehaveNet uses video frames of N individuals over T time steps to classify the animals' actions. The video frames of the N individuals are generated by SIPEC:SegNet. If only a single animal is present in the video, SIPEC:BehaveNet can be used directly

without SIPEC:SegNet. We use a recognition network to extract features from single frames analysis, based on the Xception network architecture[32]. We initialize parts of the network with ImageNet[4] weights. These features are then integrated over time by a TCN[33,34] to classify the animal's behaviour in each frame (see Methods for architectural and training details).

SIPEC behaviour recognition outperforms DLC-based approach. We compare our raw-pixel-based approach with Sturman and colleagues[20], who recently demonstrated that they can classify behaviour based on DLC-generated[2] features. In addition to a higher classification performance with fewer labels, SIPEC:BehaveNet does not require annotation and training for pose estimation if the researcher is interested in behavioural classification alone. The increased performance with fewer labels comes at the cost of a higher computational demand because we increased the dimensionality of the input data by several orders of magnitude (12 pose estimates versus 16,384 pixels). We used the data and labels from Sturman and co-workers[20] on 20 freely behaving mice in an OFT to test our performance. The behaviour of these mice was independently annotated by three different researchers on a frame-by-frame basis using the VGG video annotation tool[39]. Annotations included the following behaviours: supported rears, unsupported rears, grooming and none (unlabelled/default class). Although Sturman and colleagues[20] evaluated the performance of their behavioural event detection by averaging across chunks of time, evaluating the frame-by-frame performance is more suitable for testing the actual network performance as it was trained the same way. Such frame-by-frame analysis shows that SIPEC:BehaveNet has fewer false positives and false negatives with respect to the DLC-based approach of Sturman and co-workers[20]. We illustrate a representative example of the performance of both approaches for each of the behaviours with their respective ground truths (Fig. 3a). We further spatially resolved the events that were misclassified by Sturman et al. but correctly classified by SIPEC:BehaveNet, and vice versa (Fig. 3b). We calculated the percentage of mismatches that occurred in the centre or the surrounding area. Grooming-event mismatches in Sturman et al.[20] and SIPEC:BehaveNet occur similarly often in the centre ($41 \pm 12\%$ and $42 \pm 12\%$, respectively). Sturman et al.[20] has more mismatches occurring in the centre than SIPEC:BehaveNet for supported and unsupported rearing events (supported rears: $40 \pm 4\%$ and $37 \pm 6\%$, respectively; unsupported rears: $12 \pm 2\%$ and $7 \pm 2\%$, respectively). This indicates that the misclassifications of the pose-estimation-based approach are more biased towards the centre than those of SIPEC:BehavNet. We used leave-one-out cross-validation to quantify the behavioural classification over the whole time course of all of the videos of the 20 mice (Fig. 3c). We used the macro-averaged F1 score as a common metric to evaluate a multiclass classification task, and the 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 behaviours, SIPEC:BehaveNet achieves F1 scores of $0.6 \pm 0.16/0.49 \pm 0.21/0.84 \pm 0.04$, respectively, whereas the performance of the manually intensive approach by Sturman and colleagues[20] reaches only $0.49 \pm 0.11/0.37 \pm 0.2/0.84 \pm 0.03$, respectively, leading to a much higher performance of SIPEC:BehaveNet for the unsupported rearing (F1: $P = 1.689 \times 10^{-7}$, the Wilcoxon paired test was used as recommended[44]) as well as the grooming (F1: $P = 6.226 \times 10^{-4}$) behaviours. Although we see a higher precision only in the classification of supported rears in the DLC-based approach, SIPEC:BehaveNet has an improved recall for the supported rears as well as improved precision and recall for the other behaviours (Extended Data Fig. 4a). As expected, more stereotyped behaviours with many labels, such as supported rears, yield higher F1 scores. By comparison, less stereotypical behaviours, such as grooming with fewer labels, have lower F1 scores for SIPEC:BehaveNet and the DLC-based approach. We also 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 behavioural classification performance (Extended Data Fig. 4b). Although the overall human-to-human F1 score is 0.79 ± 0.07, SIPEC:BehaveNet classifies with an F1 score of 0.71 ± 0.07. We then grouped behaviours by integrating the classification over multiple frames as described in Sturman and colleagues[20]. This analysis results in a behaviour count per video. For these per video behaviour counts, we found no significant difference between human annotators, SIPEC:BehaveNet and Sturman and co-workers[20] (Tukey's multiple comparison test; Extended Data Fig. 6). Such classification and counting of specific behaviours per video are commonly used to compare the number of occurrences of behaviours across experimental groups. Using such analysis, Sturman et al.[20] demonstrate how video-based analysis outperforms commonly used commercial systems. Moreover, we also tested combining the outputs of pose-estimation-based classification together with the raw-pixel model (see the "Combined Model" section in the Methods and Extended Data Fig. 4). Finally, we performed a frame-ablation study and showed that SIPEC:BehaveNet needs only 114 min of labelled data to reach peak performance in behaviour classification (Fig. 3d).

Socially interacting primate behaviour classification. We used the combined outputs of SIPEC:SegNet and SIPEC:IdNet—smoothed by greedy-matching-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. An interaction event was detected whenever the masks of individual primates came sufficiently close (see Methods). We were able to rapidly annotate these videos again using the VGG video annotation tool[39] (overall 80 min of video is annotated from three videos, including the individual behaviours of object interaction, searching, social grooming and none (background class)). We then trained SIPEC:BehaveNet to classify individuals' frames and merged frames of pairs of primates socially interacting over time. We used grouped fivefold stratified cross-validation over all annotated video frames, with labelled videos being the groups. Overall, SIPEC:BehaveNet achieved a macro F1 score of 0.72 ± 0.07 across all behaviours (Fig. 4a), which is similar to the earlier mentioned mouse behavioural classification performance. The increased variance compared with the classification of mouse behaviour is expected as imaging conditions, as previously mentioned, are much more challenging and primate behaviours are much less stereotyped than mouse behaviours. This can probably be compensated with more training data.

Tracking position of primates in three-dimensional without stereo vision. By performing SIPEC:SegNet and SIPEC:IdNet inference on a full 1 h video, we built a density map of positions of individuals within the husbandry (Fig. 1a). Without stereo vision, one cannot optically acquire depth information. We instead used the output masks of SIPEC:SegNet and annotated the positions of the primates in 300 frames using a three-dimensional model (Supplementary Fig. 4). We subsequently generated six features using Isomap[45] and trained a multivariate linear regression model to predict the three-dimensional positions of the primates (Fig. 4b). Using tenfold cross-validation, our predicted positions using only a single camera have an overall RMSE of only 0.43 ± 0.01 m, that is, 0.27 ± 0.01 m in the x-direction or 6% error with regards to the room dimension in the x-direction; and 0.26 ± 0.01 m (7%) and 0.21 ± 0.01 m (7%) for the y- and z-coordinates, respectively. If an annotation is impossible, quasi depth estimates can be calculated through the mask size alone and correlate highly with the actual depth (Extended Data Fig. 5).

