DL_Track
v0.1.0

Preface

Welcome to the DL_Track python package tests. In the next roughly 20 pages, we will demonstrate how you can objectively test the functionality of DL_Track when automatically and manually analysing ultrasonography images and videos of human lower limb muscles. Because we have not (yet) integrated unit testing in the DL_Track python package, we have prepared specific instructions and provided example results. This testing tutorial works in the way that you will perform several analyses and compare your results to the test result we provided. If the results are comparable, the DL_Track python package is functional. Have fun!

Content

Good to know…

All relevant instructions and guidelines for the installation of the DL_Track software package are described in our Github repository, so please take a look there if anything is unclear.

~~So, what is DL_Track all about? The DL_Track algorithm was first presented by Neil Cronin, Olivier Seynnes and Taija Finni in 2020. The algorithm makes extensive use of fully convolutional neural networks trained on a fair amount of ultrasonography images of the human lower limb. Specifically, the dataset included longitudinal ultrasonography images from the human gastrocnemius medialis, tibialis anterior, soleus and vastus lateralis. The algorithm is able to analyse muscle architectural parameters (muscle thickness, fascicle length and pennation angle) in both, single image files as well as videos. By employing the deep learning models, the DL_Track algorithm is one of the first fully automated algorithms, requiring no user input during the analysis. Then in 2022, we (Paul Ritsche, Olivier Synnes, Neil Cronin) have updated the code and deep learning models substantially, added a graphical user interface, manual analysis and an extensive documentation. Moreover we turned everything into an openly available Pypi package.~~

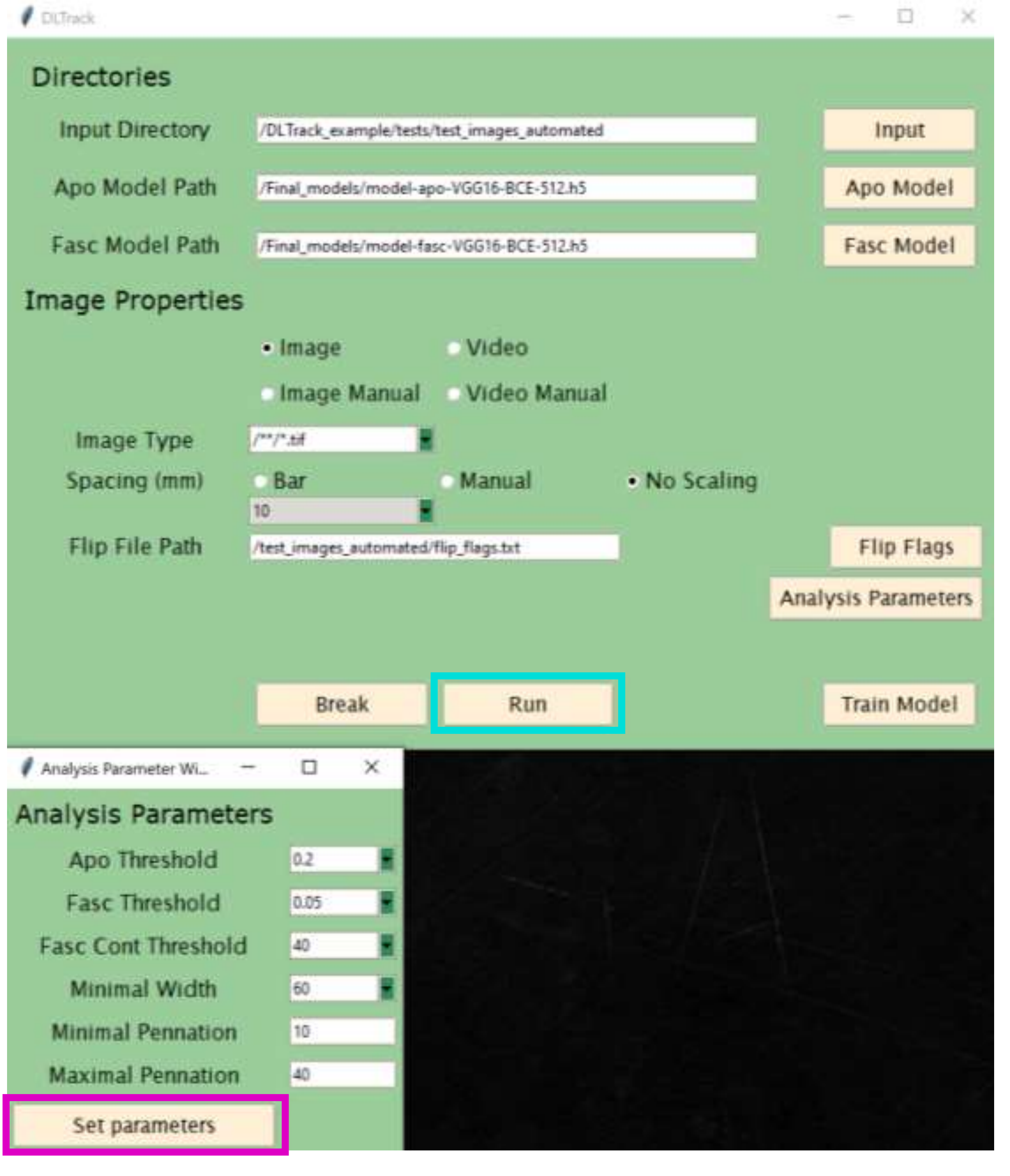
Before we start with testing, here are some important points to consider:

- Please **exactly** follow the instruction we provide!
- We assume that you have looked at the "DL_Track_tutorial.pdf" file prior to working through these test instructions. We therefore won't explain the core functionalities of the DL_Track GUI.
- All of the test files required for testing procedures are in the "DL_Track_example/tests" folder. If you have not downloaded the "DL_Track_example" folder, you can do so here. Unzip it and put it somewhere you can easily access it, you will use it extensively during testing.
- During the testing process, you will need our pre-trained models. If you have not downladed them already, you can do so here. Unzip the folder, put it somewhere you can easily access it and select the included aponeurosis and fascicle model for the testing procedures.
- In case you encounter problems during the testing process or your test results deviate from the ones we provided without reasonable explenation, please report it. Take a look at the "DL_Track_bugreport.pdf" file in this folder how to best do this. Otherwise contact us at paul.ritsche@unibas.ch, but we would prefer the other way.

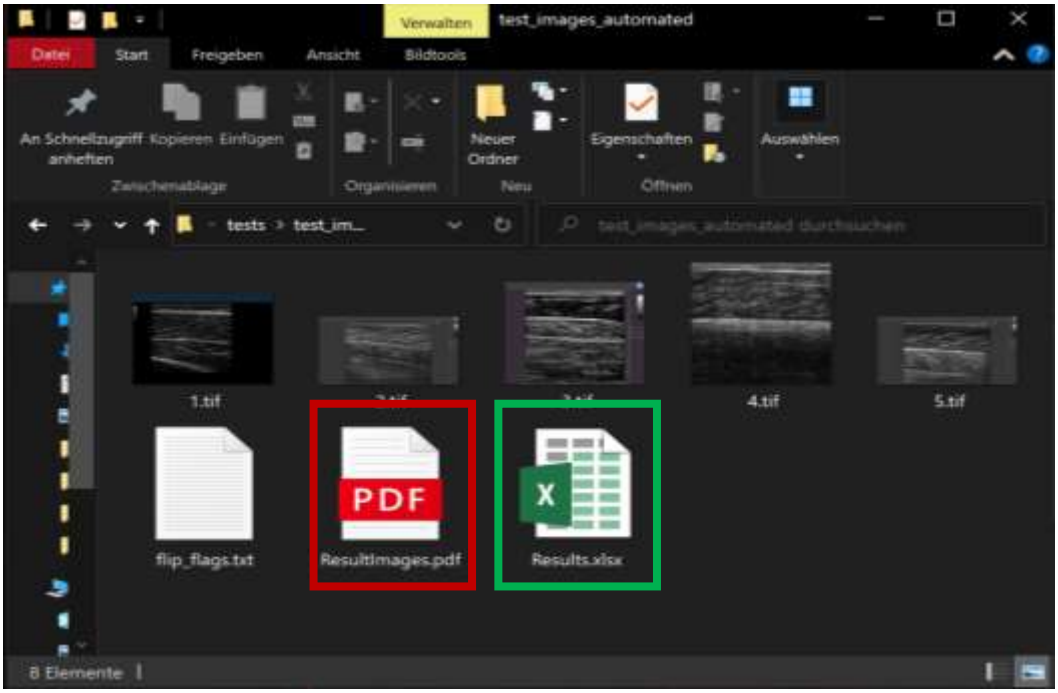
Automated Image Analysis Test

The DL_Track python software package offers several different analysis types for analysis of human lower limb longitudinal ultrasonography images. The first analysis type you are going to test is the automated image analysis. The images are evaluated without user input and may be scaled. However, we will not scale the images during testing. For this test, single images (not videos) are a prerequisite. The test images and the flip_flag.txt file you must use for this test are located in the "DL_Track_examples/tests/test_images_automatic" folder. In the next few pages, we will look at every required step to successfully perform automated image analysis testing.

