DL_Track_US v0.1.2

Preface

Welcome to the DL_Track_US python package tutorial. In the next roughly 80 pages, you will learn how to automatically and manually analyse ultrasonography images and videos of human lower limb muscles. You will do so by making extensive use of the graphical user interface provided by the in the DL_Track_US package. Moreover, you will learn how to train your own neural networks using the graphical user interface as well. Have fun!

Content

Good To Know…

All relevant instructions and guidelines for the installation of the DL_Track_US software package are described in our documentation, so please take a look there if anything is unclear. We have also provided information on what to do when you encounter problems during the installation process, encounter errors during the analysis process that are not caught by the GUI (no error message pop ups and advises you what to do), if you want to contribute to the DL_Track_US software package, and how you can reach us.

Before we start with this tutorial, here are some important tips:

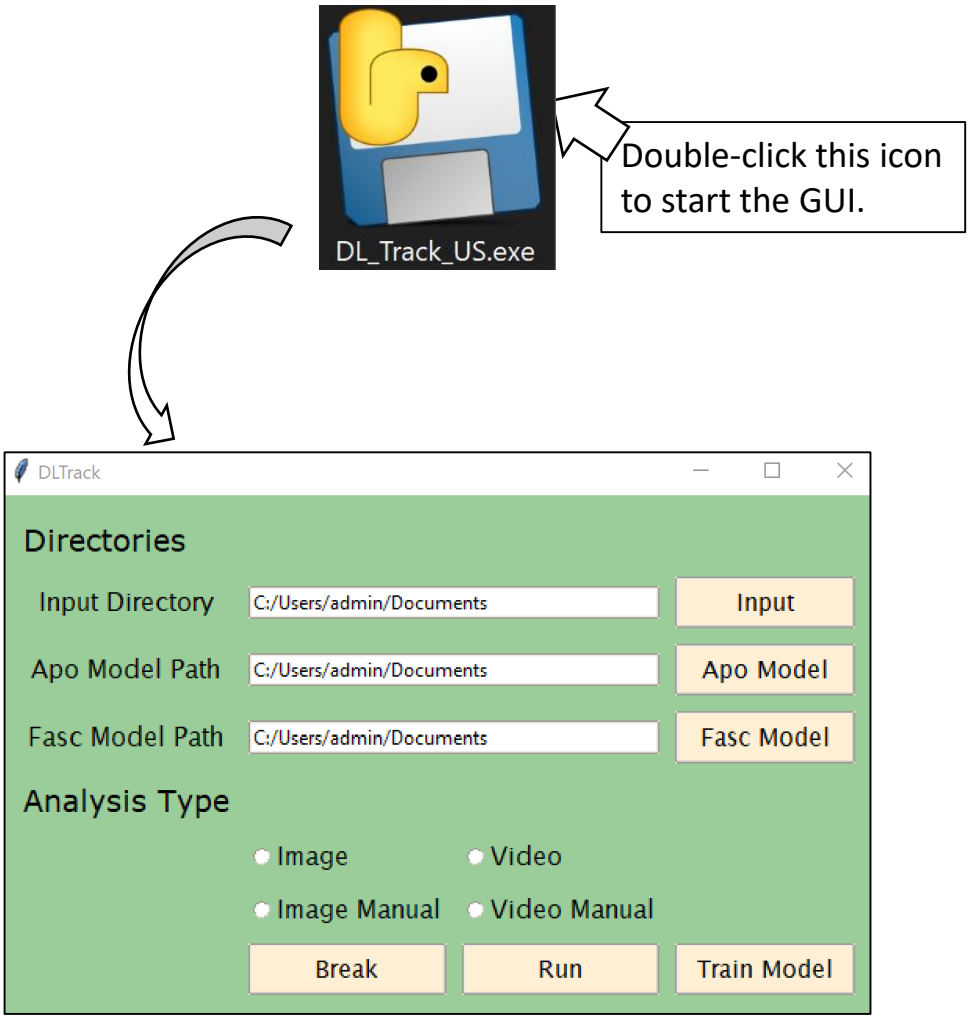
- In case you plan an analysis on images taken from different muscles, we strongly advise to test the algorithm first and in case of bad performance, train your own models. We have provided extensive documentation on how to do so in the section “Training your own network” in this tutorial.
- Although we used extensive data augmentation during the model training process, we must caution you about the generalizability of our models. Deep learning is no magic! Even though our model demonstrated good performance on unseen images during testing, we cannot confidently claim that they will work fine on all lower limb ultrasonography images. It is possible that even for images of muscles represented in our training data set, different device types, different muscle regions and even different settings of ultrasonography devices during image acquisition might offset model performance.
- Quality matters! Please pay attention that the images you want to analyse with DL_Track_US are of high quality. High quality means good image contrast, appropriate image brightness, clearly visible fascicles and aponeurosis and clear alignment of the probe with the fascicle plain. If the quality of the images you want to analyse is bad, the results will be as well.
- Bad model performance can be detected. The first and easiest step to take is to visually inspect the output of the models. If the segmentation results and the actual fascicles and aponeuroses overlap on most of the analysed images, model performance is good. If not, adapt the analysis parameters (how to do so is covered in the tutorials) or train a separate model. Secondly, you should manually analyse a few of your images and compare the model results to your manual results. If both results are similar, model performance is good. If not, adapt the analysis parameters (how to do so is covered in the tutorials) or train a separate model.
- Lastly, we advise to follow the provided testing procedures in the DL_Track_US/tests folder. This ensures that the DL_Track_US package is working properly on you computer.

