

DEEPLABCUT

Slutsky Lab

Lior de Marcos

Last update: 10.4.22

Contents

Manual Change Log.....	2
Introduction	3
Installation	4
Prepare GPU	4
Check if your GPU is Compatible:	5
GPU Engagement.....	6
Known Troubleshooting	9
Install DeepLabCut	10
Check if your Installation is OK.....	13
Known troubleshooting:.....	14
Usage.....	18
Basics	18
Entering DeepLabCut Environment.....	18
iPython.....	18
Analyzing Videos	19
Check your Video (~18 of the video time)	19
Make Your Video Smaller (~18 of the video time for each function)	20
Prepare network for analysing	22
Video Analysis (dependent on video size, and system used, usually long).....	24
Known troubleshooting:.....	26

Manual Change Log

Date	Chapter	Change
10-5-22	All	Publish

Introduction

DeepLabCut (DLC) is a python-based package for tracking key-points in a video. It was created by Mathis' lab ([Mathis, A., at el. 2018 *Nature Neuroscience*, Lauer, J. at el.2022 *Nature Methods*](#)) and build on google TensorFlow toolbox (<https://www.tensorflow.org/>).

You don't need to know any python to use it – few lines of codes (written within this manual) will let you use it.

DLC is very computational expensive – its center steps (analyzing videos & training networks) take relatively long time. It is necessary to have accelerator available for those steps, such as GPU. Without it, run time may be 10 times longer.

You can find strong GPU in cloud, analysis PC or if your personal PC have a strong one – see [Check if your GPU is Compatible](#): to check if your GPU is good enough.

If you don't have a GPU you can make your video smaller, to prepare for analysis, on any computer. The main analysis requires a GPU.

You can find demos and more info in DeepLabCut YouTube channel (https://www.youtube.com/channel/UC2HEbWpC_1v6i9RnDMY-dfA/featured), or in the GitHub repository & docs (<https://github.com/DeepLabCut/DeepLabCut>).

Happy DeepLabCutting!

Installation

If our PC has a strong GPU, install everything.

If you don't, then just skip to [Install DeepLabCut](#) – without GPU you can [Make Your Video Smaller](#) ($\sim \frac{1}{8}$ of the video time for each function) before analysis. All other analysis steps can be done on the cloud or analysis pc.

Prepare GPU

Engaging your graphic card may take relatively long, stay strong! You will probably only need to do it once.

GPU computation requires (1) a GPU, (2) Graphic driver (3) Microsoft visual studio (4) CUDA & (5) tensor flow, which all need to be compatible (see below).

For example, the latest version of CUDA is 11.5, and the latest version of visual studio is 2022. The latest Driver 511.09 (for GeoForce 3060) support up to the latest CUDA. DeepLabCut support for now the latest version of tensorflow, 2.7, however it only works with CUDA 11.2, which requires visual studio 2019. The following steps will guide through this nightmare of compatibility.

All of this must be done before installing DeepLabCut in its conda environment.

For more info about compatibility, see here:

<https://github.com/DeepLabCut/DeepLabCut/blob/master/docs/installation.md#gpu-support>

You can see more info about technical considerations here:

<https://github.com/DeepLabCut/DeepLabCut/blob/master/docs/installation.md#technical-considerations>

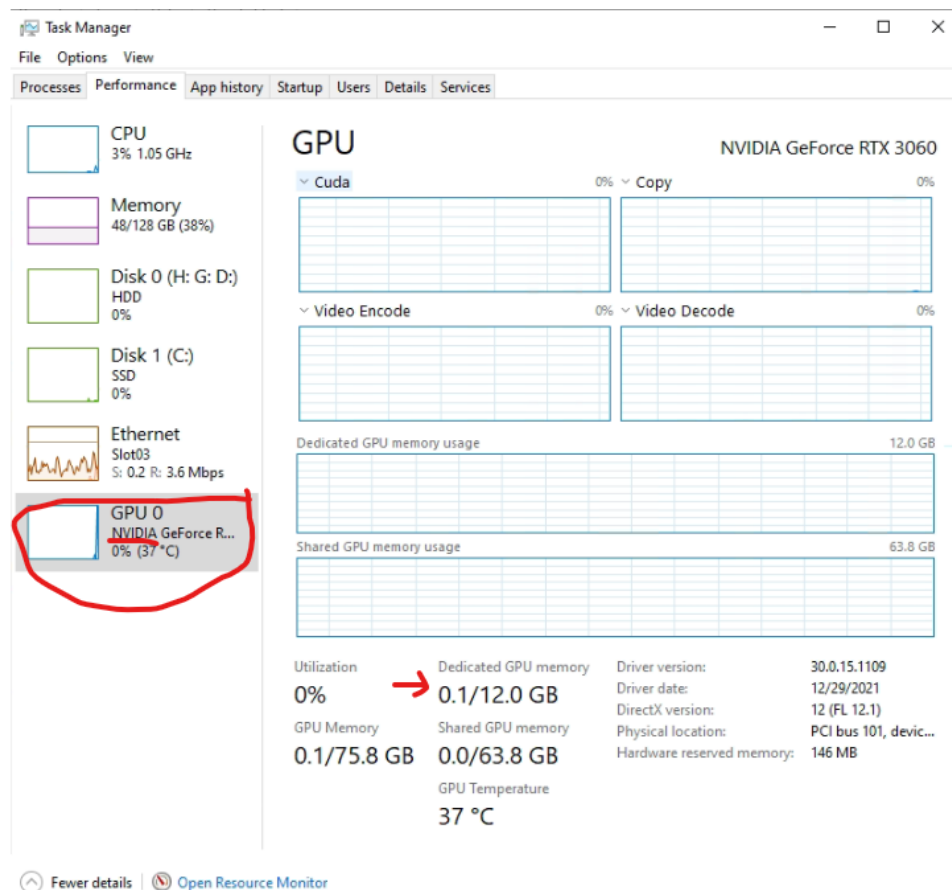
and here:

https://www.tensorflow.org/install/gpu#hardware_requirements

Check if your GPU is Compatible:

(About 5 minutes)

1. Check what is your GPU:
 - a. Open task manager, expand it if necessary.
 - b. Go to performance tab. On the left you can see your devices – you can see more info on each by clicking on them.
2. You may have more than 1 GPU. For each GPU check:
 - a. It is a Nvidia – GPU.
 - b. Dedicated GPU memory – should be at least 8 GB (see technical considerations above). Less than that may cause crashes / low speed.



- c. GPU compute ability – this must be at least 3.5. Simply search for your GPU here:
<https://developer.nvidia.com/cuda-gpus>.
3. Perform [GPU Engagement](#) for the chosen GPU. If no GPU passed the criteria in 2, you are unable to use the GPU in this computer. See what you can do [above](#).

[GPU Engagement](#)

In this step you will install everything the GPU need to connect with DeepLabCut.

(About 1-2 hours, require internet connection)

1. If you have existing graphic driver for this GPU, CUDA or Microsoft visual studio installed, remove them.
2. Uninstall Anaconda if it is already installed on the computer.
You can try to skip this step; however, a lot of time GPU isn't engaged if you install Anaconda before doing everything else for the GPU installation. If this happened, try to uninstall Anaconda & install it anew (without repeating everything else).
3. Update your GPU driver:
 - a. Enter here: <https://www.nvidia.com/Download/index.aspx>, and find GPU & Operating system. In download type, prefer studio drivers over game drivers / new feature branch.
 - b. Write down the driver version.
 - c. Follow instructions and finish downloading & any installations. A restart may be required.
4. Decide what CUDA version to use:
 - a. Find the maximal CUDA versions matching your driver here:
<https://docs.nvidia.com/cuda/cuda-toolkit-release-notes/index.html>, in table 2.
 - b. Check the installed TensorFlow DeepLabCut is using, with google Colab:
 - i. Open Colab: <https://colab.research.google.com/>, and select "New notebook".

ii. In the empty cell, write:

```
!pip install deeplabcut
```

```
!pip list -v | grep tensorflow
```

iii. Click the “play” button next to the cell and wait. The TensorFlow version will be at the end of the output, at a semi “table”, next to the text “tensorflow”. Write it down.

c. Find the CUDA versions matching each TensorFlow here:

https://www.tensorflow.org/install/source_windows#gpu.

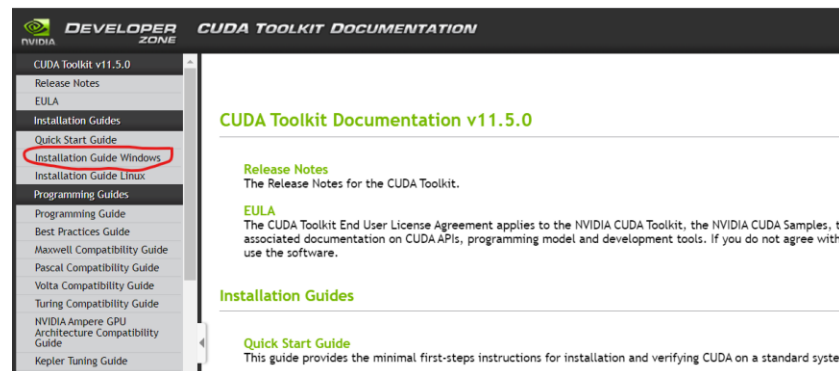
d. Integrate the information – choose the latest TensorFlow & CUDA versions that support each other and are supported by both DLC & your Driver. Below TensorFlow 2.5 some stability & speed losses may occur.

e. Write both the chosen CUDA & TensorFlow versions down, for later usage.

5. Find what type of Visual Studio to install:

a. Open the CUDA documentations matching your desired CUDA version, found here: <https://docs.nvidia.com/cuda/archive/>

b. On the menus on your left, under “Installation Guides” choose “Installation Guide Windows”.



c. From Table 2 find what Visual Studio is supported – take the newest written.

6. Download visual studio. You can find all the versions of visual studio here:

<https://visualstudio.microsoft.com/vs/older-downloads/>

For older versions, you may need to sign in to your Windows account, or register to the “free Dev Essentials program” (follow instructions in the link above). Base visual studio is usually enough, without “workloads” if offered.

7. Download the CUDA version you written; you can find all versions here:
<https://developer.nvidia.com/cuda-toolkit-archive>, or using google search. Both local & network Installers are good.

