© 2020. Published by The Company of Biologists Ltd | Journal of Experimental Biology (2020) 223, jeb226720. doi:10.1242/jeb.226720

METHODS & TECHNIQUES

Integrating XMALab and DeepLabCut for high-throughput XROMM

J.D. Laurence-Chasen[1,*], Armita R. Manafzadeh[2], Nicholas G. Hatsopoulos[1], Callum F. Ross[1] and Fritzie I. Arce-McShane[1,*]

ABSTRACT

Marker tracking is a major bottleneck in studies involving X-ray reconstruction of moving morphology (XROMM). Here, we tested whether DeepLabCut, a new deep learning package built for markerless tracking, could be applied to videoradiographic data to improve data processing throughput. Our novel workflow integrates XMALab, the existing XROMM marker tracking software, and DeepLabCut while retaining each program's utility. XMALab is used for generating training datasets, error correction and 3D reconstruction, whereas the majority of marker tracking is transferred to DeepLabCut for automatic batch processing. In the two case studies that involved an *in vivo* behavior, our workflow achieved a 6 to 13-fold increase in data throughput. In the third case study, which involved an acyclic, post-mortem manipulation, DeepLabCut struggled to generalize to the range of novel poses and did not surpass the throughput of XMALab alone. Deployed in the proper context, this new workflow facilitates large scale XROMM studies that were previously precluded by software constraints.

KEY WORDS: XMALab, DeepLabCut, XROMM, Marker tracking, Deep learning

INTRODUCTION

Data processing in kinematics workflows can be a time-consuming and laborious task, especially when three-dimensional (3D) reconstruction requires the integration of data from multiple cameras. In marker-based XROMM (X-ray reconstruction of moving morphology; Brainerd et al., 2010), every radiopaque marker in every frame of two X-ray videos must be accurately tracked. This step has been streamlined by the open-source program XMALab (Knörlein et al., 2016), which offers a suite of features for marker detection, visualization, and tracking. Marker tracking remains a major bottleneck in the XROMM workflow, however, limiting the feasibility of studies that require large sample sizes across multiple individuals or species (cf. Gintof et al., 2010; Granatosky et al., 2019; Iriarte-Diaz et al., 2017; Martinez et al., 2018).

In the past several years, deep learning, a type of machine learning, has emerged as a powerful tool for automating pose estimation in kinematics workflows (Graving et al., 2019; Insafutdinov et al., 2016; for a recent review Mathis and Mathis, 2019; Pereira et al., 2019). In particular, the open-source deep learning toolbox DeepLabCut (Mathis et al., 2018; Nath et al., 2018) has been rapidly and widely adopted in the scientific community. DeepLabCut was designed for markerless, automatic tracking of body parts in RGB/monochrome camera videos and has been used in a disparate range of study systems with impressive performance and robustness (Labuguen et al., 2019; Owen et al., 2019; Parmiani et al., 2019; Stringer et al., 2019; and many others).

The degree to which DeepLabCut's utility in digitizing visible light videos can be transferred to the biplanar videoradiographic data at the core of XROMM is not known. Whereas in visible light video different body parts are immediately distinguishable by their shape and appearance alone, in X-ray videos the markers are often identical in appearance (small black spheres), and thus only identifiable in their broader spatiotemporal context. Moreover, as many XROMM studies aim to quantify subtle motions, the desired error threshold in marker tracking is extremely small (i.e. reprojection error $\leq$1 pixel; Brainerd et al., 2010). The graphical user interface (GUI) and reconstruction features of XMALab are specifically designed for the accurate identification and tracking of markers under these challenging conditions.

The purpose of this paper is to present a workflow that integrates DeepLabCut into the existing XROMM data processing pipeline, retaining the XMALab labeling GUI and reconstruction tools while offloading initial batch tracking to DeepLabCut. We compare the performance of our pipeline to the standard XMALab workflow on three different datasets, each with different behaviors and marker sets. Strengths and weakness of the two different pipelines are discussed, and instructions and recommendations for the full implementation of our pipeline are provided.

MATERIALS AND METHODS

Software availability

Open-source Python code under the name XROMM_DLCTools and Jupyter Notebooks for the full implementation of our integrated workflow are available at github.com/jdlaurence/XROMM_DLCTools. DeepLabCut is available at github.com/AlexEMG/DeepLabCut.

Data flow

The flow of data through our pipeline is cyclic (Fig. 1). A training dataset – a relatively small subset of paired camera 1 and camera 2 frames – is tracked as accurately as possible in XMALab. Then those tracked data, in the form of 2D points and their corresponding images, are migrated to DeepLabCut where they are used to train an artificial neural network. Different videos can then be directly input to DeepLabCut for automated tracking. After the network predicts the marker locations in the new videos, the predicted 2D points are brought back into XMALab for error correction, performance

[1]Department of Organismal Biology and Anatomy, The University of Chicago, 1027 E 57th St, Chicago, IL 60637, USA. [2]Department of Ecology and Evolutionary Biology, Brown University, 80 Waterman Street, Providence, RI 02912, USA.

