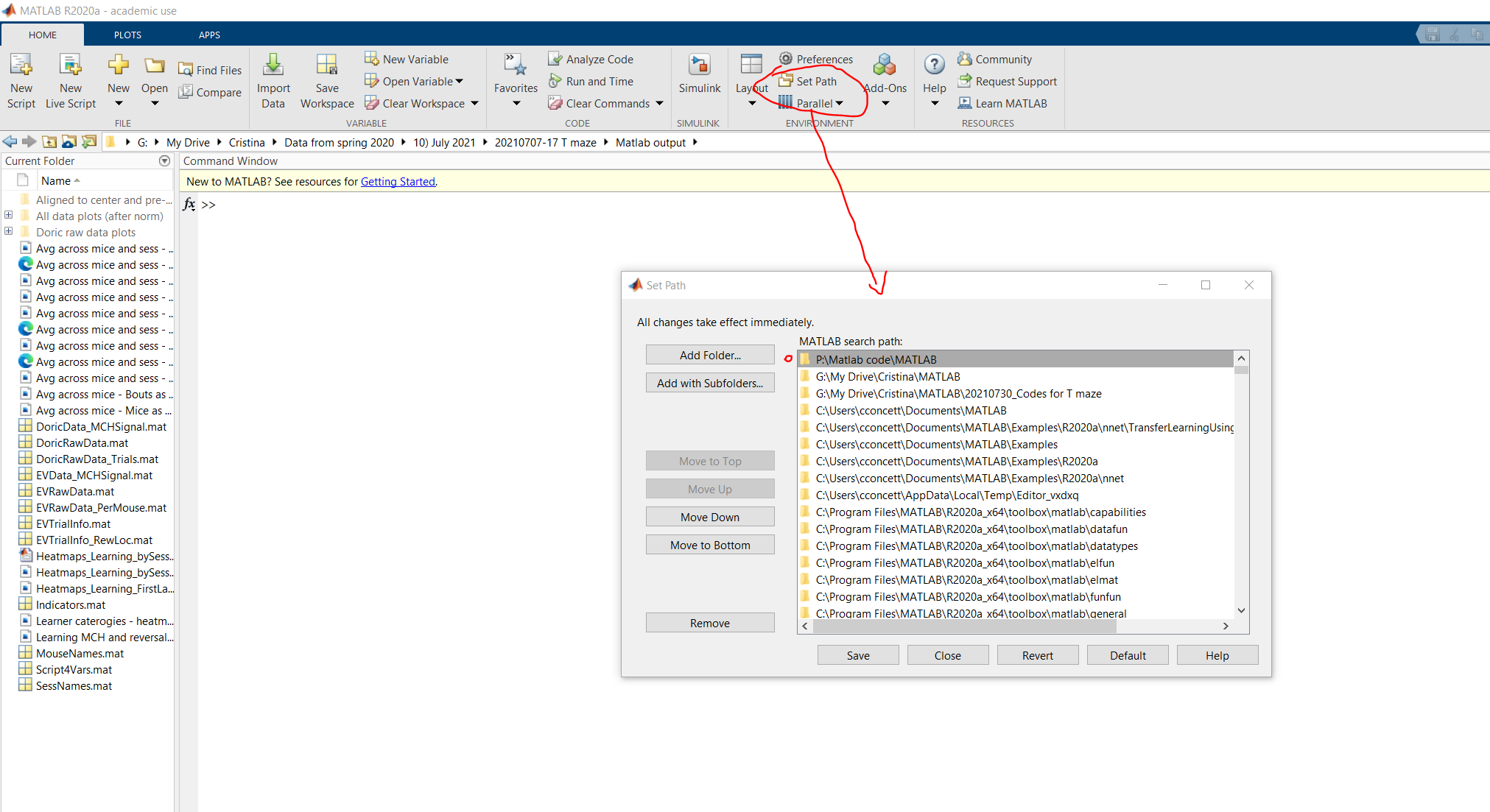
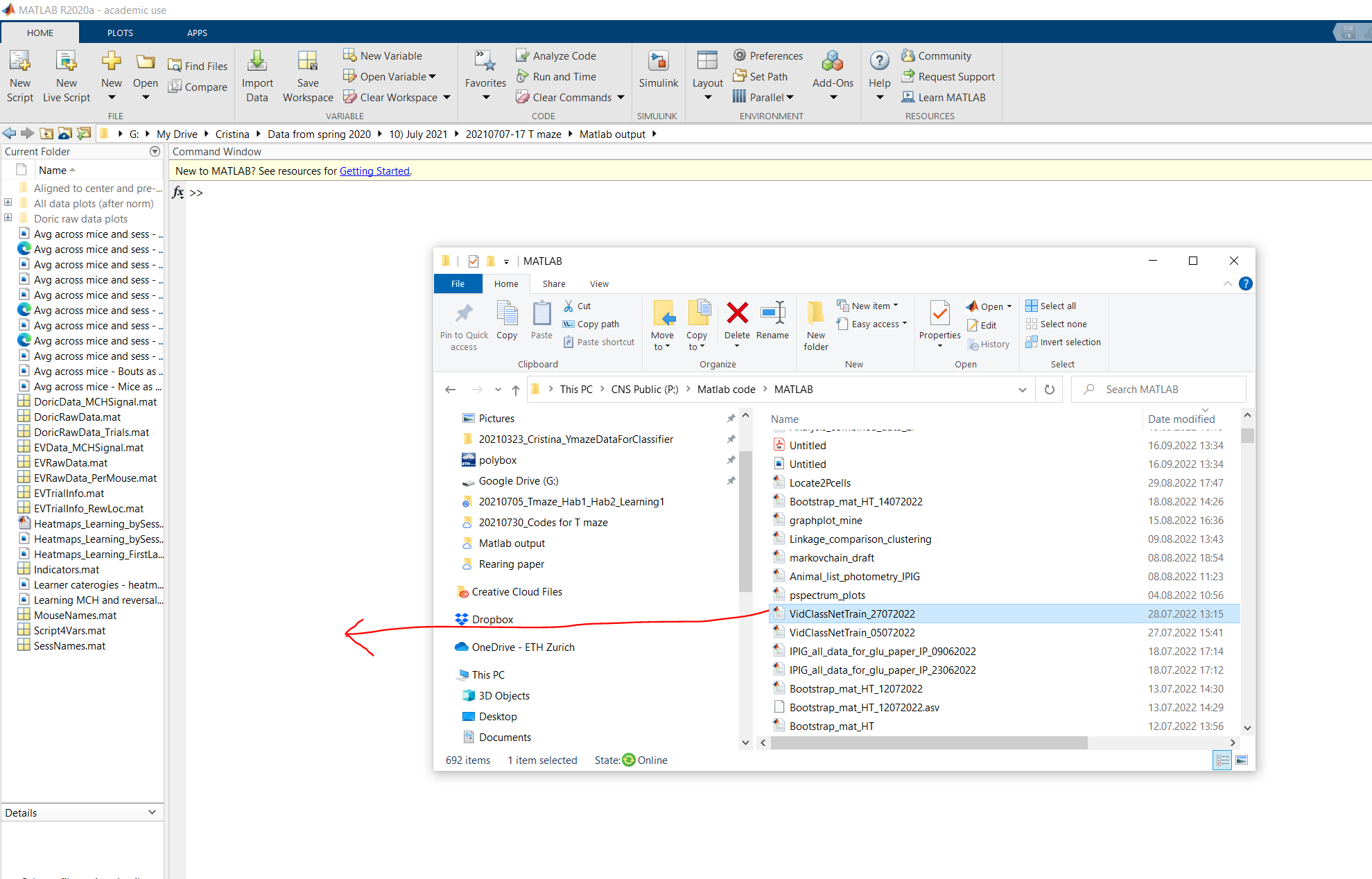
# Instructions on how to use Paulius’ MatLab Classifier software

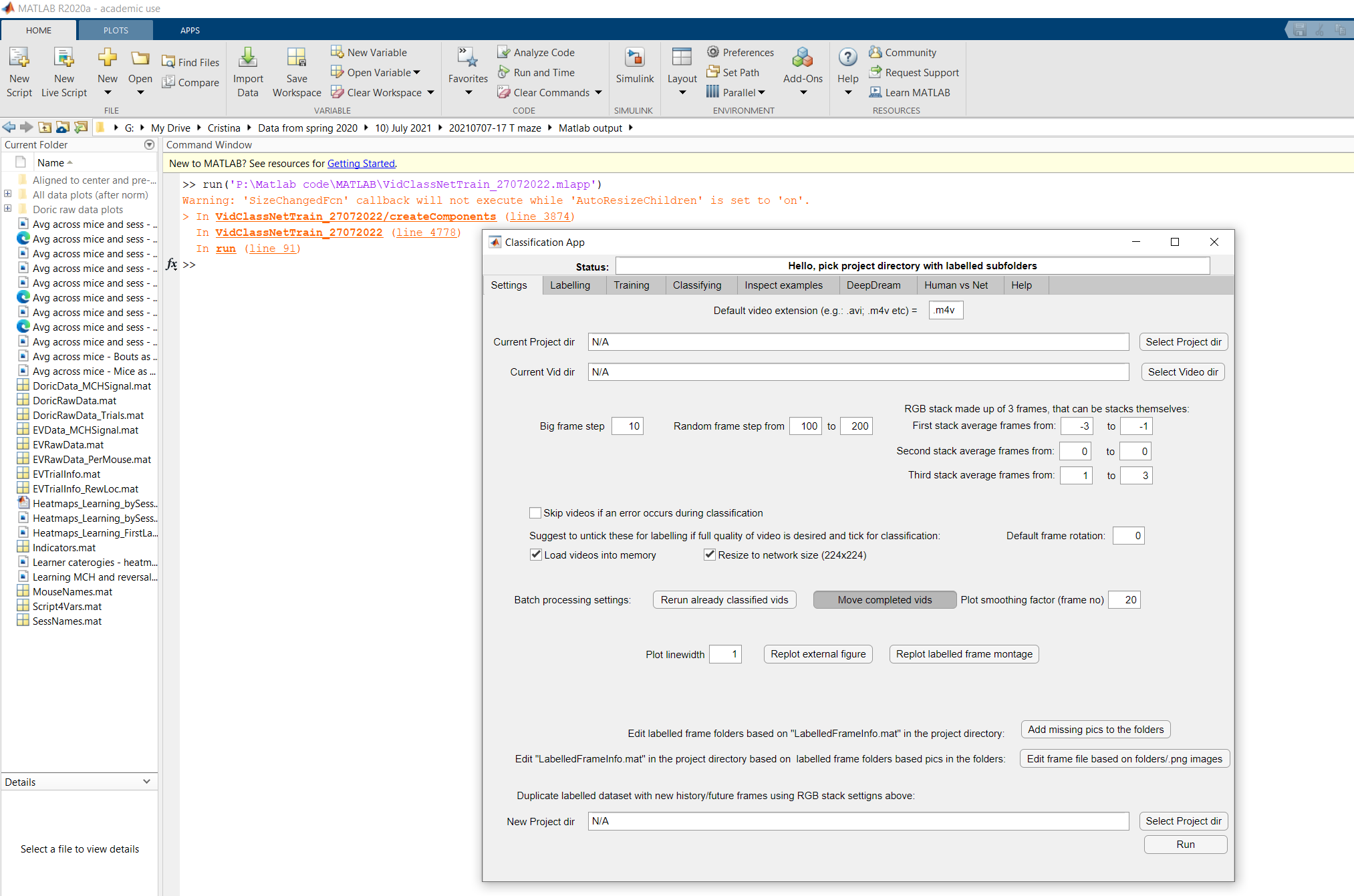
1. Set MatLab path to “P:\Matlab code\MATLAB”



