Deep Lab Cut

INSTALL

Install Deep Lab Cut 2. 0 :

Preferably on an Ubuntu system (the plan is to train the model, get the weights, and then transfer the weights to a Docker which is based on an Ubuntu OS. It's easier to transfer the weights if both are done on Ubuntu systems)

COMPLETE GUIDE

Refer: https://github.com/AlexEMG/DeepLabCut/blob/master/docs/functionDetails.md
Example notebooks: https://github.com/AlexEMG/DeepLabCut/tree/master/examples

TRAIN

- Train the model following instructions in the notebook Deeplabcut_new.ipynb in the git repo (https://github.com/ambareeshsrja16/Surgical_Tool_Tracking/blob/master/Deeplabcut_new.ipynb)

- Project config file is stored in the *path_config_file* variable - after you make the project, print this variable and store it, so that you can reuse it again if need be.

- For changing config.yaml file according to specific needs refer to https://github.com/AlexEMG/DeepLabCut/blob/master/docs/functionDetails.md (Read BOX 1)

    # Training,Evaluation and Analysis configuration
    TrainingFraction :- 0.95 # how much of the labelled data needs to be used for training
    iteration: 5 #New weights will be saved with this number in the name

- Before you run this command - *deeplabcut.train_network(path_config_file)* - make sure to change some parameters in *pose_cfg.yaml* at the following path:

C:\....\DaVinci-Ambar-2020-02-14\dlc-models\iteration-2\DaVinciFeb14-trainset40shuffle1\train

Change these two: all_joints, all_joints_names; according to the robot.

all_joints:
- - 0

- - 1
- - 2
- - 3
- - 4
- - 5
- - 6
- - 7
- - 8
- - 9
- - 10
- - 11

all_joints_names:
- PSM1-roll_2
- PSM1-pitch_1
- PSM1-pitch_2
- PSM1-pitch_3
- PSM1-yaw_1
- PSM1-yaw_2
- PSM2-roll_2
- PSM2-pitch_1
- PSM2-pitch_2
- PSM2-pitch_3
- PSM2-yaw_1
- PSM2-yaw_2

(The numbers show the order in which you'll label these parts later)

Change these two parameters to monitor the training:

display_iters: 100 (so that it prints loss values after every 100 iterations, and you can track the progress)
save_iters: 500 (so that the model can be saved after every 500 iterations or so)

- For CUDA errors that prop up, use the following:

  import tensorflow as tf
  gpu_options = tf.GPUOptions(per_process_gpu_memory_fraction=0.6)
  sess = tf.Session(config=tf.ConfigProto(gpu_options=gpu_options))