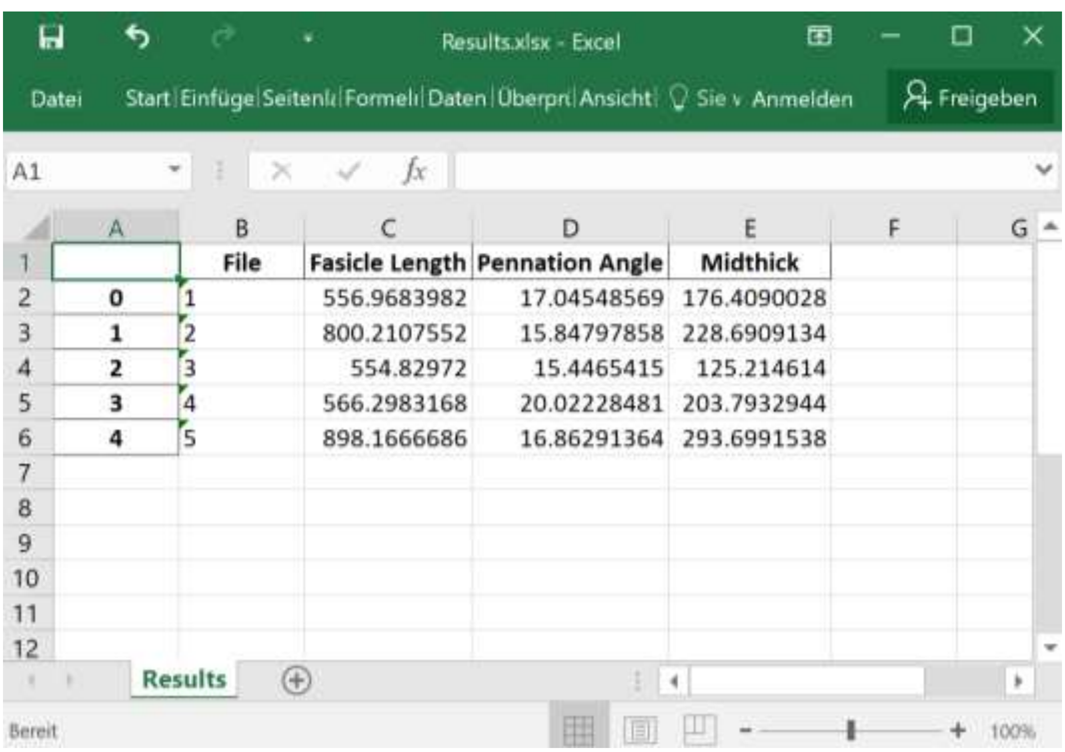
For this test make sure that the files used and parameters specified are exactly as demonstrated below. Moreover, keep the pre-specified parameter settings in the “Analysis Parameter window” as they are. Once you made sure to use the right images (“DL_Track_example/tests/test_images_automated”), the provided pre-trained models and the right flip_flag.txt file (“DL_Track_example/tests/test_images_automated/flip_flags.txt”), click the **Select parameters** button to set the analysis parameters and then the **Run** button to start the analysis.



Once the analysis is complete, two new files were created in the "DL_Track_example/tests/test_images_automated" folder. The **ResultImages.pdf** file and the **Results.xlsx** file. You can disregard the **ResultImages.pdf** for this test.



Open the **Results.xlsx** file and compare the analysis results to the ones demonstrated below. If the results are similar, the DL_Track package works properly for automated images analysis! Yay!