*Authors for correspondence (jdlaurence@uchicago.edu; fritziea@uchicago.edu)

J.D.L-C., 0000-0001-6192-175X; A.R.M., 0000-0001-5388-7942; N.G.H., 0000-0002-4913-6051; C.F.R., 0000-0001-7764-761X; F.I.A.-M., 0000-0001-6616-3564

Received 14 April 2020; Accepted 2 July 2020

Journal of Experimental Biology (2020) 223, jeb226720. doi:10.1242/jeb.226720

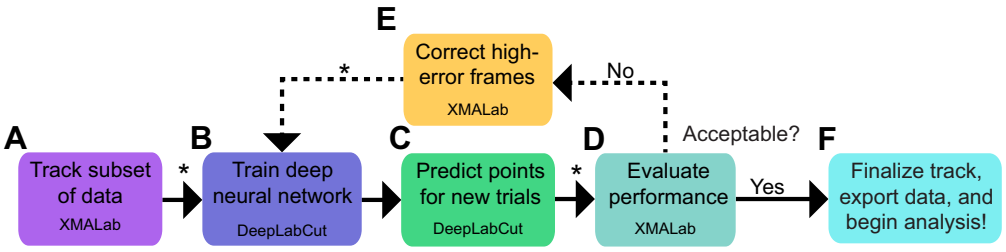

Fig. 1. Integrated XMALab and DeepLabCut workflow for marker tracking. (A) The user begins by tracking approximately 200–500 frames from the dataset in XMALab. (B) Those frames serve as the training dataset for a deep neural network trained with DeepLabCut. (C) This network can then predict 2D points for new trials. (D) The predicted points are imported back into XMALab and measures of tracking quality (e.g. reprojection and rigid body error) are used to determine whether the project-specific performance criteria are met. (E) If errors are too high, selected frames can be corrected, added to the training dataset, and steps B–D repeated. (F) Once performance is acceptable, the user corrects any remaining errors in XMALab, and can export the data (3D points and rigid body transformations) for analysis. Asterisks indicate that the step is performed by an XROMM_DLCTools function.

evaluation, and 3D reconstruction. If the network's performance is sub-optimal then areas of high error/poor performance can be manually corrected in XMALab, those corrected frames added to the training dataset, and the process repeated until the desired performance is achieved.

Training dataset generation

The composition of the training dataset is perhaps the single most important factor in DeepLabCut's performance. The network will generalize, i.e., perform well on new trials, when frames in the training dataset completely capture the variation in the full dataset. Thus, for each test case, we tracked 250–500 consecutive frames from 3–6 trials (approximately 800–2000 frames in total). We intentionally selected regions that contained the most variation in posture and/or stages of the behavior of interest. Once all frames were tracked and 2D points exported from XMALab, we created a DeepLabCut project using standard DeepLabCut practice. The project configuration file was edited to match the specifics of the dataset (i.e. marker names, file location paths, etc.).

After the project was successfully created, we used the DLCTools Python function *xma_to_dlc* to create a DeepLabCut-ready training dataset. The user specifies the location of the data and the desired size of the training dataset, and the function reads XMALab 2D points files and extracts point positions from frames with tracked data. It also extracts the corresponding video frames, either from avi files or from jpg stacks, and converts them to png images. The output of the function is identical to the output of the native DeepLabCut labeling GUI; thus, after this step the user proceeds to the established DeepLabCut workflow, starting with the function *create_training_dataset*.

Given the redundancy of postures inherent in consecutive frames of high-speed video, as well as the added computational cost of a larger training dataset, we uniformly subsampled the tracked frames by setting the 'nframes' argument to either 500 or 750. This meant the training dataset was composed of every other initially tracked frame. The choice to create an initial training dataset with more frames than the recommended number for a DeepLabCut study (~200) was made based on the inherent visual complexity and challenge of identifying multiple, visually homogeneous markers in X-ray videos. The impact of smaller and larger initial training datasets on performance is discussed in the Results section.

Network training and analysis

Once the training dataset is generated, the standard DeepLabCut workflow is followed. The functions *create_training_dataset* and *train_network* were used to train a single neural network whose weights were optimized for both camera 1 and camera 2 videos. In all cases, we used ResNet-101. We allowed training to run until DeepLabCut's native cross-entropy loss function plateaued, typically between 200,000 and 500,000 iterations. DLCTools supports the use of separate neural networks for each camera plane, if the user chooses. This would double the amount of training but may improve performance. If the user wishes to analyze visible light videos alongside the x-ray videos, a separate network should be used.

The DLCTools function *analyze_xromm_videos* calls the native *analyze_videos* function to predict points for new trials. It automatically detects the camera 1 and camera 2 videos and combines DeepLabCut's predicted points output into a single 2D file in XMALab format. The predicted 2D points files can then be imported into a XMALab file with the corresponding calibration.

Performance evaluation

We evaluated the trained neural network's tracking performance in two ways. The first method, which we do not recommend to be used exclusively, consists of DeepLabCut's *evaluate_network* function, which measures the mean 2D distance between the predicted points and the user-provided (via XMALab) 'ground-truth' points for the test fraction of the training dataset. In our experience, this native function is not a sufficient measure of tracking quality for XROMM data for several reasons: (1) camera calibration information is not used to measure 3D error (see Discussion), (2) the function can be affected by over-fitting of the network to the training dataset, and (3) the function cannot, by definition, assess the performance of a network's tracking of a novel trial. In other words, the error values provided by *evaluate_network* may not indicate that the network is ready to generalize and perform adequately on novel trials.