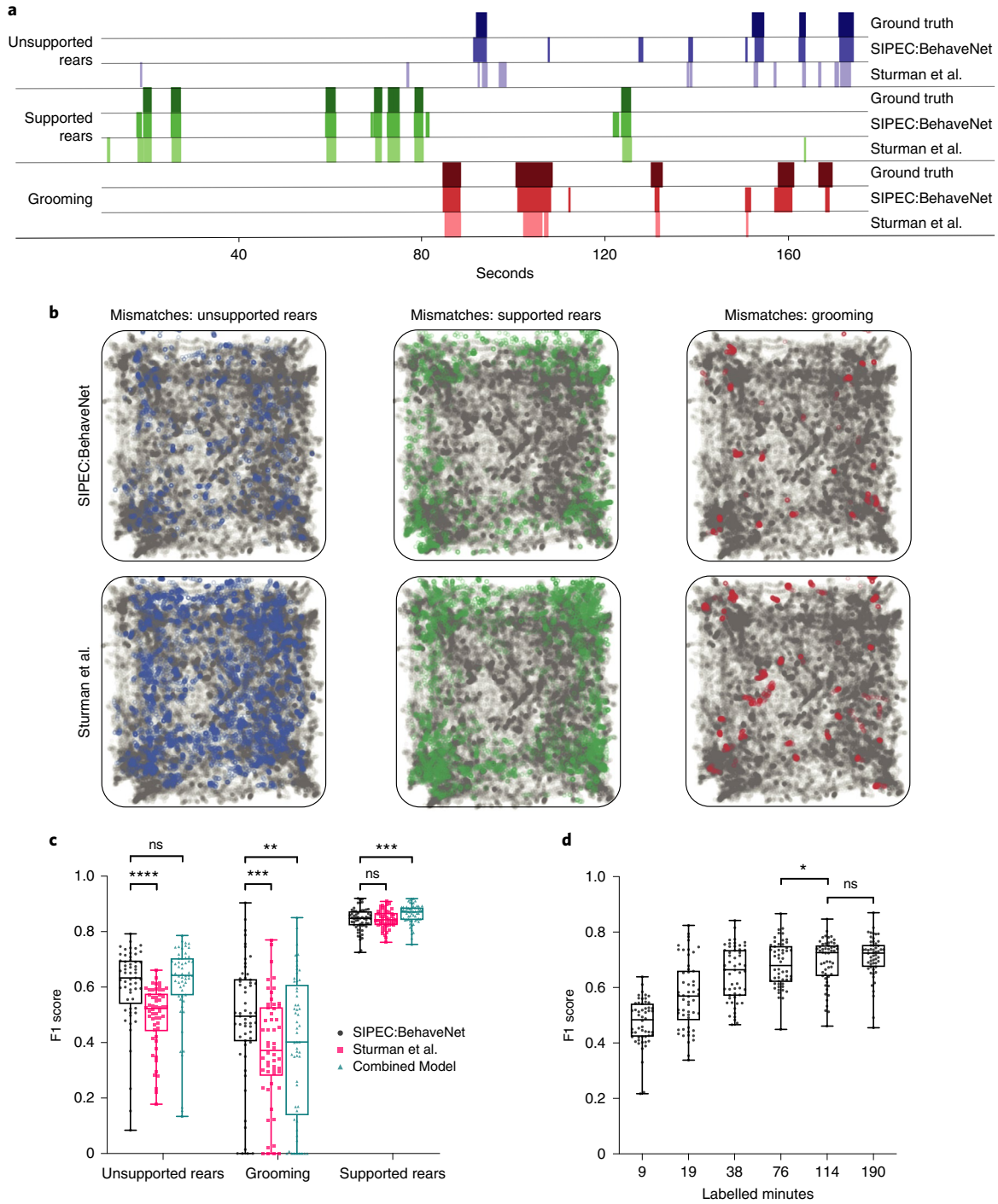


Fig. 3 | SIPEC:BehaveNet outperforms DLC. a, Comparison of behavioural classification by human annotator (ground truth), SIPEC:BehaveNet and Sturman and colleagues[20]. **b**, Errors in the classification of mouse behaviour in the open arena for SIPEC:BehaveNet versus Sturman and co-workers. Each coloured dot represents a behavioural 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 by grey dots. **c**, A frame-by-frame classification performance per video ($n = 20$ mice) compared to ground truth. **d**, SIPEC:BehaveNet classification performance as a function of labelled minutes. All data are represented by a minimum-to-maximum box-and-whisker plot, showing all points. Wilcoxon paired test: $^{*}P \leq 0.05$; $^{**}P \leq 0.01$; $^{***}P \leq 0.001$; $^{****}P \leq 0.0001$.

Discussion

We have presented SIPEC—a novel pipeline—using specialized deep neural networks to perform segmentation, identification, behavioural classification and pose estimation on individual and interacting animals. With SIPEC we address multiple key challenges in the domain of behavioural analysis. Our SIPEC:SegNet enables the segmentation of animals with only 3–30 labels (Fig. 2a–c). In combination with greedy mask matching, SIPEC:SegNet can be used to track animals' identities within one session, similar to idtracker.ai, but even in complex environments with changing lighting conditions, in which idtracker.ai fails (Supplementary Video 1).

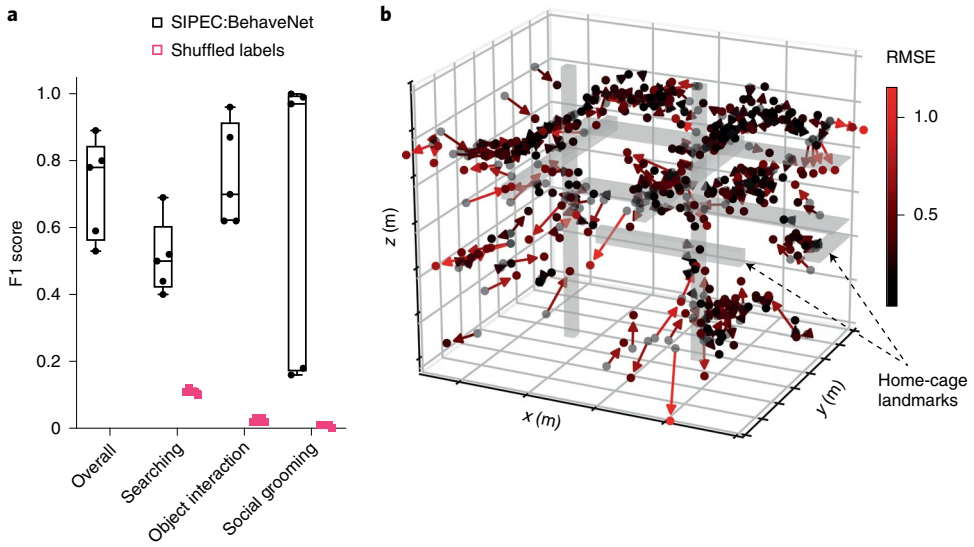


Fig. 4 | SIPEC can recognize social interactions of multiple primates and infer their three-dimensional positions using a single camera. a, Performance of SIPEC:BehaveNet for individual and social behaviours with respect to ground truth evaluated using grouped fivefold cross-validation. Behaviours include searching, object interaction and social grooming, whereas the performance is measured using F1 (which is also included for shuffled labels for comparison). All data is represented by a minimum-to-maximum box-and-whisker plot, showing all points. **b**, Evaluation of three-dimensional position estimates of primates in home-cage. Black spots mark annotated positions ($n = 300$) while predicted positions are marked as red-hued spots at the end of the solid arrows (colour-coded using a red gradient with brighter red indicating higher RMSE of predicted to true position).

