nature neuroscience

TECHNICAL REPORT

https://doi.org/10.1038/s41593-018-0209-y

DeepLabCut: markerless pose estimation of user-defined body parts with deep learning

Alexander Mathis[1,2], Pranav Mamidanna[1], Kevin M. Cury[3], Taiga Abe[3], Venkatesh N. Murthy[2], Mackenzie Weygandt Mathis[1,4,8*] and Matthias Bethge[1,5,6,7,8]

Quantifying behavior is crucial for many applications in neuroscience. Videography provides easy methods for the observation and recording of animal behavior in diverse settings, yet extracting particular aspects of a behavior for further analysis can be highly time consuming. In motor control studies, humans or other animals are often marked with reflective markers to assist with computer-based tracking, but markers are intrusive, and the number and location of the markers must be determined a priori. Here we present an efficient method for markerless pose estimation based on transfer learning with deep neural networks that achieves excellent results with minimal training data. We demonstrate the versatility of this framework by tracking various body parts in multiple species across a broad collection of behaviors. Remarkably, even when only a small number of frames are labeled (~200), the algorithm achieves excellent tracking performance on test frames that is comparable to human accuracy.

Accurate quantification of behavior is essential for understanding the brain[1–3]. Both within and beyond the field of neuroscience, there is a fruitful tradition of using cutting-edge technology to study movement. Often, the application of new technology has the potential to reveal unforeseen features of the phenomena being studied, as in the case of Muybridge's famous photography studies in the mid-19th century or modern high-speed videography that has revealed previously unknown motor sequences, such as 'tap dancing' in the songbird[2,4,5]. Historically, collected data were analyzed manually, which is a time-consuming, labor-intensive and error-prone process that is prohibitively inefficient at today's high rates of data acquisition. Conversely, advances in computer vision have consistently inspired methods of data analysis to reduce human labor[6–8].

We are particularly interested in extracting the pose of animals—i.e., the geometrical configuration of multiple body parts. The gold standard for pose estimation in the field of motor control is the combination of video recordings with easily recognizable reflective markers applied to locations of interest, which greatly simplifies subsequent analysis and allows tracking of body parts with high accuracy[9–12]. However, such systems can be expensive and potentially distracting to animals[13,14], and markers need to be placed before recording, which predefines the features that can be tracked. This mitigates one of the benefits of video data: its low level of invasiveness. One alternative to physical markers is to fit skeleton or active contour models[13–17]. These methods can work quite well and are fast, but require sophisticated skeleton models, which are difficult to develop and to fit to data, limiting the flexibility of such methods[18,19]. Another alternative is training regressors based on various computationally derived features to track particular body parts in a supervised way[6,13,20–22]. Training predictors based on features from deep neural networks also falls in this category[23,24]. Indeed, the best algorithms for challenging benchmarks in pose estimation of humans from images use deep features[19,25–29]. This suggests that deep learning architectures should also greatly improve the accuracy of pose estimation for lab applications. However, the labeled datasets for these benchmarks are large (for example, ~25,000 in the MPII Human Pose dataset[30]), which may render deep learning approaches infeasible as efficient tools at the scale of interest to neuroscience labs. Nevertheless, as a result of transfer learning[31–34], we will show that this need not be the case.

Here we demonstrate that by capitalizing on state-of-the-art methods for detecting human limb configurations, we can achieve excellent performance on pose estimation problems in the laboratory setting with minimal training data. Specifically, we investigated the feature detector architecture from DeeperCut[26,27], one of the best pose estimation algorithms, and demonstrate that a small number of training images (~200) can be sufficient to train this network to within human-level labeling accuracy. This is possible as a result of transfer learning: the feature detectors are based on extremely deep neural networks, which were pretrained on ImageNet, a massive dataset for object recognition[24]. We also show that end-to-end training the network increases performance. Thus, by labeling only a few hundred frames, one can train tailored, robust feature detectors that are capable of localizing a variety of experimentally relevant body parts. We illustrate the power of this approach by tracking the snout, ears and tail base of a mouse during an odor-guided navigation task, multiple body parts of a fruit fly behaving in a 3D chamber, and joints of individual mouse digits during a reaching task.

Results

DeeperCut achieves outstanding performance on multi-human pose detection benchmarks[27]. However, to achieve this performance, its neural network architecture has been trained on thousands of labeled images. Here we focus on a subset of DeeperCut: its feature detectors, which are variations of deep residual neural

[1]Institute for Theoretical Physics and Werner Reichardt Centre for Integrative Neuroscience, Eberhard Karls Universität Tübingen, Tübingen, Germany. [2]Department of Molecular & Cellular Biology and Center for Brain Science, Harvard University, Cambridge, MA, USA. [3]Department of Neuroscience and the Mortimer B. Zuckerman Mind Brain Behavior Institute, Columbia University, New York, NY, USA. [4]The Rowland Institute at Harvard, Harvard University, Cambridge, MA, USA. [5]Max Planck Institute for Biological Cybernetics, Tübingen, Germany. [6]Bernstein Center for Computational Neuroscience, Tübingen, Germany. [7]Center for Neuroscience and Artificial Intelligence, Baylor College of Medicine, Houston, TX, USA. [8]These authors jointly directed this work: Mackenzie Weygandt Mathis, Matthias Bethge. *e-mail: mackenzie@post.harvard.edu

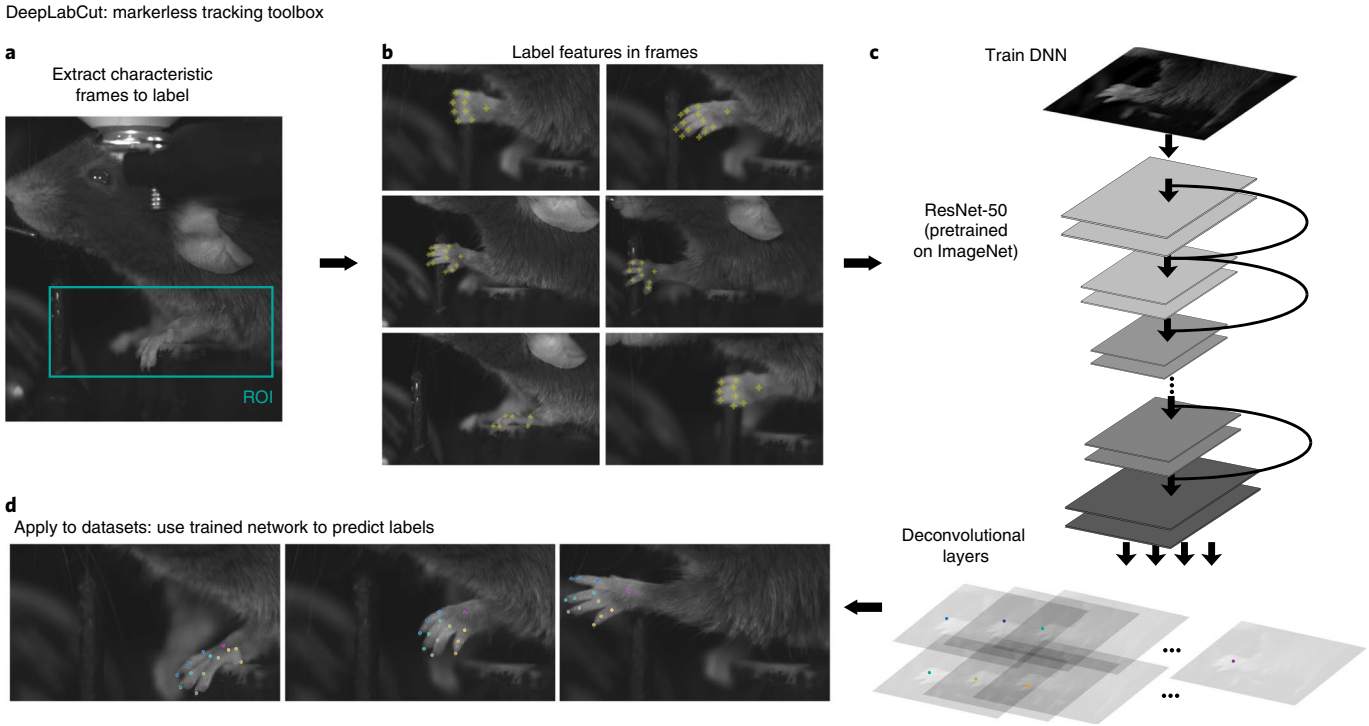

Fig. 1 | Procedure for using the DeepLabCut Toolbox. a, Training: extract images with distinct postures characteristic of the animal behavior in question. For computational efficiency, the region of interest (ROI) should be picked to be as small as possible while containing the behavior in question, which in this example is reaching. **b**, Manually localize (label) various body parts. Here, various digit joints and the wrist were selected as features of interest. **c**, Train a deep neural network (DNN) architecture to predict the body-part locations on the basis of the corresponding image. A distinct readout layer per body part is generated to predict the probability that a body part is in a particular pixel. Training adjusts both readout and DNN weights. After training the weights are stored. **d**, The trained network can be used to extract the locations of the body parts from videos. The images show the most likely body part locations for 13 labeled body parts on the hand of a mouse.

networks (ResNet)[24] with readout layers that predict the location of a body part (Fig. 1). To distinguish the feature detectors from the full DeeperCut, we refer to this autonomous portion as DeepLabCut. In this paper, we evaluate the performance of DeepLabCut for posture tracking in various laboratory behaviors, investigate the amount of required training data for good generalization, and provide an open source toolbox that is broadly accessible to the community (https://github.com/AlexEMG/DeepLabCut).