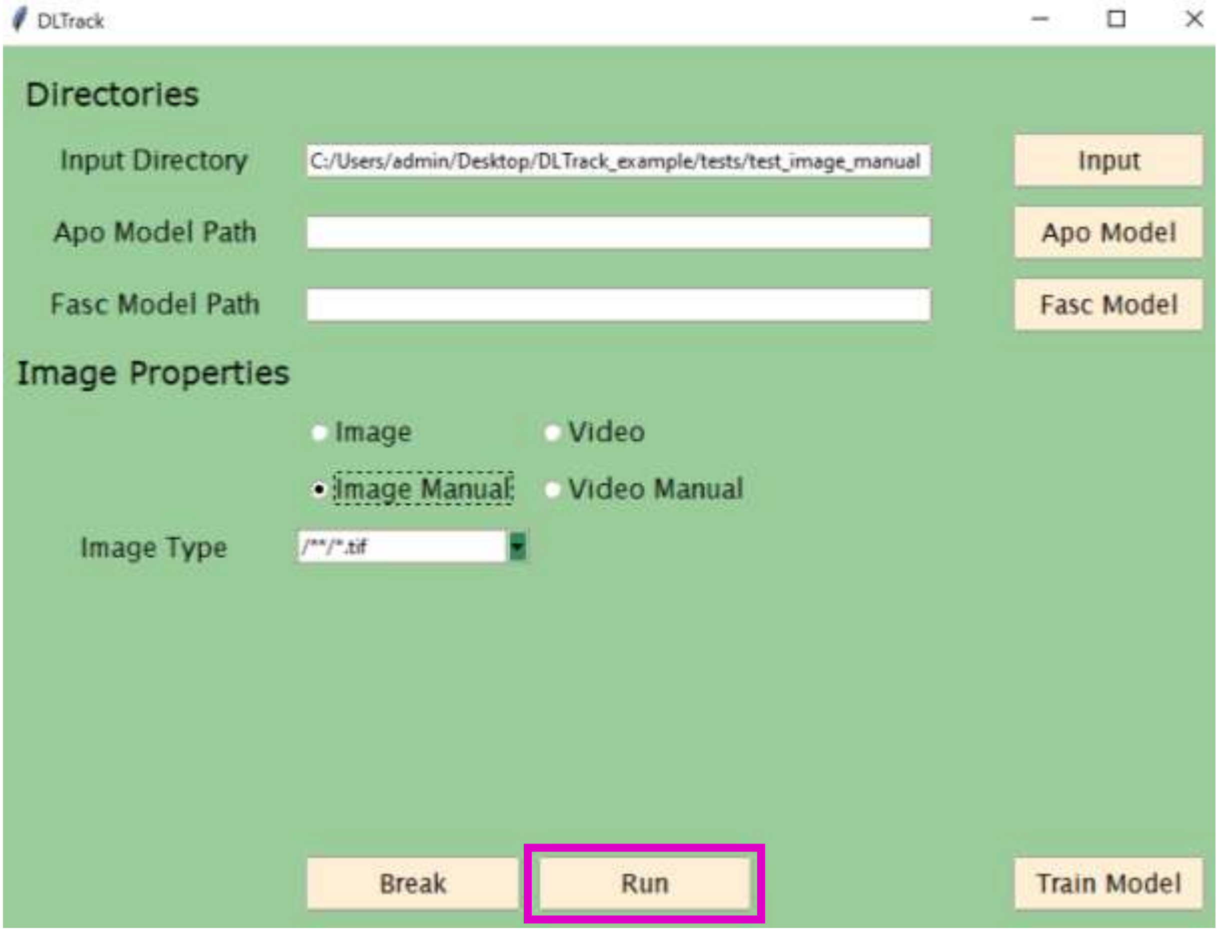


	File	Fasicle Length	Pennation Angle	Midthick
0	1	556.9683982	17.04548569	176.4090028
1	2	800.2107552	15.84797858	228.6909134
2	3	554.82972	15.4465415	125.214614
3	4	566.2983168	20.02228481	203.7932944
4	5	898.1666686	16.86291364	293.6991538

Manual Image / Video Analysis Test

The DL_Track python software package offers several different analysis types for analysis of human lower limb longitudinal ultrasonography images. The next analysis type you are going to test is the manual image / video analysis. The images are evaluated by manually segmenting the architectural parameters directly on the image. For this test, single images (not videos) are a prerequisite. The test image you must use for this test is located in the "DL_Track_examples/tests/test_images_manual" folder. In the next few pages, we will look at every required step to successfully perform manual image / video analysis testing. No worries, we do not talk about a new analysis type you haven't heard of. The analysis types manual image analysis and manual video analysis make use of the same python class (called "ManualAnalysis" and located in the manual_tracing.py file) and therefore use the same functionalities to determine architectural parameters in single images or video frames. In our opinion, it is thus not necessary to test both analysis types.

For this test make sure that the files used and parameters specified are exactly as demonstrated below. Once you made sure to use the right image ("DL_Track_example/tests/test_image_manual"), click the **Run** button to start the analysis.

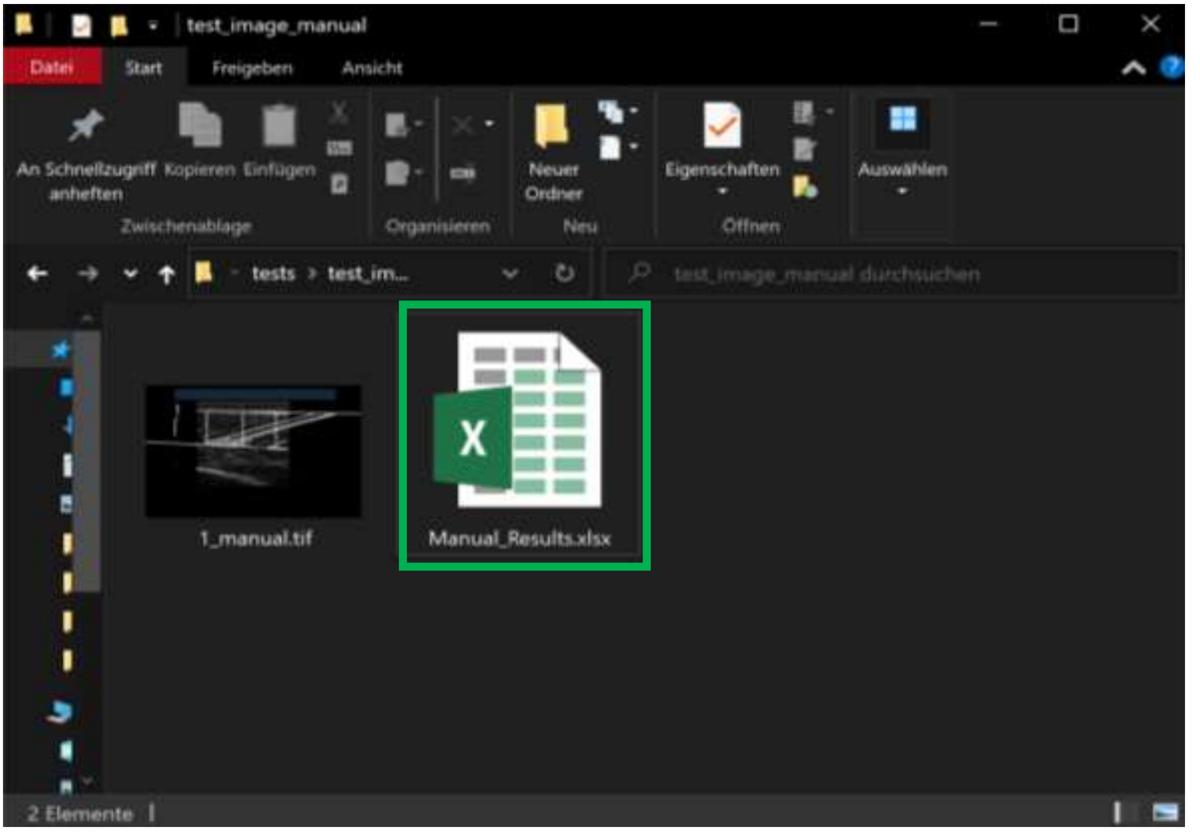


Take a look at the next page to see how to continue.

