Title: Getting started with DeepLabCut for markerless tracking – An easy installation guide

Subtitle: Why you should start using DeepLabcut in your data processing.

## What is it all about, and is DeepLabCut for me?

Probably yes. If you happen to work with videos and you want to extract information out of them, DeepLabCut is for you. If you need to track the position of a certain object on every single frame, you would benefit from using DLC. If, say, you want to track 5, 10 or 100 different points and their relative position (i.e., pose), you should be using DeepLabCut. And if it turns out these points are repeated across multiple objects or individuals, you guessed it, DLC! If you don’t happen to work with videos… you should probably start doing so.

A short remark first: This guide is about DeepLabCutTM [http://www.mackenziemathislab.org/deeplabcut] and how to install it, this is no review about the best tools available. There are several other promising open source projects for markerless tracking out there, like OpenPose, EthoVision and optiflex just to name a few. I just happened to start using DeepLabCut when other software was not yet as easily available. For a review on different approaches see Desmarais et al 2020 [https://arxiv.org/abs/2010.06449] and Mathis et al 2020 [https://www.sciencedirect.com/science/article/abs/pii/S0896627320307170].

But now back to the topic. In case you haven’t heard of it, DeepLabCut is an open source package for markerless pose estimation based on transfer learning with deep neural networks (Mathis et al., 2020). It uses algorithms from DeeperCut 26, 27, and an extremely deep neural network pretrained on a dataset for object detection (ImageNet) 24. DeepLabCut was originally developed for animal pose estimation, which does not exclude humans (Namba et al 2021 [https://www.nature.com/articles/s41598-021-83077-4]), and can also be used to track inanimate objects, as we will see in our first DLC project.

# How to get started

First of all, you will need to use python and several python related libraries (i.e., a specific python environment). Or rather, DeepLabCut will need it, don’t panic! If you have no programming experience whatsoever and not the slightest interest in learning to code, this crash course is still relevant for you. Nevertheless, I will assume either interest or minimal experience in any programming language. This post series will provide you with code snippets, jupyter notebooks and plenty of useful references to work through the python code step by step. If you are still not interested, you will learn to use your mouse and a graphic user interface with pretty buttons.

## Installation

This is arguably the hardest part, but let’s start easy.

First, you will have to install Anaconda, a free python distribution and package manager that will help you keep all your libraries up to date. The process varies slightly depending on your machine (Windows, macOS or Linux), but overall, you will have to click here [https://docs.anaconda.com/anaconda/install/], choose your system, download the installer and click through all Next, Agree and Install buttons. Once finished, you can use Anaconda either through the Navigator (a graphic user interface) or the Terminal (the scary black pop-up window).

Next, you will install DeepLabCut, and for that you will have to choose between the full GPU and the simpler CPU version (see below). On the DeepLabCut homepage [http://www.mackenziemathislab.org/deeplabcut], scroll to the bottom of the page to see the respective download files. For convenience you can find the same download links here for the CPU [http://www.mackenziemathislab.org/s/DLC-CPU.yaml] or the GPU [http://www.mackenziemathislab.org/s/DLC-GPU.yaml] version).

Because training deep neural networks is quite computationally intensive, most machine learning approaches make use of additional hardware (graphic processing units or GPUs) originally used in gaming to perform the needed computations. That means that most standard computers and laptops won’t have the processing power required, without upgrading the GPU first. But luckily, the CPU version of DeepLabCut will allow you to perform most of the pre- and post-processing on your own computer, while outsourcing the actual model training to a cloud computing environment like Google Colab (more on that later).

In case you already have a GPU or the chance to upgrade your computer you will first have to install the Nvidia drivers from here [https://www.nvidia.com/download/index.aspx] and CUDA 10 from here [https://developer.nvidia.com/cuda-10.0-download-archive].

Once you are ready open the terminal, locate your DeepLabCut installation file (e.g., in Downloads directory), and create a new virtual environment for the specific file with the following command:

ls Downloads

cd Downloads

conda env create -f DLC-CPU.yaml

conda env list

conda activate DLC-CPU

video 1 <https://youtu.be/yarI_m9F7VU>

Next, we are going to upgrade DeepLabCut to the latest beta release, as of today 2.2b8, but check for new releases here [https://github.com/DeepLabCut/DeepLabCut#versions]. As downgrade wxpython < 4.1, as some issues occure in macOS check issues here []

conda activate DLC-CPU

pip install deeplabcut==2.2b8

pip install -U wxPython==4.0.7

python -m deeplabcut

or pythonw -m deelpabcut

video 2 <https://youtu.be/8sfb0Qi2Hd0>

Windows vs macOS:

Now that DeepLabCut is installed in its own virtual environment we can start having fun with it. And you can choose how to get started! DeepLabCut can be used from the GUI by mouse clicks, from terminal with python functions or from jupyter notebooks that can easily be moved to cloud computing servers like google colab [https://github.com/DeepLabCut/DeepLabCut/blob/master/docs/UseOverviewGuide.md#the-options-for-using-deeplabcut]. Because we are working from a CPU environment and won’t be able to train our models locally, we will use a combination of jupyter notebooks and DLC GUI.

To get started, open the terminal (cmd in Windows) activate your DLC-CPU environment, and start deeplabcut:

conda activate DLC-CPU

pythonw -m deeplabcut

in Windows

conda activate DLC-CPU

python -m deeplabcut

Alternatively, you could start PythonW (or IPython in Windows) as we will be doing in the notebooks and run:

import deeplabcut

deeplabcut.launch\_dlc()

This will open the GUI and you can start a new project in the ‘Manage Project’ tab.

Graphical user interface, text, application

Description automatically generated

To start your first DeepLabCut project check out my post on “Starting your first DLC Project”. lf you feel comfortable trying by yourself, check out these resources [https://github.com/DeepLabCut/DeepLabCut/blob/master/docs/UseOverviewGuide.md#create-a-new-project] from the DeepLabCut Github page, or here [https://www.youtube.com/channel/UC2HEbWpC\_1v6i9RnDMy-dfA] youtube tutorials.