DeepLabCut is a deep convolutional network combining two key ingredients from algorithms for object recognition and semantic segmentation: pretrained ResNets and deconvolutional layers[27]. The network consists of a variant of ResNets, whose weights were trained on a popular, large-scale object recognition benchmark called ImageNet, on which it achieves excellent performance[24]. Instead of the classification layer at the output of the ResNet, deconvolutional layers are used to up-sample the visual information and produce spatial probability densities. For each body part, its probability density represents the 'evidence' that a body part is in a particular location. To fine-tune the network for a particular task, its weights are trained on labeled data, which consist of frames and the accompanying annotated body part locations (or other objects of interest in the frame). During training, the weights are adjusted in an iterative fashion such that for a given frame the network assigns high probabilities to labeled body part locations and low probabilities elsewhere (Fig. 1 and Methods). Thereby, the network is rewired and 'learns' feature detectors for the labeled body parts. As a result of the initialization with the ResNet pretrained on ImageNet, this rewiring is robust and data-efficient.

Benchmarking DeepLabCut. Analyzing videos taken in a dynamically changing environment can be challenging. Therefore, to test the utility of our toolbox, we first focused on an odor-guided navigation task for mice. Briefly, mice run freely on an 'endless' paper spool that includes an adapted ink-jet printer to deliver odor trails in real time as a mouse runs and tracks trails (further details and results will be published elsewhere). The video captured during the behavior poses several key challenges: inhomogeneous illumination; transparent side walls that appear dark; shadows around the mouse from overhead lighting; distortions due to a wide-angle lens; the frequent crossing of the mouse over the odor trail; and the common occurrence of rewards directly in front of its snout, which influences its appearance. Yet accurately tracking the snout as a mouse samples the 'odorscape' is crucial for studying odor-guided navigation. Various measures could be taken to remedy these challenges, such as performing a camera calibration to reduce distortions. However, we were interested in testing whether DeepLabCut could cope with all the challenges in the raw data without any preprocessing.

First, we extracted 1,080 distinct frames from multiple videos (across 2 cameras and 7 different mice; see Methods) and manually labeled the snout, left ear, right ear and tail base in all frames (Fig. 2a and Supplementary Fig. 1). To facilitate comparisons to ground truth and to quantify the robustness of predictors, we estimated variability (root mean square error; RMSE) of one human labeler by comparing two distinct label sets of the same data. We found the average variability for all body parts to be very small: 2.69 ± 0.1 pixels (mean $\pm$ s.e.m.; $n = 4{,}320$ body part image pairs; Supplementary Fig. 1 and Methods), which is less than the ~5-pixel width of the mouse's snout in low-resolution camera frames (Fig. 2a). The RMSE across two trials of annotating the same images is referred to as 'human variability' (note that the variability differs slightly across body parts).

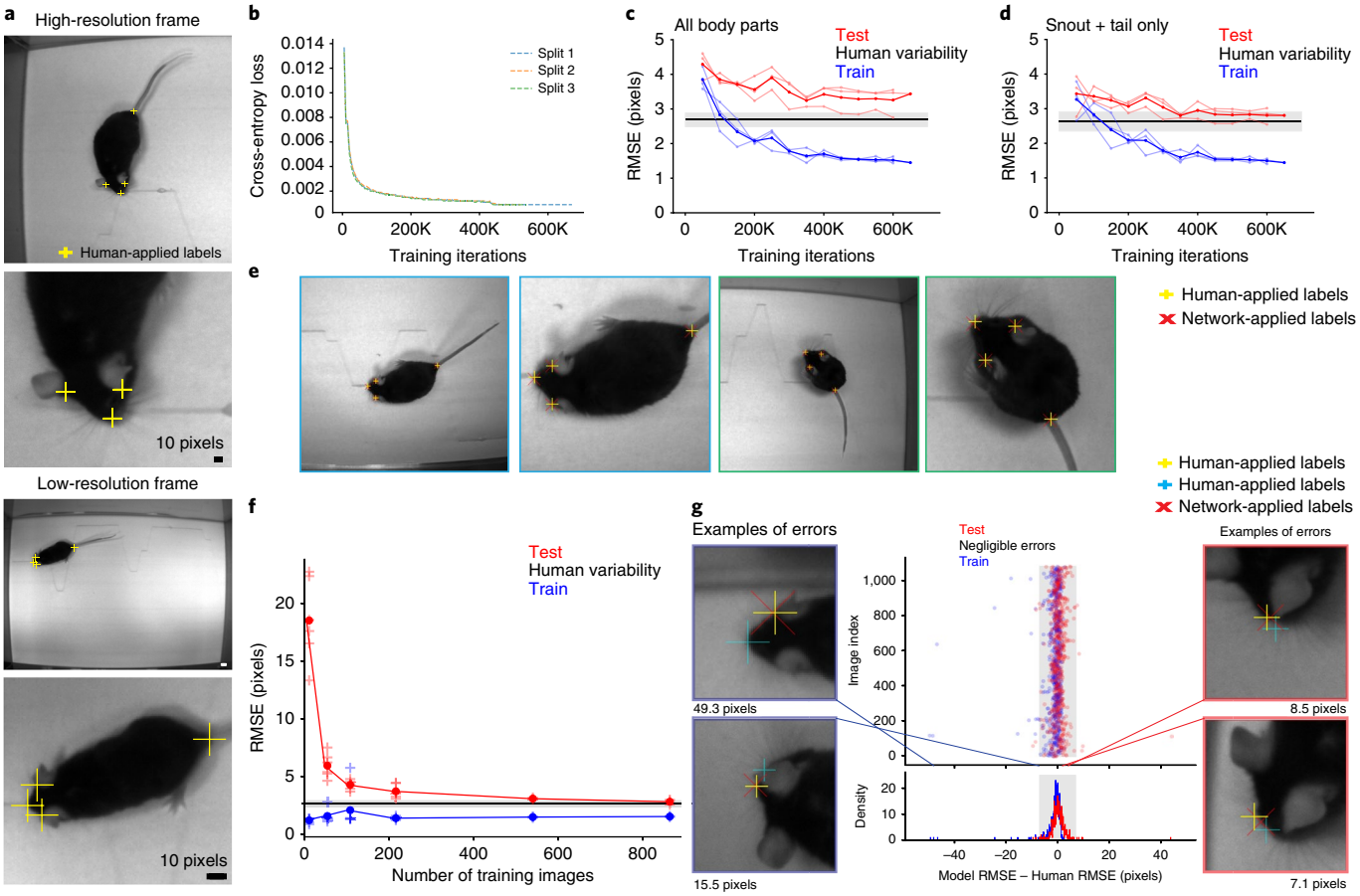

Fig. 2 | Evaluation during odor trail tracking. a, Two example frames with high-magnification images, showing the snout, ears and tail base labeled by a human. The odor trail and reward drops are visible under infrared light and introduce a time-varying visual background. **b**, Cross-entropy loss for $n=3$ splits when training with 80% of the 1,080 frames. **c**, Corresponding RMSE between the human and the predicted label on training and test images for those $n=3$ splits evaluated every 50,000 iterations (80%/20% splits). The average of those individual curves is also shown (thick line). Human variability as well as the 95% confidence interval are depicted in black and gray. **d**, Corresponding RMSE when evaluated only for snout and tail base. The algorithm reaches human level variability on a test set comprising 20% of all images. **e**, Example body part prediction for two frames, which were not used during the training (of one split in **c**). Prediction closely matches human annotator's labels. **f**, RMSE for snout and tail base for several splits of training and test data vs. number of training images compared to RMSE of a human scorer. Each split is denoted by a cross, the average by a dot. For 80% of the data, the algorithm achieves human level accuracy on the test set (**d**). As expected, test RMSE increases for fewer training images. Around 100 frames are enough to provide good average tracking performance (<5-pixel accuracy). **g**, Snout RMSE comparison between human and model per image for one split with 50% training set size. Most RMSE differences are small, with few outliers. The two extreme errors (on the left) are due to labeling errors across trials by the human.