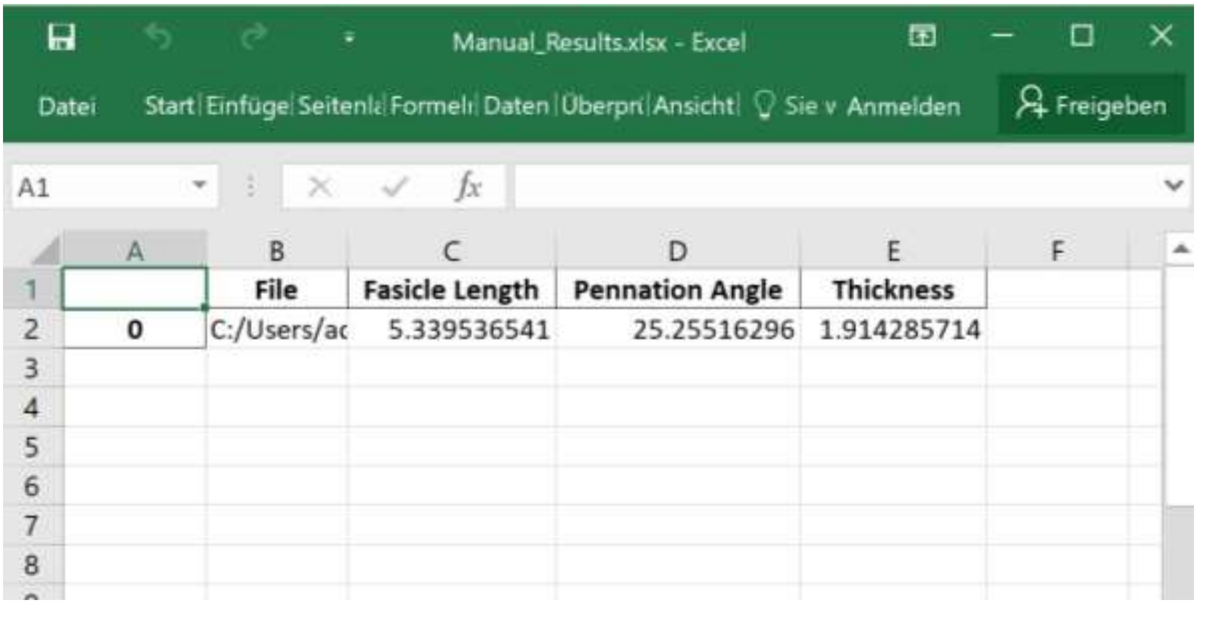
The “Manual Analysis window” should pop up containing the image as demonstrated below.

For testing the DL_Track manual image / video analysis, simply reanalyse the drawn lines. First, scale the image by following the one centimetre long scaling line in the left of the image. Then, redraw the superficial and deep aponeurosis extension lines. Subsequently, re-analyse the three vertical muscle thickness lines using one segment each, the three diagonal fascicle lines using three segments each and the three pennation angles using two segments each. Always choose the Radiobutton corresponding to the parameter you are analysing. Once you have re-analysed all the lines image, click on the Save Results button to save your analysis results.

One new file was created in the "DL_Track_example/tests/test_image_manual", the **Manual_Results.xlsx** file.



Open the **Manual_Results.xlsx** file and compare the analysis results to the ones demonstrated below. If the results are similar, the DL_Track package works properly for manual image / video analysis! Awesome!

