Protocol for Labelling

This protocol shows how to use DeepLabCut (DLC) to label videos with similar settings, including background colour, brightness, and the number of mice. Figure 1 gives an example of 22 videos in a folder, in dark red and each with a single mouse.

Videos with similar settings

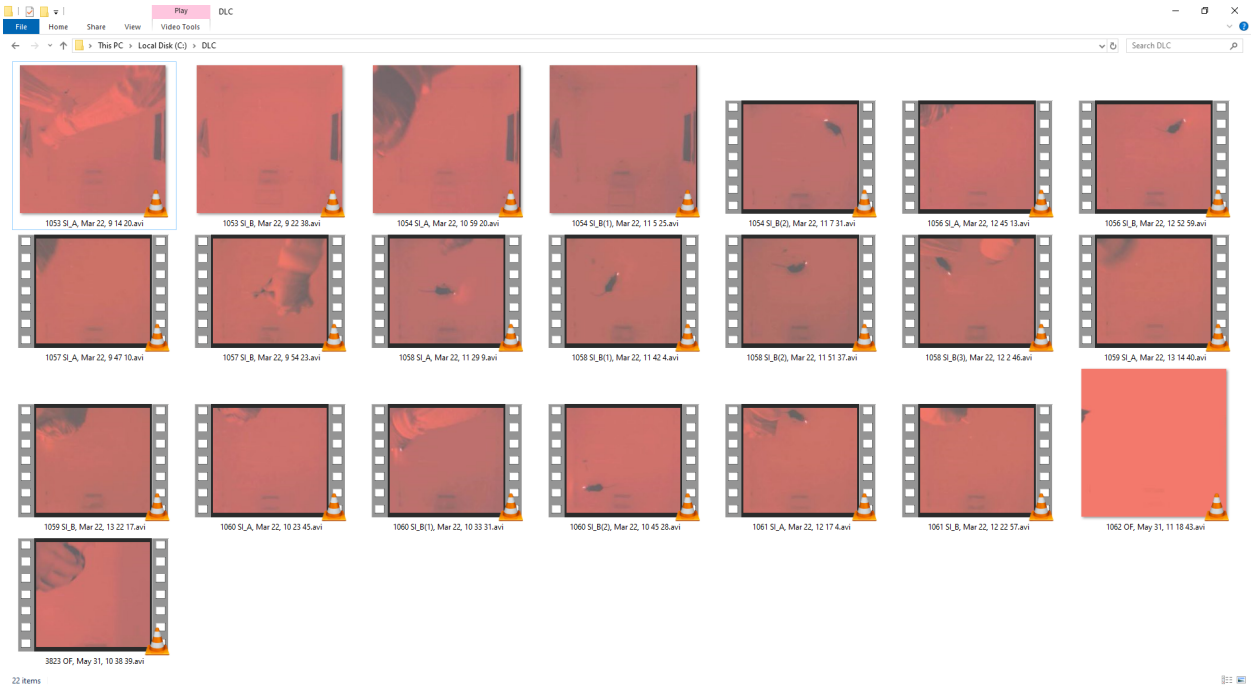

Figure 1.

A Jupyter Notebook 'labelling.ipynb' has been written to facilitate calling of DLC functions, where the four sub-headings break down the whole process into four parts. One by one they will be demonstrated below.

Create a Project

1. Run Cell 1.

```
In [1]:
# Import the required libraries
import deeplabcut as dlc
import os
```

```
# Prevent GPU memory from running out
os.environ['TF_FORCE_GPU_ALLOW_GROWTH'] = 'true'
```