To quantify the feature detector's performance, we randomly split the data into a training and test set (80% and 20%, respectively) and evaluated the performance of DeepLabCut on test images across all body parts (Fig. 2b, c) and in a subset of body parts (snout and tail base) (Fig. 2d). Unless otherwise noted, we always trained (and tested) with the labels from the first set of human labels. The test RMSE for different training/test set splits achieved average human variability (Fig. 2d). Thus, we found that when trained with 80% of the data the algorithm achieved human-level accuracy on the test set for detection of the snout and the tail base (Fig. 2d,e).

Next, we systematically varied the size of the training set and trained 30 distinct networks (three splits for 50% and 80% training set size; six splits for 1%, 5%, 10% and 20% training set fraction). As expected, the test error increases for decreasing number of training images (Fig. 2f). Yet remarkably, the test RMSE attenuates only slowly from 80% training set fraction to 10%, where one still achieves an average pixel error of less than 5 pixels. Such average errors are on the order of the size of the snout in the low-resolution camera (around 5 pixels) and much smaller than the size of the snout in the high-resolution camera (around 30 pixels). Thus, we found that even 100 frames were enough to achieve excellent generalization.

Since the RMSE is computed as an average across images, we next checked whether there were any systematic differences across images by comparing the human variability across the two splits vs. the model variability (trained with the first set of human labels; Fig. 2g: data for one split with a 50% training set size). We found that both the human and the algorithm produced only a few outliers, and no systematic error was detected (see Fig. 2g for examples).

We also tested whether data augmentation beyond rescaling (see Methods) could provide any gains in performance. On larger training sets DeepLabCut already reaches human-level accuracy, so we focused on the six splits that used only 10% of the data for training. Specifically, we augmented the data to include either several rotations or several translations plus rotations per training image (see Methods). We found minimal differences in test performance (Supplementary Fig. 2a), highlighting the data-efficiency of DeepLabCut. This also suggests that simple data augmentation cannot replace images that capture behavioral variability—i.e., adding new labeled images that have more postural variability seems to be better than augmenting a smaller subset of the data.

Thus far, we used a part detector based on the 50-layer deep ResNet-50[24,27]. We also trained deeper networks with 101 layers and

found that both the training and testing errors decreased slightly, suggesting that the performance can be further improved if required (average test RMSE for three identical splits of 50% training set fraction: ResNet-50, 3.09 ± 0.04; ResNet-101, 2.90 ± 0.09; ResNet-101 with intermediate supervision, 2.88 ± 0.06; pixel mean ± s.e.m.; see Supplementary Fig. 2b).

Overall, given the robustness and the low error rate of DeepLabCut even with small training sets, we found this to be a useful tool for studies such as odor-guided navigation. For example, Fig. 3 recapitulates a salient signature of the tracking behavior, namely that rodents swing their snout across the trail[35]. Knowing the location of the ears as well as the tail base is also important to computationally assess the orientation of the mouse (Fig. 3 and Supplementary Video 1). Furthermore, having an automated pose estimation algorithm as presented will be crucial for other video-rich experiments.

Generalization and transfer learning. We have demonstrated that DeepLabCut can accurately detect body parts across different mice, but how does it generalize to novel scenarios? First, we found that DeepLabCut generalizes to novel mice during trail tracking (Fig. 4a). Second, we tested whether the trained network could identify multiple body parts across multiple mice in the same frame (transfer learning). Notably, although the network has only been trained with images containing a single mouse, it could detect all the body parts of each mouse in images with multiple interacting mice. Although not error-free, we found that the model performed remarkably well in a social task (three mice interacting in an unevenly illuminated open field; Fig. 4b). The performance of the body part detectors could be improved by training the network with training images that include multiple mice with occlusions and/or by training image-conditioned pairwise terms between body parts to harness the power of multi-human pose estimation models[27] (see Discussion). Nonetheless, this example of multi-mouse tracking illustrates that even the feature detectors trained on a single mouse can readily transfer to extensions, as would be useful in studies of social behaviors[6,36,37].

The power of end-to-end training. As a result of the architecture of DeeperCut, the deconvolution layers are specific to each body part, but the deep network (ResNet) is shared (Fig. 1). We hypothesized that this architecture can facilitate the localization of one body part based on other labeled body parts. To test this hypothesis, we examined the performance of networks trained with only the snout and tail-base data while using the same three splits of 50% training data as in Fig. 2b. We found that the network that was trained with all body part labels simultaneously outperforms the specialized networks nearly twofold (Fig. 5). This result also demonstrates that training the weights throughout the whole network in an end-to-end fashion rather than just the readout weights substantially improves the performance. This further highlights the advantage of deep learning based models over approaches with fixed feature representations, which cannot be refined during training.

***Drosophila* in a 3D behavioral chamber.** To further demonstrate the flexibility of the DeepLabCut toolbox, we tracked the bodies of freely behaving fruit flies (*Drosophila*) exploring a small cubical environment in which one surface contained an agar-based substrate for egg laying. Freely behaving flies readily exhibit many orientations and also frequent the walls and ceiling. When viewed from a fixed perspective, these changes in orientation dramatically alter the appearance of flies as the spatial relationship of body features change or as different body parts come into or out of view. Moreover, reliably tracking features across an entire egg-laying behavioral session could potentially be challenging to DeepLabCut owing to significant changes in the background (the accumulation of new eggs or changes in the agar substrate appearance due to evaporation).

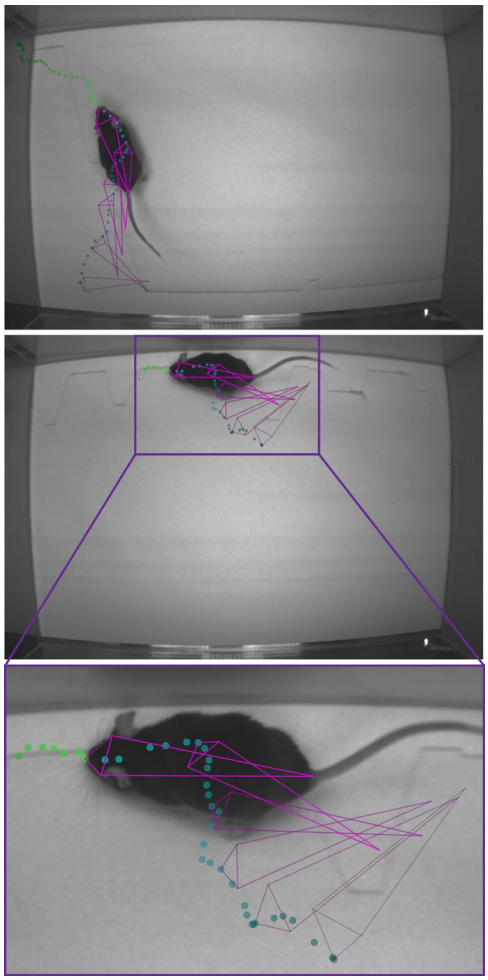

Fig. 3 | A trail-tracking mouse. Green and cyan dots show 30 time points in the future and past, respectively, of the snout positions during trail tracking. The dots are 33.3 ms apart. The body postures of the snout, ears and tail base at various past time points are depicted as magenta rhombi. Together those four points capture the body and head orientation of the mouse and illustrate the swinging head movements. The printed odor trail is visible in gray (see Supplementary Video 1).

To build toward an understanding of the behavioral patterns that surround egg-laying in an efficient way, we chose 12 distinct points on the body of the fly and labeled 589 frames of diverse orientation and posture from six different animals, labeling only those features that were visible within a given frame (see Methods).