8. Check if your GPU setting was successful:

- Open terminal – simply type cmd in the windows search (near the start icon).
- Run in the terminal:

nvidia-smi

You should get a table such as:

NVIDIA-SMI 462.59				Driver Version: 462.59		CUDA Version: 11.2	
GPU	Name	TCC/WDDM		Bus-Id	Disp.A	Volatile	Uncorr. ECC
Fan	Temp	Perf	Pwr:Usage/Cap	Memory-Usage		GPU-Util	Compute M.
						MIG	M.
0	GeForce MX330	WDDM		00000000:02:00:0	Off	N/A	
N/A	41C	P0	N/A / N/A	64MiB /	2048MiB	4%	Default
						N/A	
Processes:							
GPU	GI	CI	PID	Type	Process name	GPU Memory	
						Usage	
No running processes found							

Note the Driver Version, see that it matches what you would expect. CUDA version here may be different than expected, that is OK (see next stage & here: <https://stackoverflow.com/questions/53422407/different-cuda-versions-shown-by-nvcc-and-nvidia-smi>)

- Run in the terminal:

nvcc -V

You should get an output such as: (Note the CUDA version, should match what you expect)

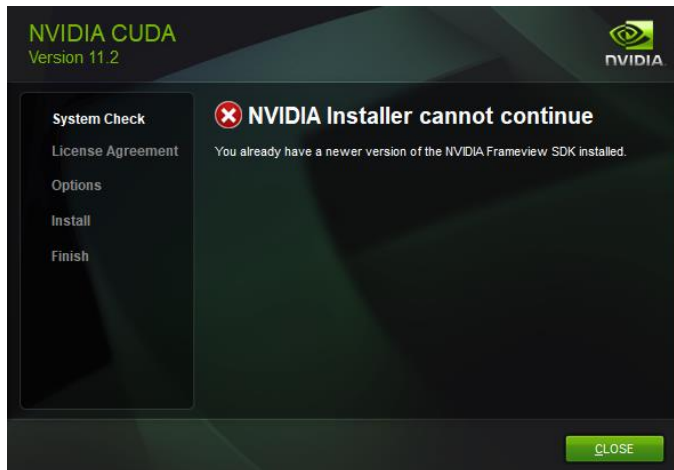
```
nvcc: NVIDIA (R) Cuda compiler driver
Copyright (c) 2005-2020 NVIDIA Corporation
Built on Mon_Nov_30_19:15:10_Pacific_Standard_Time_2020
Cuda compilation tools, release 11.2, V11.2.67
Build cuda_11.2.r11.2/compiler.29373293_0
```

- Continue to step 3: CPU Installation. Note during step 3 & 4 any “In case of GPU installation” notes.

Known Troubleshooting

- When installing CUDA, the installer gives an error “You already have a newer version of NVIDIA Frameview SDK installed” (Solution from:

<https://forums.developer.nvidia.com/t/nvidia-installer-cannot-continue/169854>)



1. Uninstall NVIDIA Frameview SDK using windows “add/remove program”.
 2. Restart the CUDA installer – it should install successfully now.
 3. Continue to the next stage, as usual.
- If driver does not match expected during:
nvvidia-smi
Or if
nvcc -V
return error such as:
**'nvcc' is not recognized as an internal or external command,
operable program or batch file.**
Installation failed.
Remove any GPU driver, CUDA & Visual Studio, using windows “add/remove program”,
and start from the beginning.
 - If CUDA version does not match when running:
nvcc -V
Remove any CUDA & Visual Studio using windows “add/remove program” and start
again from stage 6 “Download visual studio”.

Install DeepLabCut

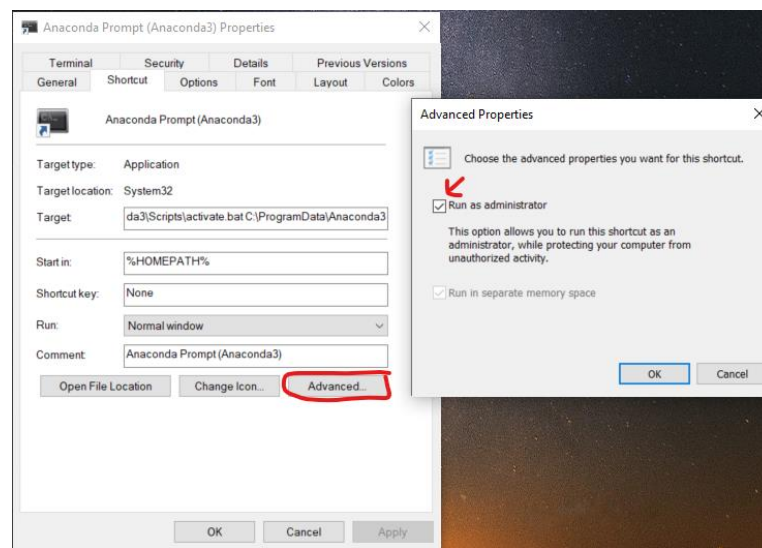
In this step you will install DeepLabCut at its own cuda environment

(About 15 minutes, require internet connection)

Video instructions (in the manual I use a few more steps, that prevent some bugs & user errors):

<https://www.youtube.com/watch?v=LYYoVTdiRQ8&list=PLipMSEOb9vRFefBwT4l6kCfAXDJ8uHJiq&index=5>

1. Download & install Anaconda from: <https://www.anaconda.com/products/individual> (will automatically find system).
2. Make Anaconda prompt to run as administrator by default from the shortcut:
 - a. Right click the desktop shortcut, and open properties.
 - b. On the Shortcut tab, click "Advanced".
 - c. Check the checkbox next to "Run as administrator".
 - d. Any time you open the Anaconda prompt, allow program to make device changes.



3. Open "Anaconda Prompt", and run:
conda update conda

4. Download the conda file from the bottom of the Mathis lab – DeepLabCut page:

<http://www.mackenziemathislab.org/deeplabcut>



5. Copy the path to folder hosting the file you downloaded. If it is “Downloads”, the easiest way is to shift-right click on “Downloads” in the quick access (in the left of the File Explorer).
6. Change to current folder (can be seen in your current prompt line) to where the downloaded file is saved:

- a. Run:

```
cd /d <conda file mother folder path>
```

Note the direction of the slash.