The second means of evaluating the network's tracking performance, which we recommend, is the suite of error measurement tools in XMALab. We use individual marker reprojection error (see Knörlein et al., 2016) as an overall heuristic for tracking performance. As the goal of marker tracking with DeepLabCut is to accelerate the process while maintaining accuracy, we determined the reprojection error value at which the measured kinematic variables did not meaningfully differ by tracking mode. These variables were joint coordinate system (JCS; Grood and Suntay, 1983) data and, in the case of the tongue data set, 3D marker positions.

For each network iteration (see following section), we tested a novel trial that had also been tracked in XMALab alone. Thus, there were two sets of data for that trial: data tracked with DeepLabCut, and data tracked with XMALab. We took the tracked data through the XROMM pipeline and then calculated the mean difference between corresponding variables across all test frames. We deemed the tracking acceptable if this value is smaller than the precision

Journal of Experimental Biology (2020) 223, jeb226720. doi:10.1242/jeb.226720

threshold for that variable (*sensu* Menegaz et al., 2015). When this threshold was reached, the reprojection error values for all points were recorded, as was the time spent for post-DeepLabCut corrections in XMALab.

This approach necessitates meticulous tracking in XMALab for the training dataset and for the comparison dataset. When evaluating DeepLabCut output, single frames or regions of frames that exhibit critically poor tracking may not be captured by the mean reprojection or rigid body error. Thus, it is essential that the predicted points data are migrated into XMALab and the reprojection error, 2D position and rigid body error plots are visually inspected for large outliers.

Training dataset augmentation

When DeepLabCut's tracking quality is not satisfactory, areas of poor performance can be manually corrected and added to the training dataset for the network to 'learn'. The user identifies high-error frames by visual inspection of the reprojection error and rigid body error traces, as well as the gestalt appearance of the tracking in the main window; i.e. are the crosshairs on the markers? The exact numbers that constitute poor performance depend on the specific study, but typically involve reprojection errors over 2 pixels, and rigid body errors over 0.5 mm. Once the user corrects all markers in a frame, they add the frame number to a frame index spreadsheet that contains the trial name and frames corrected from that trial. The DLCTools function *add_frames* reads this file, extracts the corrected frame images and their new 2D point data, and appends them to the training dataset. The user can then repeat network training and re-analyze the same videos with improved marker prediction. Exact file format and folder structure for the use of this function are detailed in the online package instructions.

Test cases

In order to assess the accuracy and limitations of our new workflow, we tested it on three previously collected datasets: pig feeding, monkey feeding and bird leg range of motion (ROM). Importantly, the datasets share few similarities; they were collected on three different biplanar radiography systems and differ in species, number of markers, marker size and marker locations. Example images from each dataset are provided (Fig. S1). While these case studies are certainly not exhaustive in terms of taxa or behaviors, their differences provide a basis for evaluating the degree to which this workflow can be generalized to future XROMM studies.

For each case study, we report the training parameters, reprojection error values, and time spent digitizing to achieve reconstructions that are statistically indistinguishable from those made from data tracked in XMALab alone (following the methods described above). As correction in XMALab is the final step in both workflows, and subject to user bias, we did not perform additional inferential statistics across conditions (e.g. mean reprojection error). The variables used to make this comparison are detailed in the following sections. All experiments were performed in compliance with Brown University and The University of Chicago's Institutional Animal Care and Use Committee protocols.

Technical details

All network training and analyses for the test cases were performed with DeepLabCut 2.1 (installed via Anaconda environment) on Windows 10 and a NVIDIA GeForce 1080 Ti GPU. ResNet-101 was used and training was stopped when the cross-entropy loss plateaued or fell below 0.005, typically at 200,000–500,000 iterations. The 'global_scale' parameter was set to 1, and 'pos_dist_threshold' was left at the default 17. The native DeepLabCut filter (arima or median) was not used; we filtered the predicted points in XMALab only. It is possible that a combination of the two filters could improve performance, and the user can choose which they wish to use. The two factors that have the largest impact on training and analysis speed are resolution and batch size, but speed is also influenced by training dataset size, number of network layers, and whether image augmentation is used ('imgaug' setting; strongly recommend, but not used here). The pig dataset took approximately 10 h of training to reach a loss plateau, whereas the monkey dataset took closer to 30 h. We recommend exploring and tuning the many DeepLabCut training parameters (e.g. 'pos_dist_threshold' or 'dataset_type') to find the settings that maximize performance for their specific dataset.

Study 1: minipig feeding

These publicly available minipig (*Sus scrofa*) feeding data were collected with C-arm videofluoroscopes (image resolution: 1024×1024 pixels) and have been used in XROMM tutorials and software testing for the last decade (Brainerd et al., 2010; Knörlein et al., 2016). Ten 1 mm tantalum markers – five in the cranium and five in the mandible – exhibit typical difficult-to-track characteristics; as the pig feeds unconstrained, the markers occasionally cross and occlude one another or enter areas of low contrast. The first iteration of the training dataset comprised a total of 500 frames from three trials of SusD feeding (dataset 2006-12-29). The network was then tested on a novel trial, specifically, the 435-frame trial from the same date that has been used for previous teaching and validation studies. We used the six degrees of freedom from the temporomandibular JCS as the output variables for performance comparisons.