SIPEC:BehaveNet enables animal behaviour recognition directly from raw video data. Raw video classification has the advantage of not requiring pre-processing adjustments or feature engineering to specific video conditions. Moreover, we show that learning task-relevant features directly from the raw video can lead to better results than pose-estimation-based approaches which train a classifier on top of the detected landmarks. In particular, we demonstrate that our network outperforms a state-of-the-art pose-estimation-based approach[13] on a well-annotated mouse behavioural dataset (Fig. 3) and reaches human-level performance for counting behavioural events (Extended Data Fig. 6). Pose-estimation can thus be skipped if researchers are solely interested in classifying behaviour. We note that our raw-pixel approach increases the input-dimensionality of the behaviour 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. SIPEC:SegNet enhances SIPEC:IdNet's high identification performance through mask-matching-based tracking and integration of identities through time. We demonstrate that identification accuracy is considerably higher than that of the identification module of state-of-art idtracker.ai and PrimNet (Fig. 2e). We note, however, that identification using deep nets is not robust to interventions that greatly affect the appearance of the mice immediately after the intervention (such as the forced-swimming test; Extended Data Fig. 2). However, even without any interventions, expert human observers have difficulty identifying mice of such similar size and colour. The effects of different interventions on the recognition performances of deep net architectures should be studied in the future. Finally, SIPEC:PosNet enables top-down pose estimation of multiple animals in complex environments, making it easy to assign pose estimates to individual animals with higher performance than DLC (Fig. 2d).

All approaches are optimized through augmentation and transfer learning, which greatly speeds up learning and reduces labelling compared with the other approaches we tested on the mouse and non-human primate datasets. We also performed ablation studies for each of the networks to estimate the number of labels necessary for successful training. The number of labels necessary can change depending on the dataset—for example, each network could require more annotated frames to be trained successfully if the background and so on are more complex. To perform well under the complex video conditions for non-human primates, SIPEC:SegNet needs about 30 labels, SIPEC:IdNet about 1,500 labels and SIPEC:BehaveNet less than 2 h of annotated video (Fig. 2c,g and Fig. 4a).

SIPEC can be used to study the behaviour of primates and their social interactions over longer periods in a naturalistic environment, as we demonstrated for social grooming (Fig. 4a). Furthermore, after initial training of SIPEC modules, they can automatically output a behavioural profile for each individual in a group, over days or weeks and therefore also be used to quantify the changes in behaviours of individuals in social contexts over time. As SIPEC is fully supervised, it may be difficult to scale it to large colonies with hundreds of animals, such as bees and ants. However, SIPEC is well suited for most other animal species beyond insects.

Finally, we show how SIPEC enables three-dimensional localization and tracking from a single-camera view, yielding an off-the-shelf solution for home-cage monitoring of primates, without the need for setting stereo vision set-ups (Fig. 4b). Estimating the three-dimensional position requires the experimenter to create a three-dimensional model and annotate three-dimensional data. However, we show a quasi-three-dimensional estimate can be generated directly from the mask size, without manual annotation, that correlates highly with the actual position of the animal (Extended Data Fig. 5).

Behaviours that were not recognized and annotated by the researcher and therefore not learned by the neural network could be picked up using complementary unsupervised approaches[12,13]. The features-vectors, embedding individual behaviours, created by SIPEC:BehaveNet can be used as input to unsupervised approaches, which can help align the outputs of unsupervised approaches with human annotation. Moreover, the output of other modules (SIPEC:SegNet, SIPEC:IdNet and SIPEC:PoseNet) can also be used after such unsupervised approaches to analyse individual animals.

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 five 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 behavioural testing 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. We refer to Sturman et al.[20] for mouse behavioural data and annotation. For each day, we randomized the recording chamber of mice used. On days 1 and 2, we recorded animals 1–8 individually. On day 3, for measuring the effect of interventions on performance, mice were forced-swim-tested in water for 5 min immediately before the recording sessions.

Acquisition of primate data. Four male rhesus macaques were recorded with a 1080p camera within their home-cage. The large indoor room was about 15 m^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. We then used the VIA video annotator[39] to draw outlines around the animals.

Generation and annotation of primate behavioural videos. For creating the dataset, three primate videos of 20–30 min were annotated using the VIA video annotator[39]. These videos were generated by previous outputs of SIPEC:SegNet and SIPEC:IdNet. Frames of primates, identified as the same over consecutive frames, were stitched together to create 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 centre-cropped the frames around them. Labelled behaviours included 'searching', 'object interacting', 'social grooming' and 'none' (background class).

Tracking by segmentation and greedy mask matching. Based on the outputs of the segmentation masks, we implemented greedy-mask-matching-based tracking. For a given frame the bounding box of a given animal is assigned to the bounding box previous frames with the largest spatial overlap, with a decaying factor for temporally distant frames. The resulting overlap can be used as a confidence of SIPEC:SegNet-based tracking of the individual. This confidence can be used as a weight when using the resulting track identities to optionally smooth the labels of SIPEC:IdNet.

Identification labelling with the SIPEC toolbox. As part of SIPEC we release a graphical user interface that allows labelling for identification when multiple animals are present (Supplementary Fig. 2). To use the graphical user interface, SIPEC:SegNet has to be trained and inference has to be performed on videos to be identity labelled. SIPEC:SegNet results can then be loaded from the graphical user interface and overlaid with the original videos. Each box then marks an instance of the species that is to be labelled in green. For each animal, a number on the keyboard can be defined, which corresponds to the permanent ID of the animal. This keyboard number is then pressed and the mask-focus jumps to the next mask until all masks in that frame are annotated. The graphical user interface then 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 graphical user interface is closed.

SIPEC top-down workflow. For a given image, if we assume that N individuals are in the field of view, the output of SIPEC:SegNet is N segmentations or masks of the image. This step is mandatory if the analysis is for multiple animals in a group since subsequent pipeline parts are applied to the individual animals. Based on the masks, the individual animals' COMs are calculated as a proxy for the animals' two-dimensional spatial positions. We next crop the original image around the COMs of each animal, thus reducing the original frame to N COMs and N square-masked cut-outs of the individuals. This output can then be passed onto other modules.

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)[46] as the basis of a convolutional backbone architecture. These features were fed to the region proposal network, which applies convolutions onto these feature maps and proposes regions of interest (ROIs). These are subsequently passed to a ROIAlign layer, which performs feature pooling while preserving the pixel-correspondence in the original image. Per level of this pyramidal ROIAlign layer, we assign an ROI feature map from the different layers of the FPN feature maps. Multiple outputs are generated from the FPN, one of which is classifying if an animal is identified. The regressor head of the FPN returns bounding-box regression offsets per ROI. Another fully convolutional layer, followed by a per-pixel sigmoid activation, performs the mask prediction, returning a binary mask for each animal ROI. The network is trained using stochastic gradient descent, minimizing a multitask loss for each ROI:

$$L = L_{\text{mask}} + L_{\text{regression}} + L_{\text{class}} \quad (1)$$

where L_{mask} is the average binary cross-entropy between predicted and ground truth segmentation mask, applied to each ROI. $L_{\text{regression}}$ is a regression loss function applied to the coordinates of the bounding boxes, modified to be outlier robust as in the original fast R-CNN paper[47]. L_{class} is calculated for each of the proposed ROIs (or anchors) as a logarithmic loss of non-animal versus animal. The learning rate was adapted by an animal specific schedule and training was done iteratively, by first training the output layers for some epochs and then incrementally including previous blocks in the training process. SIPEC:SegNet outputs segmentation masks and bounding boxes to create cut-outs or masked cut-outs of individual animals to be used by one of the downstream modules.

