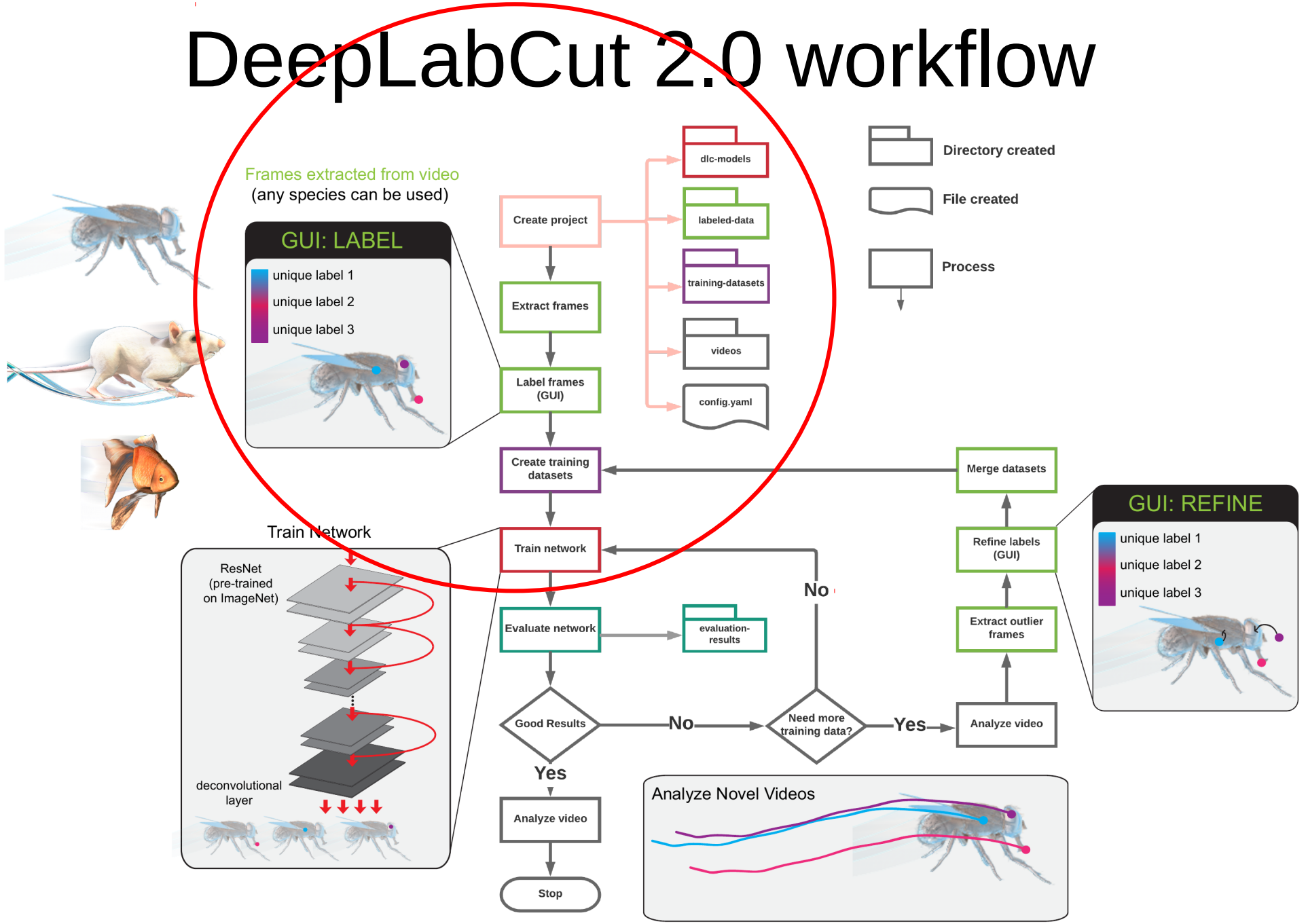
DeepLabCut workshop demo session 1

Alexander Mathis
Harvard University

December 2018
DeepLabCut Workshop - Warsaw

DeepLabCut 2.0 workflow



Nath*, Mathis*
BioRxiv 2018

User guide on BioRxiv

Using DeepLabCut for 3D markerless pose estimation across species and behaviors

Tanmay Nath[1,†], Alexander Mathis[2,3,†], An Chi Chen[4], Amir Patel[4], Matthias Bethge[2,5], & Mackenzie Weygandt Mathis[1,*]

1: The Rowland Institute at Harvard, Harvard University, Cambridge, MA USA
2: Institute for Theoretical Physics, Werner Reichardt Center for Integrative Neuroscience, Eberhard Karls Universität Tübingen, Tübingen, Germany
3: Center for Brain Science & Department of Molecular & Cellular Biology Harvard University, Cambridge, MA USA
4: Department of Electrical Engineering, University of Cape Town, South Africa
5: Max Planck Institute for Biological Cybernetics & Bernstein Center for Computational Neuroscience, Tübingen, Germany
† co-first authors and
**corresponding author: mackenzie@post.harvard.edu*

