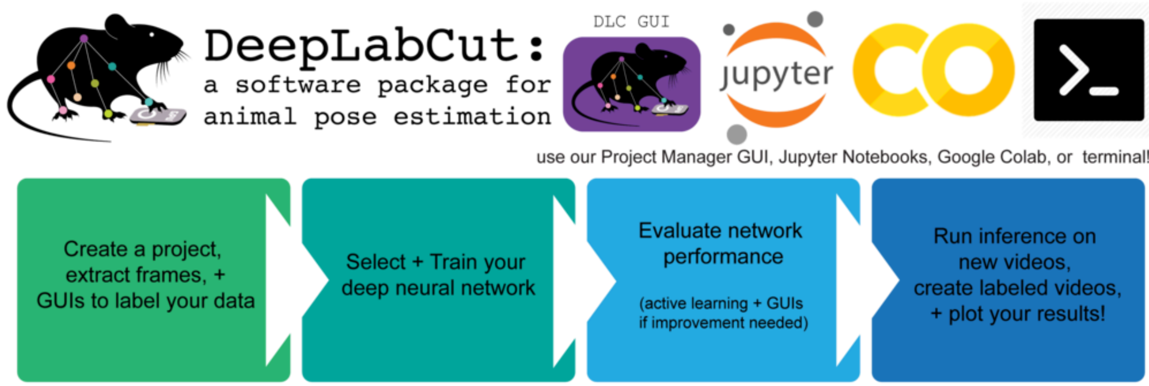
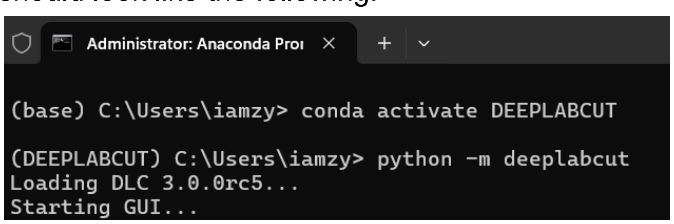
Instructions on Pose Estimation Using DeepLabCut

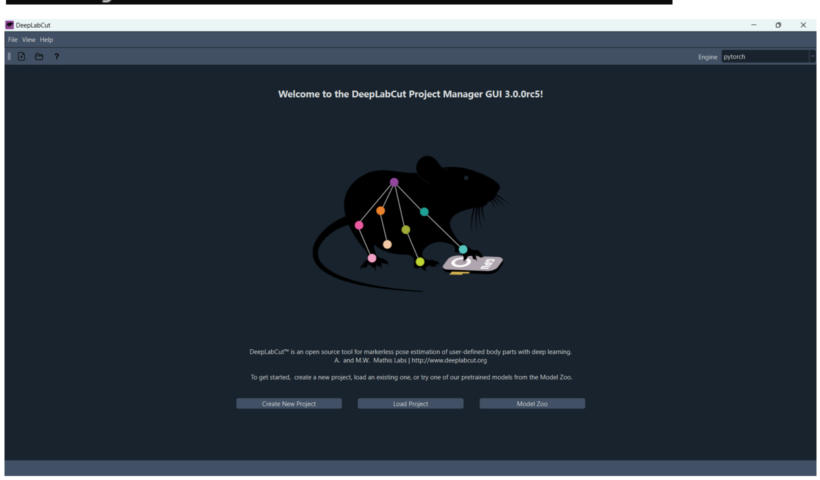


DeepLabCut is an open-source software designed for animal pose estimation using deep learning. It allows researchers to create and train neural networks to label body parts in videos. By downloading and using the DeepLabCut (DLC) GUI interface, no coding is required. The following outlines the steps for downloading and using the GUI.
For more detailed information, please visit: DeepLabCut GitHub Repository.