SIPEC:IdNet network architecture and training. SIPEC:IdNet was based on the DenseNet architecture[28] for frame-by-frame identification. It consists of four dense blocks, which consist of multiple sequences of a batch normalization layer, a rectified linear unit (ReLU) activation function and a convolutional layer. The resulting feature maps are concatenated to the outputs of the following sequences of layers (skip connections). The resulting blocks are connected through transitions, which are convolutional followed by pooling layers. After the last dense block, we connect an average pooling layer to a Dropout[48] 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 time point into a batch-normalization layer[49] followed by three layers of bidirectional gated recurrent units[29,30] with leaky ReLU activation[50,51] ($\alpha = 0.3$) followed by a Dropout[48] layer with a rate of 0.2 followed by the softmax layer. The input for SIPEC:IdNet is the output cut-outs of individuals, generated by SIPEC:SegNet (for the single-animal case background-subtracted thresholding and centred-cropping would also work). For the recurrent case, the masks of past or future frames are dilated with a frames-per-second-dependent factor that increases with distance in time to increase the field of view. We first pre-trained the not-recurrent version of SIPEC:IdNet using Adam[52] with a learning rate (lr) of 0.00025, a batch size of 16 and using a weighted cross-entropy loss. We used a learning rate scheduler in the following form:

$$lr_{\text{E}+1} = \frac{lr_{\text{E}}}{k^{\text{E}}} \quad (2)$$

where E stands for epoch and constant $k = 1.5$. We then removed the softmax layer and fixed the network's weights. We then trained the recurrent SIPEC:IdNet again using Adam[52], lr = 0.00005, $k = 1.25$ and a batch size of six.

SIPEC:BehaveNet network architecture and training. SIPEC:BehaveNet was constructed as a raw-pixel action recognition network. It consists of a feature recognition network that operates on a single frame basis and a network, which integrates these features over time. The feature recognition network (FRN) is based on the Xception[32] architecture, consisting of an entry, middle and exit flow. The entry flow initially processes the input with convolution and ReLU blocks. Subsequently, we pass the feature maps through three blocks of separable convolution layers, followed by ReLU, separable convolution, and a max-pooling layer. The outputs of these three blocks are convolved and concatenated and passed to the middle flow. The middle flow consists of eight blocks of ReLU layers followed by a separable convolution layer. The exit flow receives the feature maps from the middle flow and passes it one more entry-flow-like block, followed by separable convolution and ReLU units. Finally, these features are integrated by a global average pooling layer, followed by a dense layer and passed through the softmax activation. This FRN was first pre-trained on a frame-by-frame basis using lr = 0.00035, gradient clipping norm of 0.5 and batch size = 36 using the Adam[52] optimizer. We reduced the original Xception architecture by the first 17 layers for mouse data to speed up the 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[33]. It is non-causal because, for the classification of a behaviour at time point t, it combines features from $[t - T, t + T]$ with T being the number of time steps, therefore looking backward in time and forward. In this study, we used an T 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[52] 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', that is no labelled behaviour), we used a multiclass adaptation technique Focal loss[53]—which is commonly used for object detection—and adapt it for action recognition to discount the contribution of the background class to the overall loss:

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

We used $\gamma = 3.0$ and $\alpha = 0.5$. For evaluation, we used the commonly used the F1 score to assess multiclass classification performance while using Pearson correlation to assess temporal correlation.

SIPEC:PoseNet network architecture and training. Combined with SIPEC:SegNet we can perform top-down pose estimation with SIPEC:PoseNet. That means, instead of the pose-estimation network outputting multiple possible outputs corresponding to different animals for one landmark, 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[2]. The SIPEC:PoseNet architecture is based on an encoder–decoder design[40]. In particular, we used EfficientNet[41] as a feature detection network for a single frame. These feature maps are then deconvolved into heatmaps that regress towards the target location of that landmark. Each deconvolutional layer is followed by a batch normalization layer and a ReLU activation function layer. For processing target images for pose-regression, we convolved pose landmark locations in the image with a two-dimensional Gaussian kernel. As there were many frames with an incomplete number of labels, we defined a custom cross-entropy-based loss function, which was zero for non-existing labels.

$$L_{\text{incomplete}} = \begin{cases} \text{Cross-entropy} \\ 0, \ \text{if labels does not exist} \end{cases}$$

Combined model. To test performance effects of doing a pose-estimation-based classification in conjunction with SIPEC:BehaveNet, we pre-trained SIPEC:PoseNet (with classification layer on top) as well as SIPEC:BehavNet individually. We then removed the output layers and fixed the weights of the individual networks and trained a joint output model, which combined inputs of each stream followed by a batch normalization layer, a dense layer (64 units), and a ReLU activation layer. The resulting units were concatenated into a joint tensor followed by a batch normalization layer, a dense layer (32 units), and a ReLU activation layer. This layer was followed by a dense layer with four units for the four behavioural classes and softmax activation function. This combined model was trained using Adam[52] with a lr of 0.00075. We further offer to use optical flow as an additional input, which has been shown to enhance action recognition performance[54].

Implementation and hardware. For all neural network implementations, we used Tensorflow[55] and Keras[56]. Computations were done on either NVIDIA RTX 2080 Ti or V100 GPUs.

Three-dimensional location labelling. To annotate the three-dimensional location of a primate, we firstly create a precise model of the physical room (Supplementary Fig. 4) using Blender. For a given mask-cut-out of a primate, we place an artificial primate at an approximate location in the three-dimensional model. We can then directly read out the three-dimensional position of the primate; 300 samples are annotated, covering the most frequent parts of the primate positions.

Three-dimensional location estimation. To regress the animal positions in three dimensions, we trained a manifold embedding using Isomap[45] 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 six 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 three-dimensional position of the animal.

Metrics used. The following metrics were used:

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

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

$$\text{Precision} = \frac{\text{TP}}{\text{TP}+\text{FP}}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP}+\text{FN}}$$

Where TP, FP, TN and FN denote true positives, false positives, true negatives and false negatives, respectively.

$$F1 = 2\cdot\frac{\text{Precision}\cdot\text{Recall}}{\text{Precision}+\text{Recall}}$$

$$\text{IOU}\,(M_{\text{GT}}, M_{\text{P}}) = \frac{M_{\text{GT}}\cap M_{\text{P}}}{M_{\text{GT}}\cup M_{\text{P}}}$$

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:

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

With

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

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

$$\text{dice} = 2\cdot\frac{M_{\text{GT}}\cap M_{\text{P}}}{|M_{\text{GT}}|+|M_{\text{P}}|}$$

Reporting summary. Further information on research design is available in the Nature Research Reporting Summary linked to this article.