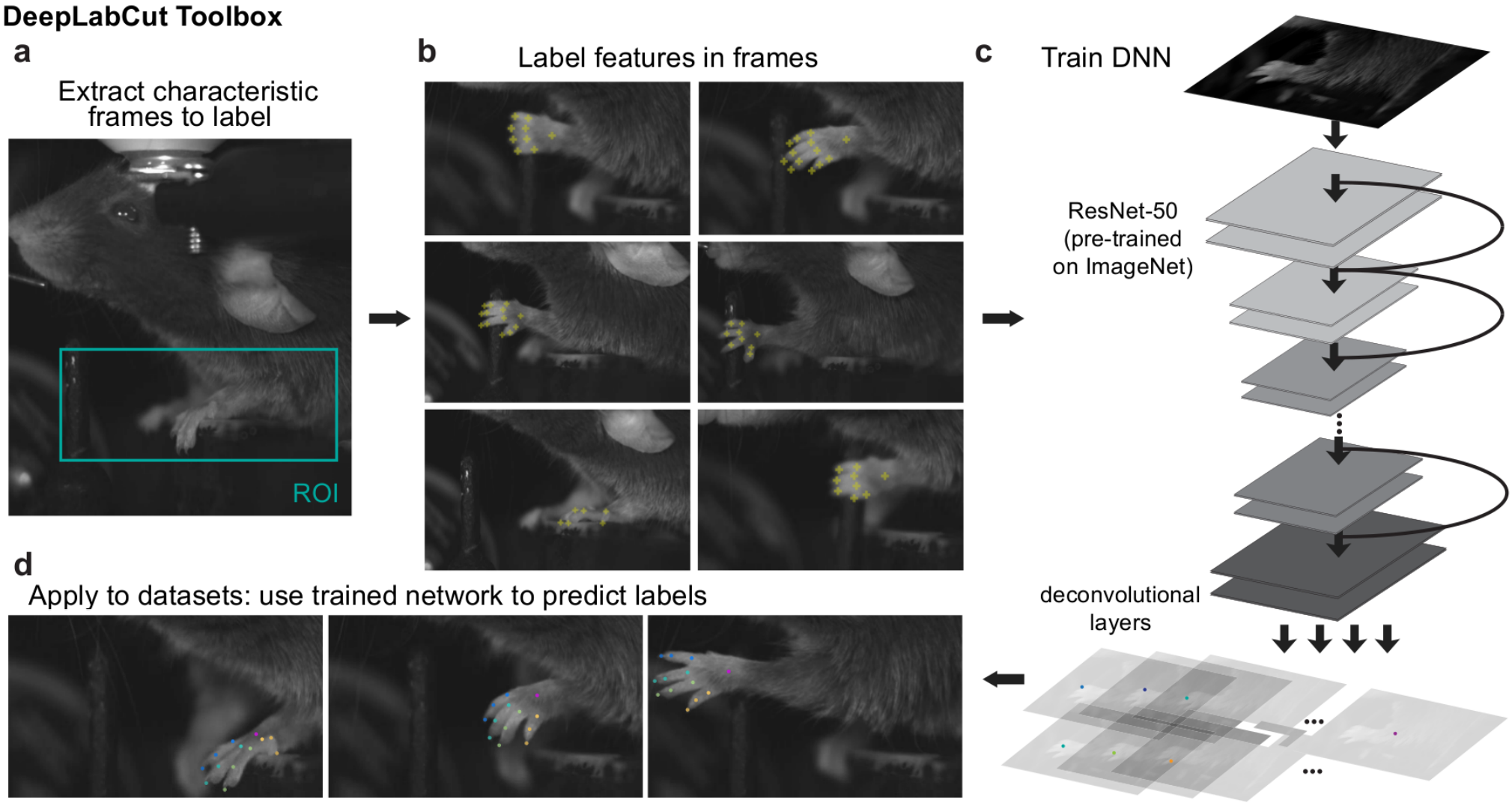
(Dated: November 21, 2018)

Noninvasive behavioral tracking of animals during experiments is crucial to many scientific pursuits. Extracting the poses of animals without using markers is often essential for measuring behavioral effects in biomechanics, genetics, ethology & neuroscience. Yet, extracting detailed poses without markers in dynamically changing backgrounds has been challenging. We recently introduced an open source toolbox called DeepLabCut that builds on a state-of-the-art human pose estimation algorithm to allow a user to train a deep neural network using limited training data to precisely track user-defined features that matches human labeling accuracy. Here, with this paper we provide an updated toolbox that is self contained within a Python package that includes new features such as graphical user interfaces and active-learning based network refinement. Lastly, we provide a step-by-step guide for using DeepLabCut.

- Contains syntax and rationale for underlying functions
https://www.biorxiv.org/content/early/2018/11/24/476531

Nath*, Mathis*
BioRxiv 2018

DeepLabCut needs your user-defined labels!



Mathis et al. Nature Neurosci Sept. 2018

Project creation

- Task name
- User name
- Each project is identified and localized by a (**full path to the config file**). All project parameters are in *config file*!

Note: project (folders) can be **moved**, but the *project_path* in the *config file* has to be adjusted!

Demo

```
alex@T440s:~/Desktop$ source activate DLC2
(DLC2) alex@T440s:~/Desktop$ ipython
Python 3.6.7 |Anaconda, Inc.| (default, Oct 23 2018, 19:16:44)
Type 'copyright', 'credits' or 'license' for more information
IPython 6.0.0 -- An enhanced Interactive Python. Type '?' for help.

In [1]: import deeplabcut

In [2]: deeplabcut.create_new_project?

In [3]: configfiledirectory=deeplabcut.create_new_project('basilisk','alex',['/h
   ...: ome/alex/Desktop/BasiliskL.avi'],copy_videos=True)
Created "/home/alex/Desktop/basilisk-alex-2018-12-15/videos"
Created "/home/alex/Desktop/basilisk-alex-2018-12-15/labeled-data"
Created "/home/alex/Desktop/basilisk-alex-2018-12-15/training-datasets"
Created "/home/alex/Desktop/basilisk-alex-2018-12-15/dlc-models"
Copying the videos
/home/alex/Desktop/BasiliskL.avi
Generated "/home/alex/Desktop/basilisk-alex-2018-12-15/config.yaml"

A new project with name basilisk-alex-2018-12-15 is created at /home/alex/Deskto
p and a configurable file (config.yaml) is stored there. Change the parameters i
n this file to adapt to your project's needs.
 Once you have changed the configuration file, use the function 'extract_frames'
 to select frames for labeling.
. [OPTIONAL] Use the function 'add_new_videos' to add new videos to your project
 (at any stage).

In [4]:
```

Config file name

The variable *configfiledirectory* will store the config file name for this project.

Demo

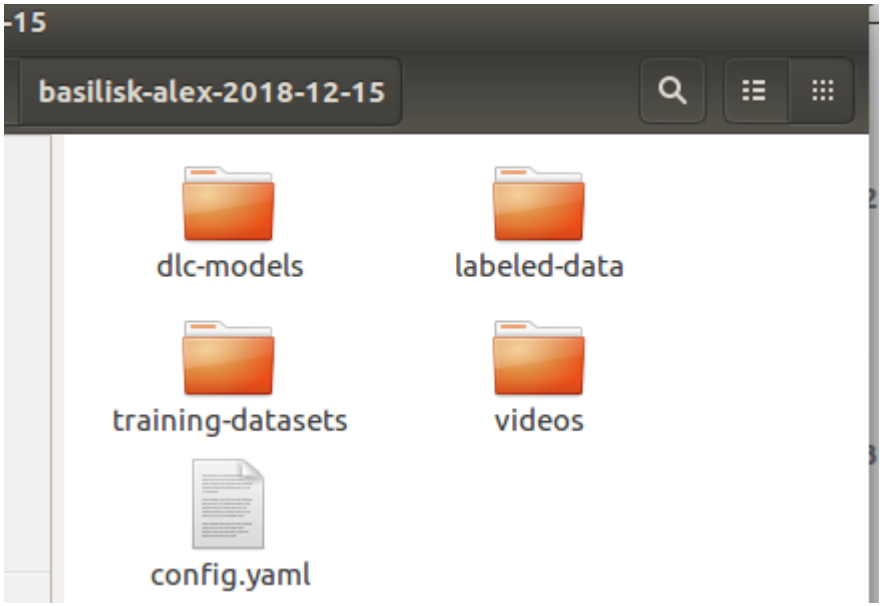
Config file

config.yaml (~/Desktop/basilisk-alex-2018-12-15) - gedit

Open

```
batch_size: 4
project_path: /home/alex/Desktop/basilisk-alex-2018-12-15
iteration: 0
colormap: hsv
alphavalue: 0.5
dotsize: 12
y2: 624
y1: 277
x2: 640
x1: 0
move2corner: false
corner2move2: !!python/tuple
- 50
- 50
pcutoff: 0.1
snapshotindex: -1
resnet: 50
TrainingFraction:
- 0.95
date: Dec15
scorer: alex
bodyparts:
- hand
- Finger1
- Finger2
- Joystick
numframes2pick: 20
stop: 1
start: 0
cropping: false
video_sets:
  /home/alex/Desktop/BasiliskL.avi:
    crop: 0, 640, 0, 480
Task: basilisk
```