2. Fill in the blanks in Cell 2 and run the Cell. Regarding the Figure 1 example, the blanks have been completed as follows:

```
In [2]:
# Change the current directory to where the videos are
os.chdir(r'C:\DLC')
# Name the project the filename of the video that comes first in alphabetical order,
without the suffix .avi/.mp4
task = '1053 SI_A, Mar 22, 9 14 20'
# Type in the filename of each video below, with the suffix
videos = ['1053 SI_A, Mar 22, 9 14 20.avi',
          '1053 SI_B, Mar 22, 9 22 38.avi',
          '1054 SI_A, Mar 22, 10 59 20.avi',
          '1054 SI_B(1), Mar 22, 11 5 25.avi',
          '1054 SI_B(2), Mar 22, 11 7 31.avi',
          '1056 SI_A, Mar 22, 12 45 13.avi',
          '1056 SI_B, Mar 22, 12 52 59.avi',
          '1057 SI_A, Mar 22, 9 47 10.avi',
          '1057 SI_B, Mar 22, 9 54 23.avi',
          '1058 SI_A, Mar 22, 11 29 9.avi',
          '1058 SI_B(1), Mar 22, 11 42 4.avi',
          '1058 SI_B(2), Mar 22, 11 51 37.avi',
          '1058 SI_B(3), Mar 22, 12 2 46.avi',
          '1059 SI_A, Mar 22, 13 14 40.avi',
          '1059 SI_B, Mar 22, 13 22 17.avi',
          '1060 SI_A, Mar 22, 10 23 45.avi',
          '1060 SI_B(1), Mar 22, 10 33 31.avi',
          '1060 SI_B(2), Mar 22, 10 45 28.avi',
          '1061 SI_A, Mar 22, 12 17 4.avi',
          '1061 SI_B, Mar 22, 12 22 57.avi',
          '1062 OF, May 31, 11 18 43.avi',
          '3823 OF, May 31, 10 38 39.avi']
```

3. Run Cell 3. This Cell will make a new folder in the video folder, which acts as the project directory (Figure 2).

```
In [3]:
path_config_file = dlc.create_new_project(task, '', videos, copy_videos = True)
```

Project directory

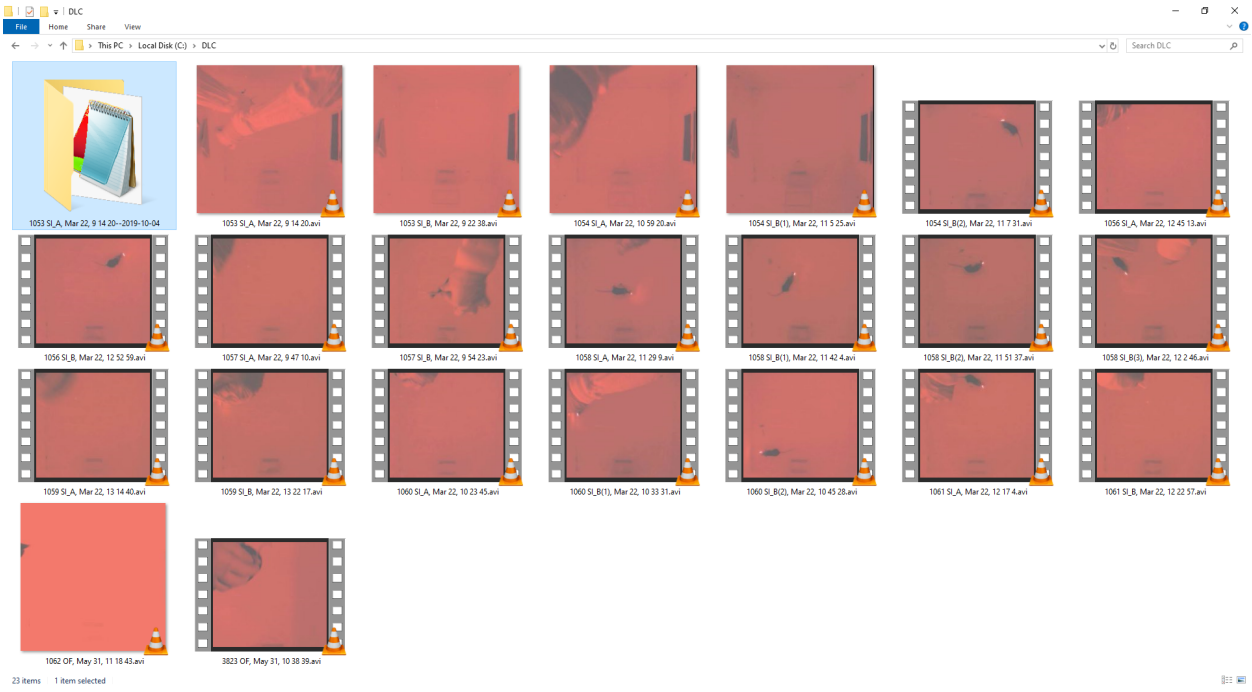

Figure 2.