Starting the Graphical User Interface

In the very first step of this tutorial, we will take a look at how to start the graphical user interface (GUI) once it was installed. We have provided two different installation procedures: 1. downloading the DL_Track_US_GUI.exe file and 2. installing the DL_Track_US python package using pip, Github and Pypi.

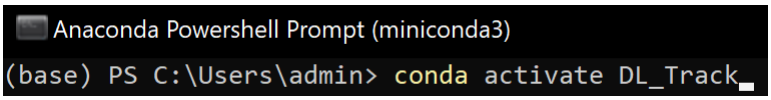
Let's begin with 1., how to start the GUI when you downloaded the DL_Track_US_GUI.exe:

- It doesn't get any easier than this. Navigate to the downloads folder and place the DL_Track_US_GUI.exe file somewhere you can easily find it again. Done so, you just have to double click the DL_Track_US_GUI.exe file with your left mouse button to start the GUI. Once you've done that, the GUI should open and you are ready to start an analysis.

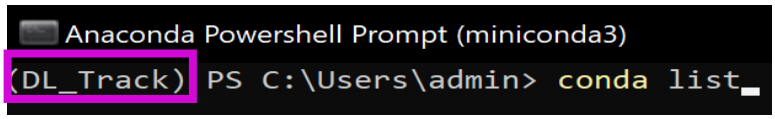


Now to 2., how to start the GUI when you installed the DL_Track_US python package via Pip, Github and Pypi:

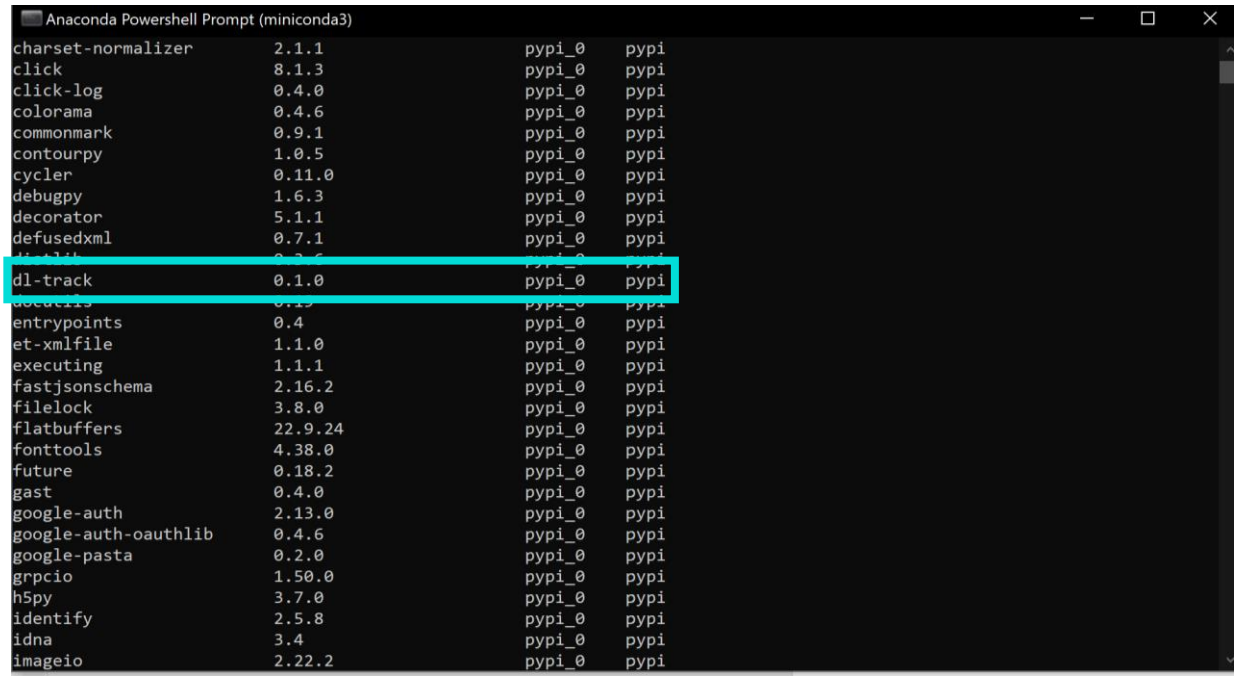
There are essentially two ways you can start the GUI. But first lets make sure that the package was correctly installed. The package should be automatically installed when you create the conda virtual environment. Therefore activate the environment by typing “conda activate DL_Track_US”:



You should see the **activated environment** now in the left round brackets. Next type “conda list” to see all packages installed in the DL_Track_US environment:

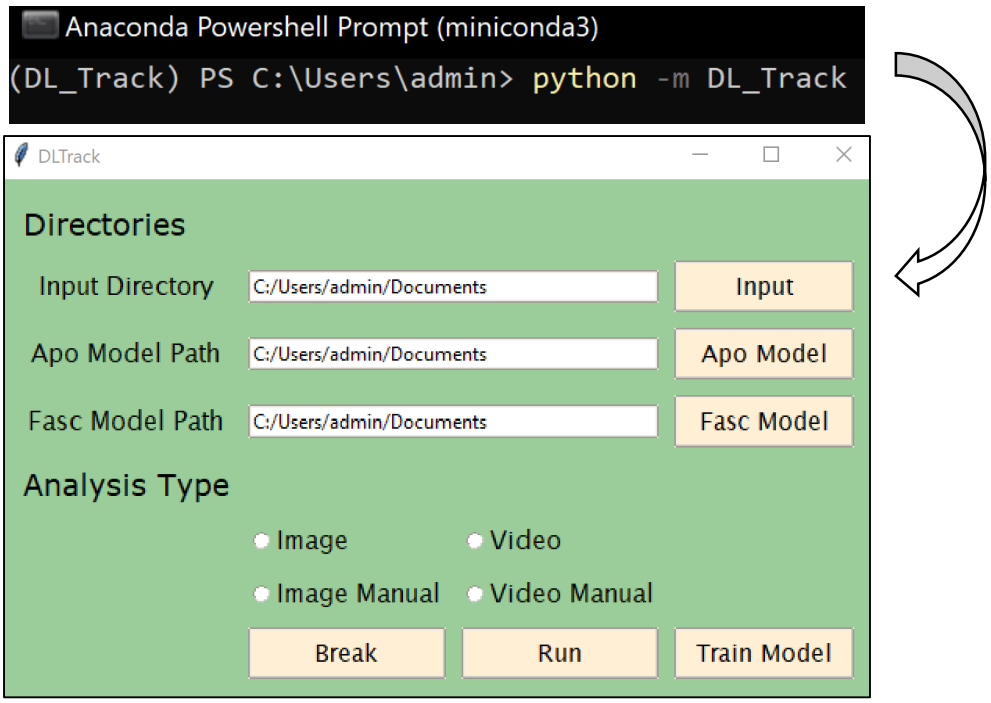


When you pressed enter, all packages installed in the environment should be listed. When **DL_Track_US** is included, you are good to go.



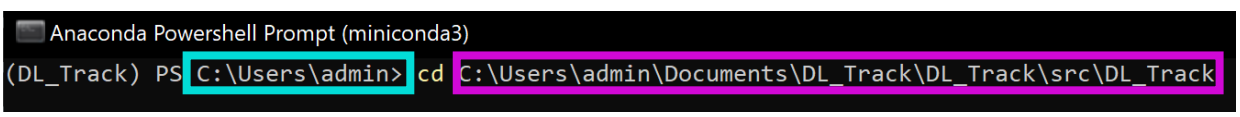
In case the DL_Track_US package is not installed in your environment, please take a look at the installation guidelines again. If you still encounter a problem, ask a question in the [Q&A discussion section of DL_Track_US] on Github and add the Label “Problem”.

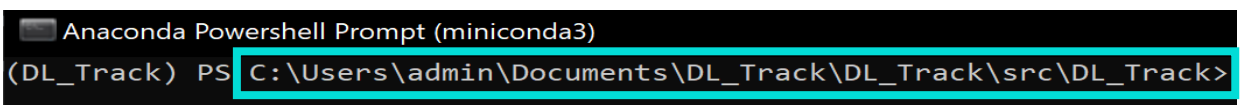
Alright, let's actually start the GUI now once you've made sure that the DL_Track_US package is included in your active environment. As mentioned, you can do this in two ways. **First option**, you can run the DL_Track_US package from the command prompt. For this, type "python –m DL_Track_US" in the command prompt. When you press enter, the GUI should open.