We trained DeepLabCut with 95% of the data and found a test error of 4.17 ± 0.32 pixels (mean ± s.e.m.; corresponding to an average training error of 1.39 ± 0.01 pixels, $n = 3$ splits, mean ± s.e.m.). For reference, the average eye diameter (top to bottom) was 36 pixels and the average femur diameter was 8.5 pixels (although owing to the 3D body movements and chamber depth, sizes change depending on the fly's location). Figure 6a depicts some example test frames with human- and network-applied labels. Generalization to flies not used in the training set was excellent, and the feature detectors were robust to changes in orientation (Fig. 6b) and background (Fig. 6c and Supplementary Video 2). Although fly bodies are relatively rigid, which simplifies tracking, there are exceptions. For instance, the proboscis changes its visual appearance substantially during feeding behaviors. Yet the feature detectors can resolve fast motor movements such as the extension

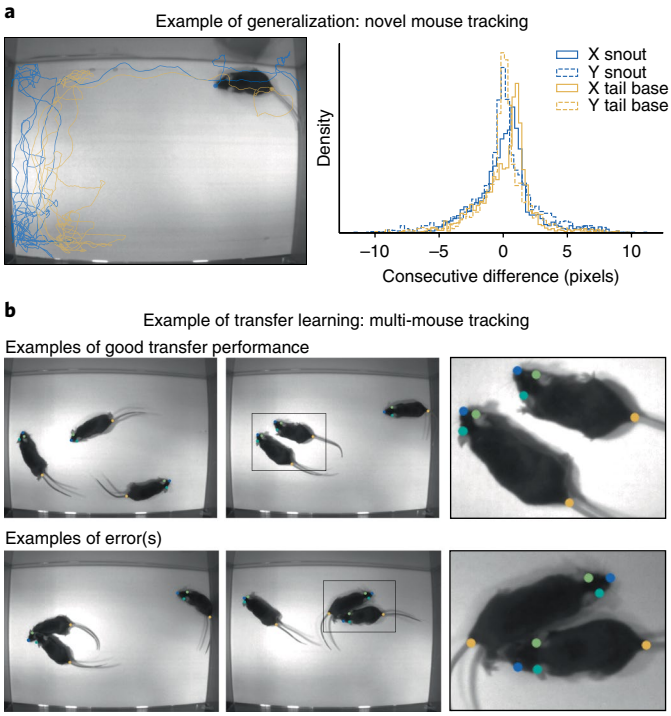

Fig. 4 | Generalization. a, Frame-by-frame extracted snout and tail-base trajectory in a novel mouse (not part of training set) during trail tracking on moving paper ground ($n = 2{,}331$ frames). Continuity of the trajectory suggests accurate training, which is also confirmed by histogram of pairwise, frame-by-frame differences in *x* and *y* coordinates. **b**, Body part predictions for images with multiple mice, showing predictions of a network that was trained with images containing only a single mouse. It readily detects body parts in images with multiple novel mice (top), unless they are occluding each other (bottom). Additionally, these mice are younger than the ones in the training set and thus have a different body shape.

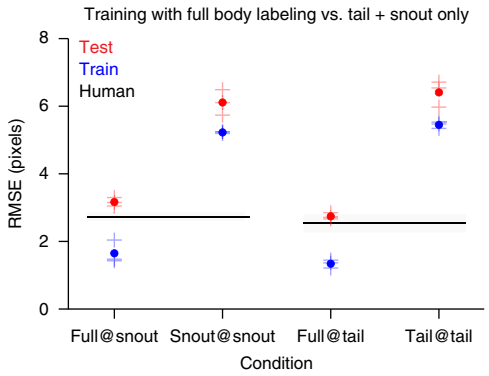

Fig. 5 | End-to-end training. We trained 'specialized' networks with only the snout or tail-base labels, respectively. We compare the RMSE against the full model that was trained on all body parts, but is also only evaluated on the snout or tail base, respectively (e.g., "full@snout" means the model trained with all body parts (full) and RMSE evaluated for the snout labels). Training (blue) and test (red) performance for the full model and specialized models trained with the same $n = 3$ 50% training set splits of the data (crosses) and average RMSE (dots). Although all networks have exactly the same information about the location of the snout or tail base during training, the network that also received information about the other body parts outperforms the 'specialized' networks.

and retraction of the proboscis (Fig. 6d). Thus, DeepLabCut allows accurate extraction of low-dimensional pose information from videos of freely behaving *Drosophila*.

Digit tracking during reaching. To further illustrate the versatility and capabilities of DeepLabCut, we tracked segments of individual digits of a mouse hand (Figs 1 and 7a). We recently established a head-fixed, skilled reaching task in mice[38], wherein mice grab a joystick with two degrees of freedom and pull it from a start location to a target location. While the joystick allows spatio-temporally accurate measurement of the joystick (hand) position during the pull, it neither constrains the hand position on the joystick nor provides position information during the reaches or between pulls (when the mice might or might not hold the joystick). Placing markers is difficult as the mouse hand is a small and highly complex structure with multiple bones, joints and muscles. Moreover, it is intrusive to mice and can disrupt performance. Therefore, in principle, markerless tracking is a promising approach for analyzing reaching dynamics. However, tracking is challenging because of the complexity of possible hand articulations, as well as the presence of the other hand in the background, making this task well suited to highlighting the generality of our DeepLabCut toolbox.

We labeled 13 body parts: 3 points per visible digit and the wrist (see Methods). Notably, we found that by using just 141 training frames we achieved an average test error of 5.21 ± 0.28 pixels (mean ± s.e.m.; corresponding to average training error 1.16 ± 0.03 pixels, $n = 3$ splits, mean ± s.e.m.). For reference, the width of a single digit was ~15 pixels. Figure 7a depicts some example test frames.

We believe that this application of hand pose estimation highlights the excellent generalization performance of DeepLabCut despite training with only a few images.

So far we have shown that the body part estimates derived from DeeperCut are highly accurate. But, in general, especially when sieving through massive datasets, the end user would like to have each point estimate accompanied by a confidence measure of the label location. The location predictions in DeeperCut are obtained by extracting the most likely region, based on a scalar field that represents the probability that a particular body part is in a particular region. In DeeperCut these probability maps are called score-maps, and predictions are generated by finding the point with the highest probability value (see Methods). The amplitude of the maximum can be used as a confidence readout to examine the strength of evidence for individual localizations of the individual parts to be detected. For instance, the peak probability of the digit tip is low when the mouse holds the joystick (in which case the finger tips are occluded). Similarly, when the features cannot be disambiguated, the likelihood becomes small (Fig. 7a). This confidence readout also works in other contexts: for instance, in the *Drosophila* example frames we only depicted the predicted body parts when the probability was larger than 10% (Fig. 6b–d). Using this threshold, the point estimate for the left leg was automatically excluded in Fig. 6c,d. Indeed, all occluded body parts are also omitted in Figs 6b and 7f.

Lastly, once a network is trained on the hand posture frames, the body part locations can be extracted from videos and used in many ways. Here we illustrate a few examples: digit positions during a reach across time (Fig. 7b; note that this trajectory comes from frame-by-frame prediction without any temporal filtering), comparison of movement patterns across body parts (Fig. 7c,d), dimensionality reduction to reveal the richness of mouse hand postures during reaching (Fig. 7e) and creating 'skeletons' based on the semantic meaning of the labels (Fig. 7f and Supplementary Video 3).

Discussion

Detecting postures from monocular images is a challenging problem. Traditionally, postures are modeled as a graph of parts, where each node encodes the local visual properties of the part in question, and then these parts are connected by spring-like links. This graph

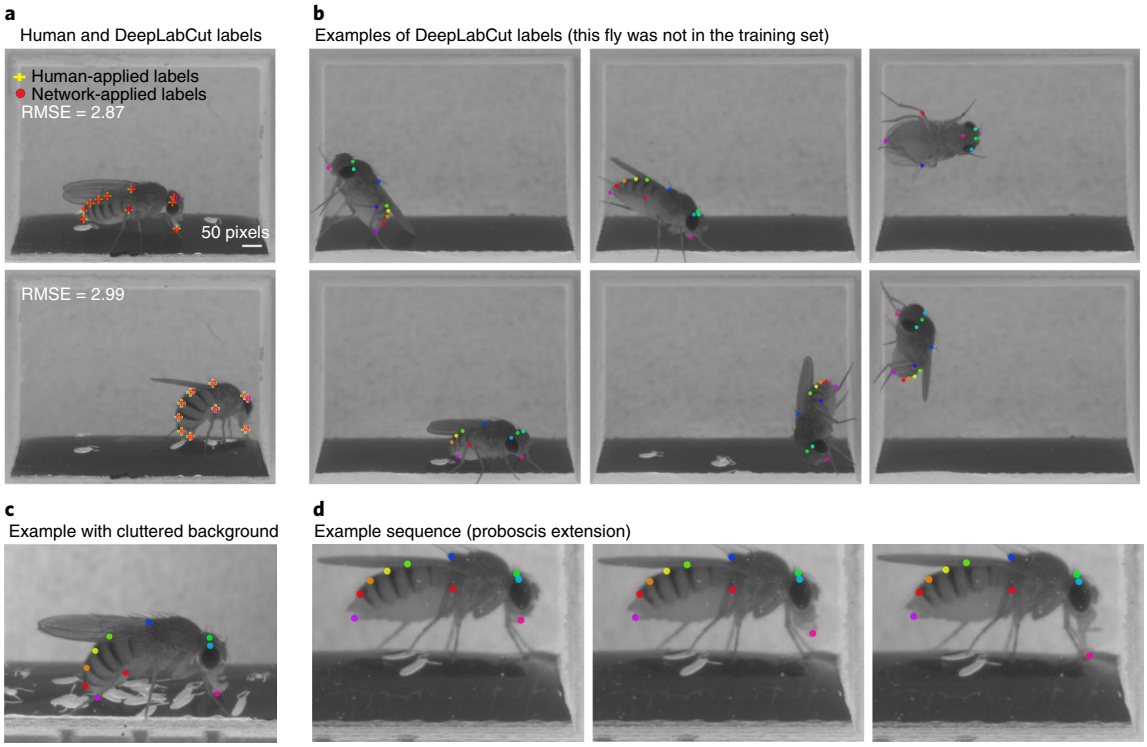

Fig. 6 | Markerless tracking of *Drosophila*. a, Example body part predictions closely match human annotator labels, shown for two frames that were not used in the training set (95% training set size). **b**, Example frames and body part predictions for a fly that was not part of the training data in various postures and orientations. **c**, Example labels for a fly against a cluttered background comprising numerous laid eggs. **d**, Example sequence of proboscis extension being automatically and accurately tracked. See Supplementary Video 2.

is then fit to images by minimizing some appearance cost function[18]. This minimization is hard to solve, but designing the model topology together with the visual appearance model is even more challenging[18,19]; this can be illustrated by considering the diversity of fruit fly orientations (Fig. 6) and hand postures we examined (Fig. 7). In contrast, casting this problem as a minimization with deep residual neural networks allows each joint predictor to have more than just local access to the image[19,25–28]. As a result of the extreme depth of ResNets, architectures such as DeeperCut have large receptive fields, which can learn to extract postures in a robust way[27].