Study 2: macaque feeding

In this study, performed at The University of Chicago XROMM Facility, a male rhesus macaque (*Macaca mulatta)* fed on grapes and gummy bears while head-fixed. The data were collected at 200 Hz (image resolution: 900×900 pixels) with an X-ray technique of 100–105 kilovolt peak (kVp) and 10–12.5 milliamperes (mA). A total of 24 tantalum markers, all 1 mm in diameter, were located as follows: 4 in the cranium, 4 in the mandible, 1 in the hyoid and 15 in the tongue. The tongue markers, being in a soft body, moved in complex ways, frequently crossing and occluding one another. The combination of numerous bone and soft tissue markers makes these data extremely difficult to track; an expert XMALab user took approximately 8 h to track a 10 s, 2000 frame trial. The first iteration of the training dataset comprised 750 frames sampled from six trials. Following sub-optimal performance on the test trial, the training dataset was augmented twice, such that the final training dataset comprised 1250 frames. Final manual corrections of output involved setting a reprojection error threshold at 2–2.5 pixels and correcting all frames that exceeded that threshold. We used the temporomandibular joint rotation values and tongue point 3D positions as the output variables for performance comparisons.

Study 3: guineafowl range of motion

In this study, performed at the W.M. Keck Foundation XROMM Facility at Brown University, the hind limb of a helmeted guineafowl (*Numida meleagris*) was physically manipulated post mortem to assess the ROM of the bird's hip, knee, and ankle joints. The data were collected at 50 Hz (image resolution: 1760×1760 pixels) with an X-ray technique of 70–85 kVp and 200 mA. A total of 12 0.8 mm zirconium oxide markers were placed in the pelvis, femur, tibiotarsus and tarsometatarsus – 3 in each element. Unlike

Journal of Experimental Biology (2020) 223, jeb226720. doi:10.1242/jeb.226720

the previous two studies, these data do not involve a cyclic behavior; in fact, the aim of the study was to explore each joint's full ROM through intentionally non-cyclic, non-repeated movements (Kambic et al., 2017; Manafzadeh and Padian, 2018). An expert XMALab user took approximately 10 h to track an 1800 frame trial. The first iteration of the training dataset comprised 750 frames from four trials. The training dataset was augmented twice, and the final training dataset comprised 1500 frames. We used the rotations at the three joints as the output variables for performance comparisons.

RESULTS

Study 1: minipig feeding

When applied to the pig feeding dataset, DeepLabCut rapidly reached XMALab-level performance. DeepLabCut's raw (i.e. pre-XMALab correction) marker predictions for a novel trial exhibited rigid body errors and JCS rotation values that fell within the precision threshold of the study (Fig. 2A,C). Mean reprojection error of individual points, however, was higher in the trial tracked with DeepLabCut (0.51± 0.25 s.d. pixels) as compared with XMALab (0.16±0.02 pixels). This difference in mean reprojection error persisted after manual correction of select, high error frames in XMALab. Nevertheless, the difference in measured JCS variables never fell outside of the error threshold. DeepLabCut was immediately robust to the cyclic crossing of select markers that consistently required manual intervention when tracking in XMALab. After training the neural network on the initial training dataset, time to fully track 1000 frames decreased from approximately 30 min with XMALab alone to 5 min with the integrated workflow, constituting a six-fold increase in throughput when tracking cranial and mandibular markers.

Study 2: macaque feeding

DeepLabCut quick achieved XMALab-level performance when tracking the markers in the two rigid bodies – the cranium and mandible. Before any manual correction, mean rigid body error was comparable to that of the trial tracked using XMALab only. Likewise, the temporomandibular JCS *y*- and *z*-axis rotations fell within the respective variable's precision thresholds (Fig. 2A,B). As in the pig dataset, DeepLabCut-predicted marker locations exhibited higher mean reprojection errors, both before and after manual correction, compared with the XMALab trial.

For every iteration of the network, the uncorrected *X*-*Y*-*Z* positions of the tongue markers did not meet the threshold for successful performance (Fig. 3A–D). The first iteration of the network produced predictions that required approximately 2 h of manual correction per 2000 frame trial to reach the error threshold. Satisfactory performance was achieved through the correction of all frames in which a marker's reprojection error exceeded 2–2.5 pixels (Fig. 3E,F). In order to reduce the amount of manual correction needed, new frames were tracked and added to the training dataset two separate times, and each iteration resulted in progressively lower reprojection errors (Fig. S2). The output of the second iteration of the network required 1 h of manual correction, and the third iteration of the network required 20–30 min, an approximately 13-fold increase in throughput, including training dataset generation time.

Study 3: guineafowl range of motion

This case study was unique in that the 'behavior' being studied, post-mortem specimen manipulation, was acyclic and designed to document the range of possible poses. In short, we were unable to achieve successful marker tracking results with DeepLabCut. After three iterative augmentations of the training dataset, reprojection and rigid body errors were still so high that it took longer to correct the output of DeepLabCut than to track the test trial from scratch in XMALab (Fig. 2A). As each trial contained novel postures, we