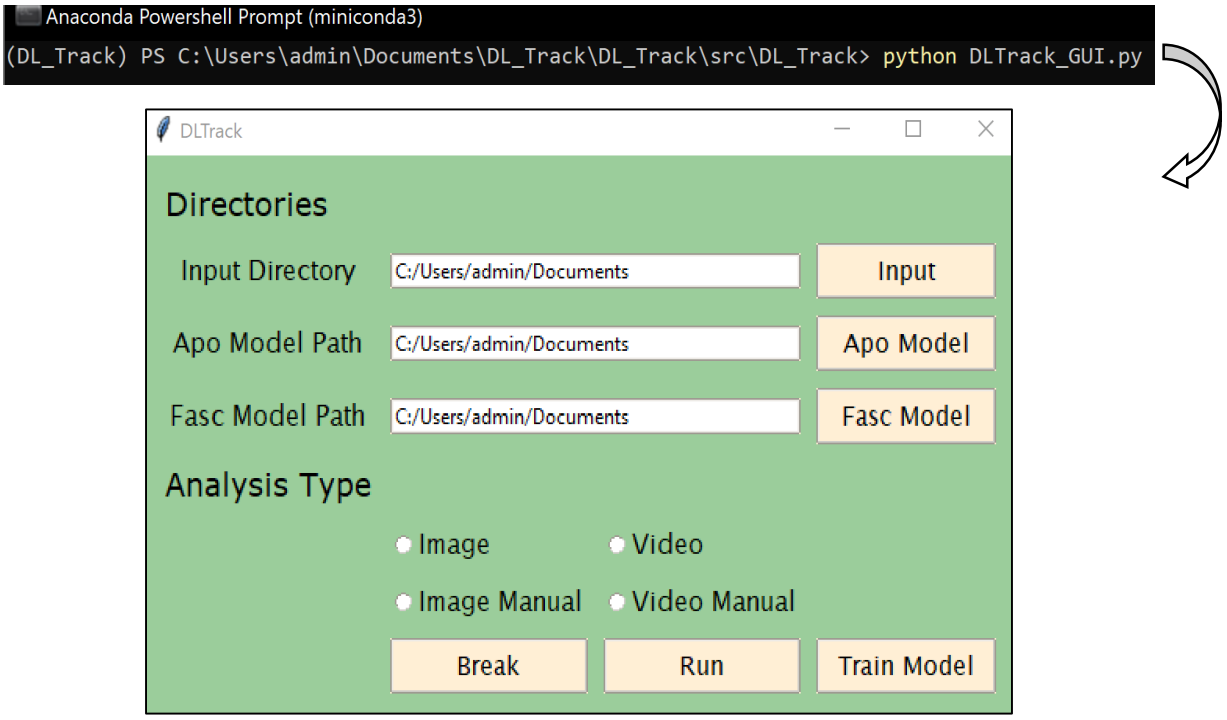
Neither to activate the environment (see previous page) nor to start the package do you need to be in a specific folder. The only prerequisite for this method is an active environment containing all dependencies for as well as the DL_Track_US package itself.

Second option, you run the python file containing the main GUI manually. Therefore you need all the source files from the Github repository. Please take a look at the installation guidelines on how to get them. It is required that the environment containing all dependencies for as well as the DL_Track_US package itself is active (see previous page) and that you are in the folder containing the DL_Track_US_GUI.py module. To navigate to the folder type "cd **yourpathtofile**" in the command prompt. By pressing enter, the path is changed to the folder containing the DL_Track_US_GUI.py file.





To start the GUI, type “python DL_Track_US_GUI.py” in the command prompt once you have activated the environment containing the DL_Track_US package and all its dependencies and navigated to the folder that contains the DL_Track_US_GUI.py file. By pressing enter, the GUI should start.