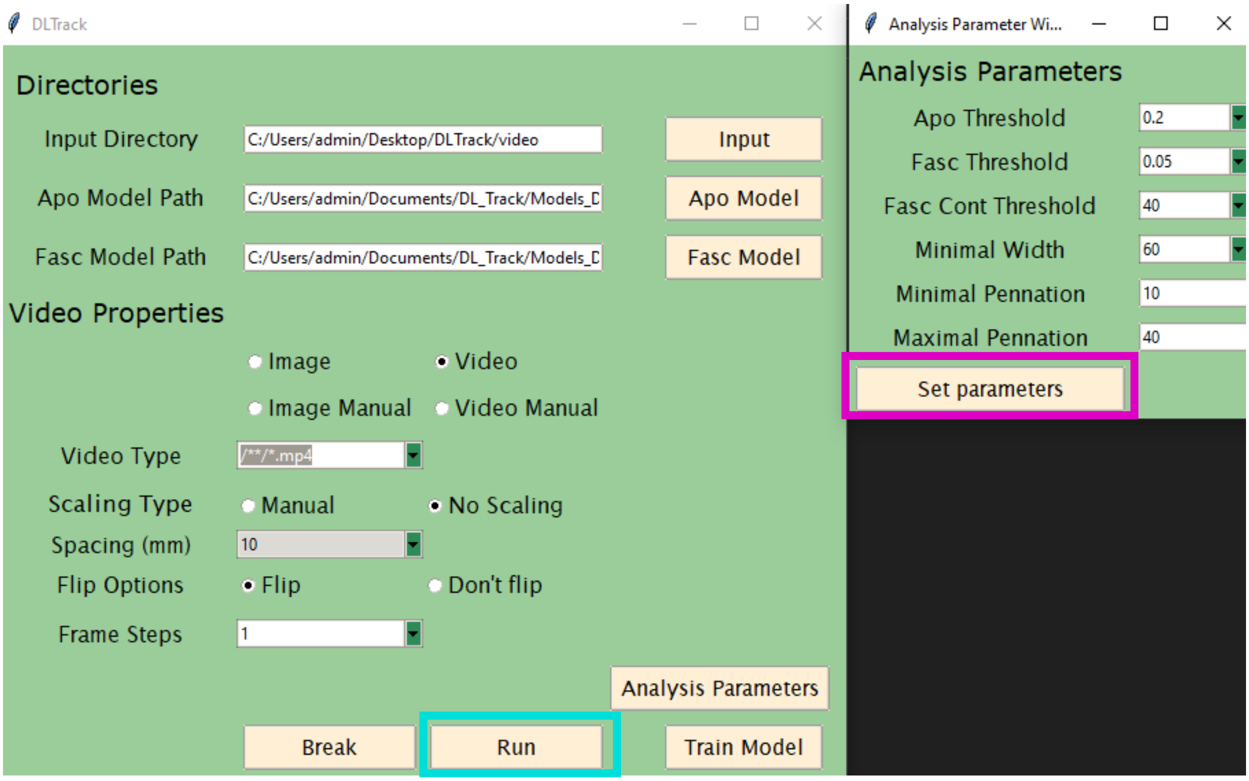


	File	Fasicle Length	Pennation Angle	Thickness
0	C:/Users/ac	5.339536541	25.25516296	1.914285714

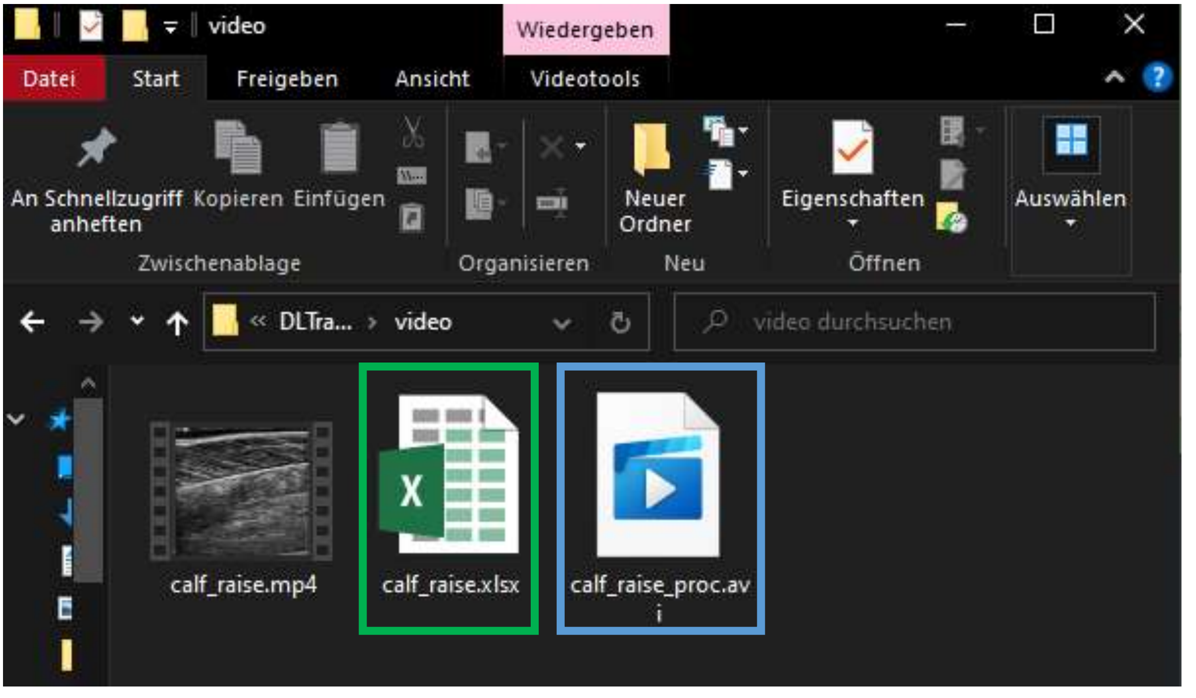
Automated Video Analysis Test

The DL_Track python software package offers several different analysis types for analysis of human lower limb longitudinal ultrasonography images. The next analysis type you are going to test is the automated video analysis. Single video frames are evaluated automatically without user input. For this test, videos are a prerequisite. The test video you must use for this test is located in the “DL_Track_examples/tests/test_video_automated” folder. In the next few pages, we will look at every required step to successfully perform automated video analysis testing.