4. As stated in Cell 4, the ‘config.yaml’ file shall be edited prior to continuing. Since there is only one mouse in the example, the template ‘conf_single.yaml’ is used. Usually, ‘bodyparts’, ‘numframes2pick’, and ‘dotsize’ will be altered (Figure 3). We call the head, body, and tail of the first mouse ‘h’, ‘b’, and ‘t’, respectively. Similarly, we name those of the second mouse ‘h2’, ‘b2’, and ‘t2’.

```
In [4]:
# Go to the project root directory and edit config.yaml according to the templates co
nf_*.yaml
```

Initial config.yaml and the template

```
bodyparts:
- bodypart1
- bodypart2
- bodypart3
- objectA
start: 0
stop: 1
numframes2pick: 20

# Plotting configuration
skeleton:
- - bodypart1
  - bodypart2
- - objectA
  - bodypart3
skeleton_color: black
pcutoff: 0.1
dotsize: 12
alphavalue: 0.7
colormap: jet
```

```
bodyparts:
- h
- b
- t

start: 0
stop: 1
numframes2pick: 50

# Plotting configuration
skeleton:
- - bodypart1
  - bodypart2
- - objectA
  - bodypart3
skeleton_color: black
pcutoff: 0.1
dotsize: 4
alphavalue: 0.7
colormap: jet
```

Figure 3.

Initial Neural Network

1. Fill in the blank in Cell 5 and run the Cell. Below the blank has been completed for the Figure 1 example, and we learnt the project root directory from Figure 2.

```
In [5]:
# Type in the full path of config.yaml, the project root directory followed by '/conf
ig.yaml'
# Skip this cell if you just created the project or haven't shutdown the notebook sin
ce then
path_config_file = r'C:\DLC\1053 SI_A, Mar 22, 9 14 20--2019-10-04\config.yaml'
```

2. Run Cell 6.

```
In [6]:
%matplotlib inline
dlc.extract_frames(path_config_file, algo = 'uniform', crop = False, userfeedback = F
alse)
```

3. Run Cell 7, and a graphical user interface (GUI) will in turn pop up.

```
In [7]:
%matplotlib inline
%gui wx
dlc.label_frames(path_config_file)
```

4. In the GUI, click ‘Load frames’, and next select the top folder or any other folder with the filename of the video of interest to you (Figure 4).

