Using DeepLabCut for 3D Pose Estimation

1. You need to have a 2D DeepLabCut network that is already analyzed.
2. Open Anaconda navigator (right click >> **run as administrator**)
3. Open environments >> dlc-windowsGpu >> open Terminal (green arrow)
4. Every time you want to use DeepLabCut in the Anaconda:

Type: python

import deeplabcut

1. If you are **starting a new project**, follow these steps. Otherwise skip to step 6.
   1. Create a New 3D Project

Type:

config\_path3d=deeplabcut.create\_new\_project\_3d('**ProjectName**','**NameofLabeler**',num\_cameras=2,working\_directory=**(r"’ProjectDirectory’")**)

This function will create a project directory with the name ProjectName+NameofLabeler+date of creation of the project+3d in the Working directory.

* 1. Process of **Camera Calibration**:

### Take a short video while you move the checkerboard around in different angles and distances. Convert the video to .jpg frames.

### Place your images into the calibration\_images folder in the project directory.

### The images should be named with the camera-# as the prefix, i.e. camera-1-01.jpg and camera-2-01.jpg for the first pair of images. (Please note, this cannot be changed after the project is created)

### **Detect bad pairs** Type: deeplabcut.calibrate\_cameras(config\_path3d, cbrow=8, cbcol=6, calibrate=False, alpha=0.9) Delete from the folder any bad pairs where the corners and their directions are not correctly detected.

### Type: deeplabcut.calibrate\_cameras(config\_path3d, cbrow=8, cbcol=6, calibrate=True, alpha=0.9) The above function brings both the camera image planes to the same plane by computing the stereo rectification.

### Check for Undistortion: In order to check how well the stereo is calibrated, it is recommended to undistort the calibration images and the corner points by typing: deeplabcut.check\_undistortion(config\_path3d, cbrow=8, cbcol=6)

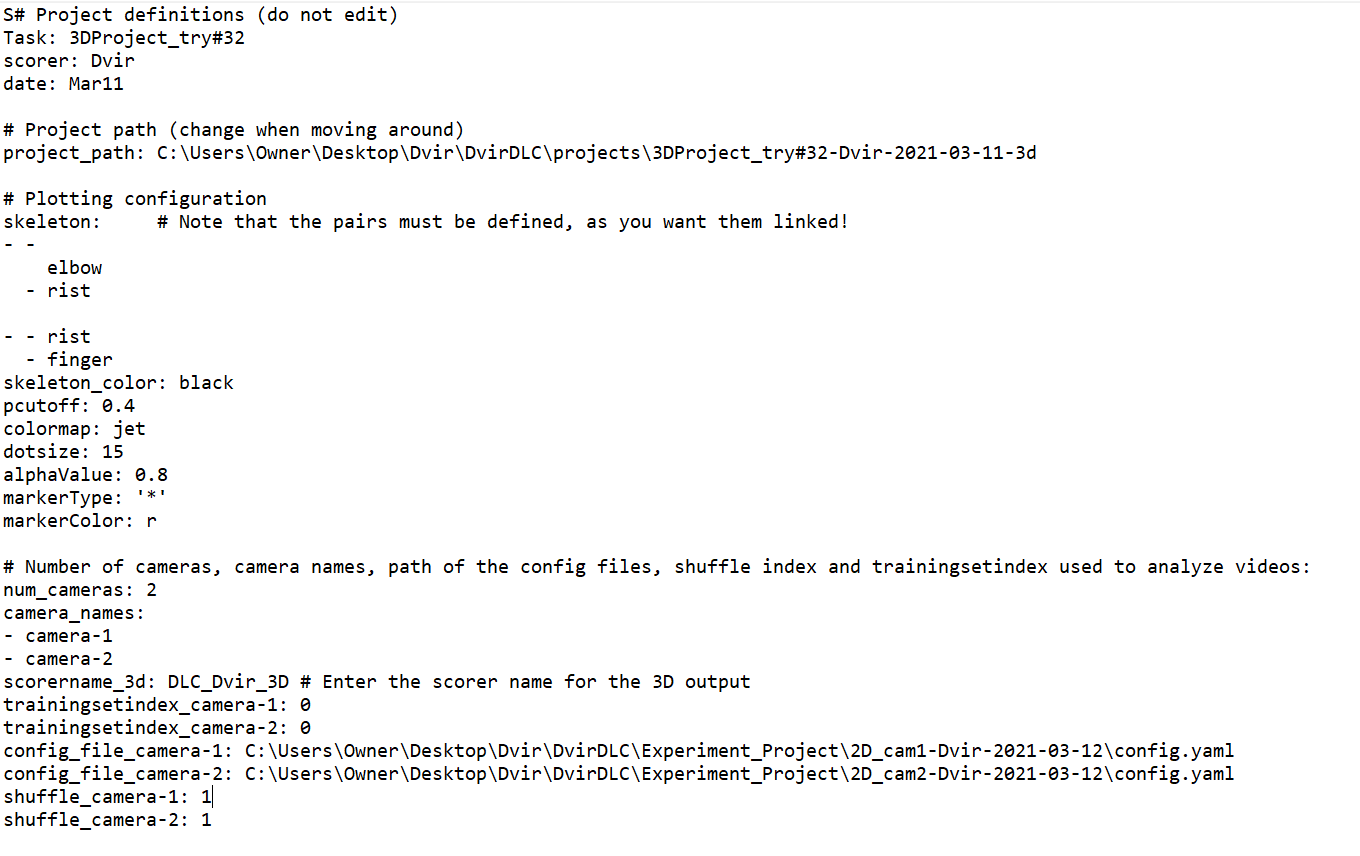
1. Re-opening the project (from a previously created project):   
   **Type**: config\_path3d=r‘enter full path of config.yaml file

Example-

config\_path3d=r"C:\Users\Owner\Desktop\Dvir\DvirDLC\projects\3DProject\_try#32-Dvir-2021-03-11-3d\config.yaml"

1. **Analyze** --> Take your 2D to 3D
   1. Open a new folder for analyze and choose video files you want to analyze.
   2. Name the video files you want to analyze in such a way that the file name contains the name of the cameras as specified in the config file. e.g. if the cameras are named camera-1 and camera-2 (or cam-1, cam-2 etc.) then the video filename must contain this name, i.e. they could be named  
      nana-1-monkey-day1-**camera-1**-date.avi and nana-1-monkey-day1-**camera-2**-date.avi.
   3. Edit the 3D project config.yaml file to denote which DeepLabCut projects have the information for the 2D views.
   4. Input the same body part names as in the config.yaml file of the 2D project.
   5. Type(change the file location of the analyze video):

deeplabcut.triangulate(config\_path3d, r"F:\nana\_060421\Analyze", filterpredictions=True,save\_as\_csv=True)



* 1. The triangulation can be done in DeepLabCut by typing:  
     deeplabcut.triangulate(config\_path3d, r"F:\nana\_060421\Analyze", filterpredictions=True,save\_as\_csv=True)

1. Visualize your 3D DeepLabCut Videos:

In order to visualize both the 2D videos with tracked points plot the pose in 3D, the user can create a 3D video for certain frames (these are large files, so I advise just looking at a subset of frames).

Type: deeplabcut.create\_labeled\_video\_3d(config\_path3d, [r"F:\nana\_060421\Analyze"], start=50, end=250)

-For more information about any of the functions, visit:

<https://github.com/DeepLabCut/DeepLabCut/blob/master/docs/Overviewof3D.md>