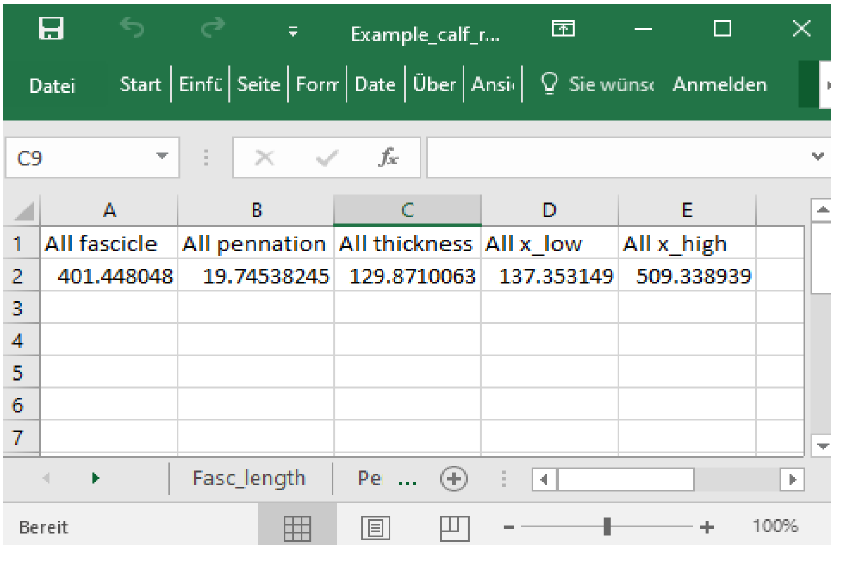
For this test make sure that the files used and parameters specified are exactly as demonstrated below. Moreover, keep the pre-specified parameter settings in the "Analysis Parameter window" as they are. Once everything looks like illustrated below and you made sure to use the right video ("DL_Track_example/tests/test_video_automated") and the provided pre-trained model files (model-apo-VGG16-BCE-512.h5 & model-fasc-VGG16-BCE-512.h5), click the **Select parameters** button to set the analysis parameters and then the **Run** button to start the analysis.



When the analysis is complete, two new files were created in the "DL_Track_example/tests/test_video_automated" folder. The calf_raise_proc.avi file and the calf_raise.xlsx file. You can disregard the calf_raise_proc.avi for this test.



Open the calf_raise.xlsx file. Take the average value from all calculated fascicle length values in all frames, all calculated pennation angles in all frames, all calculated muscle thickness values in all frames and all calculated upper (x_high) and lower (x_low) aponeuroses edge coordinates in all frames. If the results are similar to those demonstrated below, the DL_Track package works properly for automated images analysis! Wohoo!

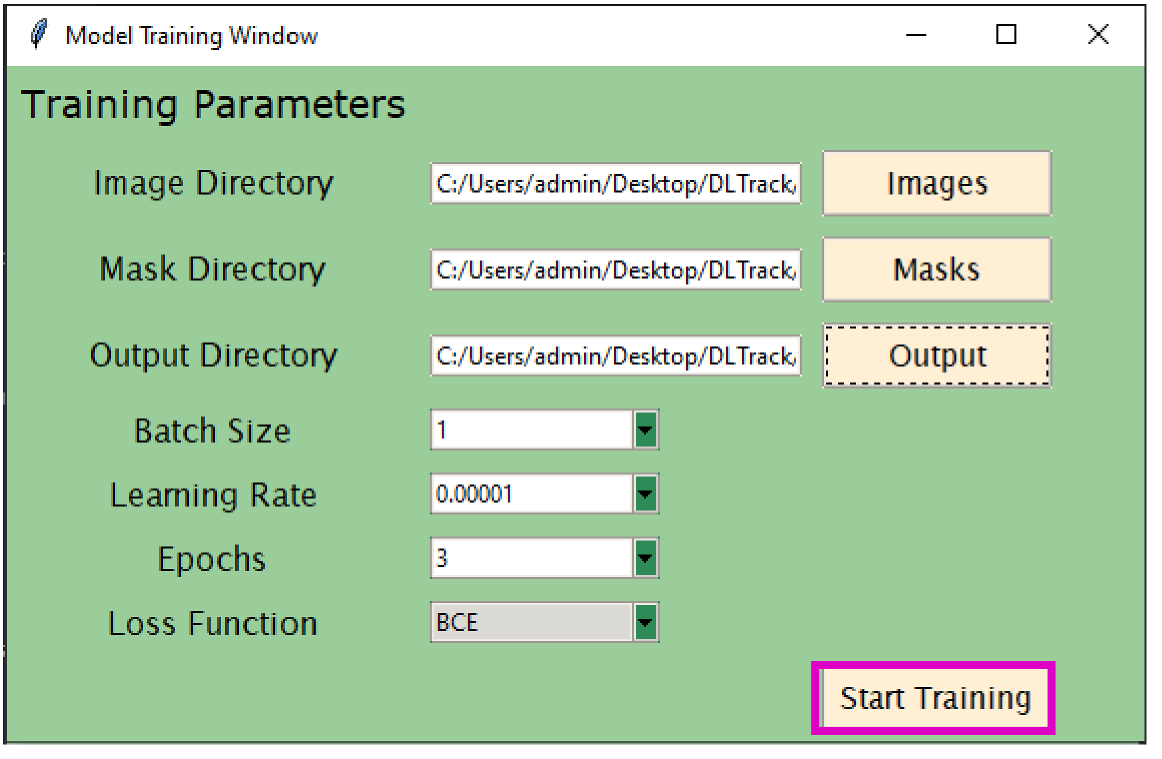


	A	B	C	D	E
1	All fascicle	All pennation	All thickness	All x_low	All x_high
2	401.448048	19.74538245	129.8710063	137.353149	509.338939