Here we demonstrate that cutting-edge deep learning models can be efficiently used in the laboratory. Specifically, we leveraged the fact that adapting pretrained models to new tasks can dramatically reduce the amount of training data required, a phenomena known as transfer learning[27,31–34]. We first estimated the accuracy of a human labeler, who could readily identify the body parts of interest for odor-guided navigation, and then demonstrated that a deep architecture can achieve similar performance on detection of body parts such as the snout or the tail base after training on only a few hundred images. Moreover, this solution requires no computational body model, stick figure, time information or sophisticated inference algorithm. Thus, it can be quickly applied to completely different behaviors that pose qualitatively distinct computer vision challenges, such as skilled reaching in mice or egg-laying in *Drosophila*.

We believe that DeepLabCut will supplement the rich literature of computational methods for video analysis[6,8,16,20–22,39–42], where powerful feature detectors of user-defined body parts need to be learned for a specific situation or where regressors based on standard image features and thresholding heuristics[6,37,41] fail to provide satisfying solutions. This is particularly the case in dynamic visual environments—for example, those with varying background and reflective walls (Figs 2 and 6)—or when tracking highly articulated objects such as the hand (Fig. 7).

Dataset labeling and fine-tuning. Deep learning algorithms are extremely powerful and can learn to associate arbitrary categories to images[33,43]. This is consistent with our own observation that the training set should be free of errors (Fig. 2g) and approximate the diversity of visual appearances. Thus, to train DeepLabCut for specific applications, we recommend labeling maximally diverse images (i.e., different poses, different individuals, luminance conditions, etc.) in a consistent way and curating the labeled data well. The training data should reflect the breadth of the experimental data to be analyzed. Even for an extremely small training set, the typical errors can be small, but large errors for test images that are quite distinct from the training set can start to dominate the average error. One limitation for generalizing to novel situations comes from stochasticity in training set selection. Given that we only select a small number of training samples (a few hundred frames), it is plausible that images representing behaviors that are especially sparse or noisy (e.g., due to motion blur) could be suboptimally sampled or entirely excluded from the training data, resulting in difficulties at test time.

Therefore, a user can expand the initial training dataset in an iterative fashion using the score-maps. Specifically, errors can be addressed via post hoc fine-tuning of the network weights, taking advantage of the fact that the network outputs confidence estimates for its own generated labels (Fig. 7a and Methods). By using these confidence estimates to select sequences of frames containing a rare behavior (by sampling around points of high probability), or to find frames where reliably captured behaviors are largely corrupted with noise (by sampling points of low probability), a user can then selectively label frames based on these confidence criteria to generate a minimal yet additional training set for fine-tuning the network. Additionally, heuristics such as the continuity of body part trajectories can be used to select frames with errors. This selectively improves model performance on edge cases, thereby extending the

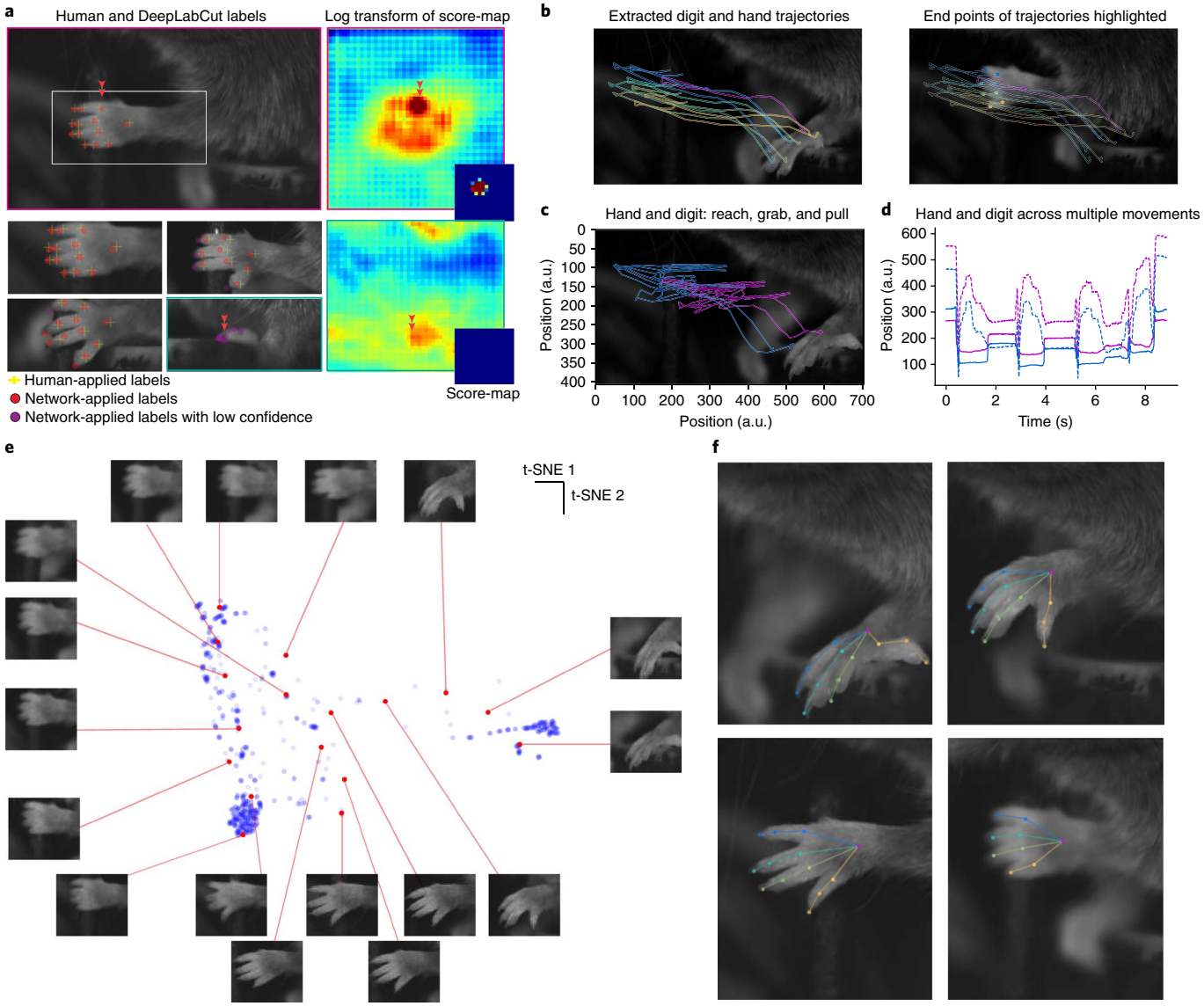

Fig. 7 | Markerless tracking of digits. a, Example test images labeled by human (yellow) and automated labeling (red or purple): the most likely joint is labeled in red if the probability of the peak in the score-map (see below) is larger than 10% at the peak, and purple otherwise. Here DeepLabCut was trained with only 141 images. In images where targets were occluded, the model typically reduces its confidence in the label position (purple), including in frames where the human was not confident enough to apply labels (bottom right) or when the digit tips are not visible (top right). Score-maps are shown to the right of the image for two example labels (highlighted by red arrows). For better visualization, log-transformed score-maps are also shown. **b**, Example analysis of trajectories after automated labeling. **c**, Extracted reach and pull trajectories from the wrist (purple) and digit 1 (blue) for three pulls; a.u., arbitrary units. **d**, Trajectories in **c**. Dashed lines are the *x* coordinate (lateral movements) and solid lines represent the *y* coordinate (pulling axis). **e**, On the basis of the predicted wrist location, we extracted $n = 1{,}139$ images of the hand from the video of one behavioral session and performed dimensionality reduction by *t*-distributed stochastic neighbor embedding (t-SNE) of those images. The blue point cloud shows the 2D embedding with several images corresponding to the red highlighted coordinates. This figure illustrates the richness of hand postures during this reaching task. **f**, Labeled body parts with connecting edges giving rise to a 'skeleton' of the hand (see Supplementary Video 3).

architecture's capacity for generalization in an efficient way. Such an active learning framework can be used to achieve a predefined level of confidence for all images with minimal labeling cost. Then, owing to the large capacity of the neural network that underlies the feature detectors, one can continue training the network with these additional examples.