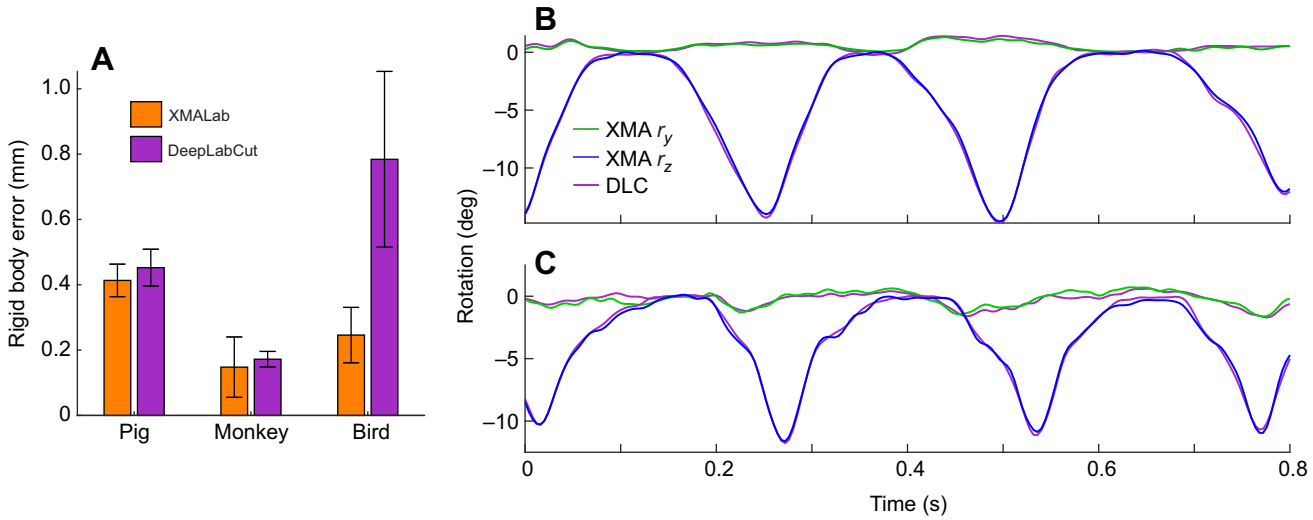

Fig. 2. Comparison of XMALab and DeepLabCut rigid body tracking performance. (A) Mean (±s.d.) rigid body error (filtered at 30 Hz) from XMALab for the test trial of each case study where markers were tracked either with XMALab (orange) or with DeepLabCut (DLC; purple). Pig and monkey errors comprise the rigid body transformations of the cranium and mandible, and the bird errors comprise the transformations of all leg bones for their respective test trials. (B) Monkey and (C) pig temporomandibular joint rotation data derived from the two sets of rigid body transformations described in A. Green lines (XMA r_y) are *y*-axis rotation, or yaw; blue lines (XMA r_z) are *z*-axis rotation, or pitch; brown lines (DLC) are the same degrees of freedom, from DeepLabCut-tracked data. Note that despite differing mean reprojection errors (see main text), mean rigid body error and the resultant rotation values for the pig and monkey were comparable to those of the same trial tracked in XMALab. Here, the DeepLabCut marker predictions were not corrected in XMALab, and the joint coordinate systems were oriented following Menegaz et al. (2015) and Orsbon et al. (2018). The r_x (*x*-axis rotation, roll) trace was omitted because it failed to exceed the established noise threshold in both tracking methods.

