Updating, re-training and refining your DeepLabCut model

Overfitting or generalizing, that is the question. What do you want to use your DLC model for? If you want to label many videos from a similar setup, an overfitted model will do very well (until you change the camera perspective, lighting conditions or maybe even the subject). On the other hand, if you wanted to train a model that detects human facial landmarks in different subjects, from different genders, ethnicities, facial hair styles and glasses and specially from different angles, then you need a model that generalizes well to all these conditions. And for that, you will need training examples, quite a few actually.

In this example we will start with a DLC model to detect facial landmarks, overfitted to a single subject, i.e., facial landmarks that look exactly like mine. We will then try to analyze new videos from different subjects of different genders and ethnicities. We will extract outliers from these misdetections, refine them manually and re-train the model. Last, just for fun, I will use anipose to triangulate multiple camera perspectives into a final 3D facial expression.

The original model was trained on 120 labeled frames of one single subject from 6 different camera angles, using resnet50 with 1.030.000 iterations

Figure 1: Video of parallel tracked face

<https://youtu.be/RoaPwEWcHF0>

It was to be expected, that this model wouldn’t generalize well to new subjects, some wearing glasses and different hair styles coving parts of the face. But we can make use of this and check were the model struggles most. Then, we can refine a few of those frames and re-train the model with the information that was apparently missing before.

Check the notebook provided [here]() for a step by step guide.

First, we start deeplabcut either as gui or over the command line and initialize the previous project. Set the path to the config file and list all new videos to be analyzed, in this case 30 new videos from 5 different subjects:

“path\_config\_file =

new\_videos =”

deeplabcut.analyze\_videos()

Extract outlier frames (numframes2pick: 5 because I was feeling lazy… and because 5x30 still is 150 frames to re-label manually…)

deeplabcut.extract\_outlier\_frames()

refine labels manually

deeplabcut.refine\_labels()

* 1. load the label folder
  2. load the corresponding h5 file with outlier predictions
  3. select likelihood threshold 0.4 (this is only to show you which labels have a specially low prediction, but as I am going to refine all labels, I recommend setting the threshold to 1 to display all markers as translucent circles )
  4. re-place labels by drag and drop with left mouse click
  5. you will notice that the workflow is minimally different than labeling original data. Instead of saving and quiting to go on labeling other files, a pop-up asking to refine other files will come up directly after clicking save.
  6. Quit will close the GUI without redirecting you to new files to refine… and that caused at least in my notebook to hang and having to restart the kernel.
  7. ProTip: DeepLabcut will often guess occluded points by their expected location. Please make sure to remove these labels when the specific body part is not visible. My facial expression model for examples “knows” where to expect the eyebrows and does so even when covered by strands of hair. Make sure to teach your model well. If something is not visible, don’t label it. This is probably the hardest part of the entire process. Suppress your expectations.

1. merge dataset
2. train model again

The result is a DLC model trained on 270 labeled frames from 6 different camera angles and 6 different subjects. Three females and three males, representing nationalities from Spain, Germany, Turkey, Iran, and China.

Figure 2: Collage of tracked facial expressions from different subjects

Extra. Recording with multiple camera angles has several benefits. First, different angles provide multiple training examples from the same recording, thus increasing your training set. Second, multiple angles avoid occlusion, e.g., when subjects roll their eyes behind your back, at least one camera will catch that. And last, multiple camera angles allow

Figuer 3: 3D video