We note, however, that not every low value in a probability score-map necessarily reflects erroneous detection. As we showed, low probabilities can also be indicative of occlusions, as in the case of the digit tips when the mouse is holding the joystick (Fig. 7). Here, multiple camera angles can be used to fully capture a behavior of interest, or heuristics (such as a body model) can be used to approximate occluded body parts using temporal and spatial information.

Speed and accuracy of DeepLabCut. Another important feature of DeepLabCut is that it can accurately transform large videos into low-dimensional time sequence data with semantic meaning, as the experimenter preselects the parts that will presumably provide the most information about the behavior being studied. In contrast to high-dimensional videos, such low-dimensional time sequence data are also highly amenable to behavioral clustering and analysis because of their computational tractability[6–8,44]. On modern hardware, pose extraction is also fast. For instance, one can process the

682 × 540 pixel frames of the *Drosophila* behavior at around 30 Hz on an NVIDIA 1080Ti GPU. Processing speeds scale with the frame size; for example, lower resolution videos with 204 × 162 pixel frames are analyzed at around 85 Hz. Such fast pose extraction can make this tool potentially amenable for real-time feedback[38,45] based on video-based posture estimates. This processing speed can be further improved by cropping input frames in an adaptive way around the animal and/or adapting the network architecture to speed up processing times.

Extensions. As presented, DeepLabCut extracts the posture data frame by frame, but one can add temporal filtering to improve performance (as for other approaches)[6,46,47]. Here we omitted such methods because of the high precision of the model without these additional steps, as well as to highlight the accurate prediction based on single frames solely driven by within-frame visual information in a variety of contexts.

While temporal information could indeed be beneficial in certain contexts, challenges remain to using end-to-end-trained deep architectures for video data to extract postures. Because of the curse of dimensionality, deep architectures on videos must rely on input images with lower spatial resolution, and thus the best-performing action recognition algorithms still rely on frame-by-frame analysis with deep networks pretrained on ImageNet as a result of hardware limitations[28,29,48]. As this is an active area of research, we believe this situation is likely to change with improvements in hardware (and in deep learning algorithms), and this should have a strong influence on pose estimation in the future. Therefore currently, in situations where occlusions are very common, such as in social behaviors, pairwise interactions could also be added to improve performance[6,13–18,27,29]. Here we have focused on the deep feature detectors alone to demonstrate remarkable transfer learning for laboratory tasks without the need for such extensions.

Conclusions

Together with this report, we provide an open source software package called DeepLabCut. The toolbox uses the feature detectors from DeeperCut and provides routines to (i) extract distinct frames from videos for labeling, (ii) generate training data based on labels, (iii) train networks to the desired feature sets, and (iv) extract these feature locations from unlabeled data (Fig. 1). The typical use case would be for an experimenter to extract distinct frames from videos and label the body parts of interest to create tailored part detectors. Then, after only a few hours of labeling and training the network, DeepLabCut can be applied to novel videos. While we demonstrate the utility of this toolbox on mice and *Drosophila*, there is no inherent limitation of this framework, and our toolbox can be applied to other model, or non-model, organisms in a diverse range of behaviors.

Methods

Methods, including statements of data availability and any associated accession codes and references, are available at https://doi.org/10.1038/s41593-018-0209-y.

Received: 8 April 2018; Accepted: 27 June 2018;
Published online: 20 August 2018

References

1. Tinbergen, N. On aims and methods of ethology. *Z. Tierpsychol.* **20**, 410–433 (1963).
2. Bernstein, N. A. *The Co-ordination and Regulation of Movements* Vol. 1 (Pergamon, Oxford and New York, 1967).
3. Krakauer, J. W., Ghazanfar, A. A., Gomez-Marin, A., MacIver, M. A. & Poeppel, D. Neuroscience needs behavior: correcting a reductionist bias. *Neuron* **93**, 480–490 (2017).
4. Ota, N., Gahr, M. & Soma, M. Tap dancing birds: the multimodal mutual courtship display of males and females in a socially monogamous songbird. *Sci. Rep.* **5**, 16614 (2015).
5. Wade, N. J. Capturing motion and depth before cinematography. *J. Hist. Neurosci.* **25**, 3–22 (2016).
6. Dell, A. I. et al. Automated image-based tracking and its application in ecology. *Trends Ecol. Evol.* **29**, 417–428 (2014).
7. Gomez-Marin, A., Paton, J. J., Kampff, A. R., Costa, R. M. & Mainen, Z. F. Big behavioral data: psychology, ethology and the foundations of neuroscience. *Nat. Neurosci.* **17**, 1455–1462 (2014).
8. Anderson, D. J. & Perona, P. Toward a science of computational ethology. *Neuron* **84**, 18–31 (2014).
9. Winter, D. A. *Biomechanics and Motor Control of Human Movement* (Wiley, Hoboken, NJ, USA, 2009).
10. Vargas-Irwin, C. E. et al. Decoding complete reach and grasp actions from local primary motor cortex populations. *J. Neurosci.* **30**, 9659–9669 (2010).
11. Wenger, N. et al. Closed-loop neuromodulation of spinal sensorimotor circuits controls refined locomotion after complete spinal cord injury. *Sci. Transl. Med.* **6**, 255ra133 (2014).
12. Maghsoudi, O. H., Tabrizi, A. V., Robertson, B. & Spence, A. Superpixels based marker tracking vs. hue thresholding in rodent biomechanics application. Preprint at https://arxiv.org/abs/1710.06473 (2017).
13. Pérez-Escudero, A., Vicente-Page, J., Hinz, R. C., Arganda, S. & de Polavieja, G. G. idTracker: tracking individuals in a group by automatic identification of unmarked animals. *Nat. Methods* **11**, 743–748 (2014).
14. Nakamura, T. et al. A markerless 3D computerized motion capture system incorporating a skeleton model for monkeys. *PLoS One* **11**, e0166154 (2016).
15. de Chaumont, F. et al. Computerized video analysis of social interactions in mice. *Nat. Methods* **9**, 410–417 (2012).
16. Matsumoto, J. et al. A 3D-video-based computerized analysis of social and sexual interactions in rats. *PLoS One* **8**, e78460 (2013).
17. Uhlmann, V., Ramdya, P., Delgado-Gonzalo, R., Benton, R. & Unser, M. FlyLimbTracker: An active contour based approach for leg segment tracking in unmarked, freely behaving *Drosophila*. *PLoS One* **12**, e0173433 (2017).
18. Felzenszwalb, P. F. & Huttenlocher, D. P. Pictorial structures for object recognition. *Int. J. Comput. Vis.* **61**, 55–79 (2005).
19. Toshev, A. & Szegedy, C. DeepPose: human pose estimation via deep neural networks. in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* 1653–1660 (IEEE, Piscataway, NJ, USA, 2014).
20. Dollár, P., Welinder, P. & Perona, P. Cascaded pose regression. in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2010* 1078–1085 (IEEE, Piscataway, NJ, USA, 2010).
21. Machado, A. S., Darmohray, D. M., Fayad, J., Marques, H. G. & Carey, M. R. A quantitative framework for whole-body coordination reveals specific deficits in freely walking ataxic mice. *Elife* **4**, e07892 (2015).
22. Guo, J. Z. et al. Cortex commands the performance of skilled movement. *Elife* **4**, e10774 (2015).
23. Krizhevsky, A., Sutskever, I. & Hinton, G. E. ImageNet classification with deep convolutional neural networks. in *Advances in Neural Information Processing Systems* Vol. 25 (eds. Pereira, F. et al.) 1097–1105 (Curran Associates, Red Hook, NY, USA, 2012).
24. 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 (IEEE, Piscataway, NJ, USA, 2016).
25. Wei, S.-E., Ramakrishna, V., Kanade, T. & Sheikh, Y. Convolutional pose machines. in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* 4724–4732 (IEEE, Piscataway, NJ, USA, 2016).
26. Pishchulin, L. et al. DeepCut: joint subset partition and labeling for multi person pose estimation. in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* 4929–4937 (IEEE, Piscataway, NJ, USA, 2016).
27. Insafutdinov, E., Pishchulin, L., Andres, B., Andriluka, M. & Schiele, B. DeeperCut: a deeper, stronger, and faster multi-person pose estimation model. in *European Conference on Computer Vision* 34–50 (Springer, New York, 2016).
28. Feichtenhofer, C., Pinz, A. & Zisserman, A. Detect to track and track to detect. in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* 3038–3046 (IEEE, Piscataway, NJ, USA, 2017).
29. Insafutdinov, E. et al. ArtTrack: articulated multi-person tracking in the wild. in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* 1293–1301 (IEEE, Piscataway, NJ, USA, 2017).
30. Andriluka, M., Pishchulin, L., Gehler, P. & Schiele, B. 2D human pose estimation: new benchmark and state of the art analysis. *in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* 3686–3693 (IEEE, Piscataway, NJ, USA, 2014).
31. Donahue, J. et al. DeCaf: a deep convolutional activation feature for generic visual recognition. in *International Conference on Machine Learning* 647–655 (PMLR, Beijing, 2014).
32. 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 (Curran Associates, Red Hook, NY, USA, 2014).
33. Goodfellow, I., Bengio, Y. & Courville, A. *Deep Learning* Vol. 1 (MIT Press, Cambridge, MA, USA, 2016).