Labelling GUI, select the first folder

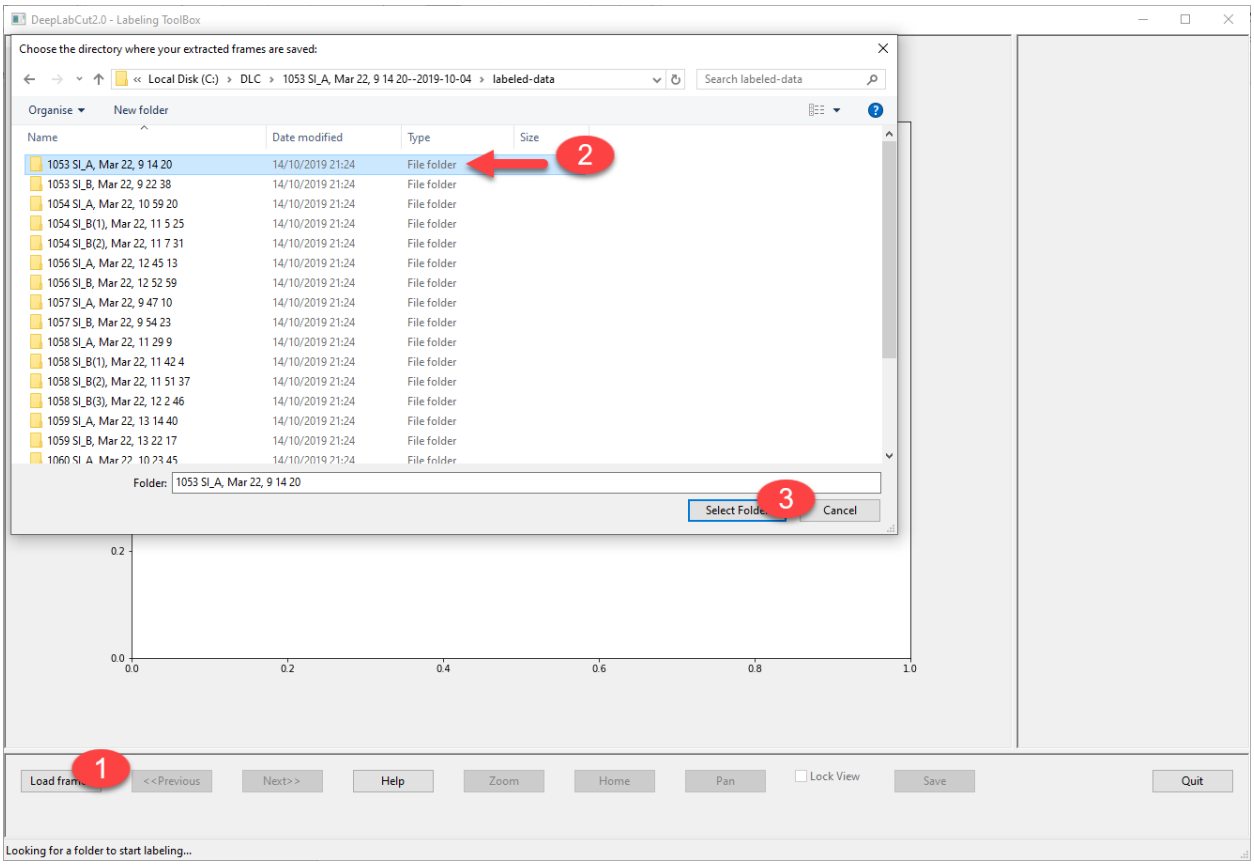

Figure 4.

5. Then right-click to add labels, middle-click to remove labels, and left-click and drag to adjust labels (Figure 5). Switch between bodyparts in the red box, and between frames in the green box. We generally deem the midpoint between two eyes as the head or ‘h’, the joint between the neck and the body as the body or ‘b’, and the joint between the body and the tail as the tail or ‘t’.