Step 0. Install DeepLabCut

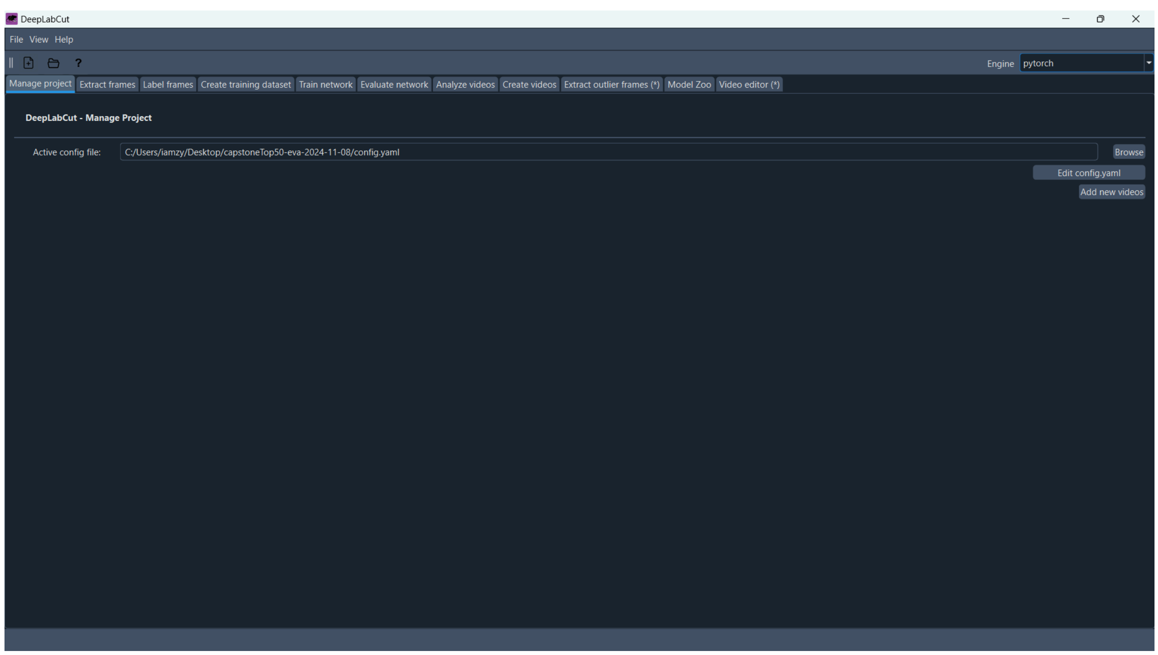
- To install the DLC GUI, follow this link: DeepLabCut Installation Guide.
- (Note: For our group, installing Anaconda and building an environment using Anaconda has proven to be the most effective method.)
- Launch the DLC GUI by using the appropriate command. If using Anaconda, this should look like the following:





Step 1. Create Project

- Click "**File**" and then "**New Project**" in the top-left corner of the interface. Select the folder containing the videos you wish to use for training. (Keep in mind that these videos will need to be manually labeled in Step 3, so avoid selecting too many for training purposes.) After creating the project, navigate to the project folder and locate the `config.yaml` file to set up the configurations. The following key features need to be adjusted:
  - "bodyparts": Specify the key body parts you want to label.
  - "skeleton": Define the connections between the labeled body parts.
- Alternatively: If a project has already been set up, you can open the existing configuration file. Specifically, click "**File**" and then "**Open**" in the top-left corner of the interface, and select the `config.yaml` file located in the project folder we shared. Our folder includes 43 videos as the training set, and in our case:
  - "bodyparts" include: nose, left ear, right ear, spine middle, tail base, tail middle, and tail tip.
  - "skeleton" connections are defined as follows:
    - Nose connected to the left ear, right ear, and spine middle.
    - Spine middle connected to the tail base.
    - Tail base connected to the tail middle.
    - Tail middle connected to the tail tip.



If you are using our shared project folder, you can directly utilize our training results. In this case, skip Steps 2 through 6 and proceed directly to Step 7: Analyze Videos.

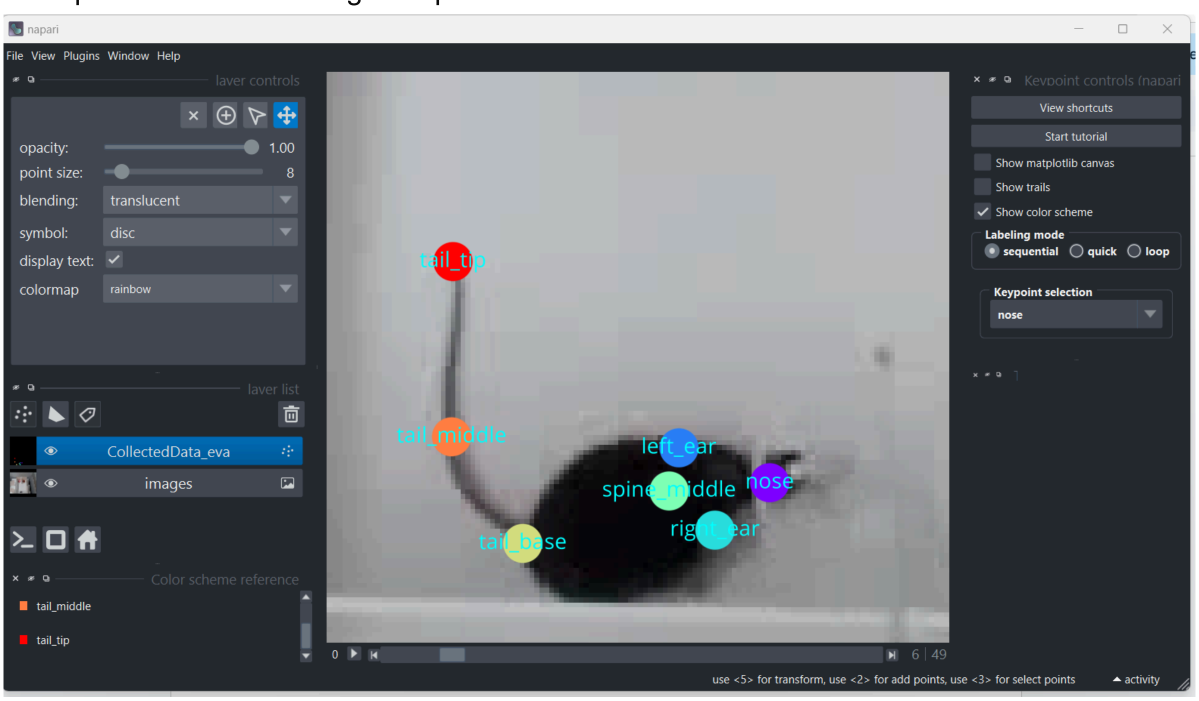
Step 2. Extract Frames

- Go to the “**Extract frames**” tab in the GUI interface. Adjust the attributes as needed (the default settings are usually sufficient), select the videos from which you want to extract frames for manual labeling, and click the “**Extract Frames**” button on the right side.
- The extracted frames will be saved in the `labeled-data` folder within the project directory. Each video’s extracted frames will be organized into a separate subfolder.