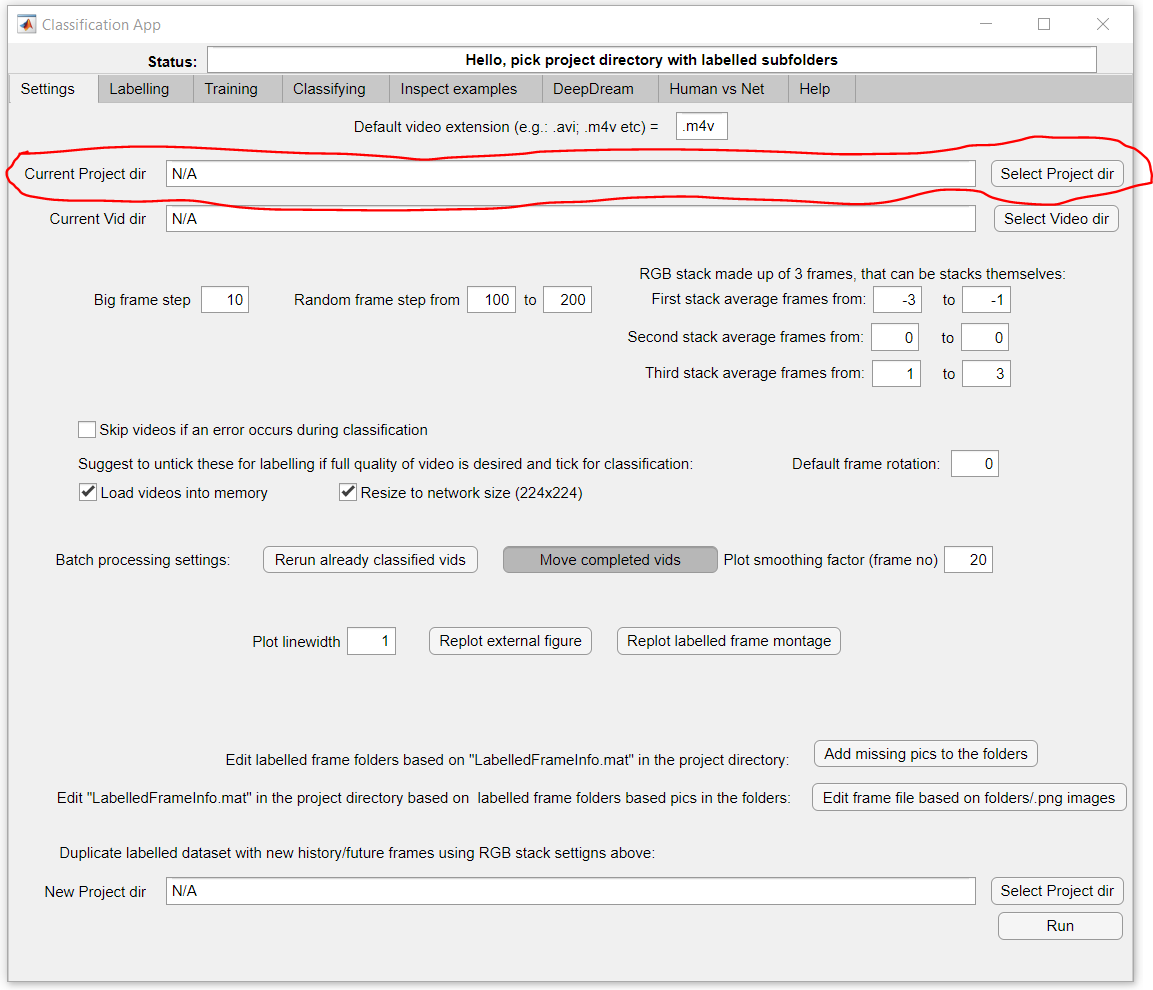
1. Navigate to “P:\Matlab code\MATLAB”, select file “VidClassNetTrain\_27072022”, drag and drop into the MatLab command window to run the app



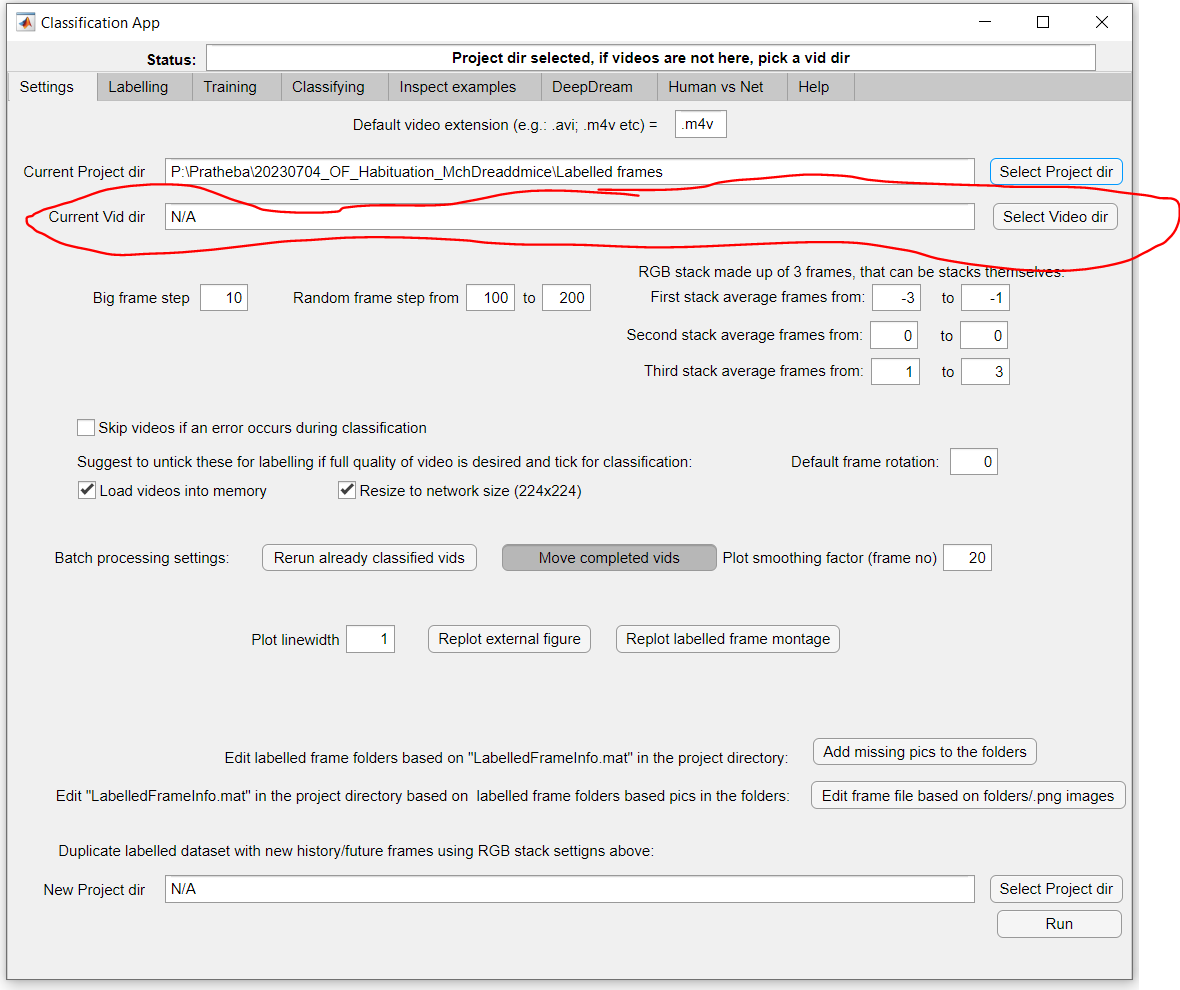
The app window will appear:



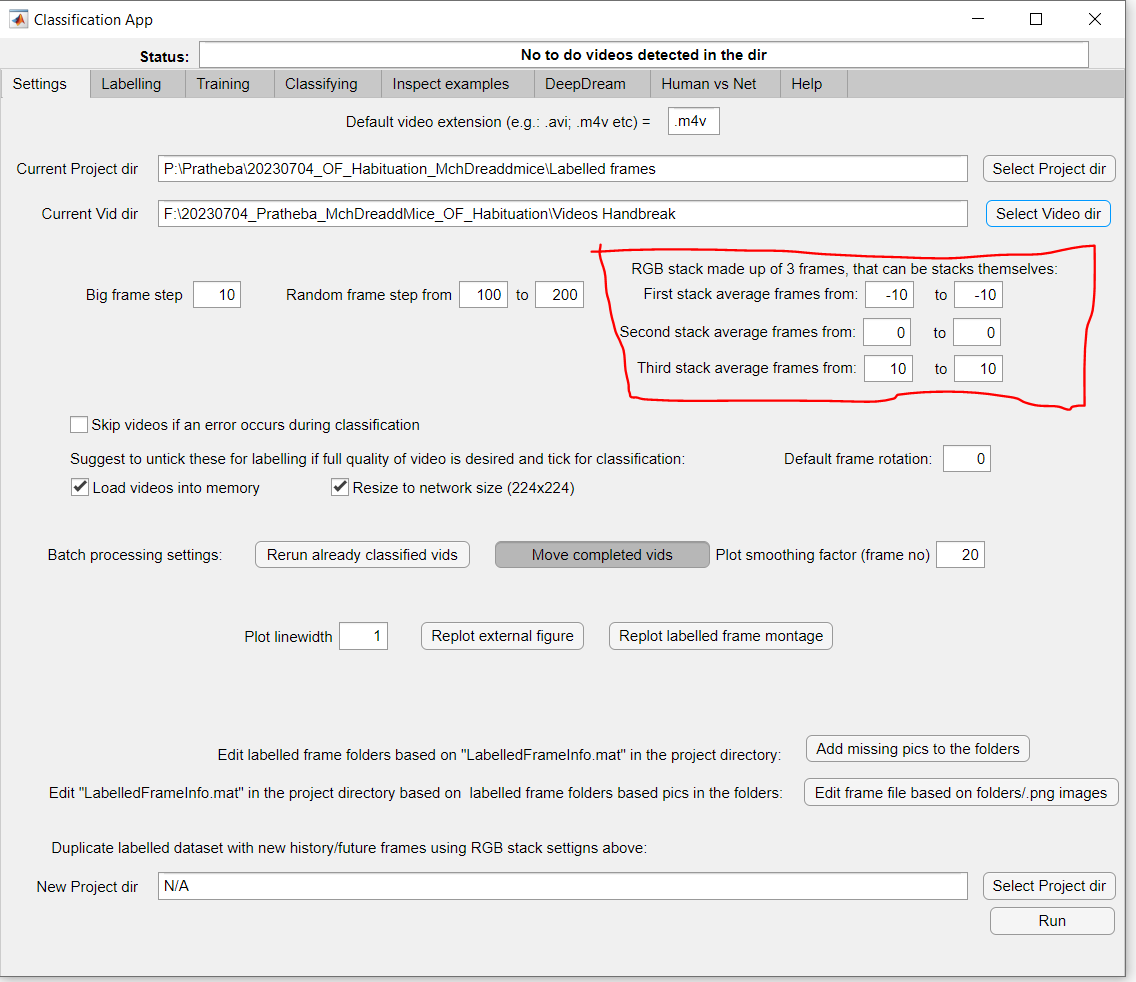
1. In the Settings (default) tab, select the Project directory (this folder need to have one subfolder for each category you want to classify, eg running, licking, etc)



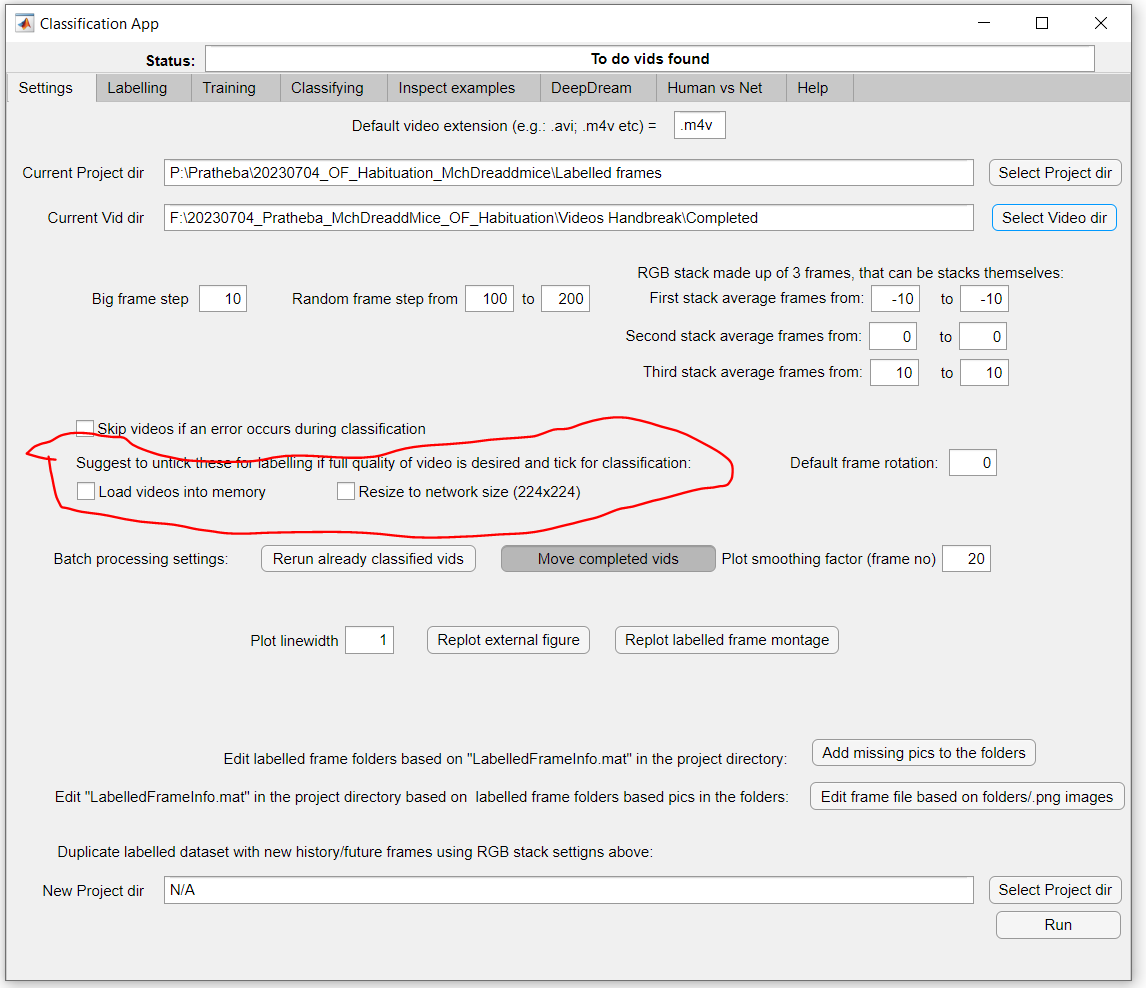
1. Now select the Video directory (this is the folder with videos you want to classify)



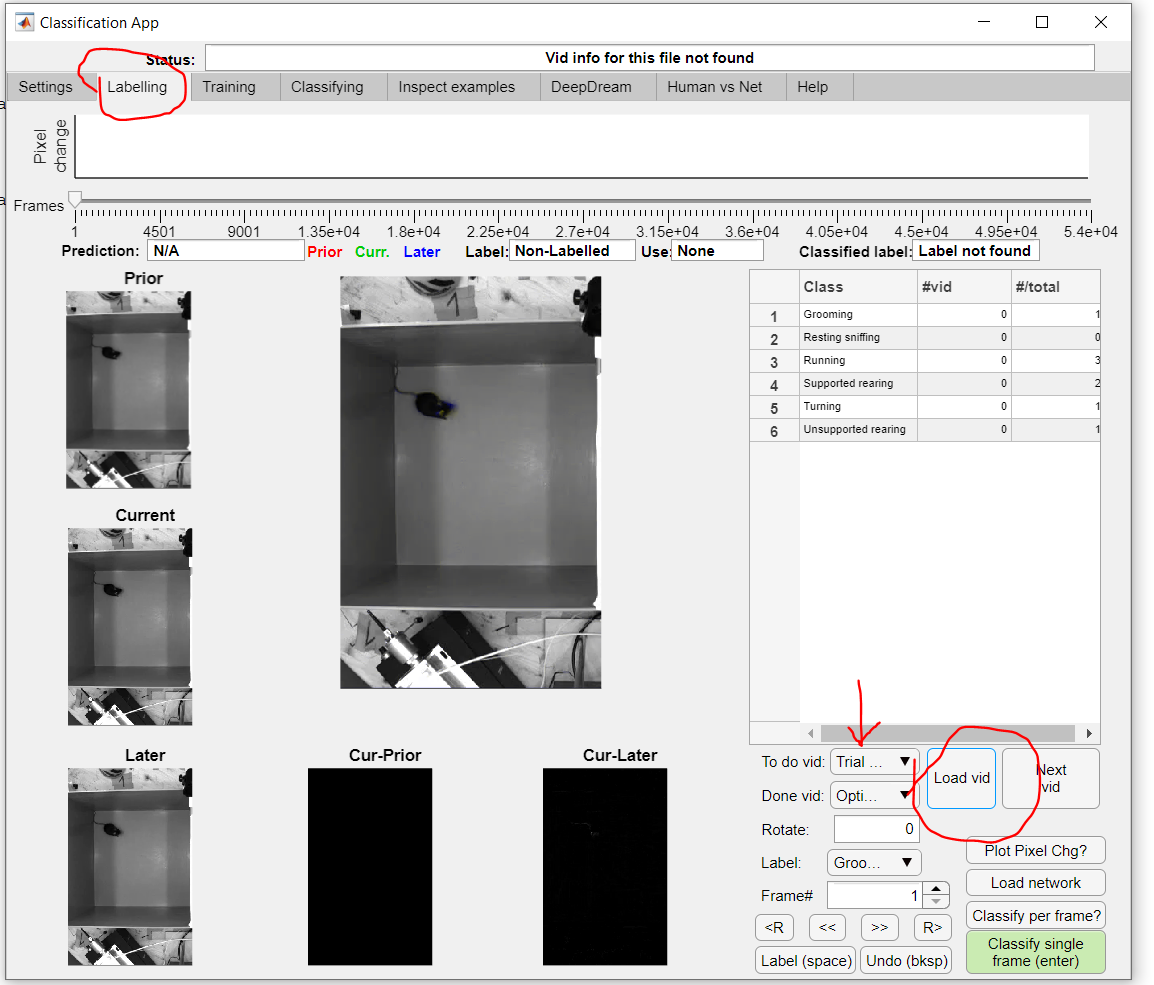
1. Now set the parameters for your RGB stack. This is the unit that will be labelled and classified within the classifier. It is made up of the current frame (or stack of frames, i.e. “second stack”), a previous frame (or stack of, ie “first stack”) and following frame (or stack of, ie “third stack”). In this example, second stack is the current frame (0), first stack is -10 frame and the third stack if +10 frame.