Journal of Experimental Biology (2020) 223, jeb226720. doi:10.1242/jeb.226720

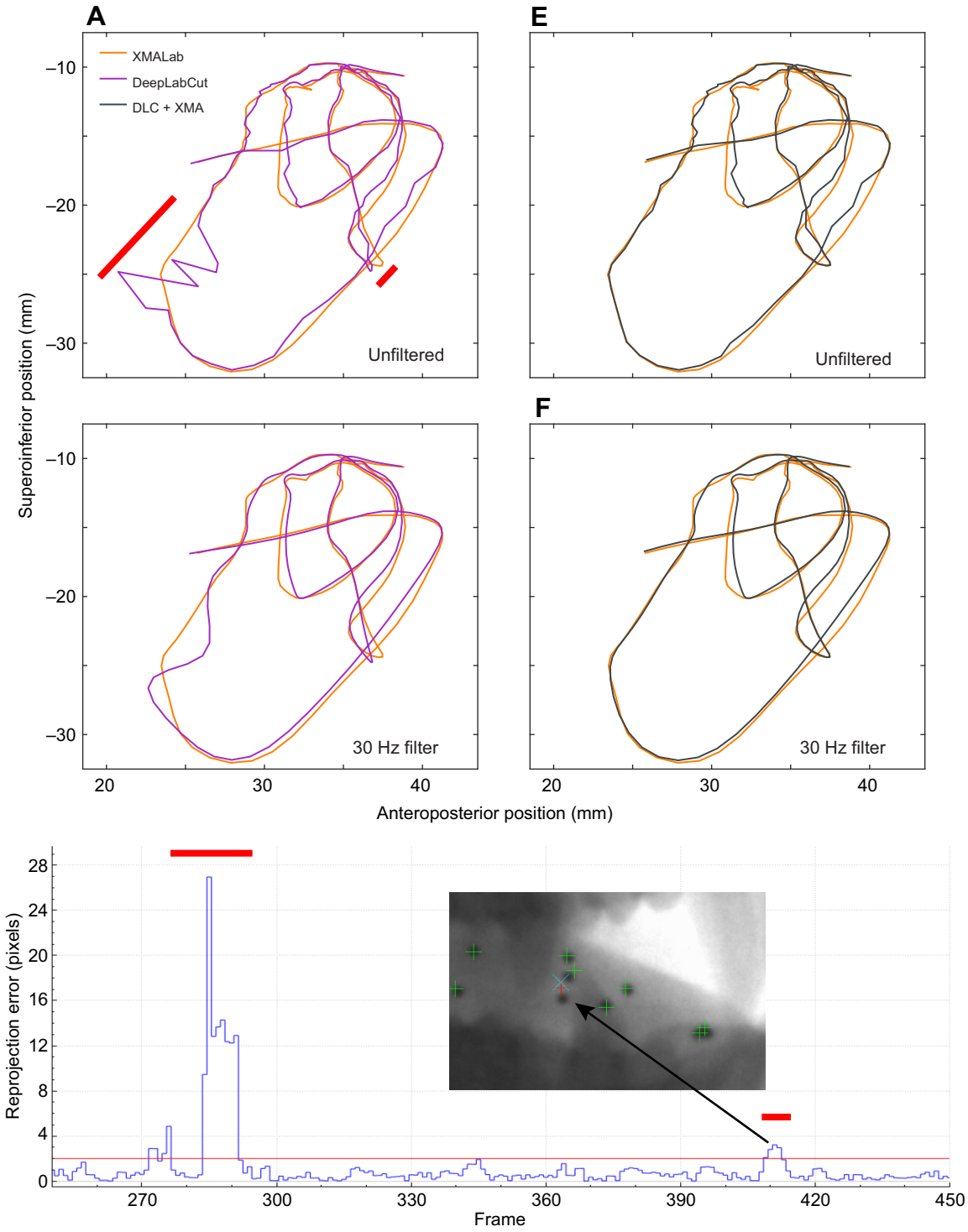

Fig. 3. Comparison of tracking methods for an example tongue marker trajectory. (A–D) DeepLabCut predicted positions (purple) for the anterior right tongue marker were at times erroenous (red bars). After importing the predicted 2D points into XMALab, those regions of poor tracking were easily identified with the reprojection error trace (C). All frames with reprojection errors higher than the established threshold (2 pixels for this study) were corrected in both cameras (D). (E,F) The resulting DeepLabCut+XMALab marker trajectory (gray) is accurate – similar to the same marker tracked in XMALab alone (orange) – and 8–13× faster to generate. The trajectories are the *X* and *Y* values taken from the marker's *X*–*Y*–*Z* coordinates that have been exported from XMALab, and transformed into an anatomical coordinate system with its origin at the posterior nasal spine (Orsbon et al., 2018). Unfiltered trajectories (A,E) and trajectories filtered with a 30 Hz low-pass butterworth filter (B,F) are shown.

found it was virtually impossible to generate a training dataset that sufficiently captured the variation in the data without tracking a majority of every trial in XMALab, defeating the purpose of the new workflow.

DISCUSSION

Comparison with XMALab

In two of the three case studies, our integrated workflow dramatically outperformed XMALab alone, in terms of overall

Journal of Experimental Biology (2020) 223, jeb226720. doi:10.1242/jeb.226720

processing time. After network optimization, per-trial marker tracking time was reduced 6-fold in the pig dataset and 13-fold in the monkey dataset. This high-throughput performance was robust to marker placement and number; the monkey dataset involved >20 markers in both rigid bodies (cranium and mandible) and soft tissue (tongue) structures. At the individual point level, DeepLabCut converged on, but never surpassed, XMALab quality. We found that mean reprojection errors of individual points were lowest when tracked in XMALab alone, but, crucially, this difference was not reflected in measured kinematic variables. After correction in XMALab, both JCS data and *X-Y-Z* marker positions did not differ meaningfully between the two tracking modes. In general, we found that what was difficult in XMALab was also more difficult for DeepLabCut; whereas DeepLabCut excelled at tracking markers in rigid bodies that followed cyclic trajectories, it had more difficulty (i.e. required more training frames) with dense and overlapping markers in soft tissue.

Establishing an error tolerance

Here, we set the error tolerance for marker tracking in DeepLabCut as the reprojection error and rigid body error values that corresponded to the point at which the measured variable (JCS data, or tongue marker positions) did not differ meaningfully from the same variable when tracked in XMALab alone. Depending on the nature of the study at hand, different performance criteria may be desired. For example, if a study is constrained to a small number of trials, noise inherent to DeepLabCut's predictions can have a magnified impact and thus more stringent error tolerances are appropriate. Likewise, in a study that seeks to quantify subtle motions (e.g. hemimandible wiggle; Bhullar et al., 2019), extra care must be taken when establishing the error tolerance.

Training dataset

Algorithmic selection of training data based on visual dissimilarity can greatly improve performance for a given training dataset size (Brust et al., 2019). DeepLabCut offers a *k*-means method for extracting frames from videos that show maximum visual differences. In theory, this approach could be used on XROMM data, however, in practice this is generally not feasible as it can be virtually impossible to accurately identify markers in single frames of XROMM data out of their temporal context. For this reason, the workflow involves tracking sub-sequences of trials and it is up to the user to identify the regions that contain different postures. In the future, an algorithmic approach to identify ideal training frames could reduce time spent augmenting the training dataset.