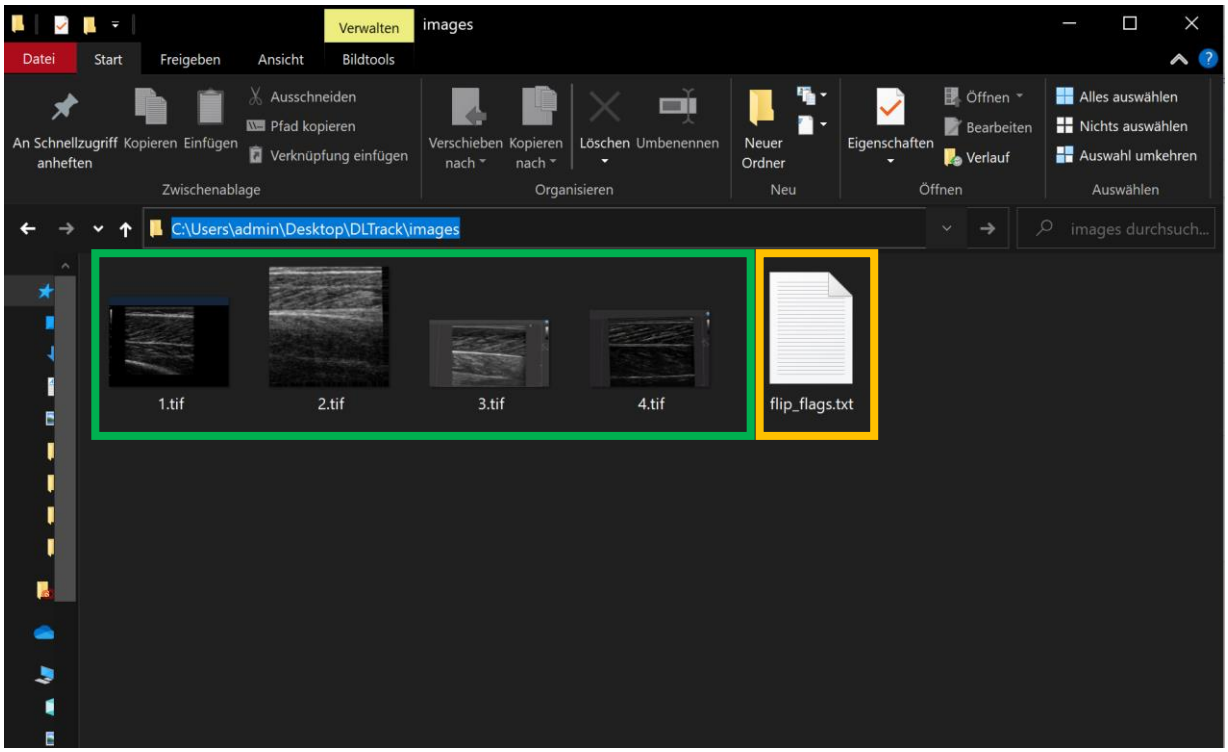
Congrats, you now know all the ways how to start the DL_Track_US GUI! Which way you use to start the GUI might be dependent on your usage intentions. If you just want to use the GUI and analyse images / videos, it is completely enough to download the DL_Track_US_GUI.exe and start right away. This is definitely the easiest way to start the GUI. However, if you want to customize the code and maybe even contribute to further releases, you are required to download all the source files in order to be able to change anything. To directly see your implemented changes use the last option explained below to start the GUI.

Automated Image Analysis

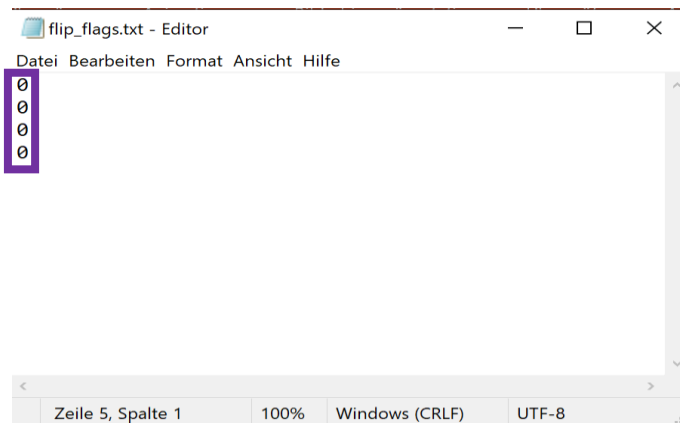
The DL_Track_US python software package offers several different analysis types for analysis of human lower limb longitudinal ultrasonography images. The first analysis type this tutorial covers is the automated image analysis. The images are evaluated without user input and may be scaled. Scaling the images will ensure estimated muscle architectural parameters are converted to centimetre units. For this type of analysis, single images (not videos) are a prerequisite. These images should be contained in a single folder, like in the "DL_Track_US_example/images" folder. If you haven't downloaded this folder, please do so now (link: DL_Track_US - Examples & Models | Zenodo). Unzip the folder and put it somewhere accessible, for example on your desktop. We will make use of the included example files extensively during this tutorial. Moreover, a FlipFlag.txt file is needed. Such a file is included in the "DL_Track_US_example/images" folder as well. In the next few pages, we will look at every required step to successfully perform automated image analysis with DL_Track_US.

1. Creating Image Directory & FlipFlag.txt File

In order for DL_Track_US to recognize your images, they should best be in a single folder (though one subfolder structure is acceptable as well). Take a look how you might structure this:

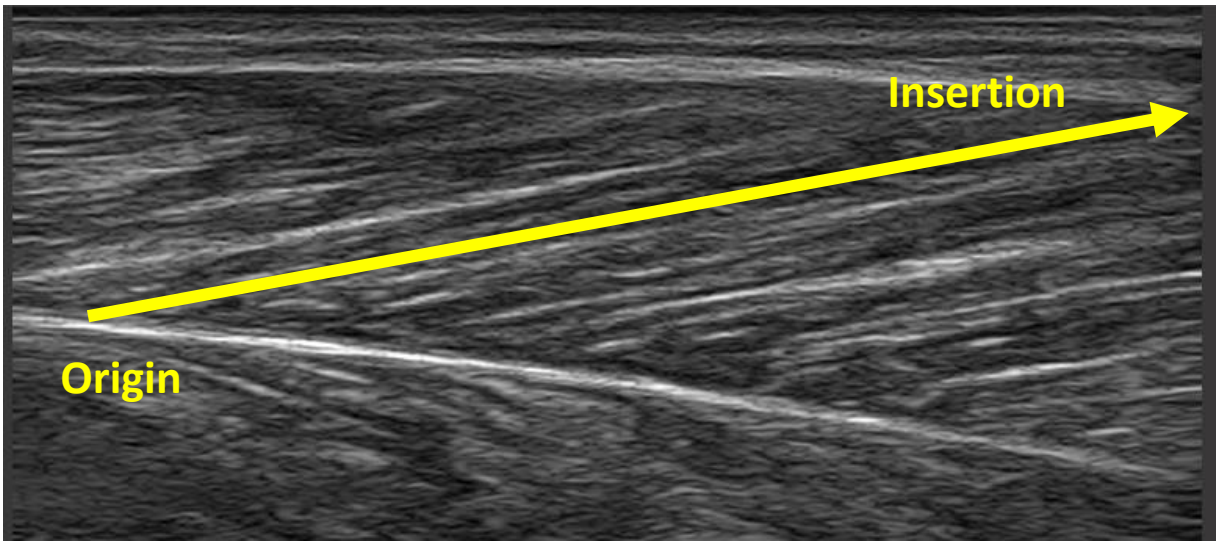


You can see in the picture above that the folder contains **5 images**, a **flip_flag.txt** file and is **located on the desktop**. This structure is already included in the "DL_Track_US_example" folder. It is not required to have the **flip_flag.txt** file in the same folder as the **images** to be analysed, but it is convenient.
Lets take a closer look at the **flip_flags.txt** file and how this should look like in order for the analysis to work. Below you can see an the **flip_flag.txt** file in the directory.



In the flip_flag.txt file you can see the respective **flip-flag** for each image. For every image there must be a **flip-flag**. If the number of **flip-flags** and **images** doesn't match, an error is raised. Another possible way to specify is displayed below. This is relevant when multiple subfolders are included, as each line then represents a subfolder.

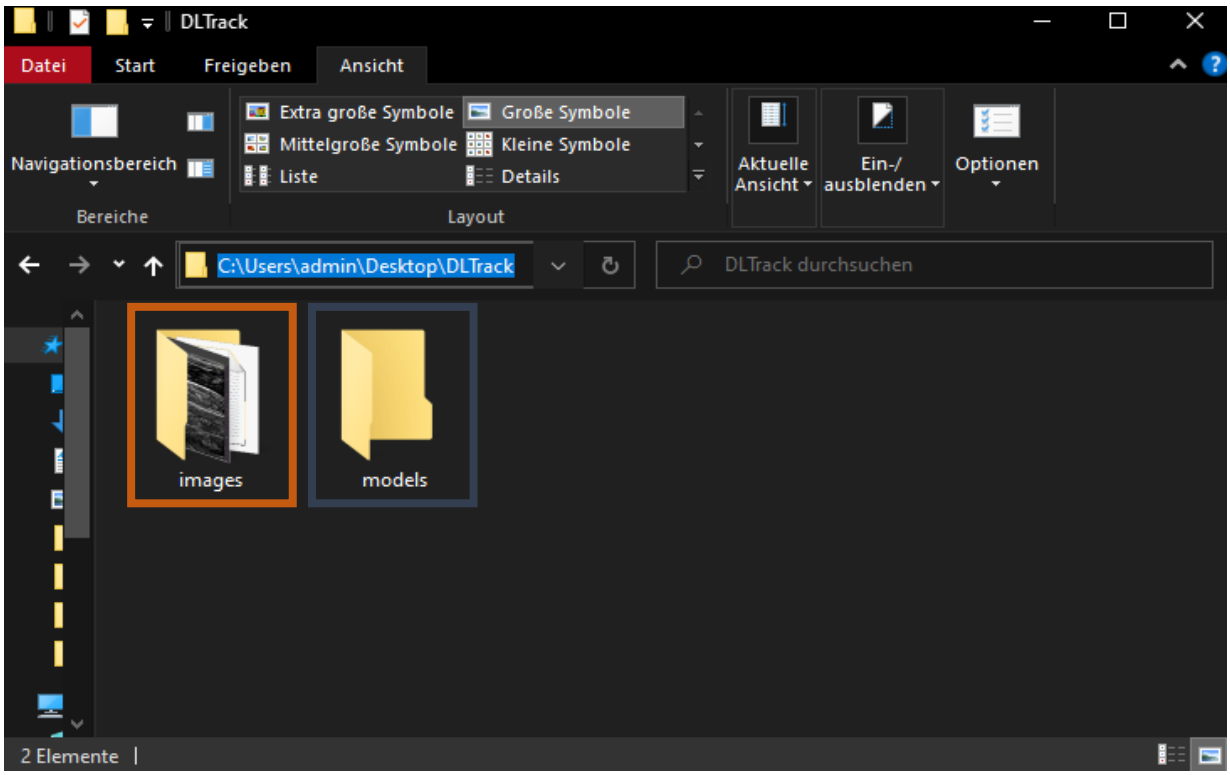
The **flip-flag** determines if an **image** is flipped during analysis or not. A "0" stands for no flipping, whereas "1" means please flip the image. The **flip-flags** enable that each **image** can be automatically flipped or not. None of the example **images** must be flipped. Their fascicle orientation is correct, with fascicles originating at the bottom left and inserting on the top right. Here is a visual representation of a **correct** fascicle orientation. If the fascicles in your **images** are orientated differently, please specify a "1" as a **flip-flag** for those **images**.