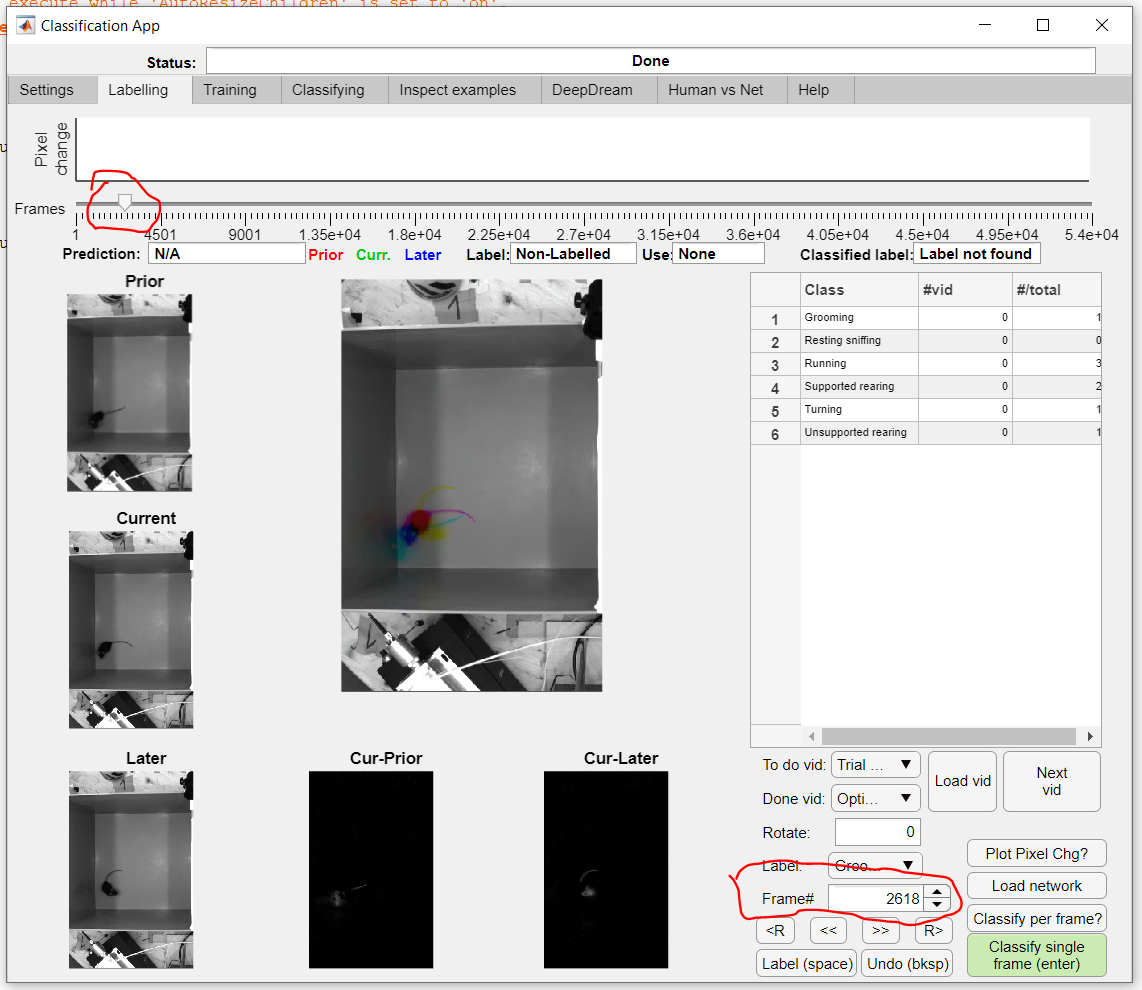
1. As suggested, untick the options “load video into memory” and “resize to network size” for labelling step (then tick them later for classification step).

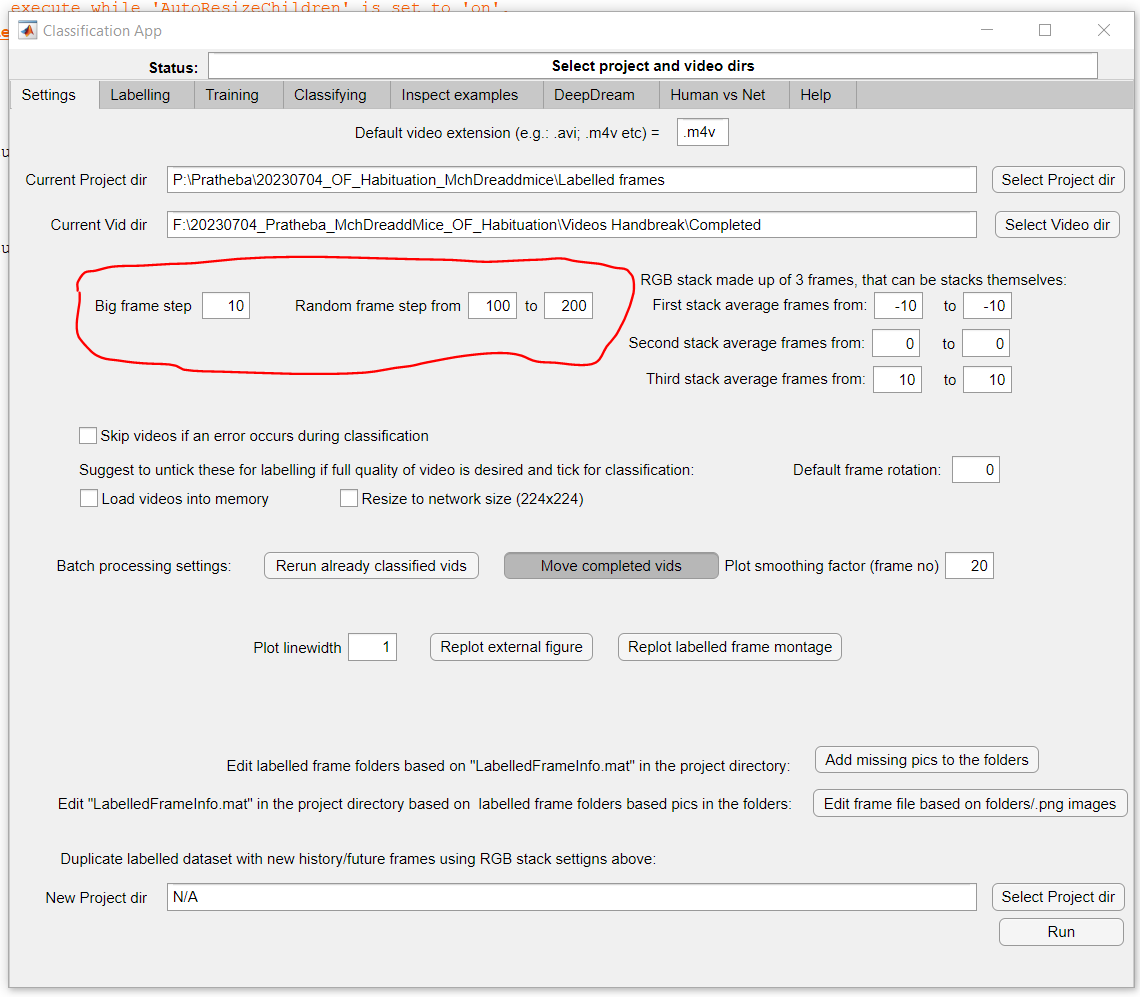


1. Move on to the Labelling tab, select your video in “To do vid:” and click “Load vid”

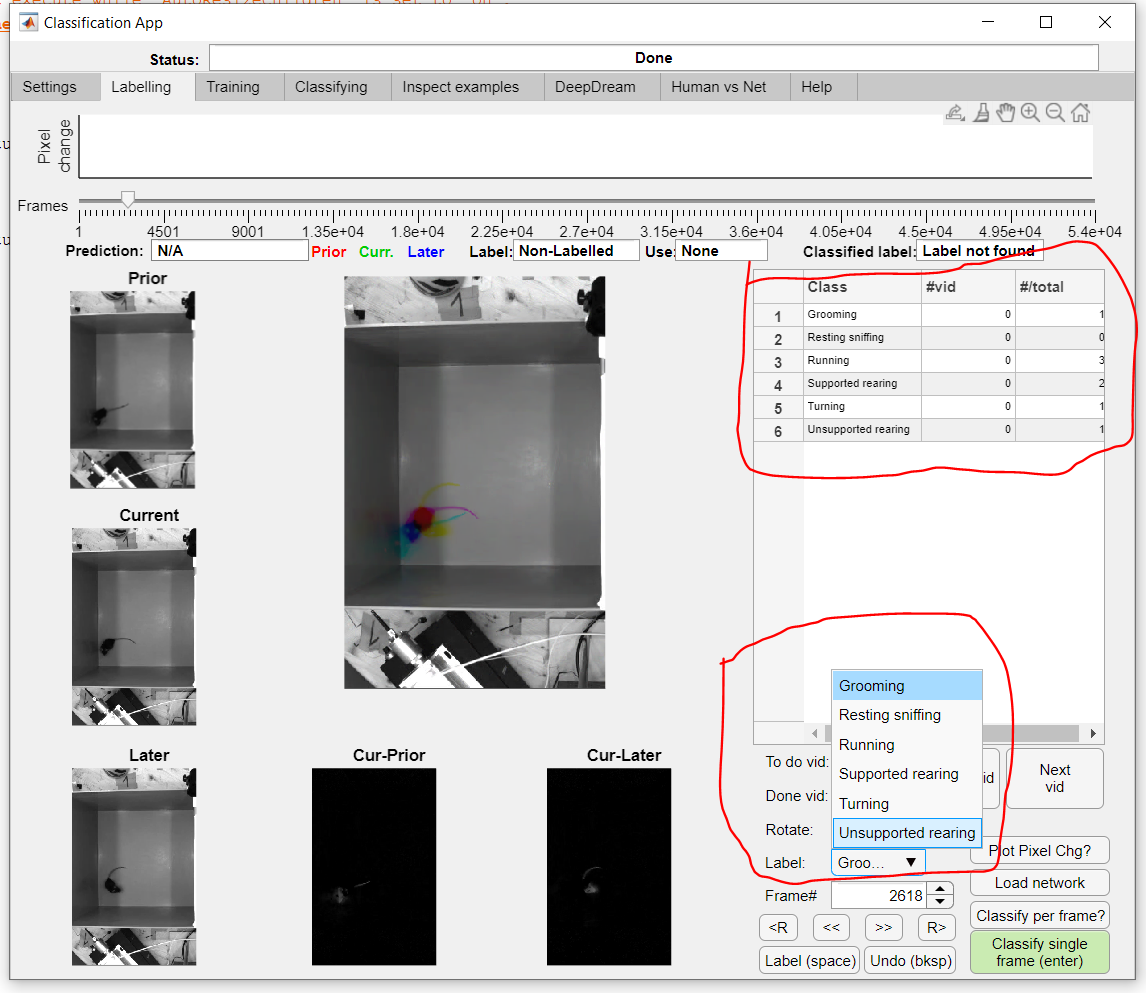


1. You can navigate to any frame of the video by clicking on the slider on top or by specifying a frame number or by jumping by a pre-set step or by a random step which you specify in the Settings tab by pressing “up” or “down” arrow keys on your keyboard (or move by 1 step, ie 1 frame, by pressing “left” and “right” arrows on your keyboard). You can find all keyboard shortcuts in the “Help” tab of the classifier.