Labelling GUI, assign labels

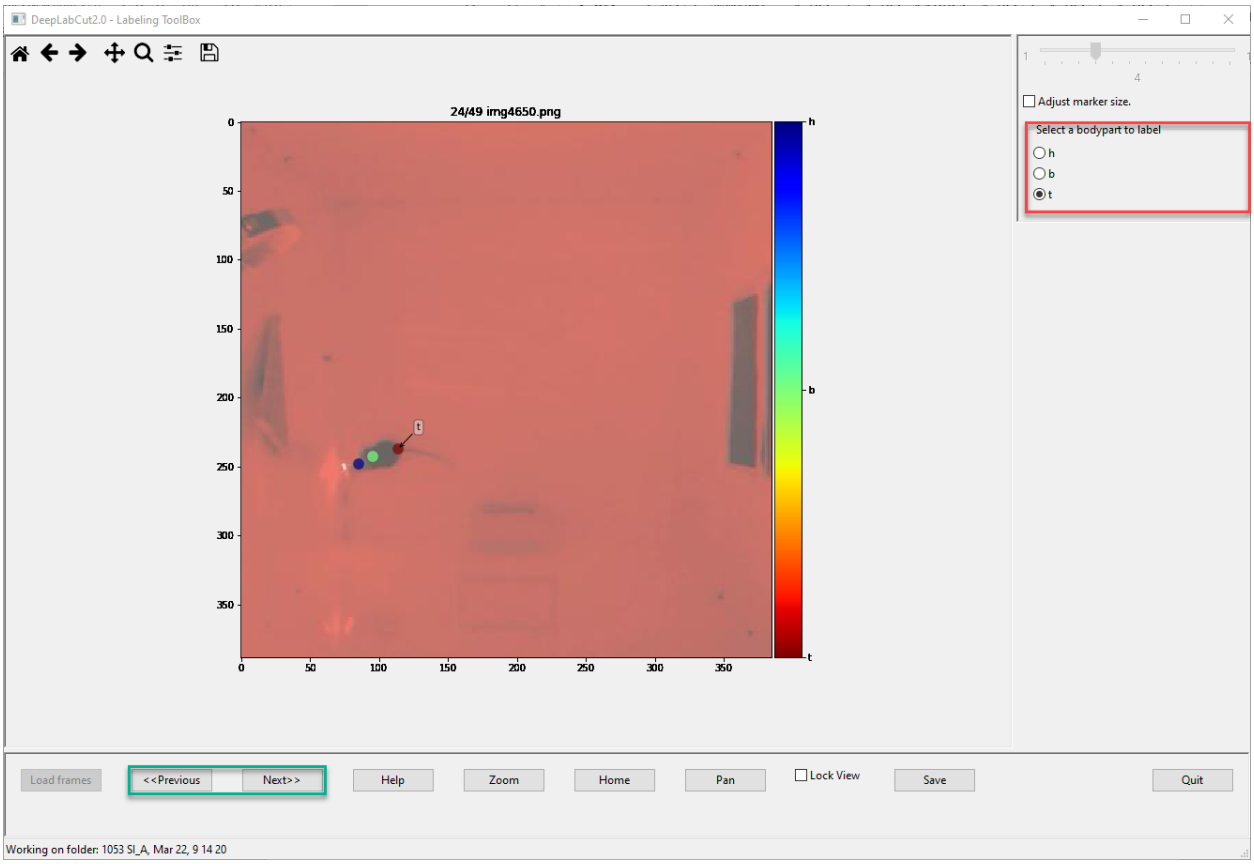

Figure 5.

6. After you label all the frames, click ‘Save’ and subsequently ‘Quit’ (Figure 6). There will be a pop-up asking for another folder with the filename of the video of interest. Choose ‘Yes’ to repeat Steps 4–6. Or choose ‘No’ to continue.