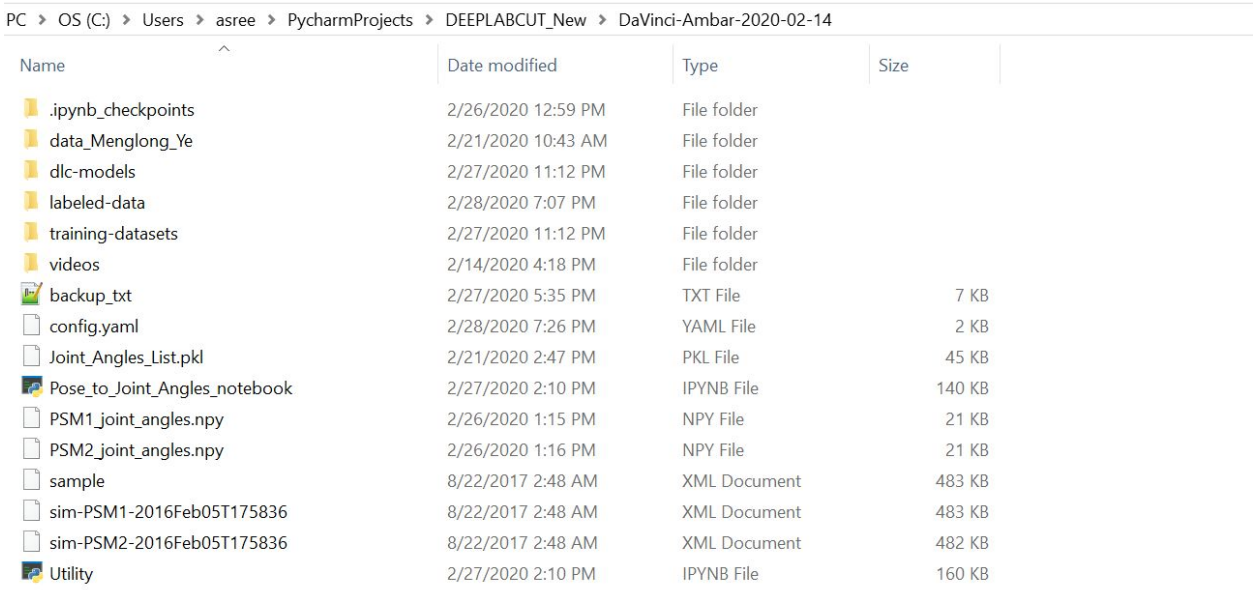
  The above prevents TensorFlow from using up the entire GPU Memory

EVALUATE

Models can be evaluated using DLC's inbuilt evaluation functions. Some additional graphs created for the SuperDeep paper were made using functions in the Deeplabcut_new.ipynb notebook

DOCKER

After training the model, you should have some file structure like this:



The project folder has to be copied to Docker for running the prediction, after the Docker container is up and running!

(The folder dlc-models has all the weights, numbered according to the iteration (you can change this in the *config.yaml* file when you are training). But we copy the entire project folder to make the other steps below easier. You can obviously change this when you understand how the whole thing works)

Clone the git repo : https://github.com/ambareeshsrja16/Surgical_Tool_Tracking
Go to (cd to) the folder containing Dockerfile, and run the following commands

1. *docker image build -t dlc_username/dlcdocker .*
   # builds the docker container from image
2. *docker run -it --gpus "device=2" dlc_username/dlcdocker bash*
   # runs the docker container
   #If you want to use all GPUs use:
   docker run -it --gpus 'all' dlc_username/dlcdocker bash

Copy the project folder to Docker using:

docker cp test-jingpei-2020-01-09 elegant_albattani:root/DLCROS_ws/src/Surgical_Tool_Tracking/ForwardPassDeepLabCut

Note:
test-jingpei-2020-01-09 - name of the project folder (in the picture above the name is *DaVinci-Ambar-2020-02-14*)
elegant_albattani - name of your docker container

Run these commands (they are hacks to prevent some errors - either in DLC, or the ROS2-Python3 binding - hopefully in the future these will clear up. For more details refer the last parts of this document, those are not formatted though)

export DLClight=True
rm -rf /opt/ros/kinetic/lib/python2.7/dist-packages/cv2.so

For running the evaluation/ prediction/ forward pass code
cd ~/DLCROS_ws/Surgical_Tool_Tracking/ForwardPassDeepLabCut
python3 generator_clipped_image…

COMMON ERRORS

- "FileNotFoundError: It seems the model for shuffle s and train ..."
  Check the config.yaml file that you copied inside the Docker, and check the **project_path** in this .yaml file, it could be incorrect

- "No module name numpy.testing.decorators"
  Uninstall numpy version 1.18.0 and install
  pip uninstall numpy
  pip install numpy==1.17.0
  https://blog.csdn.net/AiBigData/article/details/103738505

DLC : MISCELLANEOUS

The following is a rough clipboard with information that was used during the project

Test Docker

```
sudo docker run hello-world
```

Test NVIDIA, Jacob's command

sudo docker run --gpus all nvidia/cuda nvidia-smi

Added my username to the docker group:

```
sudo usermod -aG docker ambareesh
```