Project folder:



Change:
- bodyparts (before labeling)
- numframes2pick (before frame extraction)

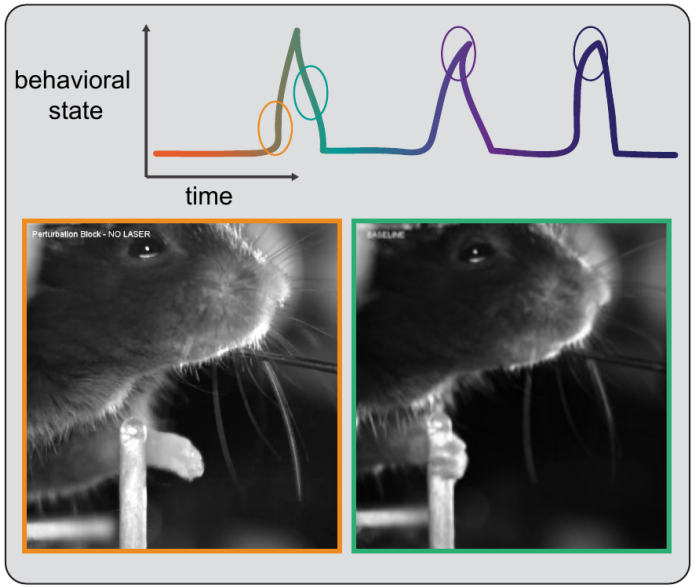
Frame extraction

Select videos to grab frames:

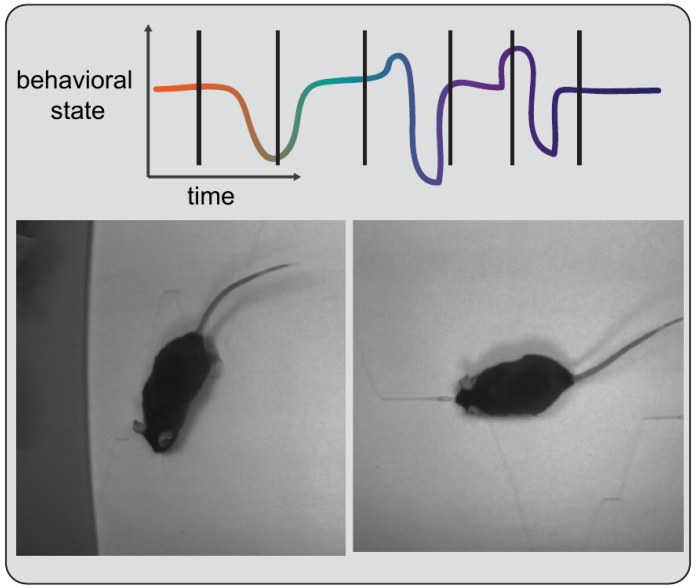
Use videos with image from:
- different sessions reflecting (if the case) varying light conditions, backgrounds, setups, and camera angles (etc).
- different individuals, especially if they look different (i.e. brown + black mice)

3 methods for frame extraction to create a labeled train/test set

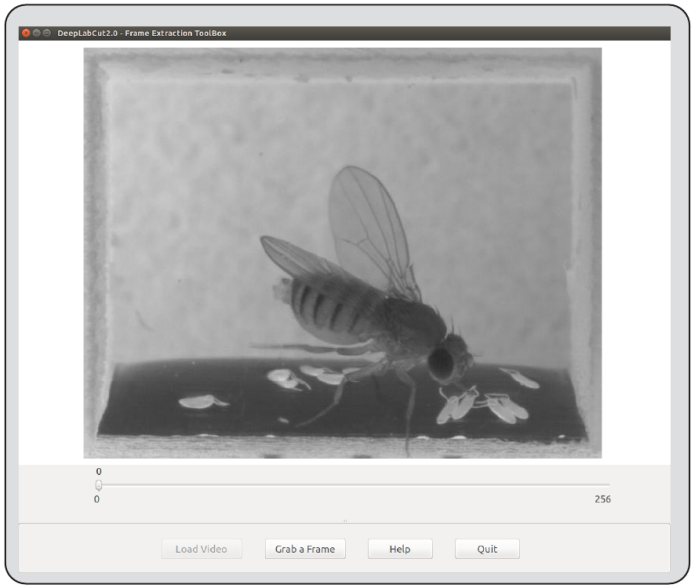
Image based clustering (k-means)



Random temporal sampling (uniform)