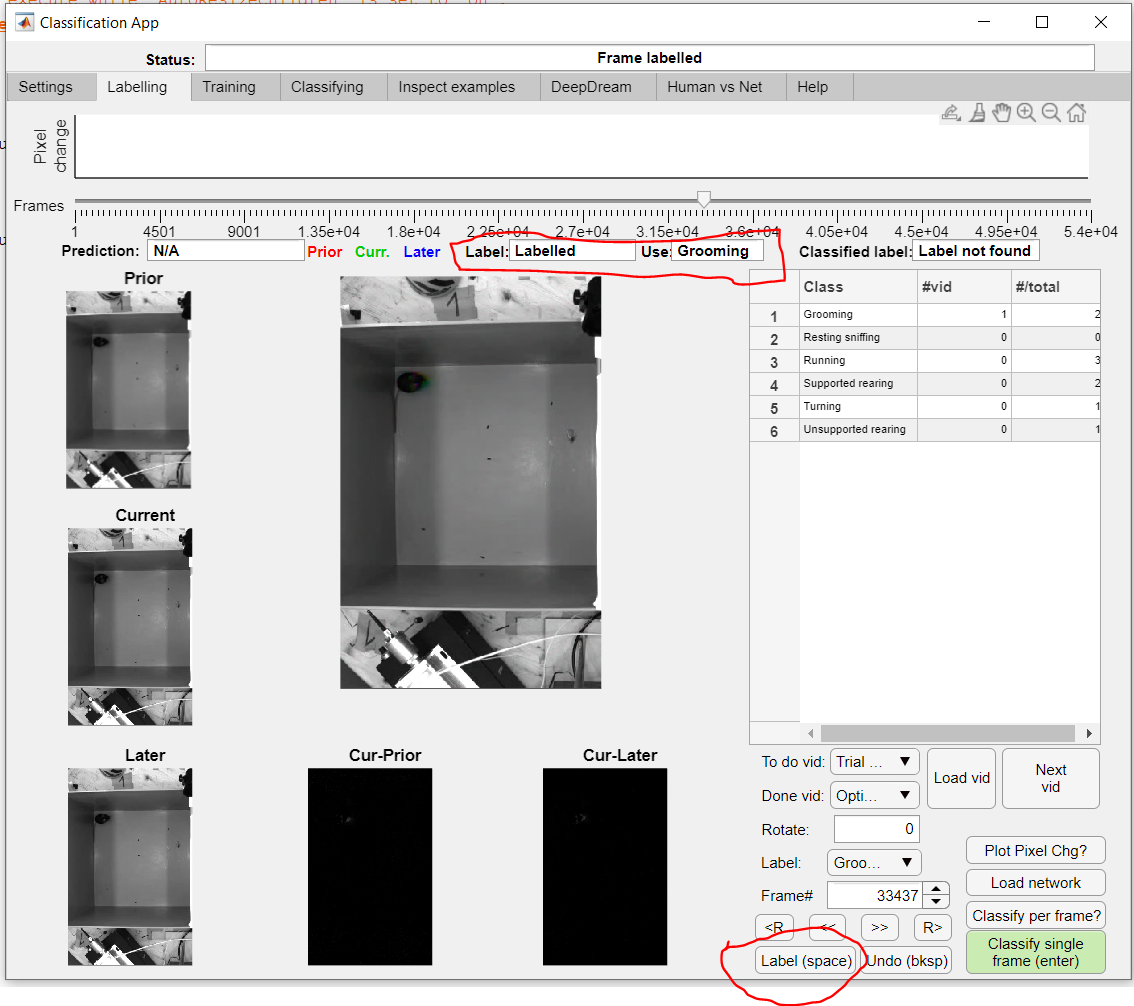




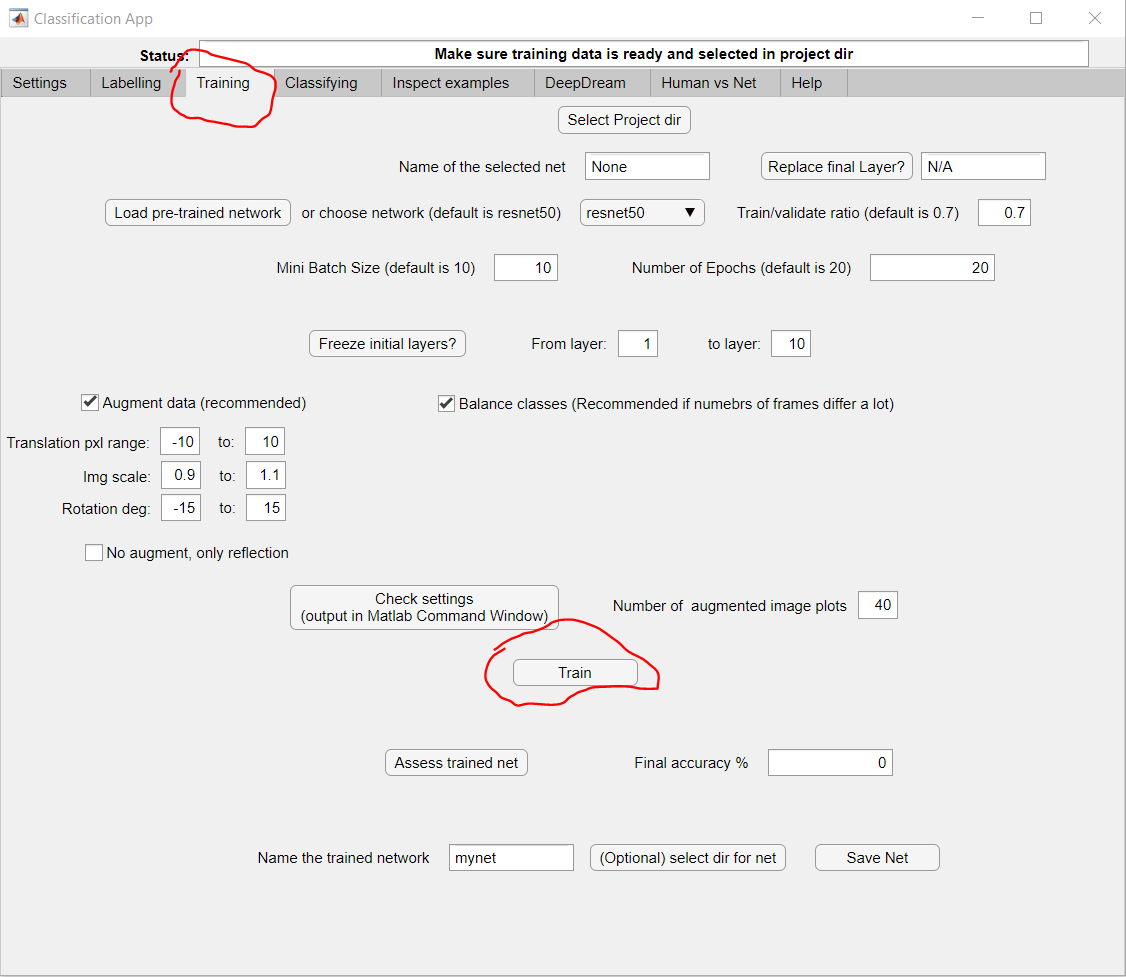
1. You can select a label (i.e. category, or class) for your frame from the drop-down menu or by using keyboard shortcut numbers. The window on the right tells you how many frames are already labelled in each category in the current video (“# vid”) and in total, ie across videos in the video folder (“# total”).