Labelling GUI, save and select the next folder

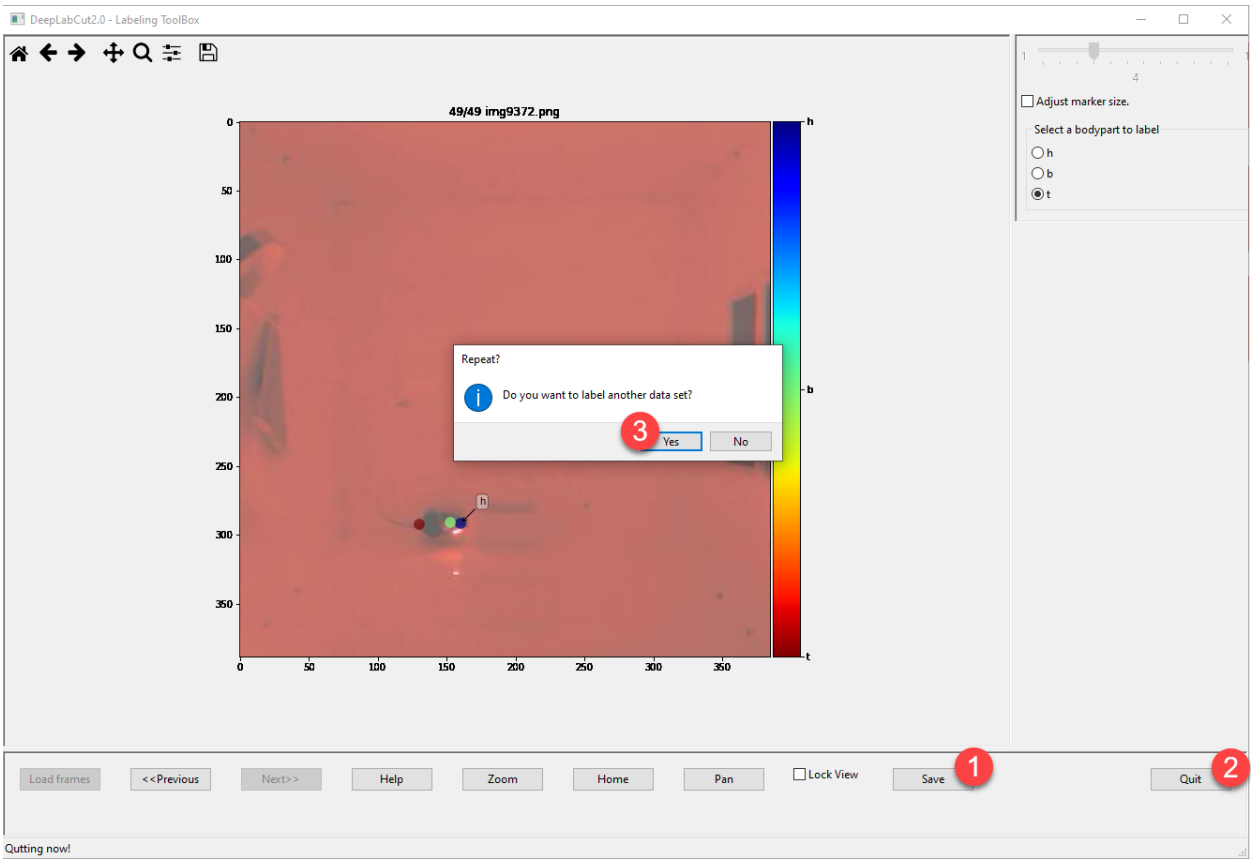

Figure 6.

7. Run Cells 8–11.

```
In [8]:
dlc.check_labels(path_config_file)
In [9]:
dlc.create_training_dataset(path_config_file)
In [10]:
# Turning on autotune speeds up the training
# maxiters: Iterations of training, recommended: 200,000
# displayiters = 1000: Display training progress per 1,000 iterations
# saveiters = 5000: Save the neural network per 5,000 iterations
dlc.train_network(path_config_file, shuffle = 1, autotune = True, displayiters = 100
0, saveiters = 50000, maxiters = 200000)
In [11]:
dlc.evaluate_network(path_config_file)
```

Label Videos Mechanically

1. Fill in the blank in Cell 12 and run the Cell, which is the same as Cell 5.

```
In [12]:
# Type in the full path of config.yaml, the project root directory followed by '/conf
ig.yaml'
# Skip this cell if you haven't shutdown the notebook since creating the project
path_config_file = r'C:\DLC\1053 SI_A, Mar 22, 9 14 20--2019-10-04\config.yaml'
```

2. Run Cell 13. If you want to label certain videos in a particular folder, complete the blanks in Cell 14 and run the Cell instead, albeit not recommended.

```
In [13]:
# Read config.yaml and get the paths of the listed videos
cnf = dlc.auxiliaryfunctions.read_config(path_config_file)
videos = cnf['video_sets'].keys()
In [14]:
# (Not recommended) Or manually type in the video directory and filenames below
os.chdir(r'VIDEO_DIRECTORY')
videos = ['VIDEO_FILENAMES.avi/mp4']
```

3. Run Cells 15–17, and check the .h5 files, labelled videos, and plots created in the 'videos' folder in the project directory (Figure 7).

```
In [15]:
dlc.analyze_videos(path_config_file, videos)
In [16]:
dlc.create_labeled_video(path_config_file, videos)
In [17]:
%matplotlib notebook
dlc.plot_trajectories(path_config_file, videos)
```