Data availability

Mouse data from Sturman and colleagues[20] are available under https://zenodo.org/record/3608658. Example mouse data for training are available through our GitHub repository. The primate videos are available to the scientific community on request to V.M. (valerio@ini.uzh.ch).

Code availability

We provide the code for SIPEC at https://github.com/SIPEC-Animal-Data-Analysis/SIPEC (https://doi.org/10.5281/zenodo.5927367) and the GUI for the identification of animals https://github.com/SIPEC-Animal-Data-Analysis/idtracking_gui.

Received: 25 October 2020; Accepted: 13 March 2022;
Published online: 21 April 2022

References

1. Datta, S. R., Anderson, D. J., Branson, K., Perona, P. & Leifer, A. Computational neuroethology: a call to action. *Neuron* **104**, 11–24 (2019).
2. Mathis, A. et al. DeepLabCut: markerless pose estimation of user-defined body parts with deep learning. *Nature* **21**, 1281–1289 (2018).
3. Geuther, B. Q. et al. Robust mouse tracking in complex environments using neural networks. *Commun. Biol.* **2**, 124 (2019).
4. Romero-Ferrero, F., Bergomi, M. G., Hinz, R. C., Heras, F. J. & de Polavieja, G. idtracker.ai: Tracking all individuals in small or large collectives of unmarked animals. *Nat. Methods* **16**, 179 (2019).
5. Forys, B. J., Xiao, D., Gupta, P. & Murphy, T. H. Real-time selective markerless tracking of forepaws of head fixed mice using deep neural networks. *eNeuro* **7**, ENEURO.0096-20.2020 (2020).
6. Pereira, T. D. et al. Fast animal pose estimation using deep neural networks. *Nat. Methods* **16**, 117 (2019).
7. Graving, J. M. et al. DeepPoseKit, a software toolkit for fast and robust animal pose estimation using deep learning. *eLife* **8**, e47994 (2019).
8. Bala, P. C. et al. Automated markerless pose estimation in freely moving macaques with OpenMonkeyStudio. *Nat. Commun.* **11**, 4560 (2020).
9. Günel, S. et al. DeepFly3D, a deep learning-based approach for 3D limb and appendage tracking in tethered, adult *Drosophila*. *eLife* **8**, e48571 (2019).
10. Chen, Z. et al. AlphaTracker: a multi-animal tracking and behavioral analysis tool. Preprint at https://www.biorxiv.org/content/10.1101/2020.12.04.405159v1 (2020).
11. Lauer, J. et al. Multi-animal pose estimation and tracking with DeepLabCut. Preprint at https://www.biorxiv.org/content/10.1101/2021.04.30.442096v1 (2021).
12. Wiltschko, A. B. et al. Mapping sub-second structure in mouse behavior. *Neuron* **88**, 1121–1135 (2015).
13. Hsu, A. I. & Yttri, E. A. B-SOiD: an open source unsupervised algorithm for discovery of spontaneous behaviors. *Nat Commun.* **12**, 5188 (2019).
14. Berman, G. J., Choi, D. M., Bialek, W. & Shaevitz, J. W. Mapping the stereotyped behaviour of freely moving fruit flies. *J. R. Soc. Interface* **11**, 20140672 (2014).
15. Whiteway, M. R. et al. Partitioning variability in animal behavioral videos using semi-supervised variational autoencoders. *PLoS Comput. Biol.* **17**, e1009439 (2021).
16. Calhoun, A. J., Pillow, J. W. & Murthy, M. Unsupervised identification of the internal states that shape natural behavior. *Nat. Neurosci.* **22**, 2040–2049 (2019).
17. Batty, E. et al. *BehaveNet: Nonlinear Embedding and Bayesian Neural Decoding of Behavioral Videos* (NeurIPS, 2019).
18. Nilsson, S. R. et al. Simple behavioral analysis (SimBA)—an open source toolkit for computer classification of complex social behaviors in experimental animals. Preprint at https://www.biorxiv.org/content/10.1101/2020.04.19.049452v2 (2020).

19. Segalin, C. et al. The Mouse Action Recognition System (MARS) software pipeline for automated analysis of social behaviors in mice. *eLife* **10**, e63720 (2021).
20. Sturman, O. et al. Deep learning-based behavioral analysis reaches human accuracy and is capable of outperforming commercial solutions. *Neuropsychopharmacology* **45**, 1942–1952 (2020).
21. Nourizonoz, A. et al. EthoLoop: automated closed-loop neuroethology in naturalistic environments. *Nat. Methods* **17**, 1052–1059 (2020).
22. Branson, K., Robie, A. A., Bender, J., Perona, P. & Dickinson, M. H. High-throughput ethomics in large groups of *Drosophila*. *Nat. Methods* **6**, 451–457 (2009).
23. Dankert, H., Wang, L., Hoopfer, E. D., Anderson, D. J. & Perona, P. Automated monitoring and analysis of social behavior in *Drosophila*. *Nat. Methods* **6**, 297–303 (2009).
24. Kabra, M., Robie, A. A., Rivera-Alba, M., Branson, S. & Branson, K. JAABA: interactive machine learning for automatic annotation of animal behavior. *Nat. Methods* **10**, 64 (2013).
25. Jhuang, H. et al. Automated home-cage behavioural phenotyping of mice. *Nat. Commun.* **1**, 68 (2010).
26. Hayden, B. Y., Park, H. S. & Zimmermann, J. Automated pose estimation in primates. *Am. J. Primatol.* https://doi.org/10.1002/ajp.23348 (2021).
27. He, K., Gkioxari, G., Dollár, P. & Girshick, R. Mask R-CNN. In *Proc. IEEE International Conference on Computer Vision* 2961–2969 (IEEE, 2017).
28. Huang, G., Liu, Z., Van Der Maaten, L. & Weinberger, K. Q. Densely connected convolutional networks. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition* 4700–4708 (IEEE, 2017).
29. Cho, K. et al. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing* (EMNLP) 1724–1734 (Association for Computational Linguistics, 2014).
30. Chung, J., Gulcehre, C., Cho, K. & Bengio, Y. Empirical evaluation of gated recurrent neural networks on sequence modeling. In *NeurIPS 2014 Workshop on Deep Learning* (2014).
31. Deb, D. et al. Face recognition: primates in the wild. Preprint at https://arxiv.org/abs/1804.08790 (2018).
32. Chollet, F. Xception: deep learning with depthwise separable convolutions. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition* 1251–1258 (IEEE, 2017).
33. Van den Oord, A. et al. WaveNet: a generative model for raw audio. Preprint at https://arxiv.org/abs/1609.03499 (2016)
34. Bai, S., Kolter, J. Z. & Koltun, V. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. Preprint at https://arxiv.org/abs/1803.01271 (2018).
35. Jung, A. B. et al. *Imgaug* (GitHub, 2020); https://github.com/aleju/imgaug
36. 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 (NeurIPS, 2014).
37. He, K., Zhang, X., Ren, S. & Sun, J. Deep residual learning for image recognition. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition* 770–778 (IEEE, 2016).
38. Lin, T.-Y. et al. Microsoft COCO: Common Objects in Context. In *European Conference on Computer Vision* 740–755 (Springer, 2014).
39. Dutta, A. & Zisserman, A. The VIA annotation software for images, audio and video. In *Proc. 27th ACM International Conference on Multimedia* (ACM, 2019); https://doi.org/10.1145/3343031.3350535
40. Xiao, B., Wu, H. & Wei, Y. Simple baselines for human pose estimation and tracking. In *Computer Vision – ECCV 2018* (eds. Ferrari, V., Hebert, M., Sminchisescu, C. & Weiss, Y.) 472–487 (Springer International Publishing, 2018).
41. Tan, M. & Le, Q. V. EfficientNet: rethinking model scaling for convolutional neural networks. Preprint at https://arxiv.org/abs/1905.11946 (2020).
42. Krizhevsky, A., Sutskever, I. & Hinton, G. E. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems* 1097–1105 (NeurIPS, 2012).
43. Vidal, M., Wolf, N., Rosenberg, B., Harris, B. P. & Mathis, A. Perspectives on individual animal identification from biology and computer vision. *Integr. Comp. Biol.* **61**, 900–916 (2021).
44. Demšar, J. Statistical comparisons of classifiers over multiple data sets. *J. Mach. Learn. Res.* **7**, 1–30 (2006).
45. Tenenbaum, J. B. A global geometric framework for nonlinear dimensionality reduction. *Science* **290**, 2319–2323 (2000).
46. 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); https://doi.org/10.1109/CVPR.2017.106
47. Girshick, R. Fast R-CNN. In *2015 IEEE International Conference on Computer Vision (ICCV)* 1440–1448 (IEEE, 2015); https://doi.org/10.1109/ICCV.2015.169
48. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. & Salakhutdinov, R. Dropout: a simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* **15**, 1929−1958 (2014).
49. Ioffe, S. & Szegedy, C. Batch normalization: accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning* Vol. 37, 448–456 (JMLR.org, 2015).
50. Maas, A. L., Hannun, A. Y. & Ng, A. Y. Rectifier nonlinearities improve neural network acoustic models. In *Proc. 30th International Conference on Machine Learning* (ICML, 2013).
51. Xu, B., Wang, N., Chen, T. & Li, M. Empirical evaluation of rectified activations in convolutional network. Preprint at https://arxiv.org/abs/1505.00853 (2015).
52. Kingma, D. P. & Ba, J. Adam: a method for stochastic optimization. In *Proceedings of the 3rd International Conference on Learning Representations* (ICLR, 2014).
53. Lin, T.-Y., Goyal, P., Girshick, R., He, K. & Dollar, P. Focal loss for Dense object detection. In *2017 IEEE International Conference on Computer Vision* (ICCV, 2017).
54. Bohnslav, J. P. et al. DeepEthogram: a machine learning pipeline for supervised behavior classification from raw pixels. *eLife* **10**, 63377 (2020).
55. Abadi, M. et al. TensorFlow: large-scale machine learning on heterogeneous distributed systems. Preprint at https://arxiv.org/abs/1603.04467 (2016).
56. Chollet, F. *Keras* (GitHub, 2015); https://github.com/fchollet/keras