Once you have created a directory containing a flip_flag.txt file with a **flip-flag** for each **image** in the directory that containes “0” for **images** with correct orientation and “1” for incorrectly orientated **images**, we can get to the analysis part. (You actually do not need to anything at this point, because you can use the example **images** folder “DL_Track_US_examples/images” with it’s contained images and **flip_flag.txt** file.)

2. Creating Neural Network Directories

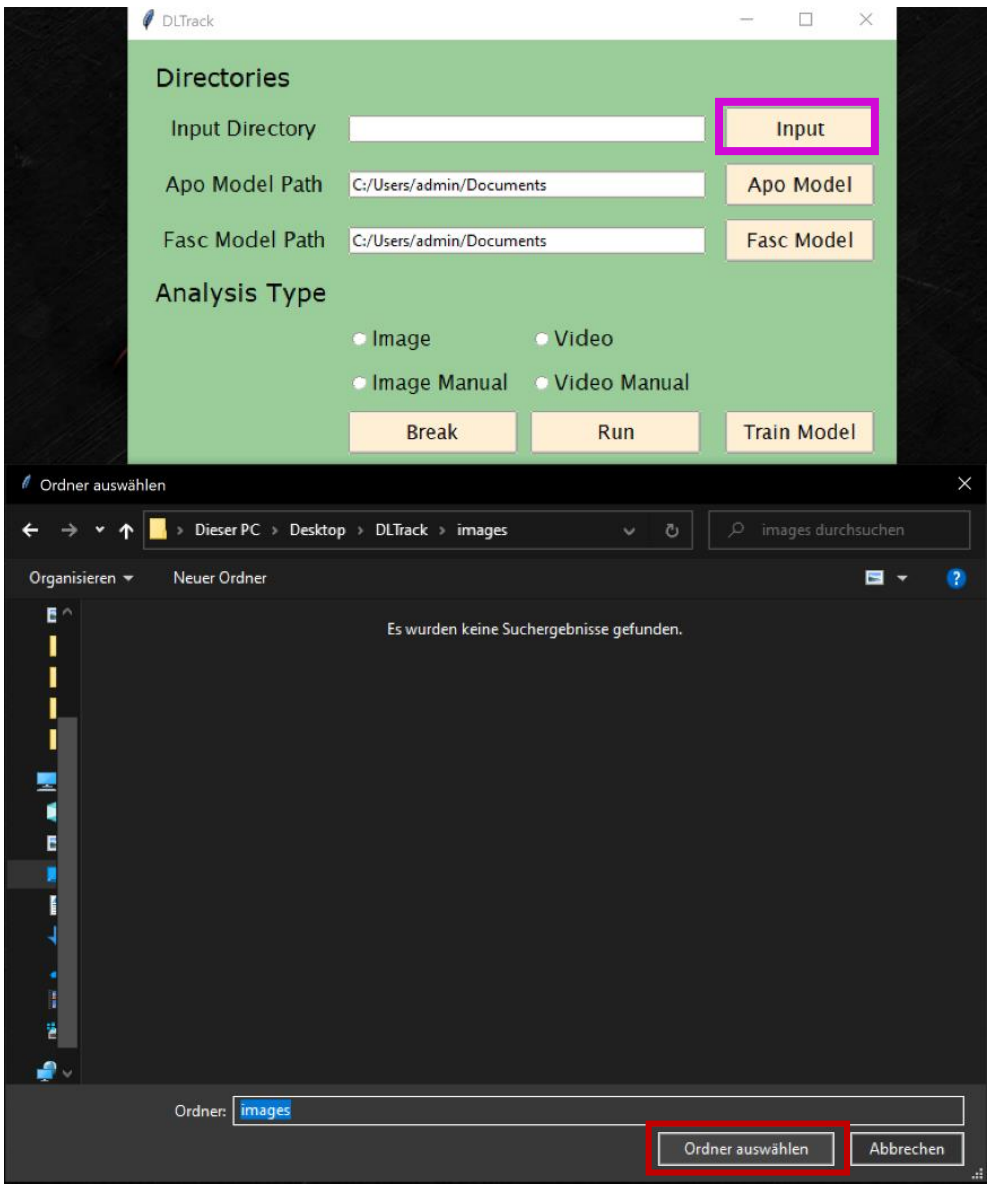
Before we start with the analysis, we need to create a directory for the pre-trained neural networks. The **folder containing the images** (in this case the “DL_Track_US_examples/images” folder) is already included in the “DL_Track_US_example” folder. We will now create a separate folder for the pre-trained **aponeurosis and fascicle neural networks**. In case you have not downloaded the models, please do so now (link: DL_Track_US - Examples & Models | Zenodo). Place them in a subfolder of the “DL_Track_US_example” folder for the moment, like “DL_Track_US_example/**models**”. You will make use of these neural networks later as well, when you analyse your own images outside of this tutorial. Of course you can move them to a different folder then. Your folder structure inside the “DL_Track_US_example” folder should now look something like this with the neural networks placed in the **models** folder.



