**A guide to machine learning methods for animal behavior analysis**

**Table of contents**

1. Overview
2. Setting up DLC
3. Running DLC
4. DLC pose labelling
5. DLC training
6. DLC inference
7. DLC refining labels
8. Obtaining behavior annotations with BORIS
9. Downstream analysis in JupyterLab

*Appendix*

A1. General computing notes on using linux, connecting with NoMachine and managing GPUs

A2. List of resources

1. **Overview**

We can separate this pipeline into two stages. The first is to obtain an estimate of an animal or set of animal’s pose from some videos. This is typically a set of tracks for each body part wanting to be tracked throughout the video. DeepLabCut is a deep-learning-based solution to obtain this data. It works by training neural networks to detect body parts in each frame in the video, and then links those detections throughout all the frames to create tracks. Labeled data on a set of training frames must be provided by the user.

DeepLabCut is well-supported, documented and relatively easy to use, as deep learning models go. There are still many steps that go into getting it to track a new dataset successfully (Figure 1). The first 5 sections cover how to use DLC.

Once a set of tracks is obtained, they can be analyzed in many ways. We can additionally label the frames of the videos with behaviors of interest, and train a machine learning model to classify when those behaviors are occurring in separate videos. This is supervised learning. To obtain behavior labels for this approach, the software BORIS makes this quite easy. This is covered in Section 7.

Notebooks to perform downstream analysis – either supervised or unsupervised learning – are described in Section 8.

Diagram, schematic

Description automatically generated

Figure DLC flowchart

1. **DLC Setup**

(DLC may already be setup in your environment, so this section is optional.)

**Install notes for DLC (v2)**  
First, obtain the yaml file from the git repo: <https://github.com/DeepLabCut/DeepLabCut>

Latest version (2.2) install notes:

* Use DEEPLABCUT.yaml file to install basic requirements (not the deeplabcut package itself, remove this line from the yaml file)  
  conda env create -f DEEPLABCUT.yaml
* Then install wxpython from conda-forge (wxpython 4.0.7)  
  conda install –c conda-forge wxpython=4.0.7
* Then install cudnn and cudatoolkit with conda  
  conda install cudnn cudatoolkit
* Then install DLC with pip:   
  pip install deeplabcut[gui]

Test GPU available with this install:

* Open ipython and type: import tensorflow as tf; tf.test.is\_gpu\_available()

Test DLC works:

* python –m deeplabcut

(Note that installing with GUI option means DLC may or may not work in headless mode. See for example: <https://issueexplorer.com/issue/DeepLabCut/DeepLabCut/1583>. If you run into this problem, the recommended solution is to install two environments, one with deeplabcut and the other with deeplabcut[gui].)

1. **Running DLC**

DLC has a few steps that require the user to label frames. These must be done through the GUI. The entire pipeline can also be performed in the GUI. To run DLC this way:

1. Login through nomachine or fastx to access a remote desktop:
   1. For fastx, in a browser, go to dnb2:3300
   2. Enter St Jude login credentials
2. Open a terminal: Activities -> Terminal
3. Type:
   1. conda activate DEEPLABCUT
   2. python -m deeplabcut

If any steps will be using the GPU, you should check which GPUs are free on the machine, with the command `nvidia-smi`. The start DLC with the command:

CUDA\_VISIBLE\_DEVICES=[GPU number] python -m deeplabcut

Where [GPU number] is whichever GPU is available. Make sure to book the use on DNB workspace (<https://workspaces.stjude.org/sites/DNB/SitePages/Home.aspx>), if it will be for a few hours or more.

1. **DLC labeling**

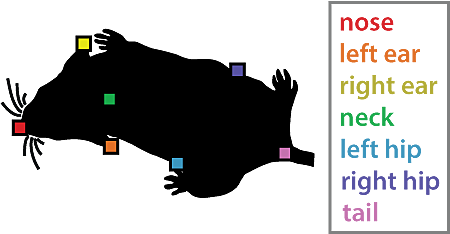
For a totally new tracking problem, pose labels will need to be provided, in order to train the DLC deep learning model. To label a training set of videos:

**Using DLC for pose labeling – generic instructions. DLC v2**

1. Open an existing project or make a new project
2. If making a new project, after the config.yaml file has been created, be sure to edit the config file for the particulars of the project. In particular, the details of the animals and body parts being tracked must be entered. Along with whether the project is multi-animal or not. Can access a text editor with Activities -> Text Editor
3. For labeling:
   1. If the mice can be reliably distinguished then they should be labeled consistently in all videos. If they are visually identical then this is not a concern. Set the identity flag in the config.yaml to True if they can be visually distinguished
   2. The skeleton should be in general overconnected to help with training
4. Once this is done, the labeled frames are turned into a training and validation dataset, and a deep learning model is trained on the training set.