GUI for manual frame grabbing



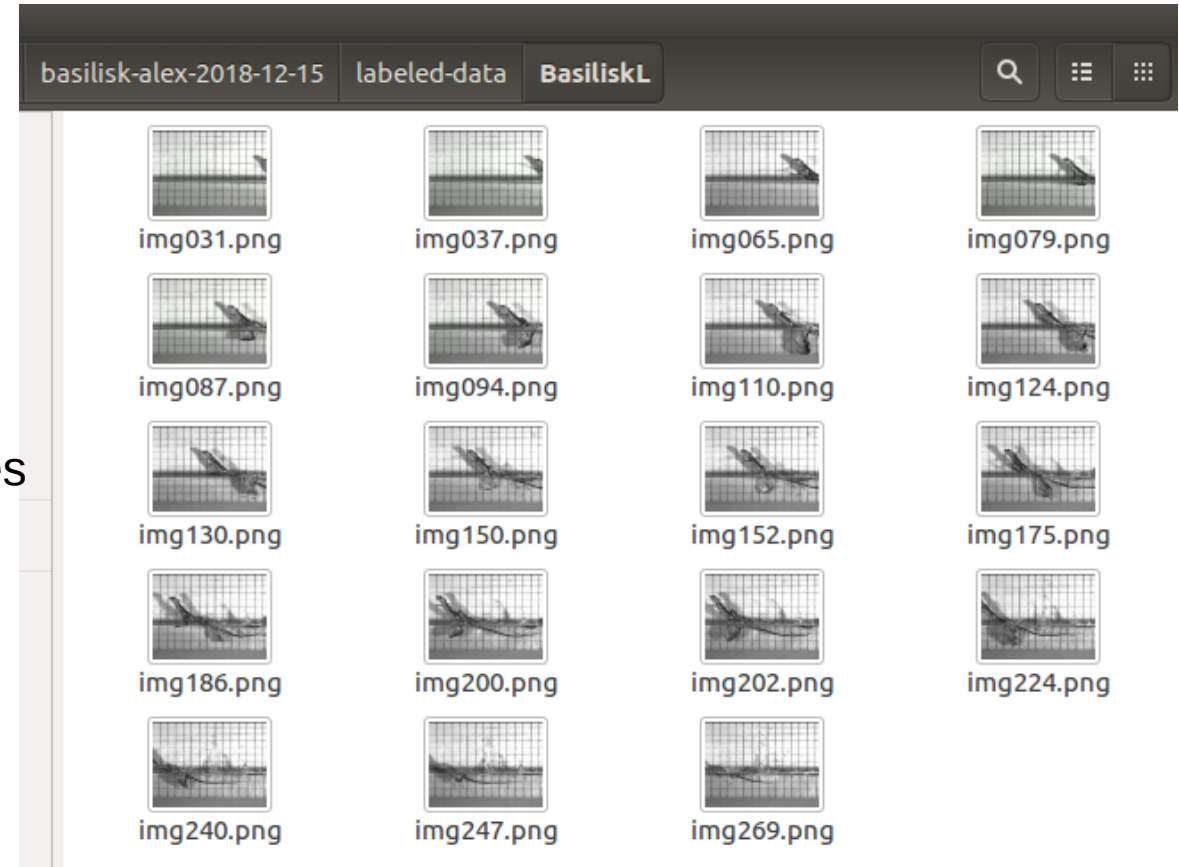
Nath*, Mathis*
BioRxiv 2018

Demo

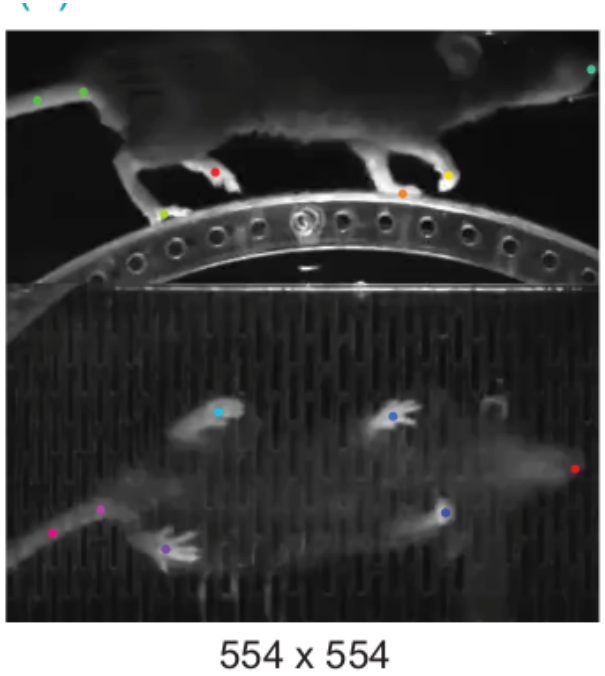
```
In [4]: deeplabcut.extract_frames(configfiledirectory)
Reading config file successfully...
Extracting frames based on kmeans ...
Kmeans-quantization based extracting of frames from 0.0  seconds to 10.01  secon
ds.
Extracting and downsampling... 300  frames from the video.
300it [00:00, 365.86it/s]
Kmeans clustering ... (this might take a while)

Frames are selected.
You can now label the frames using the function 'label_frames'.
```

Automatically extracted frames (clustering method)

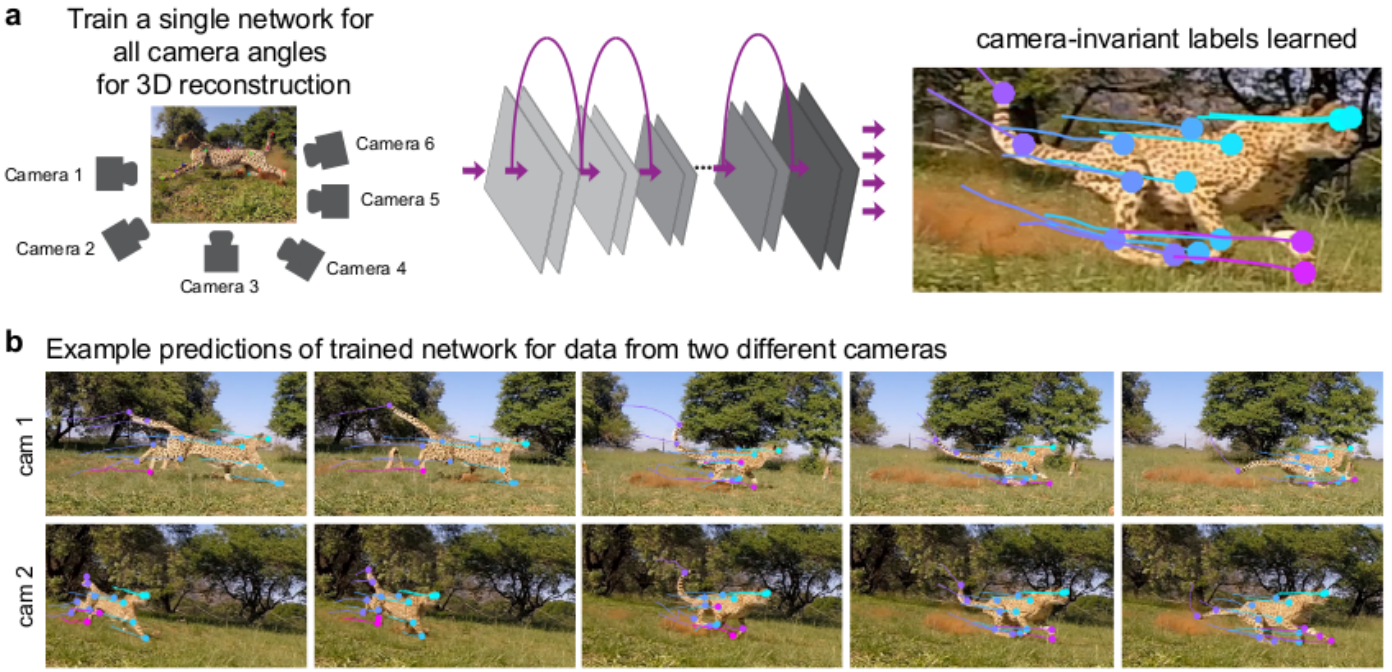


Labels can be **from/in** different views (cameras)