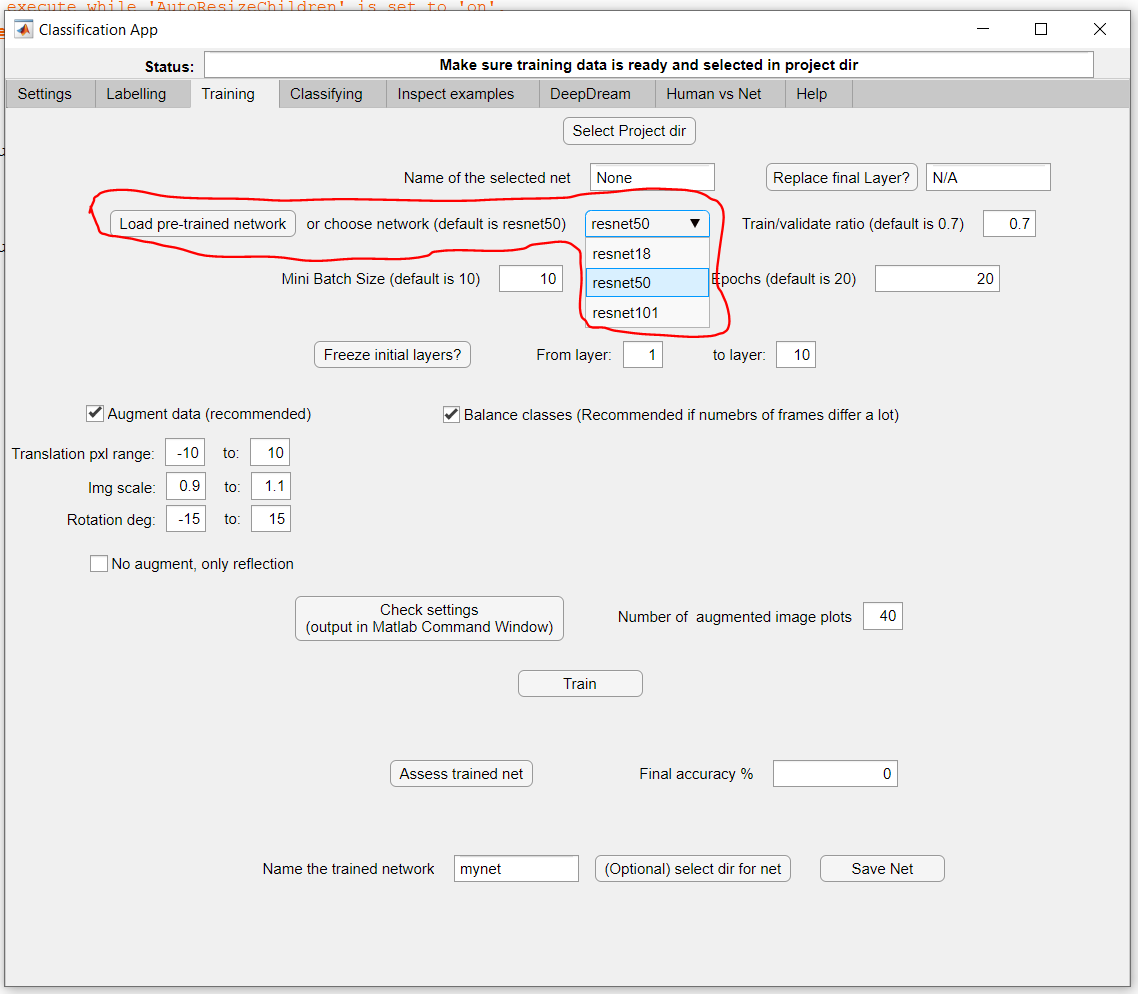
1. To label the frame, click on “Label (space)” or press the space key on your keyboard. The status of the frame will change to “Labelled” and in “Use” you can see the category that has ben labelled and the top bar with “Status” will show “Frame labelled”. If you make a mistake while labelling, you can re-label a frame in the same way and the top bar with “Status” will show “Frame RE-labelled”. The frames labelled like this are saved by the classifier in the corresponding subfolders of your project directory.



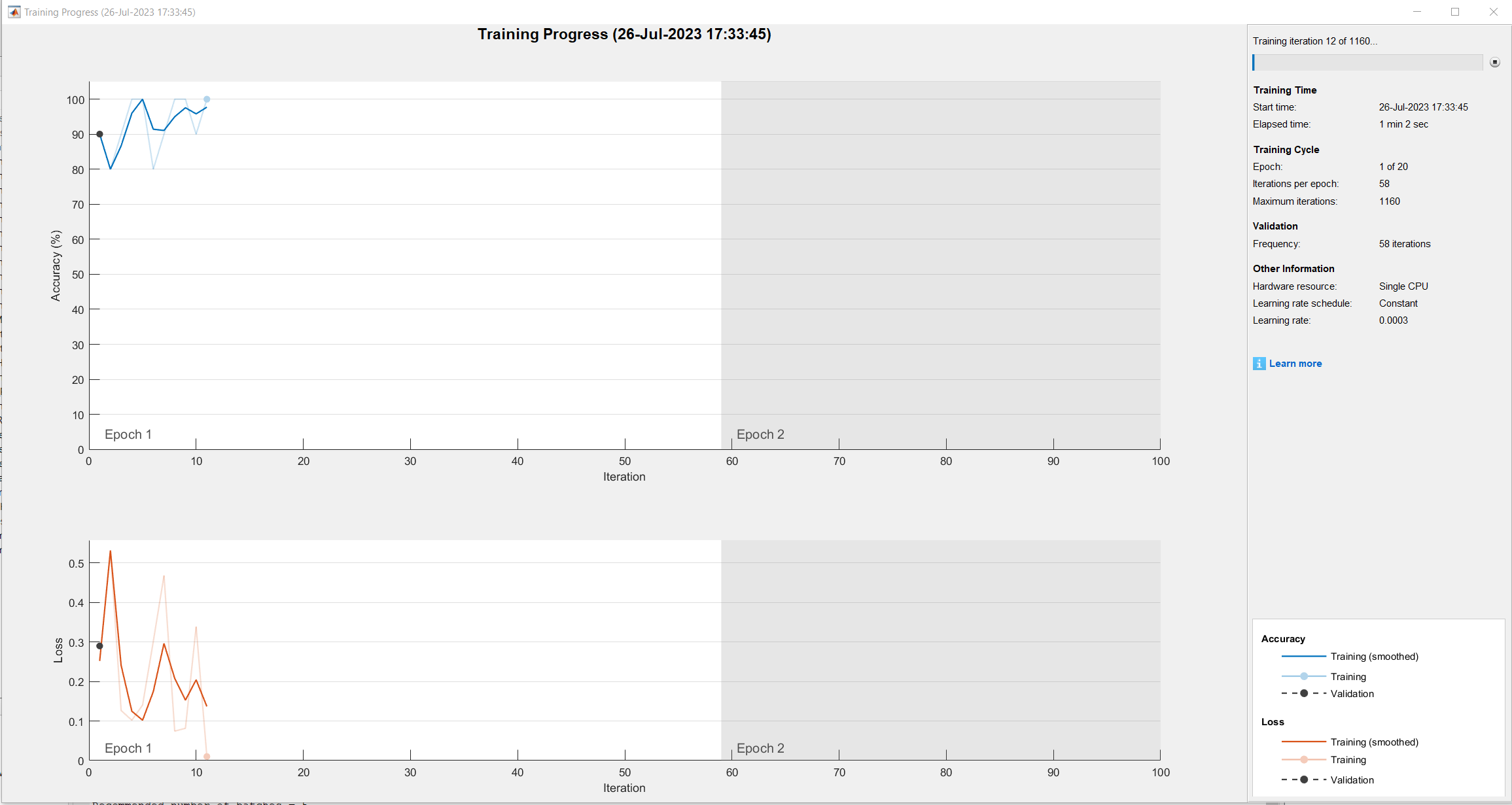
1. Once you have at least ~200 labelled frames for each category, you can move on to the “Training” tab. Select the project directory if not already selected.



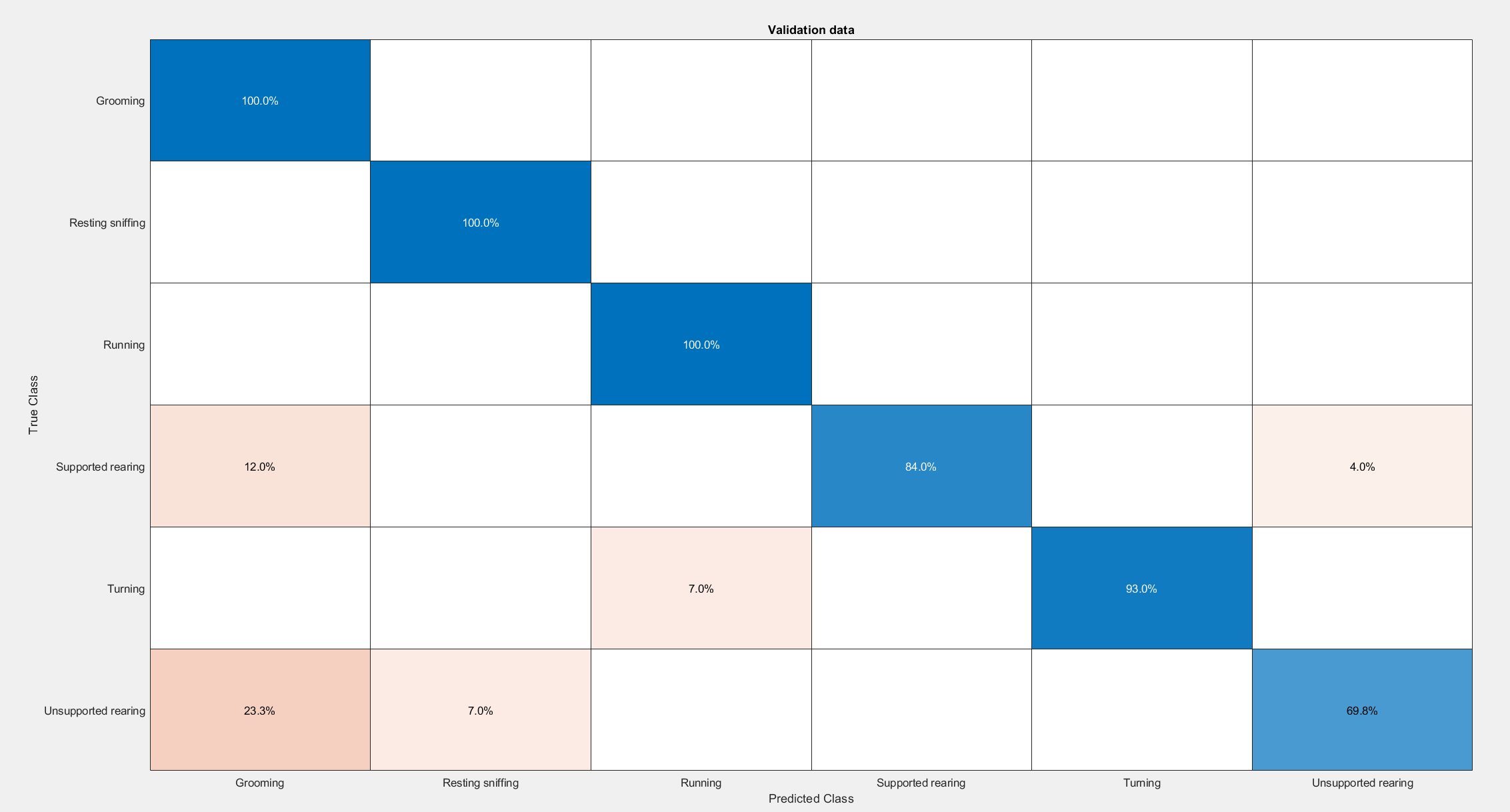
You can train your network based on a previous network, either your own or one of the ones from Google (resnet18, resnet50, resnet101 – the numbers indicate the number of layers the networks have; more layers is not necessarily better).