Place path to the folder containing the conda file instead of “<conda file mother folder path>” –

you can do it by simply right click on the console to paste the path copied earlier.

For example:

```
cd /d D:\user\Downloads
```

- b. Verify that the path to current folder (left to where you write in the prompt) is now the expected path.

7. Run:

```
conda env create -f DEEPLABCUT.yaml --name DLC
```

in the prompt, say yes to any question & wait until finished.

8. When finished, run:

```
conda activate DLC
```

In the parentheses at the beginning of the prompt, instead of “base” now it will write DLC.

9. **In case of GPU installation:** run:

```
conda install -c conda-forge cudnn
```

10. **In case of GPU installation:** If you decided on a different TensorFlow then DeepLabCut default, run:

```
pip install tensorflow==<your desired Tensorflow version>
```

Replace “<your desired Tensorflow version>” with the TensorFlow version you had written earlier ([here](#)).

For example:

```
pip install tensorflow==2.5
```

11. Run:

```
pip install --upgrade deeplabcut
```

12. Continue to Step 4: Check if your Installation is OK.

Check if your Installation is OK

1. Check your DeepLabCut version:

a. Open Anaconda prompt as administrator (should be done automatically, as you set [above](#)).

b. Enter your conda environment by running:

```
conda activate DLC
```

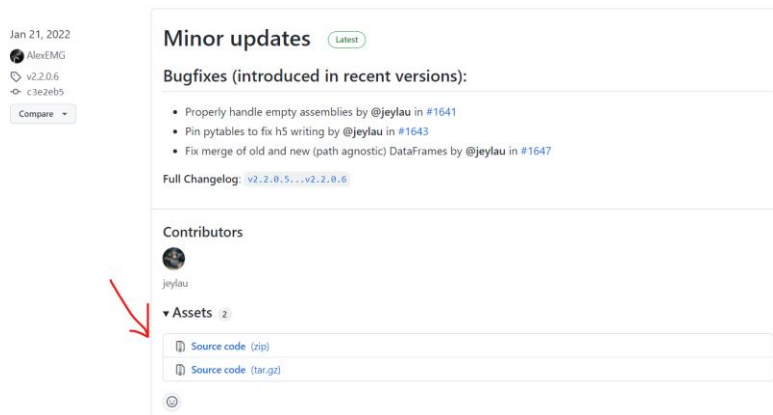
c. Run:

```
conda list deeplabcut
```

2. Download from GitHub DeepLabCut releases the source code relevant to you release.

Can be done by finding your version here:

<https://github.com/DeepLabCut/DeepLabCut/releases>, finding the relevant source code under “Assets”, download and then unzip.



3. Copy the path to the unzipped folder. Can be done simply by shift-right clicking the unzipped folder and choosing “copy as path”.

4. Navigate to the unzipped folder by running:

```
cd /d <path_to_unzipped_folder>
```

Place the copied path instead of “<path_to_unzipped_folder>”, you can do it by simply right click the console to paste the path copied earlier.

For example:

```
cd /d D:\Users\Downloads\DeepLabCut-2.2.0.5
```

5. Enter the “examples” folder, by running:

```
cd examples
```

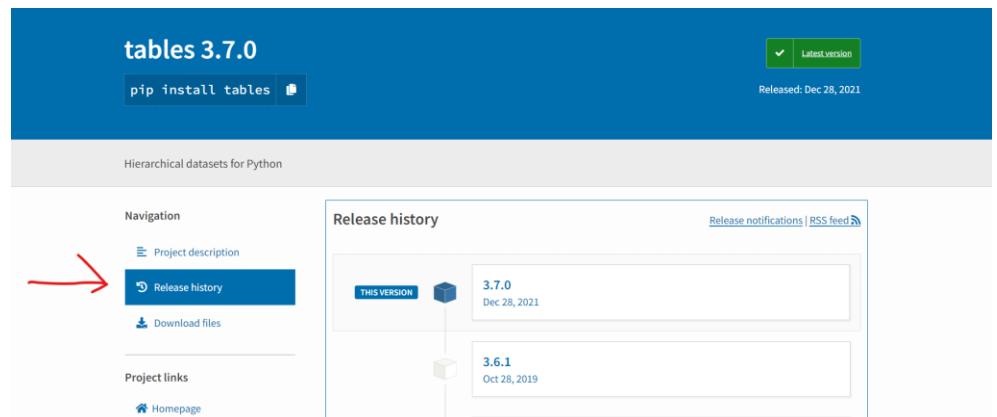
6. **In case of GPU installation:**
 - a. Open task manager, expend it if necessary.
 - b. Go to performance tab.
 - c. On your left you can see the devices – search for the GPU that was set before the installation.
 - d. During the next codes lines, you should see the dedicated memory usage rise & drop – this means you GPU is engaged correctly. If it does not happen, GPU isn't engaged – see [Known troubleshooting](#): below for possible solutions.
7. Run:
python testscript.py
and wait until prompt return.
8. If you see:
ALL DONE!!! - default cases are functional
then your installation is OK. Delete the created folder, its name begins with "TEST-ALEX".
9. Run:
python testscript_multianimal.py
and wait until prompt return.
10. If you see:
ALL DONE!!! - default cases are functional
then your installation is OK. Delete the created folder, its name begins with "multi_mouse- dlc_team".