**Using DLC for pose labeling for mouse social interactions – matching with MARS annotations.**

Same as the above steps, except uses the following labeling scheme. The body parts should be the following. Don’t use spaces, use underscores for whitespace in the body part names. See also Figure 2A and 2C of this paper for examples of exact ground truth body part positions: <https://www.biorxiv.org/content/10.1101/2020.07.26.222299v1.full.pdf>



1. **DLC training**
   1. After frames are labeled, these are turned into a training dataset.
   2. On the create training dataset tab, can keep all options their defaults. Click on OK
   3. Train network. Update ‘Save iterations’ and ‘Max iterations’. 50000 max iterations is a good starting point.
   4. Evaluate network. Keep defaults. RUN: Evaluate Network
2. **DLC inference**

Once a DLC model is trained and evaluated, we can compute poses for a new set of videos (with the same setup – same body parts to be tracked, similar experimental setup, etc.).

This can be split into two steps. For a totally new set of videos, we want to perform some basic quality control (QC) to make sure the inference is giving acceptable results. So, the first step is to go through the process in the DLC GUI for a few videos, making videos of the detections and tracks to examine the performance.

Once we’re confident the model is performing well enough on these new videos, the second step is to run the inference for the whole batch of videos. This can be done either through the GUI or through a Jupyter notebook (see Section 9) (less user intervention is required this way, easier to run and forget).

**Part 0 – obtain the DLC trained models**

Some pretrained models are available at this repository:

<https://git.stjude.org/scm/~blansdel/behavior-ml-analysis.git>

(Issues with size of repository. Download from here for now: <https://github.com/benlansdell/behavior-ml-analysis.git>)

1. Open a terminal (through NoMachine, or through ssh)

2. Type: *git clone* [*https://github.com/benlansdell/behavior-ml-analysis.git*](https://github.com/benlansdell/behavior-ml-analysis.git)This will download the files. This is the best way to get the files, downloading from the webpage will miss some of the larger files in the repo.

3. Move the directory if you want: *mv behavior-ml-analysis projects/behavior-ml-analysis*

**Part 1 – QC**

1. Open DLC, as in `DLC labeling` steps above. (Through NoMachine and the command line)
2. Load the DLC config file for the DLC project with the trained model we want to use.
3. Run the inference. In the ‘Analyze videos’ tab:
   1. Select the videos to run the inference on. Only choose a few
   2. Keep most options to their defaults. Set ‘Create video for checking detections’ to Yes
   3. Run Step 1 and then Step 2.
4. Examine the output videos from these steps to see how well the method is picking out individual body parts. Look for false detections in the frame that could be distracting the model – e.g. reflections. If tracking performance is sub-optimal, these could be the cause.
5. Make tracks, correct identity swaps. In the `Refine tracklets` tab:
   1. Select the tracklet pickle file (for the ellipse tracker – default – this will end in .el.pickle) of one of the videos
   2. Set number of animals accordingly, and `Step1: Create tracks`
   3. (Optional, generally needed if more than one animal) Select the corresponding video for this tracklet file and `Refine tracks`
      1. This will open up a separate interface to explore the tracking. If there are identity swaps in the tracking, they can be corrected here.
   4. `Filter tracks`
6. Check the quality of the final DLC output. In the `Create videos` tab:
   1. Select the videos we just ran the inference on
   2. Generally helpful to set:
      1. Include the skeleton in the video: yes
      2. Create video with animal ID colored? Yes (if more than one animal)
      3. Use filtered predictions: yes
   3. RUN

* This will make videos showing the final tracking. Examine these for errors

Without having additional labeling, we just have these videos to evaluate performance. If there are significant errors, or consistently a certain type of tracking error is made, additional training of the model may be needed. See `DLC model refinement`.

Some notes:

* Can play with the pcutoff. Probably want this higher (~0.7) for the ‘Create video for checking detections’ step, and lower for the final video creation step (0)
* If the number of animals is different from the training data, DLC will name the animals ind1, ind2, ind3, etc.   
  To avoid errors when creating the videos, it’s best to update the config file with these animal names.

**Part 2 – Batch run**

Assuming the tracking is fine after following the above steps, we can run the model on the rest of the videos. If there are multiple animals, and the identity matters, then for each video, the `refine labels` step will need to be run to correct for animal swaps. Since this step must use the GUI, we may as well use the GUI for the whole inference. In this case, the steps to run the inference on the full set of videos is the same as the QC steps above. Except: we don’t need to set the `Create video for checking detections` to Yes in step 3b and Step 6 is optional.

If we don’t need to correct for identity swaps, then no user-input is required halfway through the above steps, and the whole pipeline can be run automatically through a python notebook. Can either use the GUI for this, or Jupyter notebooks, in JupyterLab. See Section 9.