Other factors influencing throughput

Image resolution has a dramatic impact on DeepLabCut processing time (Mathis et al., 2018). As such, best practice is to down-sample large images before processing. We chose to omit any image down-sampling due to the small size of XROMM markers (5–10 pixels diameter) and our desire to maximize precision. For studies where markers are larger or processing time is of greater concern, down-sampling the raw X-ray data may yield better results. The hardware on which a user runs DeepLabCut should also be considered. Without a dedicated GPU, processing full-sized XROMM images becomes practically infeasible. Here, we performed data analysis on a single individual from each case study as different individuals had different marker locations and marker numbers. Whether or not a single network can be generalized across multiple days of data collection probably depends on the variation in the day-to-day setup, as well as the marker locations.

Areas for improvement

XMALab and DeepLabCut utilize fundamentally different mechanisms for marker tracking. XMALab uses a point's velocity to make a prediction about where it will be in the following frame, then searches for the point using a template. Additionally, it uses camera calibration information such as reprojection error and rigid body error for user visualization as well as to establish thresholds at which to stop tracking. In contrast, DeepLabCut uses neither camera calibration information nor velocity when tracking. DeepLabCut evaluates each frame of video in isolation, essentially pattern-matching the appearance of the frame at-hand with frames from the training dataset. The lack of communication between the two camera views means that DeepLabCut might make a highly-erroneous prediction when, to a user looking at both camera views simultaneously, it is obvious that marker correspondence is incorrect. If DeepLabCut employed a filter based on reprojection error (see open-source software DLTdv8; Hedrick, 2008) and used a marker's velocity, tracking performance might improve.

Concluding remarks

We showed that a marker tracking workflow that integrates deep learning can dramatically outperform the existing XROMM workflow in terms of throughput. Importantly, the throughput increase occurred when the behavior at hand was cyclic and when ROM was constrained experimentally. For this reason, we believe it is best to think about the present workflow as one that enables large scale studies, the likes of which were previously impossible, when such experimental design criteria are met. This workflow is not, however, a panacea for digitizing XROMM data. In cases where the sample size is small or the behavior is acyclic, the established XMALab only marker tracking workflow is still more efficient. As deep learning algorithms improve, however, and when DeepLabCut incorporates camera calibration into its marker prediction, this balance will likely shift.

Acknowledgements

We thank Greg Shakhnarovich and Steven Basart for helpful discussions on using deep neural networks for XROMM data processing, Ben Knorlein for helpful discussions on integrating DeepLabCut with XMALab, Alexander Mathis and Mackenzie Mathis for DeepLabCut, and David Baier for XROMM_MayaTools. Rebecca Junod provided superb assistance with monkey data collection. We also thank Victoria Hosack, Madison Jewell, Jared Luckas, Emma Lesser, Tricia Nicholson and Derrick Tang for XROMM data processing assistance.

Competing interests

The authors declare no competing or financial interests.

Author contributions

Conceptualization: J.D.L.-C., F.I.A.-M.; Methodology: J.D.L.-C., F.I.A.-M.; Software: J.D.L.-C.; Validation: J.D.L.-C.; Formal analysis: J.D.L.-C.; Investigation: J.D.L.-C., A.R.M.; Writing - original draft: J.D.L.-C.; Writing - review & editing: J.D.L.-C., A.R.M., N.G.H., C.F.R., F.I.A.-M.; Visualization: J.D.L.-C.; Supervision: C.F.R., F.I.A.-M.; Funding acquisition: F.I.A.-M., N.G.H., C.F.R.

Funding

This work was funded by the National Institutes of Health (R01DE027236, 1UL1TR002389-01) and a National Science Foundation Graduate Research Fellowship (J.D.L.-C. and A.R.M.). Deposited in PMC for release after 12 months.

Supplementary information

Supplementary information available online at https://jeb.biologists.org/lookup/doi/10.1242/jeb.226720.supplemental

References

Bhullar, B.-A. S., Manafzadeh, A. R., Miyamae, J. A., Hoffman, E. A., Brainerd, E. L., Musinsky, C. and Crompton, A. W. (2019). Rolling of the jaw is essential

for mammalian chewing and tribosphenic molar function. *Nature* **566**, 528-532. doi:10.1038/s41586-019-0940-x

Brainerd, E. L., Baier, D. B., Gatesy, S. M., Hedrick, T. L., Metzger, K. A., Gilbert, S. L. and Crisco, J. J. (2010). X-ray reconstruction of moving morphology (XROMM): precision, accuracy and applications in comparative biomechanics research. *J. Exp. Zool. A Ecol. Genet. Physiol.* **313**, 262-279. doi:10.1002/jez.589

Brust, C. A., Käding, C. and Denzler, J. (2019). Active learning for deep object detection. Proceedings of the 14th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications. doi:10.5220/0007248601810190

Gintof, C., Konow, N., Ross, C. F. and Sanford, C. P. J. (2010). Rhythmic chewing with oral jaws in teleost fishes: a comparison with amniotes. *J. Exp. Biol.* **213**, 1868-1875. doi:10.1242/jeb.041012

Granatosky, M. C., McElroy, E. J., Laird, M. F., Iriarte-Diaz, J., Reilly, S. M., Taylor, A. B. and Ross, C. F. (2019). Joint angular excursions during cyclical behaviors differ between tetrapod feeding and locomotor systems. *J. Exp. Biol.* **222**, jeb200451. doi:10.1242/jeb.200451