Step 3. Label Frames

- Navigate to the “**Label frames**” tab in the GUI interface. For each training video, click the “**Label Frames**” button and select the folder containing the extracted frames to begin manual labeling. The napari interface will launch automatically.
- For each extracted frame, use the napari interface to label all the defined “bodyparts” as long as they are visible. The interface provides tools to place points on the frame corresponding to each body part. For more guidance, refer to this short video tutorial: [How to Label Frames in DeepLabCut](#).
- Before closing the labeling interface, ensure that all manual labels are saved. The results will be stored as:
  - A CSV file containing the x and y coordinates for each labeled body part.
  - An H5 file storing the same data in a hierarchical format.

Example of a frame labeling in napari interface:



Step 4. Create Training Dataset

- Go to the “**Create training dataset**” tab in the GUI interface. Set a shuffle index, adjust the attributes as needed (the default settings with network architecture as resnet_50 are usually sufficient), and click the “**Create Training Dataset**” button on the right side.
- The generated training dataset is stored in the `training-datasets` folder in the project folder. This folder contains the training data split, including labeled frames and associated metadata required for training.

Step 5. Train Network

- Go to the “**Train network**” tab in the GUI interface. Use the same shuffle index when creating the training dataset, adjust the attributes as needed (note that the default settings may result in a long training time; in our case, we set maximum epochs to 15 and save epochs to 5), and click the “**Train Network**” button on the right side.
- Monitor the training progress, including the number of iterations and training loss, in (Anaconda) Terminal.
- The trained models and associated checkpoints are stored in the `dlc-models` or `dlc-models-pytorch` folder (depending on the selected engine). These files include weights, configurations, and logs from the training process.

Step 6. Evaluate Network

- Go to the “**Evaluate network**” tab in the GUI interface. Select the shuffle index corresponding to the model you want to evaluate. Ensure the appropriate net type (e.g., `resnet_50`) and engine (e.g., `pytorch`) are displayed. Click the “**Evaluate Network**” button on the right side to start the evaluation.
- Optional settings:
    - Check “**Plot predictions**” to generate visual predictions for evaluation.
    - Check “**Compare all bodyparts**” to compare predictions for all labeled body parts.
- The progress can be monitored via the Terminal. The evaluation results, including performance metrics and accuracy plots, will be stored in the `evaluation-results-pytorch` folder within the project directory. Use these results to assess the model's accuracy and make improvements if necessary.

Step 7. Analyze Videos

- Go to the “**Analyze videos**” tab in the GUI interface. Select the new videos you would like to process for pose estimation using the “**Select videos**” button. Set the shuffle index to the trained model you wish to use for analysis. Click the “**Analyze Videos**” button on the right side to start analyzing.
- Ensure you check the following options:

- "**Save result(s) as csv**": This will save the pose estimation results as a CSV file.
- "**Filter predictions**": This refines the output from the pose estimation model by smoothing and improving the accuracy of the predicted body part coordinates.
  (The filtered CSV file will be used for the behavior annotation part.)

- Optional Settings:
  - Check "**Dynamically crop bodyparts**" to enable dynamic cropping of body parts to focus on specific areas of interest.
  - Check "**Plot trajectories**" to visualize the movement trajectories of the body parts over time.
  - Check "**Show trajectory plots**" to display trajectory plots after the analysis is complete.
- The progress can be monitored via the Terminal. The results, including the filtered CSV file, will be saved in the same directory as the selected new videos. The filtered CSV file is a data frame containing x and y coordinates and likelihood for each body part per frame.

Step 8. Create Videos (Optional)

- Navigate to the "**Create Videos**" tab in the GUI interface. Select the videos you want to overlay pose estimations on by clicking "**Select videos**".
- Adjust the following attributes as needed:
  - Use "**Shuffle**" to set the shuffle index for the model used.
  - Check "**Plot all bodyparts**" to include all labeled body parts in the video.
  - Check "**Draw skeleton**" to connect labeled body parts based on the skeleton structure.
  - Check "**Use filtered data**" to apply filtered predictions for cleaner visualizations.
  - Check "**Plotting confidence cutoff (pcutoff)**" to adjust the threshold for confidence scores (default is 0.60).
  - Check "**High quality video (slow)**" to generate a higher-resolution video (slower process).
- Click the "**Create videos**" button on the right side to generate the overlaid video.
- The output videos will be saved in the same directory as the original videos.