Usually, you have to restart, to make sure the changes are reflected, but check on a new terminal if you are able to run the hello-world docker test without sudo, if yes, you are good

Final

Ran 1, 2

1)
docker run --gpus all -t -d -p 2351:8888 -e USER_HOME=$HOME/Docker4DeepLabCut2.0 dlc_username/dlcdocker bash

-d = detach, run container in bg, and print id
-p = attach the port
-e = set environment variables
-t = allocate a psuedo tty (Exits without this for some reason)
dlc_username/dlcdocker = image
bash -> add at the end to keep the container running!

Serverhost:2351 linked to container:8888

2)

docker exec -it sweet_burnell /bin/bash

-i= interactive

clever_rosalind= random name, (if you use --user name when you start the container you can choose your own)

docker exec --user ambareesh -it sweet_burnell /bin/bash
unable to find user ambareesh: no matching entries in passwd file?

3) `cd ../../../home/`
4) `export DLClight=True`
5)`ipython -> import deeplabcut`

6) For Jupyter notebook

Connect your localhost:2351 to the serverhost:2351

ssh -L localhost:2351:localhost:2351 ambareesh@unicorn2.ucsd.edu

Run jupyter-notebook --allow-root from inside container
You'll get a token (but with 8888, this is what the container is generating, this is routed to 2351 of server host and in turn to your 2351, so copy-paste the token to your browser and replace 8888 with 2351)
Other

docker ps (show only running ones) -a (show everything)
docker container prune
docker stop <container_name>

Accessing the Server

ambareesh@unicorn2.ucsd.edu
 perception

Image building

```
docker image build -t dlc_username/dlcdocker .
```

Running the docker

- docker run -it dlc_username/dlcdocker
- docker run -it --gpus "device=2" dlc_username/dlcdocker bash

- docker run -it --gpus 'all' dlc_username/dlcdocker bash

Do first if you have added CMD[“bash”] in the docker file!, if not then do the second one

Copy stuff from docker to desktop
Docker cp <name>:~/path host_path

ROS

To show what is being published
Rosrun image_view image_view image:=/dlc_prediction_topic

To record a view from the bag
Rosrun image_view video_recorder _filename:=”/home/.._video.avi” image:=/(topic)

Copy files from ssh to my system
scp ambareesh@unicorn2.ucsd.edu:~/version_1_Docker/Docker4DeepLabCut2.0/Dockerfile .

scp ambareesh@unicorn2.ucsd.edu:~/version_1_Docker/Docker4DeepLabCut2.0/Dockerfile .
scp
Users/ambareeshsnjayakumari/Desktop/Surgical_Robot/Git/Surgical_Tool_Tracking/ForwardPassDeepLabCut/ ambareesh@unicorn2.ucsd.edu:~/version_1_Docker/Docker4DeepLabCut2.0/

Method 1

1. Remove last two lines for sourcing the ros stuff from the docker file, and then build it

Now run using gpus, export DLClight=True
Try python3, import deeplabcut (should work with np warnings), and then import cv_bridge (wouldn’t work),
Import sys
sys.path.append(“/opt/ros/kinetic/lib/python2.7/dist-packages”) now import cv_bridge and it should work. When opening new terminals to make the ROS stuff work make sure that you source the stuff correctly.

2. Add the last two lines in the dockerfile as such and then remove cv2.so from the /opt/ros/kinetic/lib/python2.7/dist-packages folder, this might work.