[Known troubleshooting:](#)

- **Always** make sure you delete the folders created during the test runs, before rerunning tests again. Do it manually, or by typing in cmd:
rm -r TEST*
rm -r multi_mouse*
- If you ever stuck in a never-ending loop, you can abort using ctrl+C. No harm done.
- If the testscripts raise some error & do not finish running:

1. Your installation may still usually work – it may have a problem with some specific case.
2. Search your issue online. Good sources are DeepLabCut GitHub issues:
<https://github.com/DeepLabCut/DeepLabCut/issues?q=is%3Aissue>
or in DeepLabCut forum:
<https://forum.image.sc/tag/deeplabcut>
3. Try to restart the computer, to finalize any changes done, and rerun the testscripts.
4. Try to solve it like a dependency issue:
 - a. Try to see if the error has some name of a python package. Usually, you can find the problematic package in the error message such as:
**ImportError: Missing optional dependency 'tables'.
Use pip or conda to install tables.**
Point to error in the dependency “tables”.
 - b. If error did not include any package name, you can see what package raised the error in the error lines that start with “File”, in the file path, after “site-packages”. For example:
File ~\anaconda3\envs\DEEPLABCUT\lib\site-packages\pandas\compat_optional.py:118
Point to error in the python package “pandas”.
 - c. If you notice a package other then DeepLabCut that raised the error, you can try to update /downgrade this package by:
 - I. Look for the package versions online, usually it is easiest by searching the package (search project) in <https://pypi.org/> - after

entering the package click “Release history” on your left.



II. Enter the DeepLabCut conda environment (like

III. To see what package version is installed, run:

```
conda list <Package>
```

substitute “<Package>” with the package name you think cause the problems. For example:

```
conda list tables
```

IV. If the package installed is in its newest version, try downgrading by 1 version. The other way is also true – if the package already installed have a newer version, try upgrading it.

To change the package version (both downscaling & upscaling) run:

```
pip install <Package>==<VersionNum>
```

substitute “<Package>” with the package name you think cause the problems, and “<VersionNum>” with the wanted version.

For example:

```
pip install tables==3.6.1
```

- If GPU is not engaged correctly, try (in this order, after every step run the testscripts again to see if the GPU was engaged - any single option may solve the problem):

1. Restart your computer to make sure all the changes were finished.
2. Remove your DeepLabCut environment by running:

```
conda env remove --name DLC
```

Verify removal was successful by:

conda env list

Then reinstall DeepLabCut again, starting at stage [11](#).

3. If this didn't work, remove your DeepLabCut environment as above, uninstall Anaconda, and redo [Install DeepLabCut](#).
 4. If this didn't work, you will have to start from scratch. Remove your DeepLabCut environment as above, uninstall Anaconda, CUDA, GPU driver & Visual studio using windows "Add / Remove program", and redo from [Prepare GPU](#).
- See <https://github.com/DeepLabCut/DeepLabCut/wiki/Troubleshooting-Tips> & <https://deeplabcut.github.io/DeepLabCut/docs/recipes/installTips.html#installation-tips>

Usage

Basics

Entering DeepLabCut Environment

1. You must activate its environment before you are using DeepLabCut. To activate the DLC environment, run in the conda prompt (which always opened as administrator, see [Install DeepLabCut](#) step 2):
conda activate DLC
2. When finished with DeepLabCut, simply close the prompt.

See <https://docs.conda.io/en/latest/> for more information about conda environment, and this nice cheat sheet for different conda commands:

https://docs.conda.io/projects/conda/en/4.6.0/_downloads/52a95608c49671267e40c689e0bc00ca/conda-cheatsheet.pdf.

iPython

1. Almost all the code needs to be run in iPython – simply, after entering the DeepLabCut environment, run:
ipython
2. The prompt will change to green. Now import DeepLabCut:
import deeplabcut
Now you have access to all of DeepLabCut functions. You can see them by writing (notice the dot):
deeplabcut.
and press tab. For any function you can add “?” after it to see docstring, for example:
deeplabcut.create_new_project?
3. To exist iPython, just run:
Exit()
or close the prompt.

For more info on iPython, see <https://ipython.org/>.

Analyzing Videos

Note 1: Any paths used should include file suffix (i.e “.yaml” or “.mp4”).

Note 2: In windows, any path used inside iPython should be given as a raw string:
<https://www.journaldev.com/23598/python-raw-string>. Simply place an ‘r’ before the path, and enclose the path in quotes (“ ” or ‘ ’).

Note 3: Everything needs to be run inside DeepLabCut environment and, unless explicitly said otherwise, in iPython. See [Basics](#) for more info.

Analyzing videos is made of the following steps:

1. Make sure your video metadata is not corrupted.
2. Make your video as small as possible, by cropping & down-sampling.
3. Prepare a network according to the number of animals in you video (help with some bugs).
4. Analyze the video.
5. (optional) move your data into MATLAB.

[Check your Video \(\$\sim \frac{1}{8}\$ of the video time\)](#)

It is recommended to verify your video integrity if it was moved between computers (by cloud or USB). If video is corrupted, DeepLabCut will perform analysis (which is usually the slowest part), however it won’t detect anything.

To check your video integrity:

1. Run in iPython:

```
from deeplabcut.utils.auxfun_videos import VideoReader  
vid = VideoReader(<path_to_your_video>)  
vid.check_integrity()
```

Instead of “<path_to_your_video>” insert the path to your video. For example:

```
vid = VideoReader(r"G:\Data\Videos\My_Vid.mp4")
```

For more info on video corruption errors, See here:

<https://github.com/DeepLabCut/DeepLabCut/issues/982#issuecomment-723480607>

2. If no additional text appears in console – all is good. Else:

a. Exit iPython.

b. Copy the path to the folder holding your video.

c. Run:

```
cd /d <path_to_the_folder_holding_your_video>  
ffmpeg -i <video_name> -c:v h264 -crf 18 -preset fast  
<fixed_video_name>
```

Instead of <path_to_the_folder_holding_your_video> insert the path you copied.