554 x 554

Mathis & Warren 2018

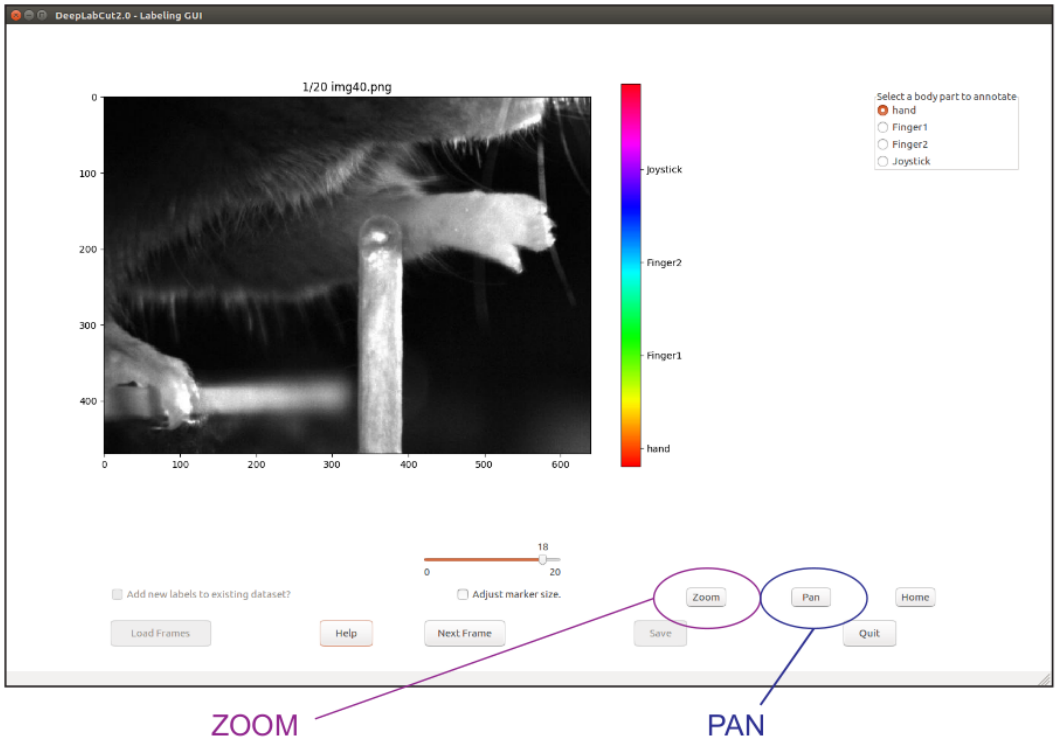


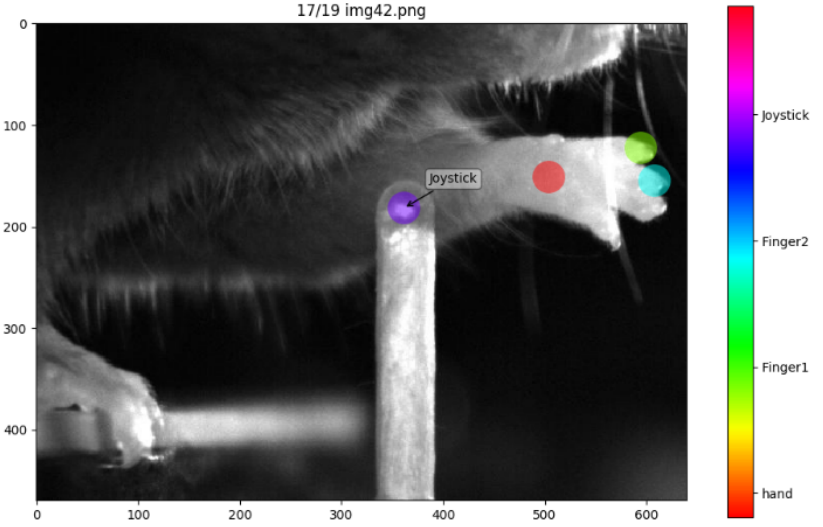
Nath*, Mathis* et al. 2018

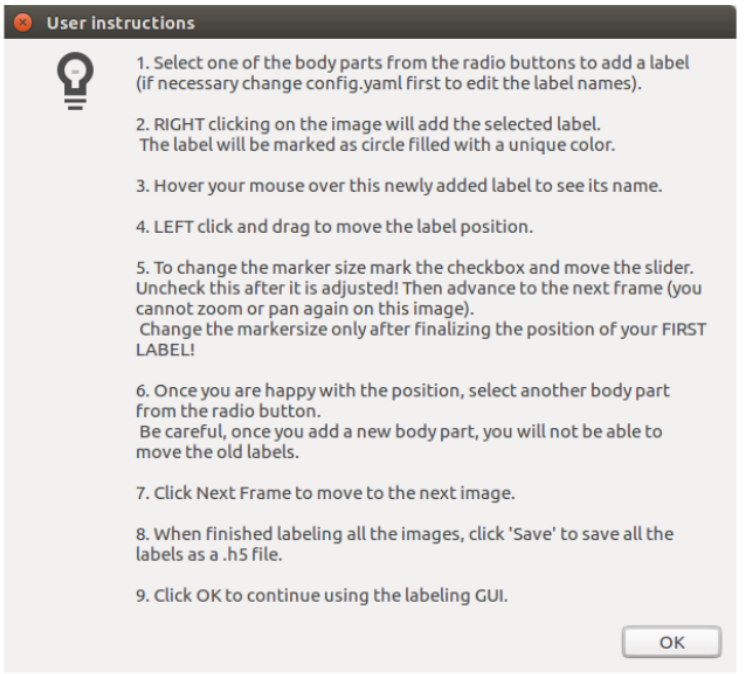
Labeling GUI

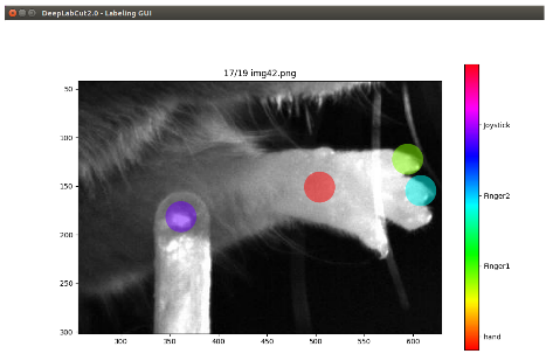
Label Frames using interactive GUI:

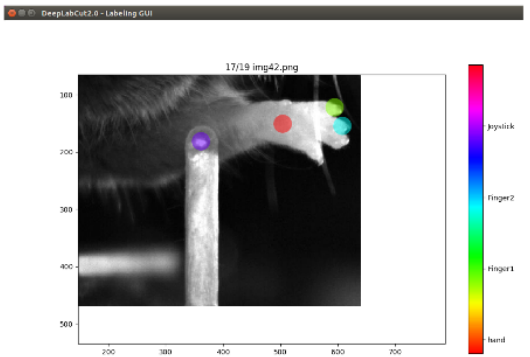
```
deeplabcut.label_frames(config_path)
```







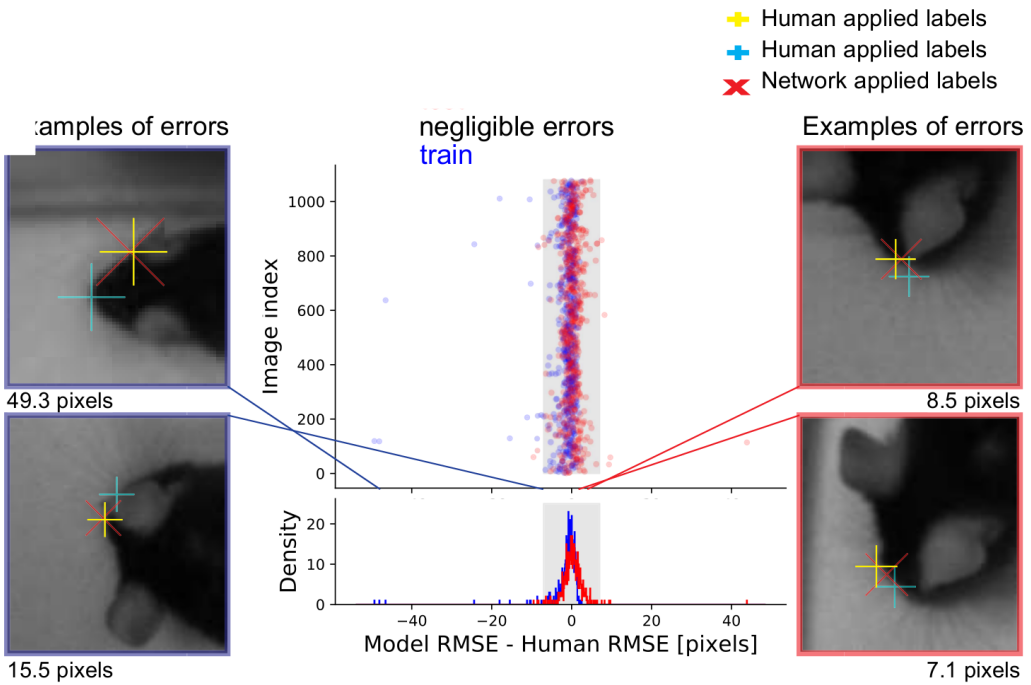




Labeling GUI features

- Zooming, Pan, auto advance (some hot keys)
- Only visible features have to be labeled
- The label (names) are defined in the config file
- The meaning is provided by the user
- Additional (new) labels can be added later to a project