TO RUN

scp weights to host
docker cp
/home/jingpei/test-jingpei-2020-01-09/dlc-models/iteration-0/testJan9-trainset95shuffle1/train/snapshot-7100.meta elegant_albattani:root/weights/snapshot-7100.meta

docker image build -t dlc_username/dlcdocker .
docker run -it --gpus "device=2" dlc_username/dlcdocker bash

export DLClight=True
rm -rf /opt/ros/kinetic/lib/python2.7/dist-packages/cv2.so
cd ~/DLCROS_ws/Surgical_Tool_Tracking/ForwardPassDeepLabCut

python3 prediction_v1.py

cp prediction_v1.py prediction_v2.py
vim prediction_v2.py

"/root/DLCROS_ws/Surgical_Tool_Tracking/ForwardPassDeepLabCut/DaVinci-Ambar-2019-10-31/config.yaml"

vim /usr/local/lib/python3.6/dist-packages/deeplabcut/utils/auxiliaryfunctions.py
Use four spaces

python3 prediction_v2.py

vim
~/DLCROS_ws/Surgical_Tool_Tracking/ForwardPassDeepLabCut/DaVinci-Ambar-2019-10-31/config.yaml

project_path
/root/DLCROS_ws/Surgical_Tool_Tracking/ForwardPassDeepLabCut/DaVinci-Ambar-2019-10-31

video_sets
/root/DLCROS_ws/Surgical_Tool_Tracking/ForwardPassDeepLabCut/DaVinci-Ambar-2019-10-31/videos/video_1.avi

Snapshot index problem

~/DLCROS_ws/Surgical_Tool_Tracking/ForwardPassDeepLabCut/DaVinci-Ambar-2019-10-31/dlc-models/iteration-2/DaVinciOct31-trainset95shuffle1/train
Added this to dockerfile, scp ed the file to host, and copied it in.

Io.imread problem for PNG files, try the below thing
image = io.imread(imagename, plugin='matplotlib')

ssh ambareesh@unicorn2.ucsd.edu

DLC
Report error in
ruamel.yaml.scanner.ScannerError: mapping values are not allowed here
  in
"/root/DLCROS_ws/Surgical_Tool_Tracking/ForwardPassDeepLabCut/DaVinci-Ambar-2019-10-31/config.yaml", line 9, column 116

"/usr/local/lib/python3.6/dist-packages/deeplabcut/utils/auxiliaryfunctions.py", line 120, in read_config
    cfg = ruamelFile.load(f)

Need to write an else statement for other errors

MULTIPLE TERMINALS

You can run `docker exec -it <container> bash` from multiple terminals to launch several sessions connected to the same container.

No module name numpy.testing.decorators
Uninstall numpy version 1.18.0 and install
pip uninstall numpy

pip install numpy == 1.17.0

https://blog.csdn.net/AiBigData/article/details/103738505

Running on a fresh system

- Download/clone from git https://github.com/ambareeshsrja16/Surgical_Tool_Tracking

- Run
    - docker image build -t dlc_username/dlcdocker .
    - docker run -it dlc_username/dlcdocker

OR

docker run -it --gpus “device=2” dlc_username/dlcdocker bash

OR

docker run -it --gpus ‘all’ dlc_username/dlcdocker bash

- Will face this error: *Python 3.6 PPA removed from public access*
  Solve using: https://github.com/DeepLabCut/Docker4DeepLabCut2.0/issues/21
- If intel-openmp distribution error occurs do this:
  https://github.com/DeepLabCut/Docker4DeepLabCut2.0/issues/25

- Copy test folder (test-Jingpe….) to the docker inside ForwardPassDeepLabcut
- Try Python3 generator_clipped_image…
  - Import cv2 error: remove cv2.so from the /opt/ros/kinetic/lib/python2.7/dist-packages folder
  - No module named “wx”
    `export DLClight=True`
  - No module name numpy.testing.decorators
    Uninstall numpy version 1.18.0 and install
    pip uninstall numpy

    pip install numpy==1.17.0

    https://blog.csdn.net/AiBigData/article/details/103738505

  - `export DLClight=True`
  - `Copy test JingPei folder`

Copy Weights to Docker!

docker cp test-jingpei-2020-01-09 elegant_albattani:root/DLCROS_ws/src/Surgical_Tool_Tracking/ForwardPassDeepLabCut