Instead <video_name> write the name of the video you want to fix, and instead of <fixed_video_name> write the name you wish the fixed video to have.

For example:

```
cd /d G:\Data\Videos  
ffmpeg -i My_Vid.mp4 -c:v h264 -crf 18 -preset fast  
My_Vid_fixed.mp4
```

d. Work only with the fixed video from now.

[Make Your Video Smaller \(\$\sim \frac{1}{8}\$ of the video time for each function\)](#)

The smaller the video, the faster the analysis. Therefore, you should crop and/or down-sample your video if possible. Usually, will remain good (see also Mathis & Warren 2018, On the inference speed and video-compression robustness of DeepLabCut, <https://doi.org/10.1101/457242>)

It is not recommended to perform this step in online servers (such as Google Colab), as they tend to have weak CPU, and this step will run slowly. Usually, any local computer will be faster.

To Crop the video:

1. Run:

```
deeplabcut.auxfun_videos.CropVideo(<Path_to_video>,
```

useGUI=True)

Substitute "<Path_to_video>" with your video path.

For example:

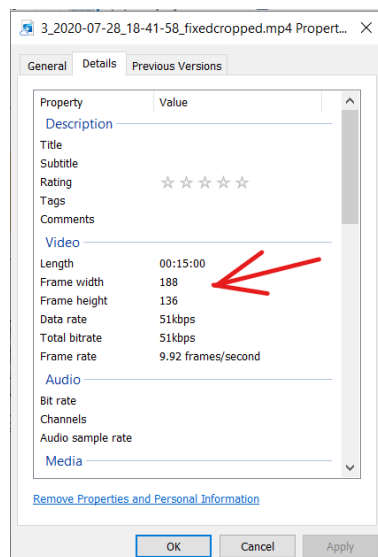
```
deeplabcut.auxfun_videos.CropVideo(  
r"G:\Data\Videos\My_Vid.mp4", useGUI=True)
```

2. Select in the GUI only the area the mice can be in
3. The cropped video will be saved in the same folder as the original. Use this video from now on.

Issue: Sometimes the cropping GUI isn't working at first use. Simply close it using X and try the command again – usually now it will work.

To Down-sample the video:

1. If you want to also crop the video, do it before down-sampling.
2. Find the video proportions:
 - a. Right click the video.
 - b. Choose "properties".
 - c. Select "Details" tab. Write the video width & height, under the "Video" section.



3. Choose a new height for your video: Both height & width will be divided by the same factor, to reach this new height.

Usually you should down-sample as much as you can – choose a small height. However,

usually you shouldn't go below about 62.5% of the original video size (before cropping), and if video is of poor quality (blurry, already very pixelated, dark) consider keeping frames bigger.

4. Run:

```
deeplabcut.auxfun_videos.DownSampleVideo(<Path_to_video>,  
height=<wanted height>)
```

Instead of "<path_to_your_video>" insert the path to your video.

Instead of "<wanted height>" write the wanted height you choose.

For example:








```
deeplabcut.auxfun_videos.DownSampleVideo(  
r"G:\Data\Videos\My_Vid.mp4", height=512)
```

If "My_Vid" was 1280 X 1024 (width X height), the down-sampled version will be 640 X 512.

The down-sampled video will be saved in the same folder as the original. [Use this video from now on.](#)

Prepare network for analysing

Network is packed in a folder, with a lot of data regarding its training & abilities. The folder will look like that:

Name	Date modified	Type	Size
 analyse_res	11/2/2022 09:12	File folder	
 dlc-models	9/2/2022 07:39	File folder	
 evaluation-results	11/2/2022 08:13	File folder	
 labeled-data	9/2/2022 07:39	File folder	
 training-datasets	9/2/2022 07:39	File folder	
 videos	9/2/2022 07:39	File folder	
 config.yaml	13/4/2022 16:50	YAML File	5 KB

It will always include:

- config.yaml file.
- dlc-models folder
- evaluation-results folder
- training-datasets folder

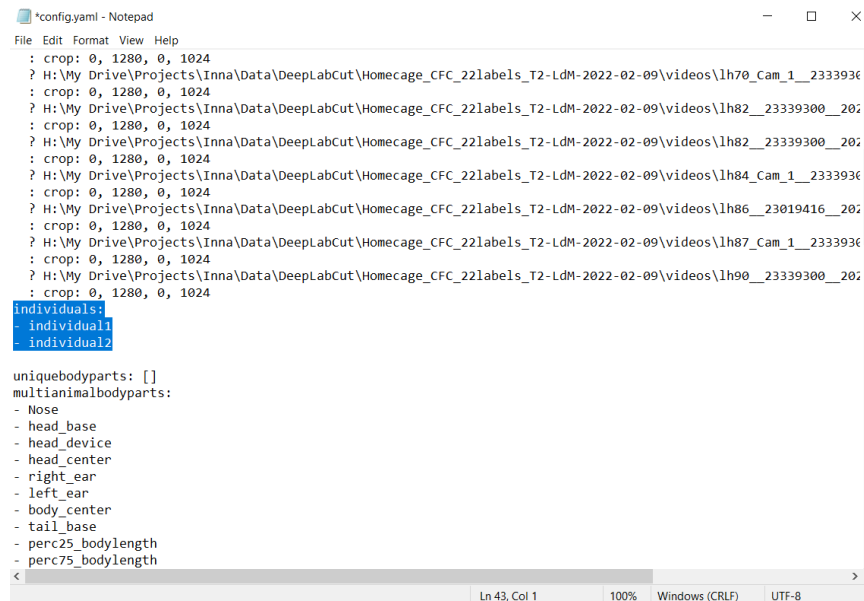
more files and/or folders may be included.