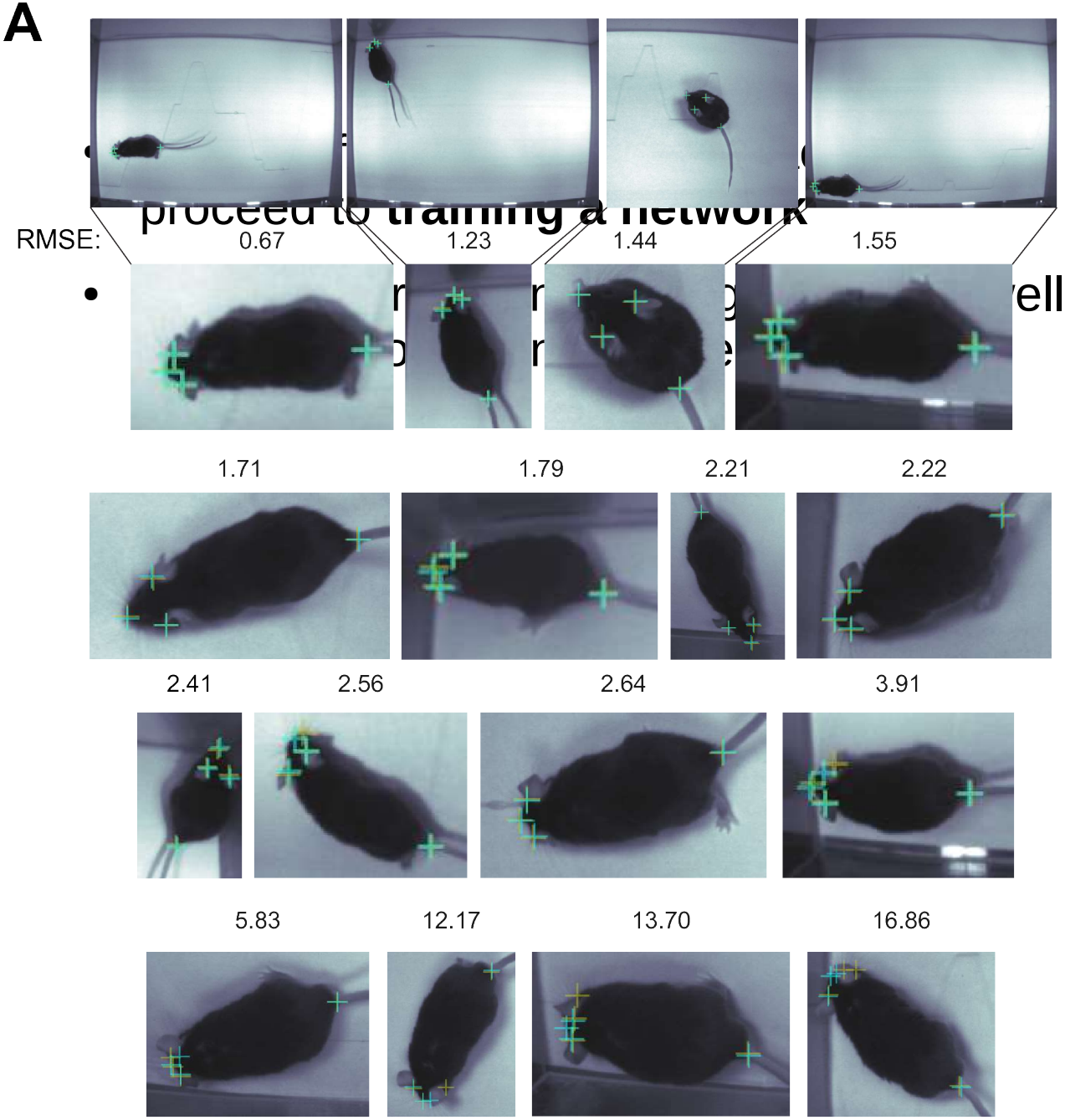
Labeling considerations



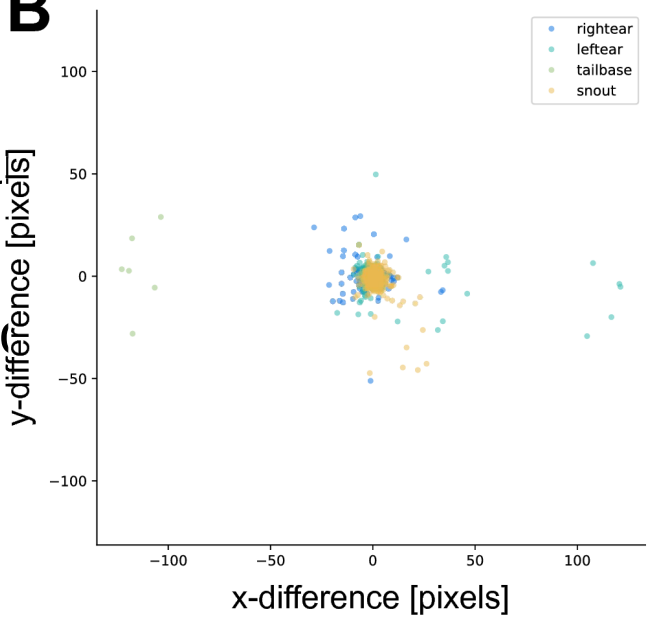
- Label (feature locations) consistently
- Errors are bad! (especially when only a few images are labeled)
- Left vs. right body parts can be if full animal visible (due to large rec. fields), but flipped labels are problematic
- You can check labels by plotting all the figures (always do this before training!)
- You can correct wrong labels manually with the refinement GUI!
- (currently adding wrongly not labeled body parts is not implemented; but body parts labels can be removed)