34. Kümmerer, M., Wallis, T. S. & Bethge, M. DeepGaze II: reading fixations from deep features trained on object recognition. Preprint at https://arxiv.org/abs/1610.01563 (2016).
35. Khan, A. G., Sarangi, M. & Bhalla, U. S. Rats track odour trails accurately using a multi-layered strategy with near-optimal sampling. *Nat. Commun.* **3**, 703 (2012).
36. Li, Y. et al. Neuronal representation of social information in the medial amygdala of awake behaving mice. *Cell* **171**, 1176–1190.e17 (2017).
37. Robie, A. A., Seagraves, K. M., Egnor, S. E. & Branson, K. Machine vision methods for analyzing social interactions. *J. Exp. Biol.* **220**, 25–34 (2017).
38. Mathis, M. W., Mathis, A. & Uchida, N. Somatosensory cortex plays an essential role in forelimb motor adaptation in mice. *Neuron* **93**, 1493–1503.e6 (2017).
39. Drai, D. & Golani, I. SEE: a tool for the visualization and analysis of rodent exploratory behavior. *Neurosci. Biobehav. Rev.* **25**, 409–426 (2001).
40. Sousa, N., Almeida, O. F. X. & Wotjak, C. T. A hitchhiker's guide to behavioral analysis in laboratory rodents. *Genes Brain Behav.* **5** (Suppl. 2), 5–24 (2006).
41. Gomez-Marin, A., Partoune, N., Stephens, G. J., Louis, M. & Brembs, B. Automated tracking of animal posture and movement during exploration and sensory orientation behaviors. *PLoS One* **7**, e41642 (2012).
42. Ben-Shaul, Y. OptiMouse: a comprehensive open source program for reliable detection and analysis of mouse body and nose positions. *BMC Biol.* **15**, 41 (2017).
43. Zhang, C., Bengio, S., Hardt, M., Recht, B. & Vinyals, O. Understanding deep learning requires rethinking generalization. Preprint at https://arxiv.org/abs/1611.03530 (2016).
44. Berman, G. J. Measuring behavior across scales. *BMC Biol.* **16**, 23 (2018).
45. Kim, C. K., Adhikari, A. & Deisseroth, K. Integration of optogenetics with complementary methodologies in systems neuroscience. *Nat. Rev. Neurosci.* **18**, 222–235 (2017).
46. Stauffer, C. & Grimson, W.E.L. Adaptive background mixture models for real-time tracking. in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1999* Vol. 2, 246–252 (IEEE, Piscataway, NJ, USA, 1999).
47. Ristic, B., Arulampalam, S. & Gordon, N. *Beyond the Kalman Filter: Particle Filters for Tracking Applications* (Artech House, Norwood, MA, USA, 2003).
48. Carreira, J. & Zisserman, A. Quo vadis, action recognition? A new model and the kinetics dataset. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* 4724–4733 (IEEE, Piscataway, NJ, USA, 2017).

Acknowledgements

We are grateful to E. Insafutdinov and C. Lassner for suggestions on how to best use the TensorFlow implementation of DeeperCut. We thank N. Uchida for generously providing resources for the joystick behavior and R. Axel for generously providing resources for the *Drosophila* research. We also thank A. Hoffmann, J. Rauber, T. Nath, D. Klindt and T. DeWolf for a critical reading of the manuscript, as well as members of the Bethge lab, especially M. Kümmerer, for discussions. We also thank the β-testers for trying our toolbox and sharing their results with us. Funding: Marie Sklodowska-Curie International Fellowship within the 7th European Community Framework Program under grant agreement No. 622943 and DFG grant MA 6176/1-1 (A.M.); Project ALS (Women and the Brain Fellowship for Advancement in Neuroscience) and a Rowland Fellowship from the Rowland Institute at Harvard (M.W.M.); German Science Foundation (DFG) through the CRC 1233 on "Robust Vision" and from IARPA through the MICrONS program (M.B.).

Author contributions

Conceptualization: A.M., M.W.M. and M.B. Software: A.M. and M.W.M. Formal analysis: A.M. Experiments: A.M. and V.N.M. (trail-tracking), M.W.M. (mouse reaching), K.M.C. (*Drosophila*). Labeling: P.M., K.M.C., T.A., M.W.M., A.M. Writing: A.M. and M.W.M. with input from all authors.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary information is available for this paper at https://doi.org/10.1038/s41593-018-0209-y.

Reprints and permissions information is available at www.nature.com/reprints.

Correspondence and requests for materials should be addressed to M.W.M.

Publisher's note: Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Methods

DeepLabCut toolbox. This publication is accompanied by open source Python code for selecting training frames, checking human annotator labels, generating training data in the required format, and evaluating the performance on test frames. The toolbox also contains code to extract postures from novel videos with trained feature detectors. Thus, this toolbox allows one to train a tailored network based on labeled images and to perform automatic labeling for novel data. See https://github.com/AlexEMG/DeepLabCut for details.

While the presented behaviors were recorded in grayscale under infrared or normal lighting conditions, DeepLabCut can also be used for color videos. There is no inherent limitation to the cameras that can be used to collect videos that can subsequently be analyzed with our toolbox. Please see http://www.mousemotorlab.org/deeplabcut for more example behaviors (including examples of color videos).

Mouse odor trail-tracking. The trail-tracking behavior is part of an investigation into odor-guided navigation wherein one or more wild-type (C57BL/6J) male mice run on a paper spool following odor trails. These experiments were carried out in the laboratory of Venkatesh Murthy at Harvard University and will be published elsewhere. For trail-tracking, we extracted 1,080 random, distinct frames from multiple experimental sessions observing 7 different mice. Data were recorded at 30 Hz by two different cameras: the 640×480 pixels images were acquired with a Point Grey Firefly FMVU-03MTM-CS and the ~1,700×1,200 pixel images with a Point Grey Grasshopper 3 4.1MP Mono USB3 Vision, CMOSIS CMV4000-3E12. The latter images are prohibitively large to process without downsampling, and therefore we cropped around mice to generate images that were approximately 800×800 pixels. One human annotator was instructed to localize the snout, the tip of the left and right ear and the base of the tail in the example images on two different occasions (using Fiji[49]), which generated two distinct label sets (>1 month apart to reduce memory bias; see Fig. 1).

Mouse reach and pull joystick task. Experimental procedures for the training of the joystick behavior and the construction of the behavioral set-up can be found in Mathis et al.[38]. In brief, head-fixed mice were trained to reach, grab and pull a joystick for a liquid reward. To generate a train/test set of images, we labeled 159 frames at 13 locations: 3 points per digit—the digit tip, the joint in the middle and the base of the digit (which roughly correspond to the proximal interphalangeal joint and the metacarpophalangeal joint, respectively)—as well as the base of the hand (wrist). The data were collected across 5 different mice (C57BL/6J, male and female) and were recorded at 2,048×1,088 resolution with a frame rate of 100–320 Hz. For tracking the digits, we used the supplied toolbox code to crop the data to extract only regions of interest containing the movement of the forelimb to limit the size of the input image to the network.

All surgical and experimental procedures for mice were in accordance with the National Institutes of Health Guide for the Care and Use of Laboratory Animals and approved by the Harvard Institutional Animal Care and Use Committee.