Graving, J. M., Chae, D., Naik, H., Li, L., Koger, B., Costelloe, B. R. and Couzin, I. D. (2019). DeepPoseKit, a software toolkit for fast and robust animal pose estimation using deep learning. *eLife* **8**, e47994. doi:10.7554/eLife.47994

Grood, E. S. and Suntay, W. J. (1983). A joint coordinate system for the clinical description of three-dimensional motions: application to the knee. *J. Biomech. Eng.* **105**, 136-144. doi:10.1115/1.3138397

Hedrick, T. L. (2008). Software techniques for two- and three-dimensional kinematic measurements of biological and biomimetic systems. *Bioinspir. Biomim.* **3**, 034001. doi:10.1088/1748-3182/3/3/034001

Insafutdinov, E., Pishchulin, L., Andres, B., Andriluka, M. and Schiele, B. (2016). Deepercut: a deeper, stronger, and faster multi-person pose estimation model. *Eur. Conf. Comput. Vis.* **9910**, 34-50. doi:10.1007/978-3-319-46466-4_3

Iriarte-Diaz, J., Terhune, C. E., Taylor, A. B. and Ross, C. F. (2017). Functional correlates of the position of the axis of rotation of the mandible during chewing in non-human primates. *Zoology* **124**, 106-118. doi:10.1016/j.zool.2017.08.006

Kambic, R. E., Roberts, T. J. and Gatesy, S. M. (2017). 3-D range of motion envelopes reveal interacting degrees of freedom in avian hind limb joints. *J. Anat.* **231**, 906-920. doi:10.1111/joa.12680

Knörlein, B. J., Baier, D. B., Gatesy, S. M., Laurence-Chasen, J. D. and Brainerd, E. L. (2016). Validation of XMALab software for marker-based XROMM. *J. Exp. Biol.* **219**, 3701-3711. doi:10.1242/jeb.145383

Labuguen, R., Bardeloza, D. K., Negrete, S. B., Matsumoto, J., Inoue, K. and Shibata, T. (2019). Primate markerless pose estimation and movement analysis using DeepLabCut. 2019 Joint 8th International Conference on Informatics, Electronics & Vision (ICIEV) and 2019 3rd International Conference on Imaging, Vision & Pattern Recognition (icIVPR). Spokane, WA, USA. doi:10.1109/ICIEV.2019.8858533

Manafzadeh, A. R. and Padian, K. (2018). ROM mapping of ligamentous constraints on avian hip mobility: implications for extinct ornithodirans. *Proc. R. Soc. B* **285**, 20180727. doi:10.1098/rspb.2018.0727

Martinez, C. M., McGee, M. D., Borstein, S. R. and Wainwright, P. C. (2018). Feeding ecology underlies the evolution of cichlid jaw mobility. *Evolution* **72**, 1645-1655. doi:10.1111/evo.13518

Mathis, M. W. and Mathis, A. (2019). Deep learning tools for the measurement of animal behavior in neuroscience. *Curr. Opin. Neurobiol.* **60**, 1-11. doi:10.1016/j.conb.2019.10.008

Mathis, A., Mamidanna, P., Cury, K. M., Abe, T., Murthy, V. N., Mathis, M. W. and Bethge, M. (2018). DeepLabCut: markerless pose estimation of user-defined body parts with deep learning. *Nat. Neurosci.* **21**, 1281-1289. doi:10.1038/s41593-018-0209-y

Menegaz, R. A., Baier, D. B., Metzger, K. A., Herring, S. W. and Brainerd, E. L. (2015). XROMM analysis of tooth occlusion and temporomandibular joint kinematics during feeding in juvenile miniature pigs. *J. Exp. Biol.* **218**, 2573-2584. doi:10.1242/jeb.119438

Nath, T., Mathis, A., Chen, A. C., Patel, A., Bethge, M. and Mathis, M. W. (2018). Using DeepLabCut for 3D markerless pose estimation across species and behaviors. *Nat. Protoc.* **14**, 2152-2176. doi:10.1038/s41596-019-0176-0

Orsbon, C. P., Gidmark, N. J. and Ross, C. F. (2018). Dynamic musculoskeletal functional morphology: integrating diceCT and XROMM. *Anat. Rec.* **301**, 378-406. doi:10.1002/ar.23714

Owen, S. F., Liu, M. H. and Kreitzer, A. C. (2019). Thermal constraints on in vivo optogenetic manipulations. *Nat. Neurosci.* **22**, 1061-1065 doi:10.1038/s41593-019-0422-3

Parmiani, P., Lucchetti, C., Bonifazzi, C. and Franchi, G. (2019). A kinematic study of skilled reaching movement in rat. *J. Neurosci. Methods* **328**, 108404. doi:10.1016/j.jneumeth.2019.108404

Pereira, T. D., Aldarondo, D. E., Willmore, L., Kislin, M., Wang, S. S.-H., Murthy, M. and Shaevitz, J. W. (2019). Fast animal pose estimation using deep neural networks. *Nat. Methods* **16**, 117-125. doi:10.1038/s41592-018-0234-5

Stringer, C., Pachitariu, M., Steinmetz, N., Reddy, C. B., Carandini, M. and Harris, K. D. (2019). Spontaneous behaviors drive multidimensional, brainwide activity. *Science* **346**, 255. doi:10.1126/science.aav7893