Note: in the future, when we will have multiple networks, a file named Net_info.txt will be added, with information about network designation & how to use it.

Changes you will need to do, for the network to work properly with your number of animals:

1. Open the file named “config.yaml” – any text editor (for example Notepad) will do.
2. Under “individuals:”, make sure there is an individual for each mouse in your video.

Delete any unnecessary individuals. The beginning “- ” (notice the space after) must be before every individual marked. **Do not change anything else in the file.**

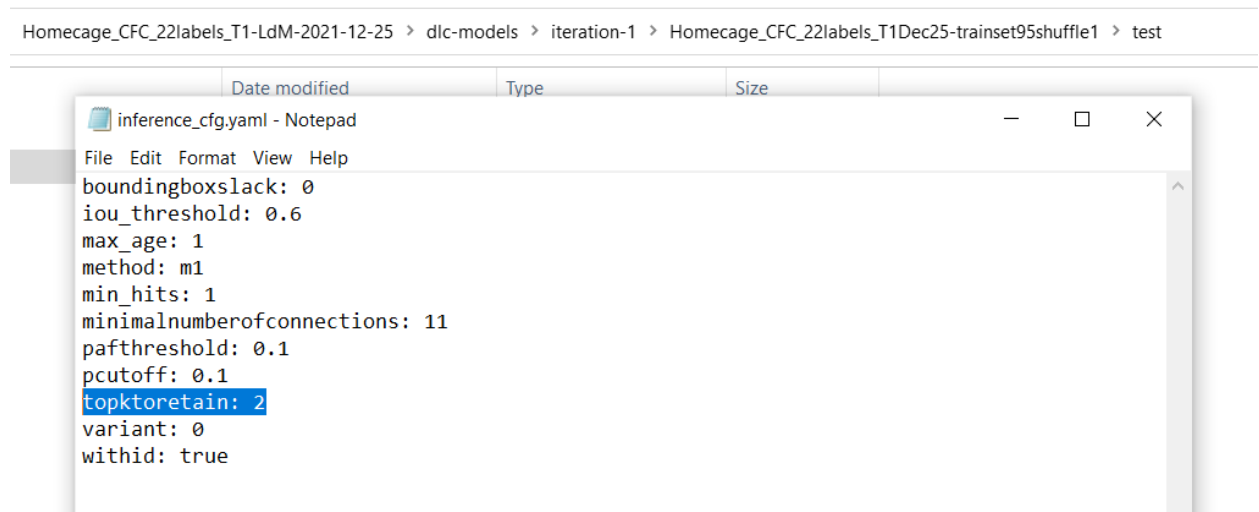


```
*config.yaml - Notepad
File Edit Format View Help
: crop: 0, 1280, 0, 1024
? H:\My Drive\Projects\Inna\Data\DeepLabCut\Home cage_CFC_22labels_T2-LdM-2022-02-09\videos\lh70_Cam_1__2333936
: crop: 0, 1280, 0, 1024
? H:\My Drive\Projects\Inna\Data\DeepLabCut\Home cage_CFC_22labels_T2-LdM-2022-02-09\videos\lh82__23339300__20
: crop: 0, 1280, 0, 1024
? H:\My Drive\Projects\Inna\Data\DeepLabCut\Home cage_CFC_22labels_T2-LdM-2022-02-09\videos\lh82__23339300__20
: crop: 0, 1280, 0, 1024
? H:\My Drive\Projects\Inna\Data\DeepLabCut\Home cage_CFC_22labels_T2-LdM-2022-02-09\videos\lh84_Cam_1__2333936
: crop: 0, 1280, 0, 1024
? H:\My Drive\Projects\Inna\Data\DeepLabCut\Home cage_CFC_22labels_T2-LdM-2022-02-09\videos\lh86__23019416__20
: crop: 0, 1280, 0, 1024
? H:\My Drive\Projects\Inna\Data\DeepLabCut\Home cage_CFC_22labels_T2-LdM-2022-02-09\videos\lh87_Cam_1__2333936
: crop: 0, 1280, 0, 1024
? H:\My Drive\Projects\Inna\Data\DeepLabCut\Home cage_CFC_22labels_T2-LdM-2022-02-09\videos\lh90__23339300__20
: crop: 0, 1280, 0, 1024
individuals:
- individual1
- individual2

uniquebodyparts: []
multianimalbodyparts:
- Nose
- head_base
- head_device
- head_center
- right_ear
- left_ear
- body_center
- tail_base
- perc25_bodylength
- perc75_bodylength
```

3. Save the path to the config file – this will be the network identifier.
4. Change the inference_cfg.yaml file, under:
dlc-models > iteration-1 > folder that ends with “shuffle1” > test > inference_cfg.yaml

Change the number after “topktoretain” to match the number of animals in your video.



Video Analysis (dependent on video size, and system used, usually long)

This step requires a GPU.