Acknowledgements

This project was funded by the Swiss Federal Institute of Technology (ETH) Zurich and the European Research Council (ERC) under the ERC Consolidator Award (grant no. 818179 to MFY), SNSF (grant no. CRSII5_198739/1 to MFY; grant no. 310030_172889/1 to J.B., grant no. PP00P3_157539 to V.M.) ETH Research Grant (grant no. ETH-20 19-1 to J.B.), 3RCC (grant no. OC-2019-009 to J.B. and M.F.Y.), the Simons Foundation (award nos. 328189 and 543013 to V.M.) and the Botnar Foundation (to J.B.). We would like to thank P. Tornmalm and V. de La Rochefoucauld for annotating primate data and feedback on primate behaviour, and P. Johnson, B. Yasar, B. Wu, and A. 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 three-dimensional-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.

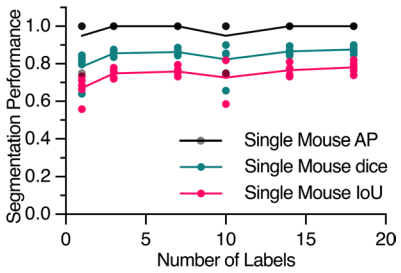
Additional information

Extended data is available for this paper at https://doi.org/10.1038/s42256-022-00477-5.

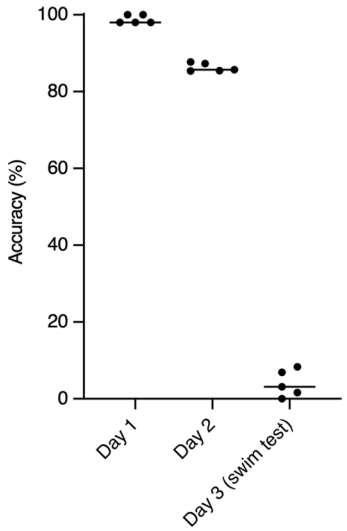
Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s42256-022-00477-5.

Correspondence and requests for materials should be addressed to Mehmet Fatih Yanik.

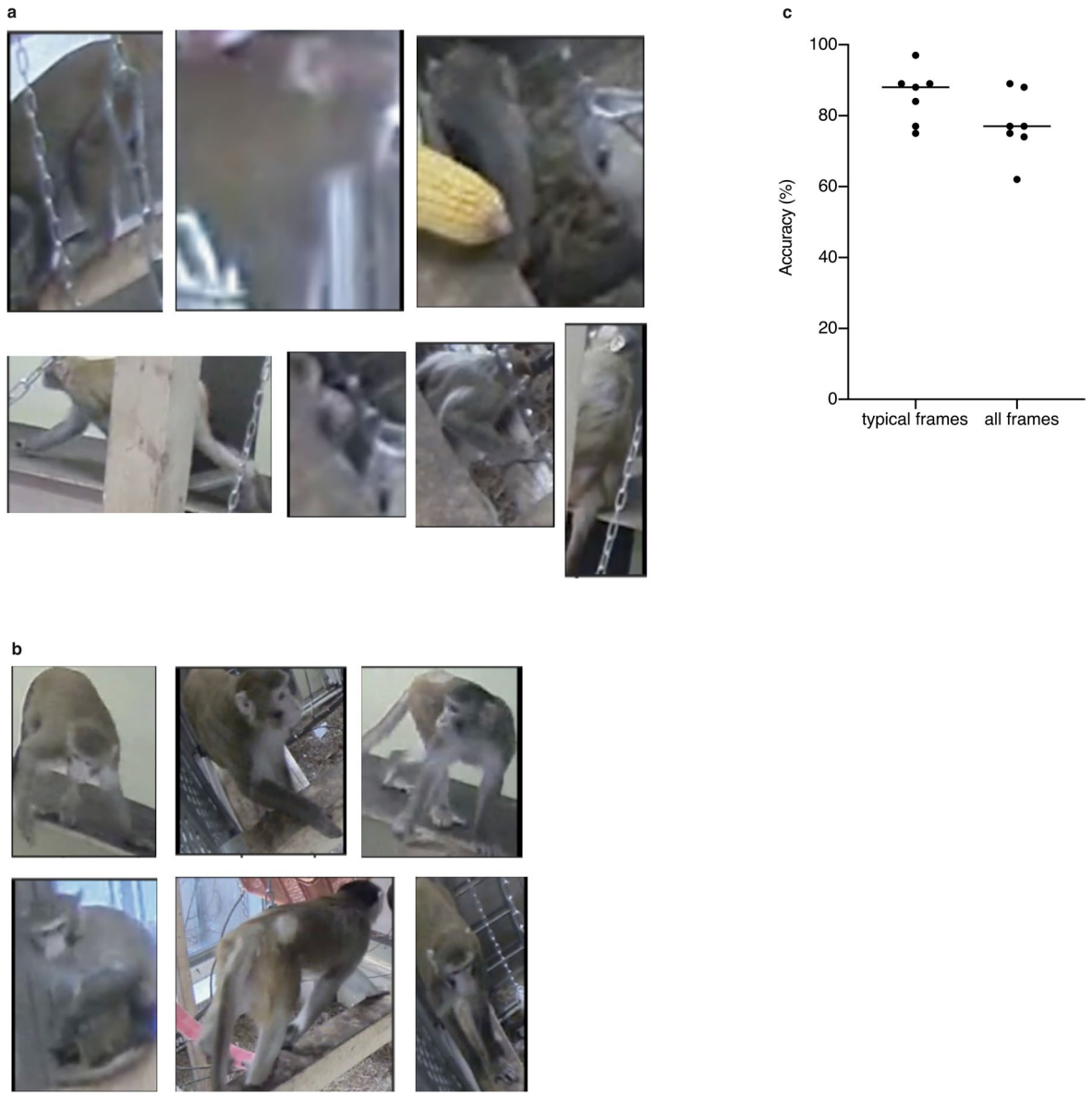
Peer review information *Nature Machine Intelligence* thanks Adam Kepecs and the other, anonymous, reviewer(s) for their contribution to the peer review of this work.