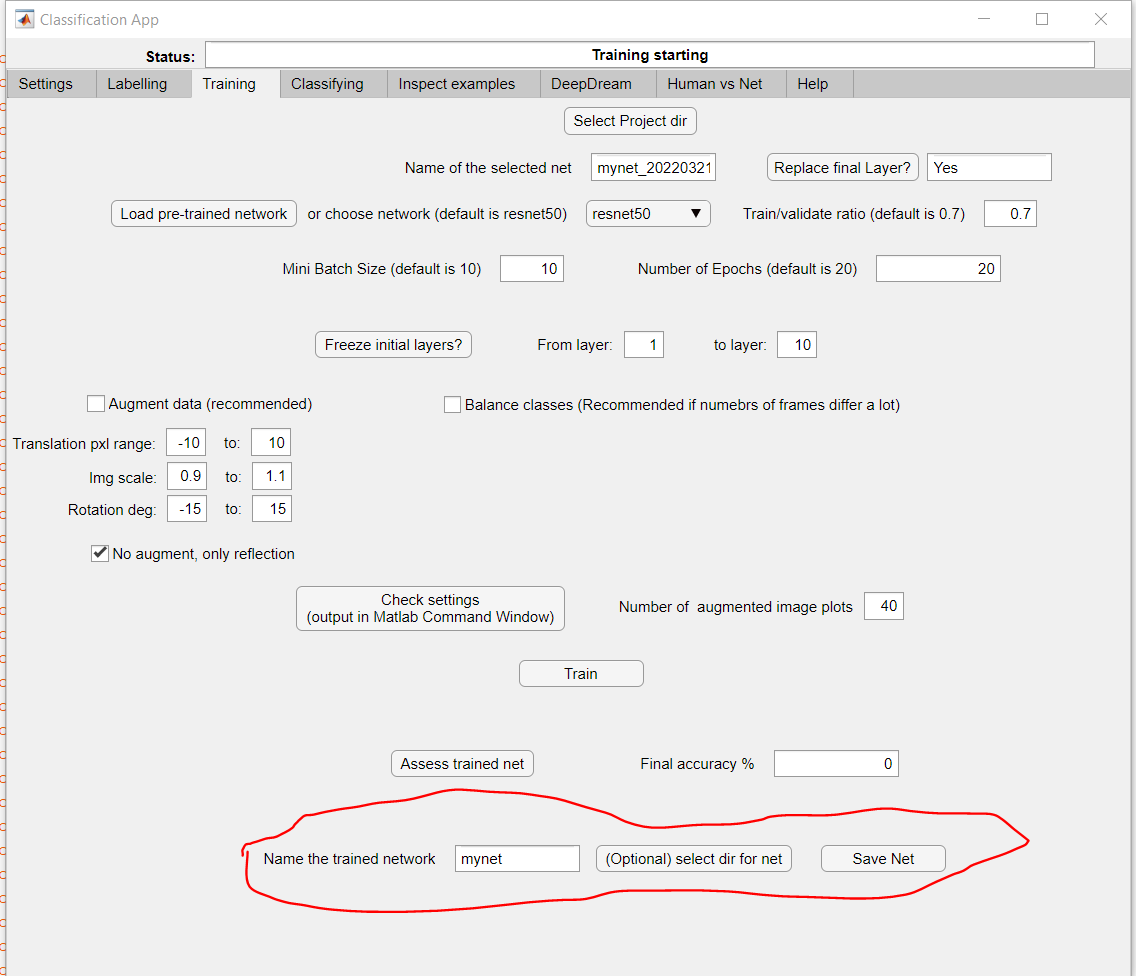
You can set your “train/validate ratio” (ie which fraction of labelled frames will the network be trained on, vs which fraction will be held back for validation). “Mini batch size” and “Number of epochs” determine the length of training, ie how many iterations will the network undergo. Once you click on “Train”, a Training Progress window will appear which will update in real time as the network is trained (in this example, accuracy is already start at the beginning because I selected a pre-trained network; what matters is the accuracy at the end of the training, ie final accuracy). This step uses GPU so it will be faster if run on the workstation computer (~10-20 minutes on workstation, ~45 min on a normal desktop pc in the office).



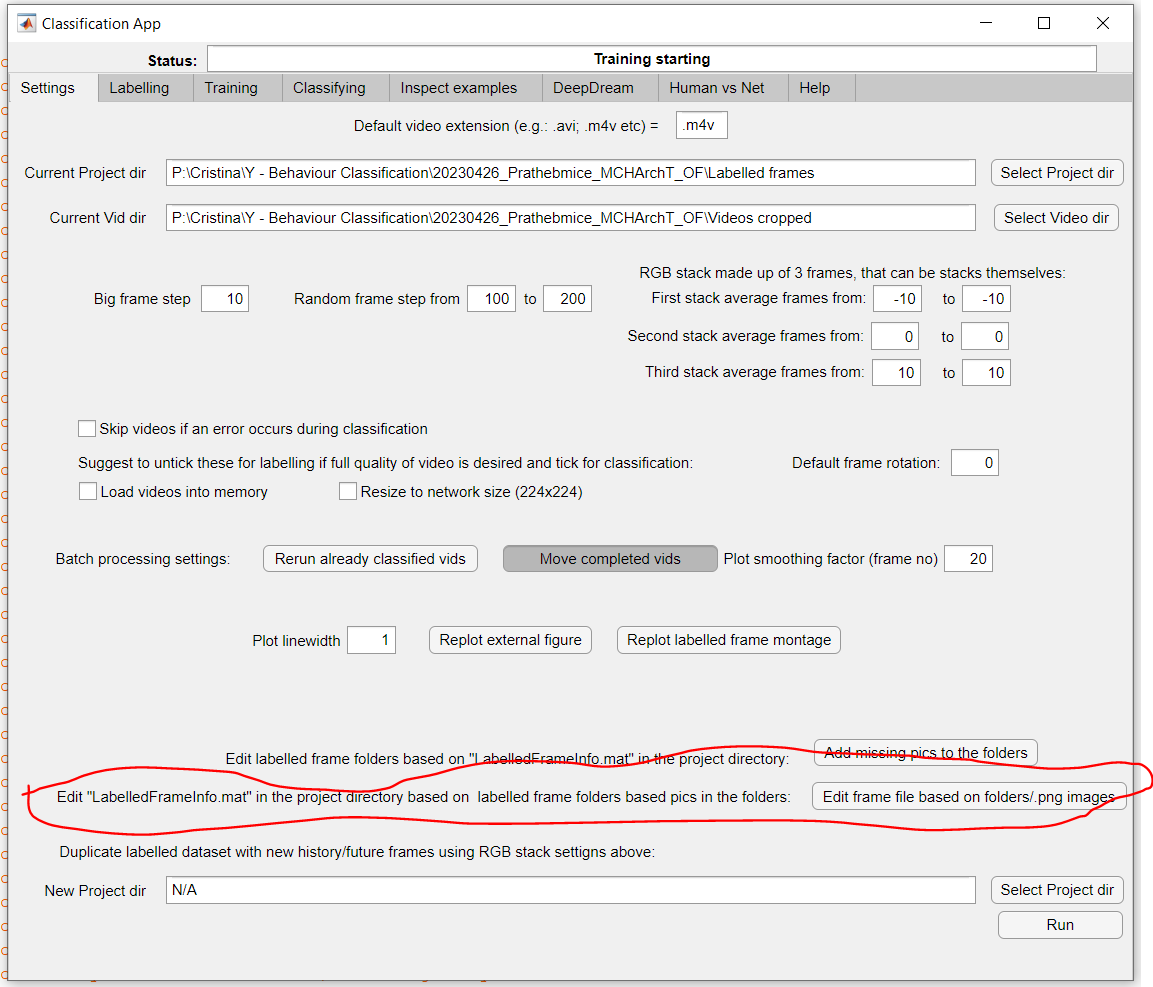
At the end of the training, you can see the overall final accuracy and also the accuracy broken down for each class, as in the example below.



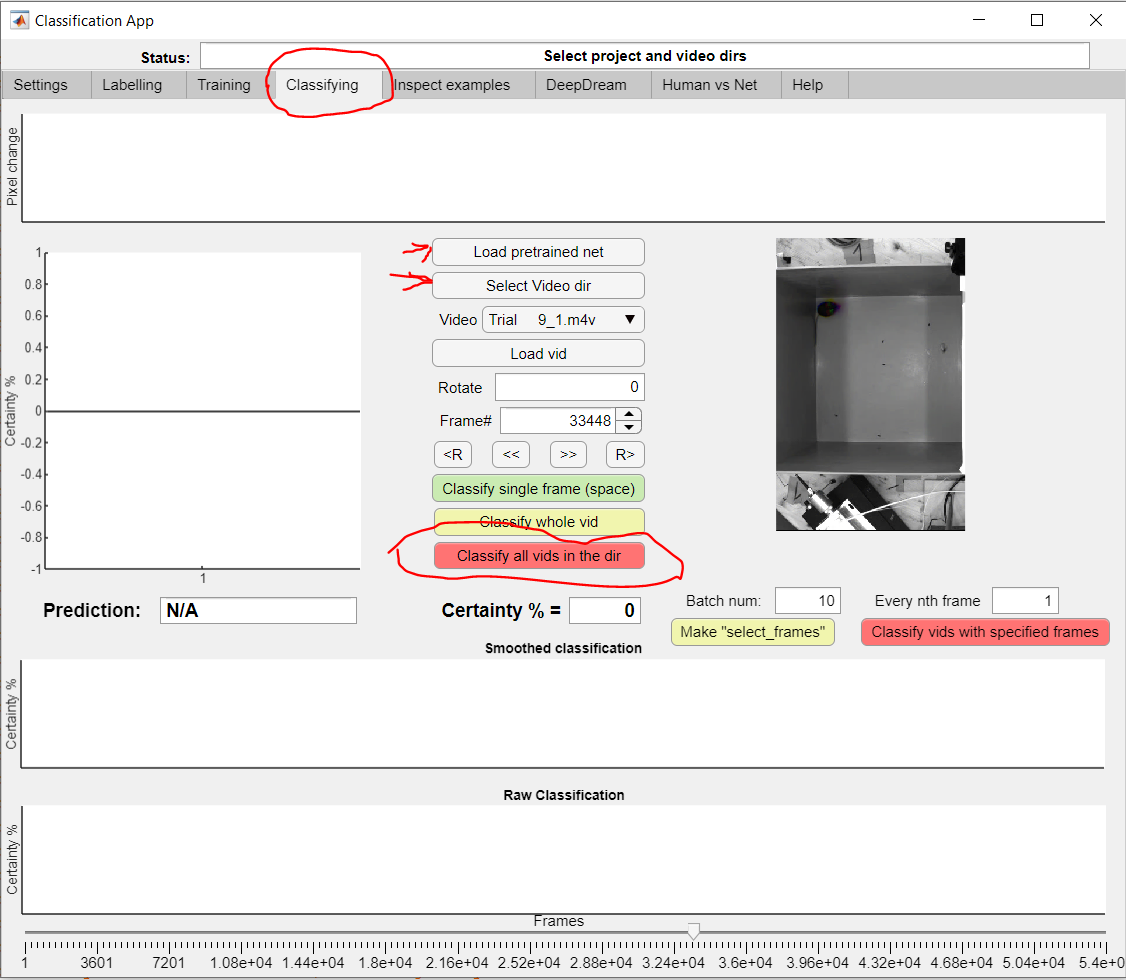
Once the training is done, inspect the final accuracy and optionally save the network:



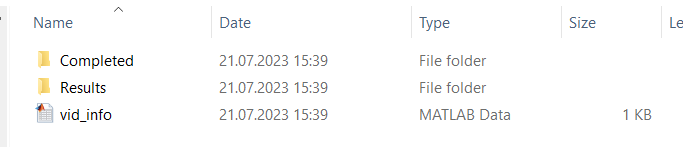
A good accuracy will be around 85-95%, with minimal errors across classes. If the accuracy is not good, inspect labelled frames and, if you find mistakes in the labelled frames, delete or move them to the correct folder and update the classifier in the Settings tab, then train again.



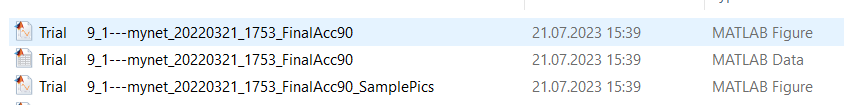
Once you have an accurate enough network, move on to the “Classifying” tab. Load the network if not already loaded, load the video folder if not already loaded, and click on “Classify all vids in the dir”. This step takes a few hours (depending on how many videos and how long they are), so best to run overnight.



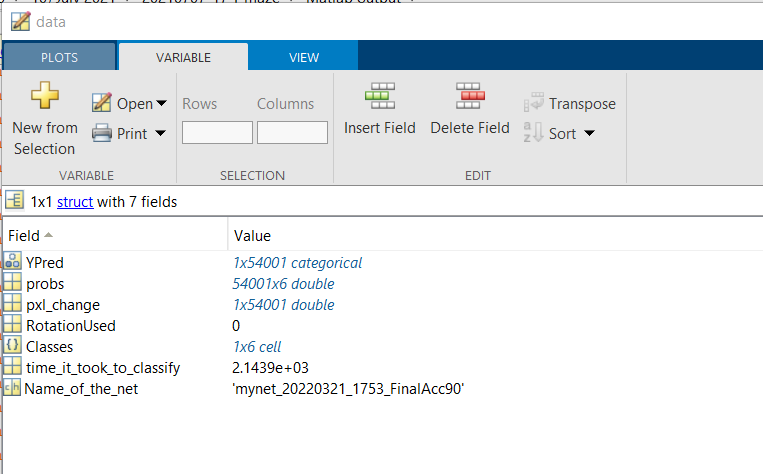
The classifier will generate 2 folders within your video folder, one named “Completed” where videos will be moved once they are classified, and a “Results” folder with classification results.

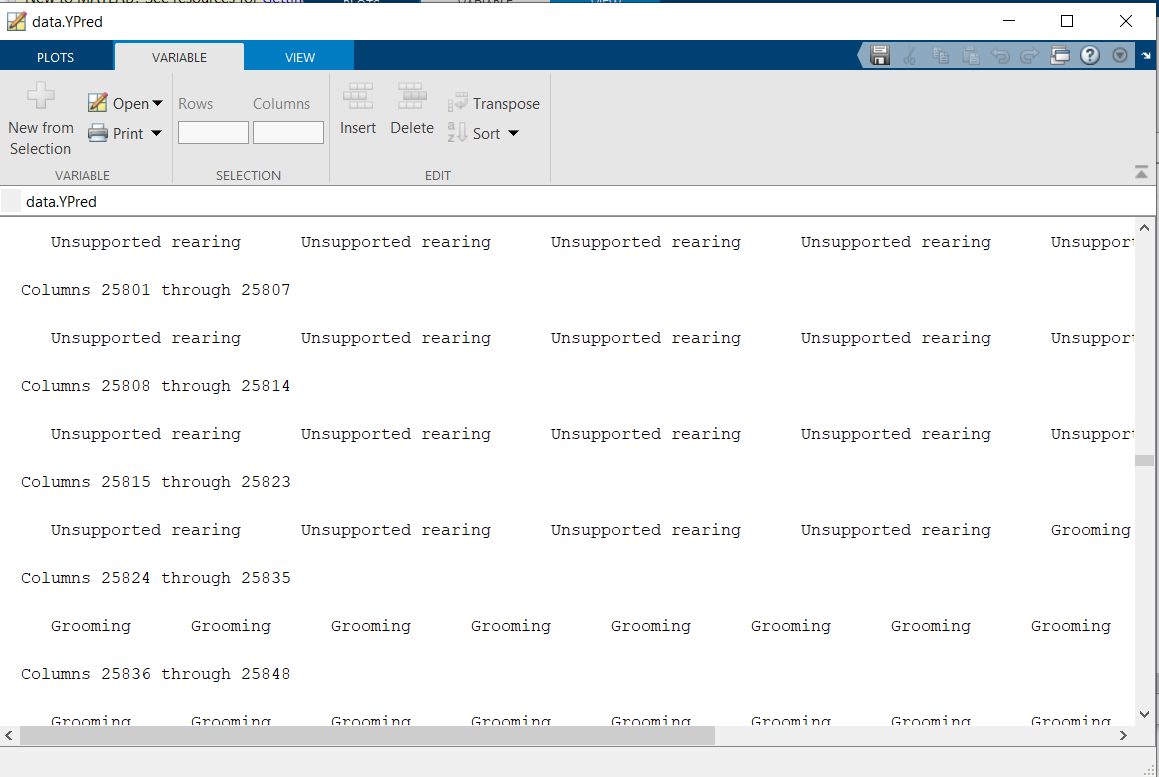


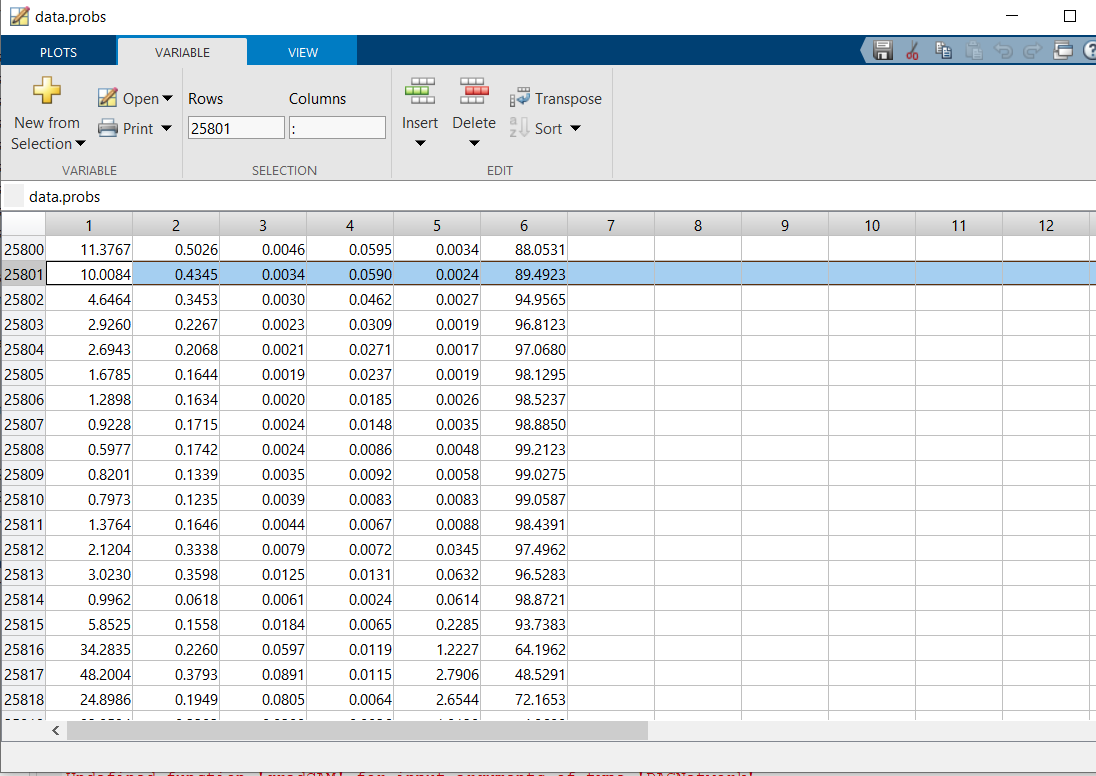
Inside the Results folder, there will be 3 files for each video: 2 .fig files (ie matlab figures of the whole video and sample pictures), and one .mat data file.



Load the .mat file into MatLab to analyse data. It is a structure like the following:



“Classes” are the names of classified categories. “Ypred” is the prediction of the category for each frame: 

“probs” are the probabilities of each class for each frame (for example, frame 25801 has Unsupported rearing in Ypred because it’s the class with the highest probability of 89%).

The data structure also contains a pixel change matrix.