Benchmarking - Labeler accuracy

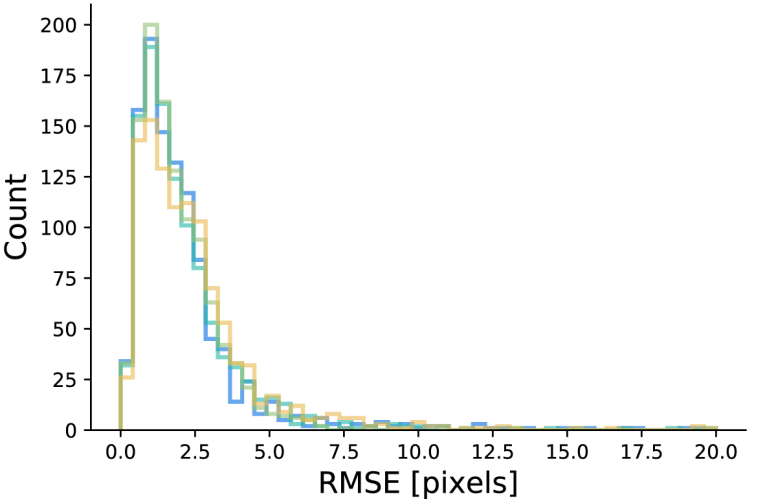
A



B



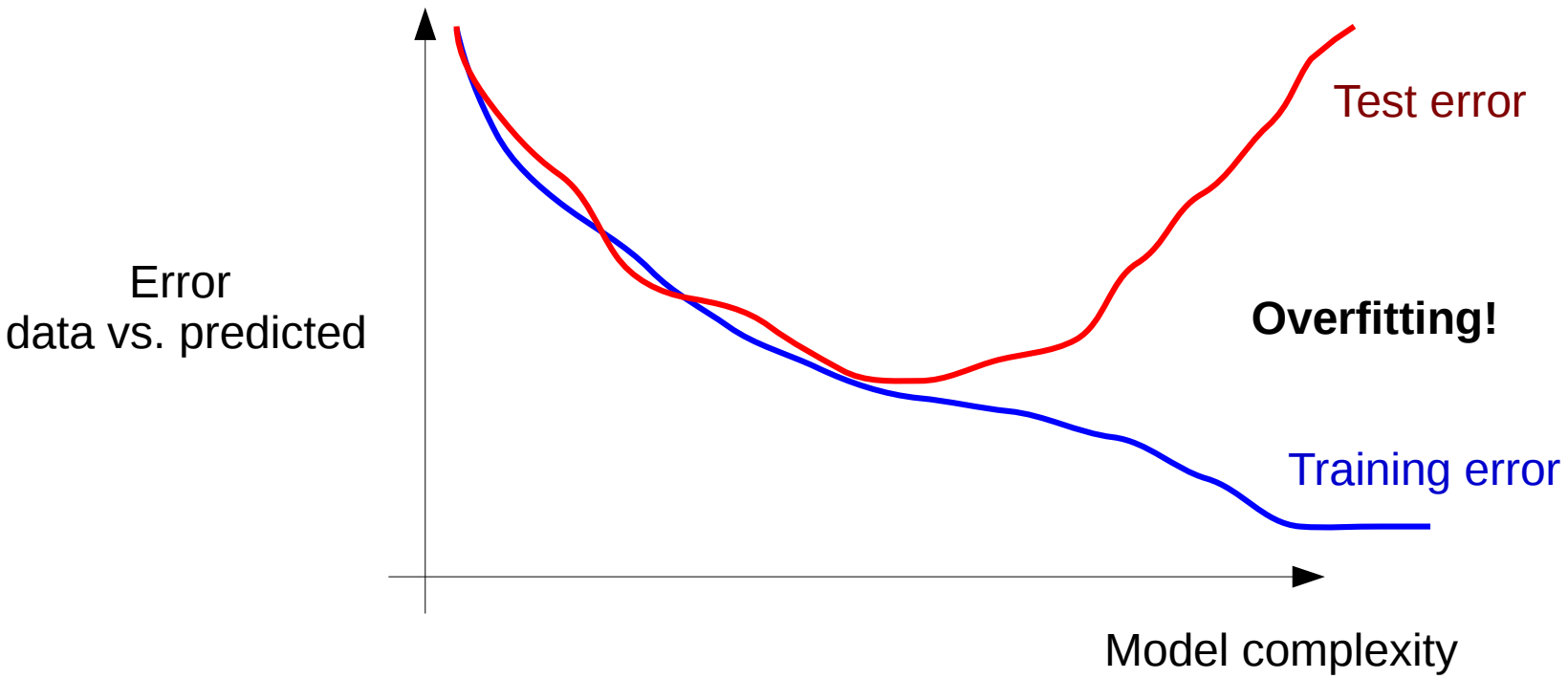
C



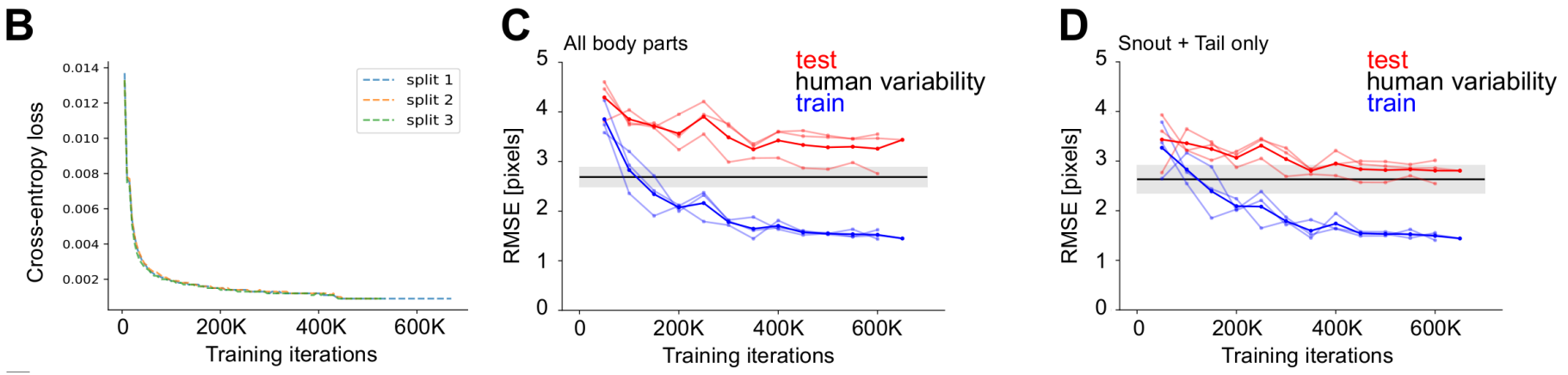
Mathis et al. Nature Neurosci Sept. 2018

Creating a training set

- Once all frames from the extracted folders are labeled, one can proceed to **training a network**
- To make sure the network generalizes well to unseen images, split the data into training and test set.

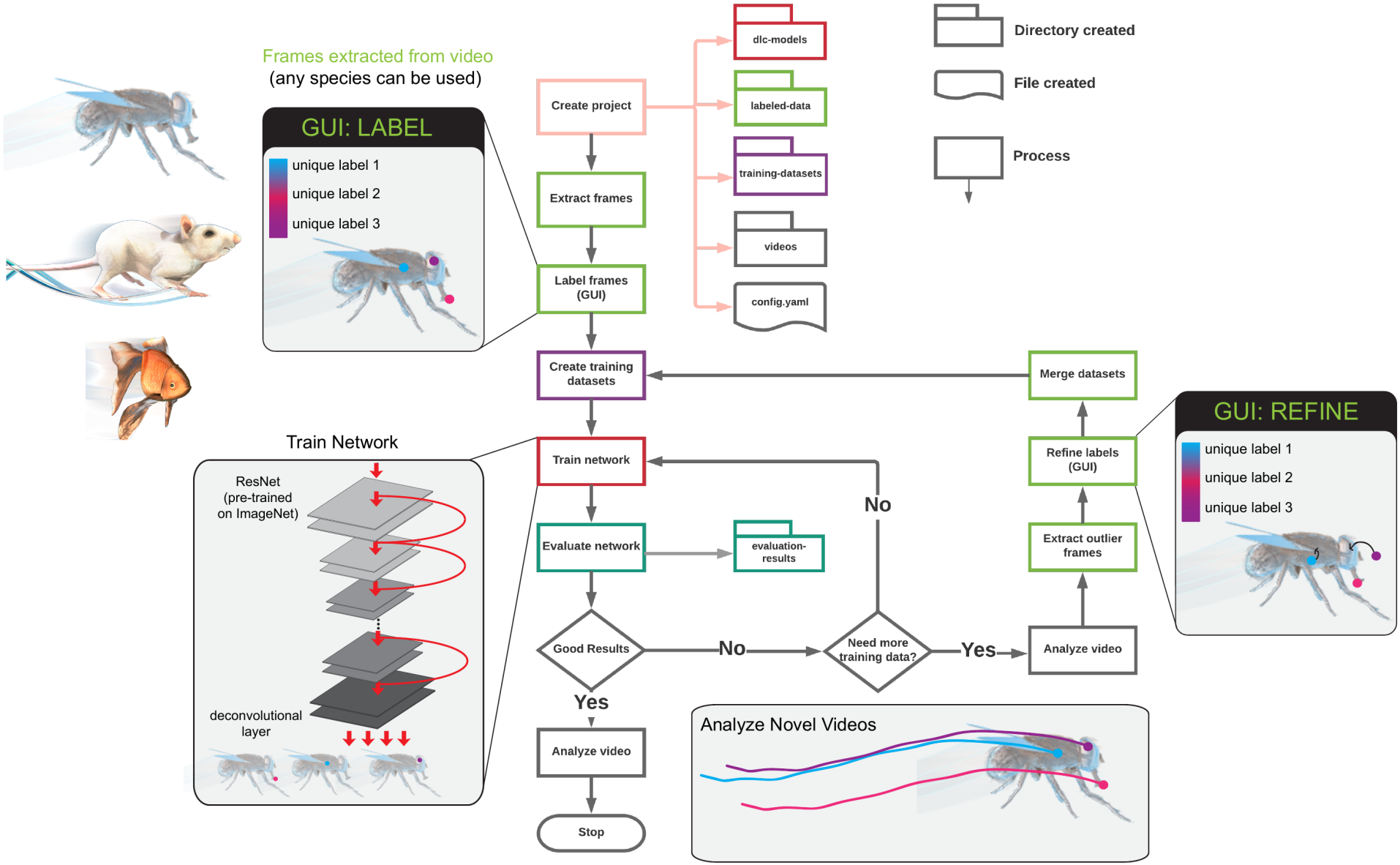


Training considerations



- Cross entropy loss minimized (measures difference in probabilities between epsilon disk around ground-truth and predicted scoremap)
- Training speed (mostly) depends on frame size! (and GPU, CPU type)
- Try to train until the “loss flattens”
- Store as many **snapshots** as necessary (intermediate weight steps; consider early stopping)
- Consider using augmentation (scale, auto-crop, *rotation, gamma*)

Workflow….



Nath*, Mathis*
BioRxiv 2018

Further demo code

There are complete demos (including labeled data) at:

- https://github.com/AlexEMG/DeepLabCut/tree/master/examples