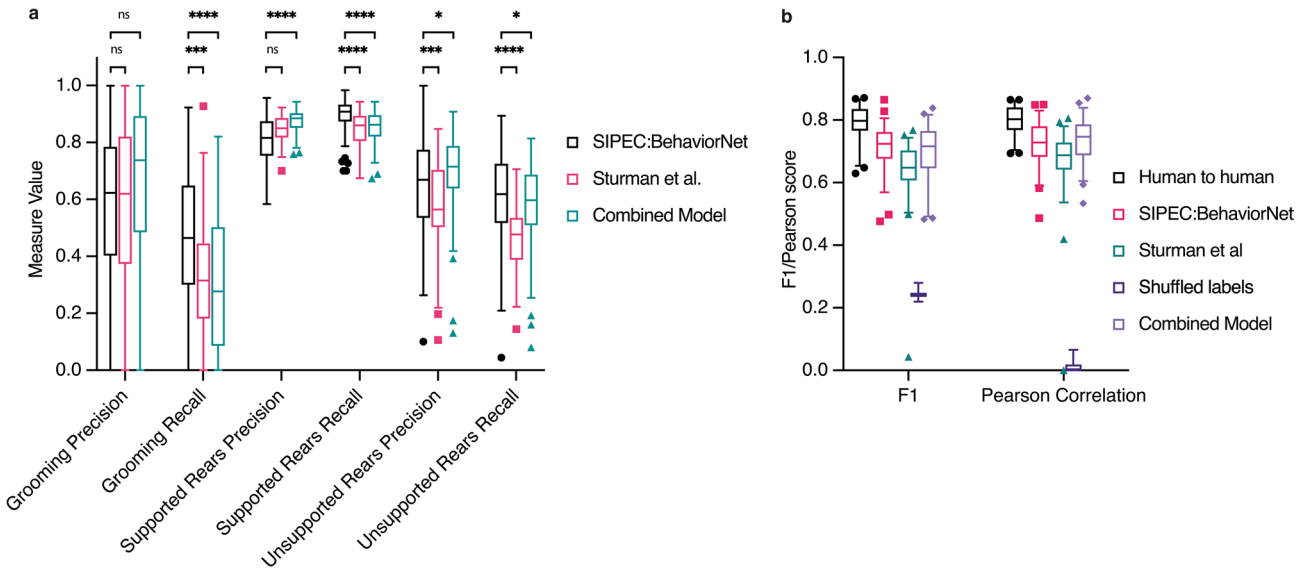
Extended Data Fig. 1 | Individual mouse segmentation. For mice, SIPEC:SegNet performance in mAP, dice and IoU for single mouse as a function of the number of labels. The lines indicate the means for 5-fold CV while circles, squares, triangles indicate the mAP, dice, and IoU, respectively, for individual folds. All data is represented by mean, showing all points.



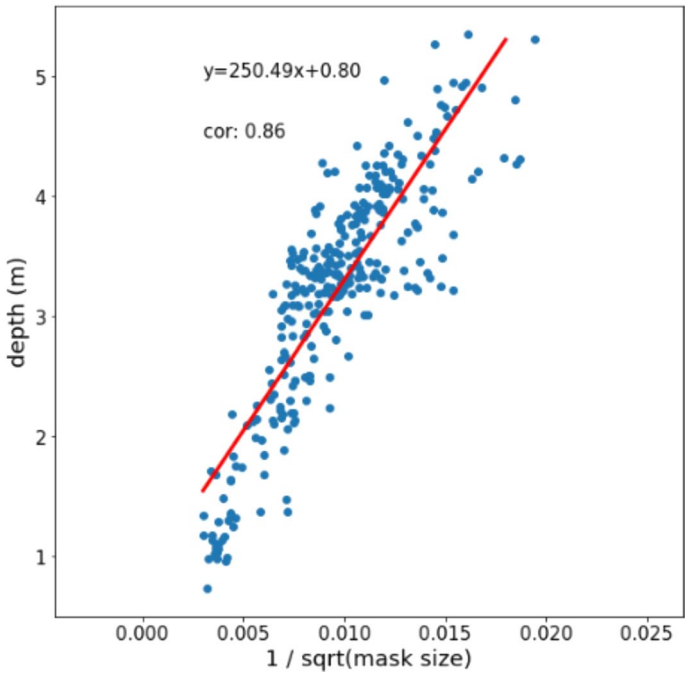
Extended Data Fig. 2 | Identification performance of mice across days and interventions. Identification accuracy across days for models trained on day 1. While the performance for the day the model is trained on is very high it drops when tested on day 2 but is still significantly above chance level.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



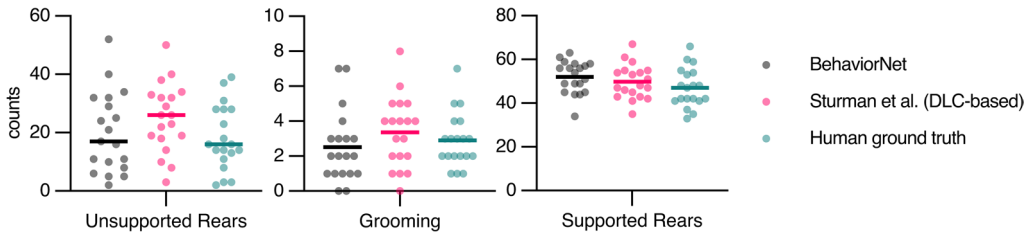
Extended Data Fig. 3 | Identification of typical vs difficult frames. a) Examples of very difficult frames, which are also beyond human single-frame recognition, are excluded for the 'typical' frame evaluation. **b**) Example frames used for the 'typical' frame analysis. **c**) Identification performance is significantly higher on 'typical' frames than on all frames. All data is represented by mean, showing all points.