1. Run:

```
deeplabcut.analyze_videos(<config_file>,[<video_path>],
use_shelve=True, auto_track=True, destfolder = <output_location>,
identity_only=True, n_tracks=<nMice>)
```

replace:

- “<config_file>” - with the path to the config file that you copied (see [Prepare network](#) step 3).
- “<video_path>” – with the path to the video/s you want to analyze (note that crop, down-sampling, and integrity checking do not support multi-video at one run, only this function). Do not forget the square brackets, even for one video – this function accepts videos path only as list!
- “<output_location>” – with the path to where you desire the output to be.
- “<nMice>” – number of mice to detect in your video.

For example:

```
deeplabcut.analyze_videos(r"D:\Code\DLC_nets\HomeCage_CFC_22label1
```

```
s_T1-LdM-2021-12-25\config.yaml",[  
r"G:\Data\Videos\My_Vid_fixedcroppeddownsampled.mp4"],  
use_shelve=True, auto_track=True, destfolder = r"G:\Data\Videos",  
identity_only=True, n_tracks=1)
```

This will analyze the video "My_Vid_fixedcroppeddownsampled" (which is "My_vid" after it was fixed, cropped & down-sampled), that have only 1 animal, and will put the result (an h5 file) in "G:\Data\Videos".

2. Convert the h5 file to csv file (which is easier to work with):

- a. If the video you analyze exist in the same folder as the analyze result (h5 file), continue to the next stage. If not, create a simple empty text file in the folder with the results, with the same name as the video (without the suffix) and the suffix "txt". For example: "My_Vid.txt".

- b. Run:

```
deeplabcut.analyze_videos_convert_h5_to_csv(  
<output_location>, videotype = <Video_Suffix>)
```

Substitute "<output_location>" with the same output location used during analysis.

Substitute "<Video_Suffix>" with the suffix of your video if it is in the same folder as "<output_location>", or with ".txt" if not.

For example, if "My_Vid" is in "<output_location>", use:

```
deeplabcut.analyze_videos_convert_h5_to_csv(  
r"G:\Data\Videos", videotype = ".mp4")
```

Else:

```
deeplabcut.analyze_videos_convert_h5_to_csv(  
r"G:\Data\Videos", videotype = ".txt")
```

The CSV will be in "<output_location>".

3. (optional) you can create a video with all detections using:

```
deeplabcut.create_video_with_all_detections(<config_file>,[<video  
_path>], destfolder = <output_location>)
```

All parameters here have to be same as when analyzed (see [above](#)). For example:

```
deeplabcut.create_video_with_all_detections(r"D:\Code\DLC_nets\Homepage_CFC_22labels_T1-LdM-2021-12-25\config.yaml",[
r"G:\Data\Videos\My_Vid_fixedcroppeddownsampled"], destfolder =
r"G:\Data\Videos")
```

The video will be in the output folder.

4. (optional) Import the data into MATLAB:

- a. Copy the path to the CSV file.
- b. All functions are in "[slutsky ECInVivo](#)".
- c. Run in MATLAB:

```
[labels_pos,body_parts] = read_label_file(<Path2CSV>);
```

instead of "<Path2CSV>" place the path of the csv you created in step 2 ([above](#))

In "**labels_pos**" you will find the location of each body part, in a 4d array:

Frame X Mice X body_part X (x,y,likelihood) – The last dimension is in the order specified, likelihood is how sure the network is the body part is there.

In "**body_parts**" you can find a list of body part the network can recognize, in the same order they appear in "**labels_pos**" – use it to filterer unwanted body parts.

For example:

```
[labels_pos,body_parts] = read_label_file("G:\Data\Videos\
My_Vid_fixedcroppeddownsampledDLC_resnet152_Homepage_CFC_22l
abels_T1Dec25shuffle1_40000_e1.csv");
```

- d. Mass Center is a simple mean without any of the "tail" body parts.

```
[mass_loc] =
points2loc(labels_file,nMice,exclude_parts,p_cutoff)
```

Known troubleshooting:

- If you get errors: "No tracklets were found" or "unable to track X mice" when running analysis, or results are bad in the CSV file:
Analysis may have succeeded but "**auto_track**" have failed. You can try performing the tracking without analyzing again (which is a lot faster), by following steps:

5. Run:

```
deeplabcut.convert_detections2tracklets(<config_file>,[<video_path>], destfolder = <output_location>,  
identity_only=<True/False>, overwrite = True)
```

except “**identity_only**”, all parameters should be the same as used when analyzed (see [above](#)). For example:

```
deeplabcut.convert_detections2tracklets  
(r"D:\Code\DLC_nets\Home cage_CFC_22labels_T1-LdM-2021-12-25\config.yaml",[  
r"G:\Data\Videos\My_Vid_fixedcroppeddownsampled"],  
destfolder = r"G:\Data\Videos", identity_only=True,  
overwrite = True)
```

Try switching “**identity_only**” and see if analyze succeed / or gives better results after running all steps.

6. Run:

```
deeplabcut.stitch_tracklets (<config_file>, [<video_path>],  
destfolder = <output_location>, n_tracks = <number/None>)
```

except “**n_tracks**”, all parameters should be the same as used when analyzed (see [above](#)).

Try changing “**n_tracks**” to a different number then the number of mice in you video (while bigger may find unreal mice, lower may skip existing mice). You can also try changing it to “None”, so it should find the number of mice automatically.

For example:

```
deeplabcut.stitch_tracklets  
r"D:\Code\DLC_nets\Home cage_CFC_22labels_T1-LdM-2021-12-25\config.yaml",[  
r"G:\Data\Videos\My_Vid_fixedcroppeddownsampled"],  
destfolder = r"G:\Data\Videos", n_tracks=2)
```