.h5, labelled videos, and plots

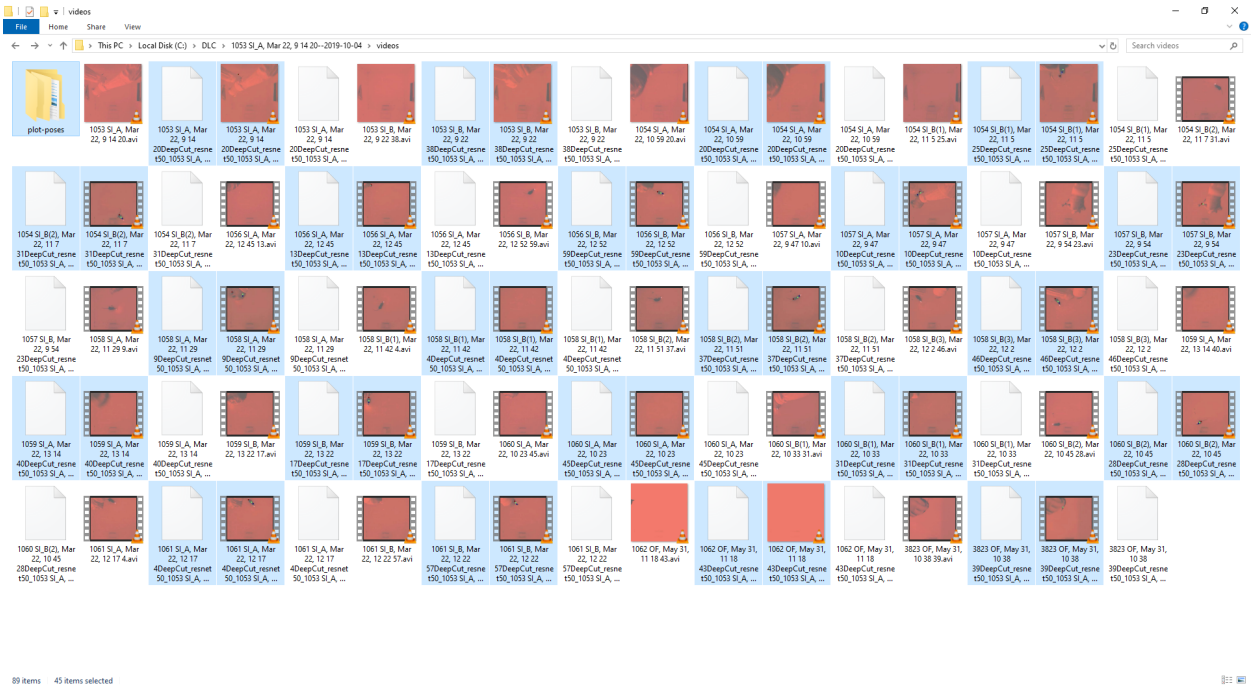

Figure 7.

Refine Neural Network

1. Fill in the blank in Cell 18 and run the Cell, which is identical to Cell 5.

```
In [18]:
# Type in the full path of config.yaml, the project root directory followed by '/conf
ig.yaml'
# Skip this cell if you haven't shutdown the notebook since creating the project
path_config_file = r'C:\DLC\1053 SI_A, Mar 22, 9 14 20--2019-10-04\config.yaml'
```

2. Run Cell 19. If you have labelled certain videos in a specific folder, complete the blanks in Cell 20 and run the Cell instead, albeit not recommended.

```
In [19]:
# Read config.yaml and get the paths of the listed videos
cnf = dlc.auxiliaryfunctions.read_config(path_config_file)
videos = cnf['video_sets'].keys()
In [20]:
# (Not recommended) Or manually type in the video directory and filenames below
os.chdir(r'VIDEO_DIRECTORY')
videos = ['VIDEO_FILENAMES.avi/mp4']
```

3. Run Cells 21–22, and a GUI that resembles the one in the ‘Initial Neural Network’ part will pop up. Left-click and drag to move labels to correct positions, or middle-click to remove labels if there is no mouse, or the mouse had yet to be placed in the field.

```
In [21]:
dlc.extract_outlier_frames(path_config_file, videos)
In [22]:
%matplotlib notebook
%gui wx
dlc.refine_labels(path_config_file)
```

4. Run Cells 23–26.

```
In [23]:
dlc.merge_datasets(path_config_file)
In [24]:
dlc.create_training_dataset(path_config_file)
In [25]:
# Turning on autotune speeds up the training
# maxiters: Iterations of training, recommended: 200,000
# displayiters = 1000: Display training progress per 1,000 iterations
# saveiters = 5000: Save the neural network per 5,000 iterations
dlc.train_network(path_config_file, shuffle = 1, autotune = True, displayiters = 100
0, saveiters = 50000, maxiters = 200000)
In [26]:
dlc.evaluate_network(path_config_file)
```

5. Go back and redo the ‘Label Videos Mechanically’ part.