Extended Data Fig. 4 | Additional behavioural 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 as well as combined model. Using pose estimates in conjunction with raw-pixel classification increases precision in comparison with solely raw-pixel classification while suffering from a decrease in recall. All data is represented by a Tukey box-and-whisker plot, showing all points. Wilcoxon paired test: *P≤0.05; ***P≤0.001; ****P≤0.0001.



Extended Data Fig. 5 | 3D depth estimates based on mask size. The inverse of the square root of the mask size (based on SIPEC:SegNet output) highly correlates with the depth of the individual in 3D space.



Extended Data Fig. 6 | Comparison of counts of behaviours between SIPEC:BehaveNet, pose estimation based approach and human raters. Unsupported and supported rears and grooming events were counted per video for n = 20 different mice videos. Behaviours were integrated over multiple frames, as described in Sturman et al. Behavioural counts of 3 different human expert annotators were averaged (in legend as 'human ground truth'). No significant differences were found for comparing the number of behaviours between SIPEC:BehaveNet and human annotators or Sturman et al. and human annotators (Tukey's multiple comparison test). All data is represented by mean, showing all points.

nature research

Corresponding author(s): Prof. Dr. Mehmet Fatih Yanik

Last updated by author(s): 02.02.2022

Reporting Summary

Nature Research wishes to improve the reproducibility of the work that we publish. This form provides structure for consistency and transparency in reporting. For further information on Nature Research policies, see our Editorial Policies and the Editorial Policy Checklist.

Statistics

For all statistical analyses, confirm that the following items are present in the figure legend, table legend, main text, or Methods section.

n/a	Confirmed	
☐	☒	The exact sample size (*n*) for each experimental group/condition, given as a discrete number and unit of measurement
☐	☒	A statement on whether measurements were taken from distinct samples or whether the same sample was measured repeatedly
☐	☒	The statistical test(s) used AND whether they are one- or two-sided *Only common tests should be described solely by name; describe more complex techniques in the Methods section.*
☒	☐	A description of all covariates tested
☒	☐	A description of any assumptions or corrections, such as tests of normality and adjustment for multiple comparisons
☐	☒	A full description of the statistical parameters including central tendency (e.g. means) or other basic estimates (e.g. regression coefficient) AND variation (e.g. standard deviation) or associated estimates of uncertainty (e.g. confidence intervals)
☐	☒	For null hypothesis testing, the test statistic (e.g. *F*, *t*, *r*) with confidence intervals, effect sizes, degrees of freedom and *P* value noted *Give P values as exact values whenever suitable.*
☒	☐	For Bayesian analysis, information on the choice of priors and Markov chain Monte Carlo settings
☒	☐	For hierarchical and complex designs, identification of the appropriate level for tests and full reporting of outcomes
☐	☒	Estimates of effect sizes (e.g. Cohen's *d*, Pearson's *r*), indicating how they were calculated

Our web collection on statistics for biologists contains articles on many of the points above.

Software and code

Policy information about availability of computer code

Data collection	- primate videos: Videos were acquired using a Bosch Autodome IP starlight 7000 HD camera with 1080p resolution at 50 Hz - mouse videos: videos were acquired like in Sturman et al. with the TSE Multi Conditioning System - 3D annotations: the 3D room model used for annotations of positions was created using the Blender software
Data analysis	- machine learning was conducted using custom written python software using the following open source software (available via pip package manager): python=3.7, tensorflow-gpu=1.14.0, keras=2.3.1, opencv-contrib-python, ffmpeg, scikit-video, numpy, imgaug, pandas, scikit-learn, scipy, scikit-image, imblearn, tqdm, joblib, seaborn, PIL (Pillow) All the code used to analyze the data is custom made and open-source and can be found at https://github.com/SIPEC-Animal-Data-Analysis/SIPEC and https://github.com/SIPEC-Animal-Data-Analysis/idtracking_gui - plotting: we used Graphpad Prism 9 for creating the plots and Adobe Illustrator 24 for arranging them

For manuscripts utilizing custom algorithms or software that are central to the research but not yet described in published literature, software must be made available to editors and reviewers. We strongly encourage code deposition in a community repository (e.g. GitHub). See the Nature Research guidelines for submitting code & software for further information.

Data

Policy information about availability of data

All manuscripts must include a data availability statement. This statement should provide the following information, where applicable:

- Accession codes, unique identifiers, or web links for publicly available datasets
- A list of figures that have associated raw data
- A description of any restrictions on data availability

Mouse data from Sturman et al.14 is available under https://zenodo.org/record/3608658. Primate data available upon reasonable request from authors. Exemplary data for training is available through our github repository.

Field-specific reporting

Please select the one below that is the best fit for your research. If you are not sure, read the appropriate sections before making your selection.

☒ Life sciences ☐ Behavioural & social sciences ☐ Ecological, evolutionary & environmental sciences

For a reference copy of the document with all sections, see nature.com/documents/nr-reporting-summary-flat.pdf

Life sciences study design

All studies must disclose on these points even when the disclosure is negative.

Sample size	We performed no sample size calculation. After establishing performance baselines for each of the SIPEC modules we systematically performed ablation studies for estimating the number of samples necessary to reach a certain performance.
Data exclusions	No data was excluded.
Replication	All attempts at replication were successful. Cross-validation was used to indicate replicability.
Randomization	Our cross-validation based assessment of performance did not rely on allocation of individuals to different groups. No random allocation to groups was performed.
Blinding	Our cross-validation based assessment of performance did not rely on allocation of individuals to different groups. Therefore no blinding was necessary.

Reporting for specific materials, systems and methods

We require information from authors about some types of materials, experimental systems and methods used in many studies. Here, indicate whether each material, system or method listed is relevant to your study. If you are not sure if a list item applies to your research, read the appropriate section before selecting a response.

Materials & experimental systems

n/a	Involved in the study
☒	☐ Antibodies
☒	☐ Eukaryotic cell lines
☒	☐ Palaeontology and archaeology
☐	☒ Animals and other organisms
☒	☐ Human research participants
☒	☐ Clinical data
☒	☐ Dual use research of concern

Methods

n/a	Involved in the study
☒	☐ ChIP-seq
☒	☐ Flow cytometry
☒	☐ MRI-based neuroimaging

Animals and other organisms

Policy information about studies involving animals; ARRIVE guidelines recommended for reporting animal research

Laboratory animals	C57BL/6J (C57BL/6JRj) mice (male, 2.5 months of age) were obtained from Janvier (France). We used 4 male rhesus macaques (Macaca, mulatta), who were 5 years old.
Wild animals	No wild animals were involved in this study.
Field-collected samples	No field-collected samples were involved in this study.
Ethics oversight	The animal procedures of these studies were approved by the local veterinary authorities of the Canton Zurich, Switzerland, and

Ethics oversight	carried out in accordance with the guidelines published in the European Communities Council Directive of November 24, 1986 (86/609/EEC).

Note that full information on the approval of the study protocol must also be provided in the manuscript.