***Drosophila* egg-laying behavior.** Experiments were carried out in the laboratory of Richard Axel at Columbia University and will be published elsewhere. In brief, egg-laying behavior in female *Drosophila* (Canton-S strain) was observed in custom-designed 3D-printed chambers (Protolabs). Individual chambers were 4.1 mm deep and tapered from top to bottom, with the top dimensions 7.3 mm×5.8 mm and the bottom dimensions 6.7 mm×4.3 mm. One side of the chamber opened to a reservoir within which 1% agar was poured and allowed to set. Small acrylic windows were slid into place along grooves at the top and bottom to enclose the fly within the chamber and to allow viewing. The chamber was illuminated by a 2-inch off-axis ring light (Metaphase) and video recording was performed from above the chamber using an infrared-sensitive CMOS camera (Basler) with a 0.5× telecentric lens (Edmund Optics) at 20 Hz (682×540 pixels). We identified 12 distinct points of interest to quantify the behavior of interest on the body of the fly. One human annotator manually extracted 589 distinct and informative frames from six different animals, labeling only those features that were visible within a given frame. The 12 points comprise 4 points on the head (the dorsal tip of each compound eye, the ocellus and the tip of the proboscis), the posterior tip of the scutellum on the thorax, the joint between the femur and tibia on each metathoracic leg, the abdominal stripes on the four most posterior abdominal segments (A3–A6) and the ovipositor.

Labeled dataset set selection. No statistical methods were used to predetermine sample sizes for labeled frames, but our sample sizes are similar to those reported in previous publications. The labelers were blinded to whether the frames would be assigned to training or test datasets (as the frames were randomized across splits). For each replicate (i.e., split of the dataset), frames were randomly assigned to the test or training set. No data or experimental animals (mice or *Drosophila*) were excluded from the study.

Deep feature detector architecture. We employ strong body part detectors, which are part of state-of-the art algorithms for human pose estimation called DeeperCut[26,27,29]. Those part detectors build on state-of-the-art object recognition architectures, namely extremely deep residual networks (ResNet)[24]. Specifically, we use a variant of the ResNet with 50 layers, which achieved outstanding performance in object recognition competitions[24]. In the DeeperCut implementation, the ResNets were adapted to represent the images with higher spatial resolution, and the softmax layer used in the original architecture after the conv5 bank (as would be appropriate for object classification) was replaced by 'deconvolutional layers' that produce a scalar field of activation values corresponding to regions in original image. This output is also connected to the conv3 bank to make use of finer features generated earlier in the ResNet architecture[27]. For each body part, there is a corresponding output layer whose activity represents probability 'score-maps'. These score-maps represent the probability that a body part is at a particular pixel[26,27]. During training, a score-map with positive label 1 (unity probability) is generated for all locations up to ϵ pixels away from the ground truth per body part (distance variable). The ResNet architecture used to generate features is initialized with weights trained on ImageNet[24], and the cross-entropy loss between the predicted score-map and the ground-truth score-map is minimized by stochastic gradient descent[27]. Around 500,000 training steps were enough for convergence in the presented cases, and training takes up to 24 h on a GPU (NVIDIA GTX 1080 Ti; note that typically the loss starts to slowly decay early in training; see Fig. 2b). We used a batch size of 1, which allows us to have images of different sizes, decreased the learning rate over training and performed data augmentation during training by rescaling the images (as in DeeperCut, but we used a range of 50% to 150%). We also tested further data augmentation by additionally training with 7 rotated frames per training image (rotation group—angles independently and uniformly sampled (uid) from [−8,8] degrees) as well as 9 rotated and 14 partial images per training images (rotation and translation group—angles uid from [−10,10] degrees, as well as uid subimages amounting to relative shifts). Unless otherwise noted, we used a distance variable $\epsilon = 17$ (pixel radius) and scale factor 0.8 (which affects the ratio of the input image to output score-map). We cross-validated the choice of ϵ for a higher resolution output (scale factor = 1) and found that the test performance was not improved when varying ϵ widely, but the rate of performance improvement was strongly decreased for small ϵ (Supplementary Fig. 2). We also evaluated deeper networks with 101 layers, ResNet-101, as well as ResNet-101ws (with intermediate supervision, Supplementary Fig. 2b); more technical details can be found in Insafutdinov et al.[27].

Evaluation and error measures. The trained network can be used to predict body part locations. At any state of training, the network can be presented with novel frames, for which the prediction of the location of a particular body part is given by the peak of the corresponding score-map. This estimate is further refined on the basis of learned correspondences between the score-map grid and ground truth joint positions[26,27,29]. In the case of multiple mice, the local maxima of the score-map are extracted as predictions of the body part locations (Fig. 4).

As discussed in the main text, a user can continue to fine-tune the network for increasing generalization to large datasets to reduce errors. One can use features of the score-maps such as the amplitude and width, or heuristics such as the continuity of body part trajectories, to identify images for which the decoder might make large errors. Images with insufficient automatic labeling performance that are identified in this way can then be manually labeled to increase the training set and iteratively improve the feature detectors.

To compare between datasets generated by the human scorer, as well as with or between model-generated labels, we used the Euclidean distance (root mean square error, RMSE) calculated pairwise per body part. Depending on the context, this metric is either shown for a specific body part, averaged over all body parts, or averaged over a set of images. To quantify the error across learning, we stored snapshots of the weights in TensorFlow[50] (usually every 50,000 iterations) and evaluated the RMSE for predictions generated by these frozen networks post hoc. Note that the RMSE is not the loss function minimized during training. However, the RMSE is the relevant performance metric for assessing labeling precision in pixels.

The RMSE between the first and second annotation is referred to as human variability. In figures we also depict the 95% confidence interval for this RMSE, whose limits are given as mean ± 1.96 times the s.e.m. (Figs 2c,d,f, 4 and 5 and Supplementary Fig. 2a–d). Depending on the figure, the RMSE is averaged over all or just a subset of body parts. For the *Drosophila* and the mouse hand data, we report the average test RMSE for all body parts with likelihood larger than 10%.

In Fig. 7 we extracted cropped images of the hand from full frames ($n = 1{,}139$) by centering it using the predicted wrist position. We then performed dimensionality reduction by t-SNE embedding of those images[51] and randomly selected certain sufficiently distant points to illustrate the corresponding hand postures.

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

Code availability. We provide the code for the DeepLabCut toolbox at https://github.com/AlexEMG/DeepLabCut.

Data availability. Data are available from the corresponding author upon reasonable request.

References

49. Schindelin, J. et al. Fiji: an open-source platform for biological-image analysis. *Nat. Methods* **9**, 676–682 (2012).
50. Abadi, M. et al. TensorFlow: a system for large-scale machine learning. Preprint at https://arxiv.org/abs/1605.08695 (2016).
51. Pedregosa, F. et al. Scikit-learn: machine learning in Python. *J. Mach. Learn. Res.* **12**, 2825–2830 (2011).

nature research

Corresponding author(s): MATHIS

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 Authors & Referees and the Editorial Policy Checklist.

Statistical parameters

When statistical analyses are reported, confirm that the following items are present in the relevant location (e.g. 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
☐	☒	An indication of 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 statistics 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
☐	☒	Clearly defined error bars *State explicitly what error bars represent (e.g. SD, SE, CI)*

Our web collection on statistics for biologists may be useful.

Software and code

Policy information about availability of computer code

Data collection	custom written scripts in LabView 2013 was used to collect the odor-guided navigation, social mice videos, and mouse reaching data. Video recording of the Drosophila was performed from above the chamber using an IR-sensitive CMOS camera (Basler).
Data analysis	All analysis and model generation code is deposited at https://github.com/AlexEMG/deeplabcut

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/reviewers upon request. 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

There is a Data Availability section that provides the link to the github.com code repository.

Field-specific reporting

Please select 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/authors/policies/ReportingSummary-flat.pdf

Life sciences study design

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

Sample size	Aside from empirically testing how many frames are required to reach human-level accuracy (Figure 2), no predetermined sample size criteria was used.
Data exclusions	No data was excluded.
Replication	Videos taken from multiple flies and/or multiple mice in each behavior was recorded. Model analysis was always performed with multiple randomized train/test splits, and run at least three times (i.e. see Figure 2) to validate the conclusions. All attempts at replication were successful.
Randomization	Data was randomly assigned to the training or test splits.
Blinding	Labelers were not told which videos or labeled data would be included in the training or test splits.

Reporting for specific materials, systems and methods

Materials & experimental systems

n/a	Involved in the study
☒	☐ Unique biological materials
☒	☐ Antibodies
☒	☐ Eukaryotic cell lines
☒	☐ Palaeontology
☐	☒ Animals and other organisms
☒	☐ Human research participants

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	Mice: with a C57BL\6J background genotype of both sexes (female and male) were used, aged P60 to P360. Number of animals are reported in the methods section. All mice experiments were conducted with IUCAC approval at Harvard. For the Drosophila melanogaster, experiments were performed on adult females that were 5-6 days post-eclosion (Canton-S strain).
Wild animals	The study did not involve wild animals.
Field-collected samples	No animals were collected from the field.