Model Training Test

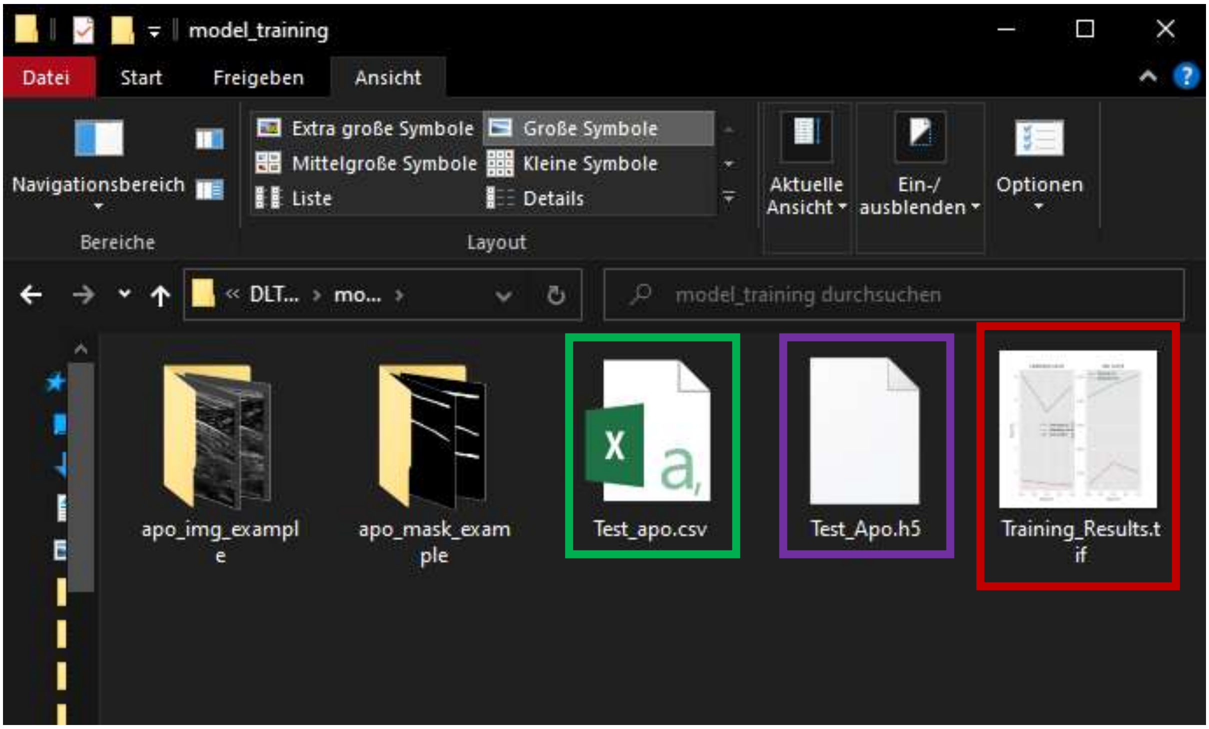
Not only does the DL_Track python software package offer four different analysis types for musculoskeletal ultrasonography images and videos of human lower limb muscles. The package also includes the possibility to train your own neural networks that may be better suited for images with different characteristics than those in our training data. This is also embedded in the GUI. It is advantageous for model training testing to have a working GPU setup, otherwise model training takes much longer. How to setup you GUI for DL_Track is described in the installation guidelines of our Github repository. In the next few pages, you will test the neural network training functionality of DL_Track. The test training images and masks you must use for this test are located at “DL_Track_example/tests/model_training” folder. This is the last part of the test instructions. After completion of this chapter, you will have tested DL_Track thoroughly!

For this test make sure that the files used and parameters specified are exactly as demonstrated below. Since you will only make use of the "Model Training window" you can disregard the main GUI. Everything specified there is irrelevant for model training. Keep the pre-specified parameter settings in the "Model Training window" as they are shown below. Especially make sure that the number of Epochs is 3 (otherwise training for test purposes takes to long) Once everything looks like illustrated below and you made sure to use the right training images and masks ("DL_Track_example/tests/model_training/apo_img_example" & "DL_Track_example/tests/model_training/apo_mask_example") click the **Start Training** button to start the training process.



Several messageboxes will appear during the training process. Always click "OK". The messageboxes simply tell you that the images and masks have successfully been loaded, the model was successfully compiled and that the analysis was successfully completed.

When the analysis is complete, three new files were created in the "DL_Track_example/tests/model_training" folder. The **Test_apo.xlsx**, the **Test_apo.h5** file and the **Training_results.tif** file.



Since each training process results in slightly different models, we cannot directly compare your results to ours. However, if the three files were created in the "DL_Track_example/tests/model_training" folder, the DL_Track package works properly model training! Yipiee!

Closing remarks

Thanks for checking out the DL_Track python package test instructions. We sincerely hope that all tests were successful and you were also able to enjoy it a bit. Moreover, we hope our instructions were clear, concise and easy to follow when you have worked through the DL_Track_turorial.pdf file before. In case something was not clearly illustrated at some point, please let us know. Don't hesitate to report this in the Q&A section in the DL_Track discussion forum. Otherwise, you can contact us by email at paul.ritsche@unibas.ch, but we would prefer the other way.