1. **DLC refining the model**

To refine labels you need:

* An already trained DLC model, along with its training frames from the original set.
* A new video to run the model on, extract frames and retrain. You must have already run 'analyze\_videos' on these videos.

Steps:

* Update the config file:
  + Update the path to the project in the config.yaml
  + Update the number of frames to extract – can start with 50
  + Update the individuals: e.g. add ‘ind1’
* Run the extract outlier frames step.
  + Select all the videos you want to use for refinement. Best to pick all of these at once.
  + In the labeled\_frames folder of the DLC project, this should add a .h5 file
* Refine labels -> Open the refine labels GUI for each of the videos.
  + When done with this, click on the merge datasets button. Now ready to retrain
* Then go back to the create training dataset tab
  + Keep most things default. Can change the network to dlc\_rnet\_ms5.
  + Click ok
  + May give a permission error (see this issue). If so, have to add the videos to the config file manually.
* Go to train network. Change the 'save\_iterations' and 'max\_iterations' to appropriate values
  + Max\_iterations can be 50000, save\_iterations can be 10000
  + Click ok. Will take an hour or more
* Evaluate network.
  + Also will take ~20-30 minutes, depending on training set size.
  + It may throw an error about number of columns in the dataset.
  + Can ignore this error.
* Ready to analyze new videos.
  + Can re-analyze the videos you refined the labels on, to confirm it’s corrected the mistakes you identified
  + The real test is to analyze a totally novel set of videos and see if it makes the same errors as before on those

1. **Behavior analysis – supervised learning**

In addition to pose labels, if wanting to predict the occurrence of a specific behavior, annotations of training videos that label that behavior need to be provided. The following are instructions to obtain these labels though the annotation software BORIS.

**Using BORIS for behavior annotation**

1. Either login to dnb2 using fastx as above to access remote server, or have videos files accessible from a local machine and install BORIS on that machine (may be more responsive for the video watching)
2. On linux machine:
   1. Open BORIS
      1. Activities -> BORIS
3. Create a new project/open an existing one
   1. Add behaviors to be labeled, and the animals being labeled, setup keyboard shortcuts
   2. Add observations (name for one recording sessions)
      1. Add media
   3. Start observation to start labeling
   4. Export: Observations -> Export events

Information about BORIS video interface: <https://boris.readthedocs.io/en/latest/#coding-your-media>

**BORIS on Windows**

* BORIS can be run on windows. The portable version doesn’t require any installation, just extracting the files, and running ‘start\_boris.bat’
* Download here: <https://www.boris.unito.it/pages/download_win.html>

1. **Downstream analysis in JupyterLab**

JupyterLab provides an environment to run python code with already configured python packages. The DEEPLABCUT environment comes with deeplabcut and the package ‘behaveml’ that can perform some downstream analysis using DLC data.

To run a python notebook in this way:

1. Go to JupyterHub. In the browser (from St Jude or through the VPN), go to dnb2:8000

2. Login with SJ credentials to access the JupyterHub

3. Navigate to the project directory in the file browser on LHS

4. Open a notebook, e.g. dlc\_inference.ipynb

5. Update paths and options in the notebook and run through the commands

Some notebooks to do DLC inference, as well as supervised and unsupervised learning are available in the repository: <https://github.com/benlansdell/behavior-ml-analysis.git>

**Appendix**

**A1. General notes**

**Notes on the Linux environment**

* Can access a basic text editor with Activities -> Text editor
* A file browser can be accessed with Activities -> File Browser
  + St Jude network storage is accessed from the file browser in the left tab, listed under ‘remote’
  + In this list, the drive named /research/rgs01/dept/DNB is the same as Z:\ResearchHome\Departments\DevelopmentalNeurobiology

**Getting NoMachine**

* <https://www.nomachine.com/download>

**JupyterHub**

* Can be accessed here: dnb2:8000
  + Sometimes gives a POST error. If you see this, try logging out and in again.

**A2. Useful resources**

* The DLC github repo houses the latest version, along with links to its documentation, publications, etc  
  <https://github.com/DeepLabCut/DeepLabCut>
* The chatroom gitter has a DLC channel: <https://gitter.im/DeepLabCut/community> that is pretty responsive. Can ask questions here
* Similarly with the image analysis forums image.sc: <https://forum.image.sc/>
* BORIS can be obtained here: <https://www.boris.unito.it/>
* DNB analysis notebooks, DLC models for our data and other resources are housed here: <https://git.stjude.org/users/blansdel/repos/behavior-ml-analysis/browse> (issues with repo size to be figured out. Download from here for now: <https://github.com/benlansdell/behavior-ml-analysis.git>)