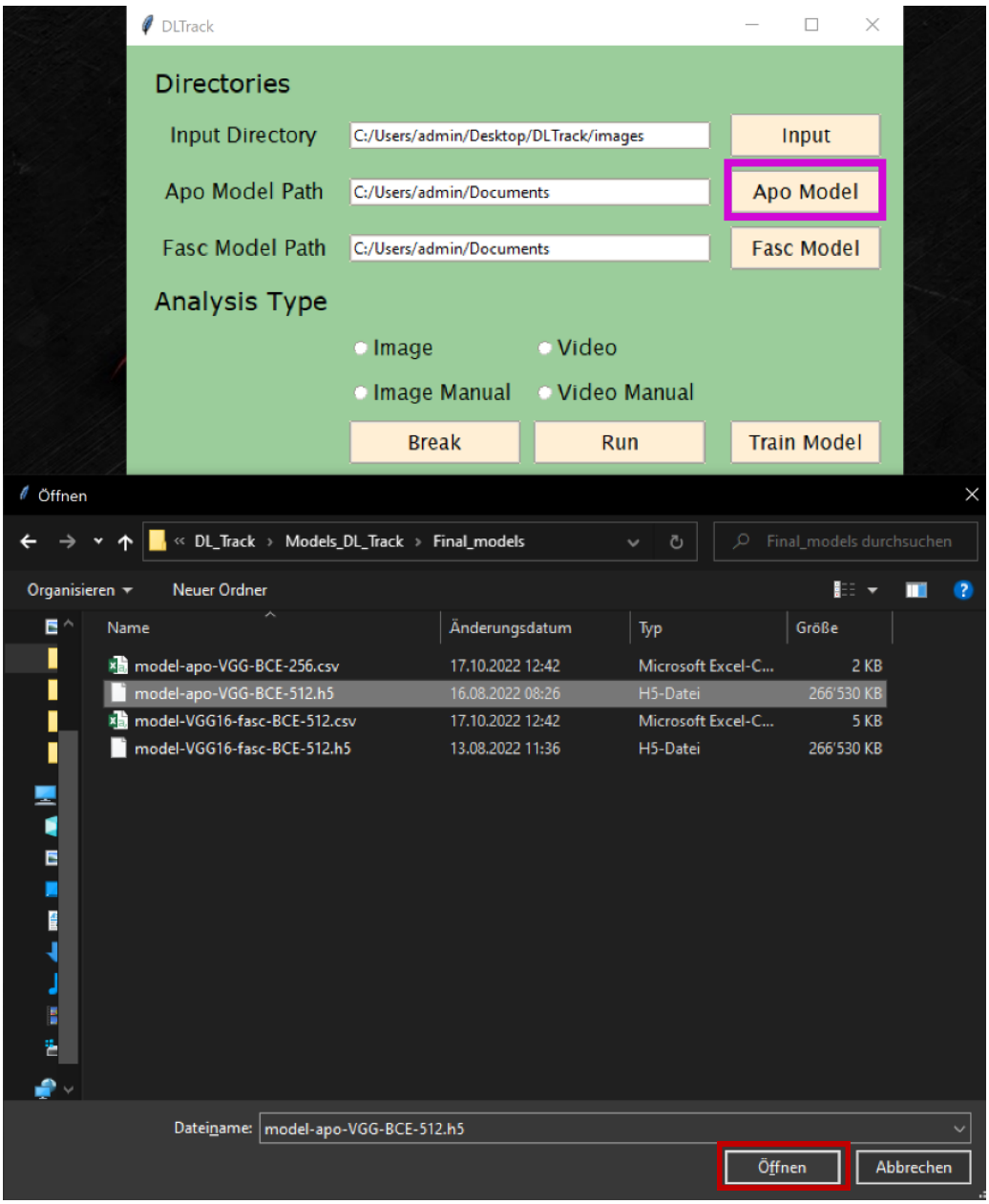
3. Specifying Input Directories in the GUI

Finally you can start with the actual analysis! The first step of every analysis type in DL_Track_US is to specify the input directories in the graphical user interface (GUI). We assume that you have already opened the GUI. (If not, take a look at the previous chapter of this document “Starting the GUI”.).

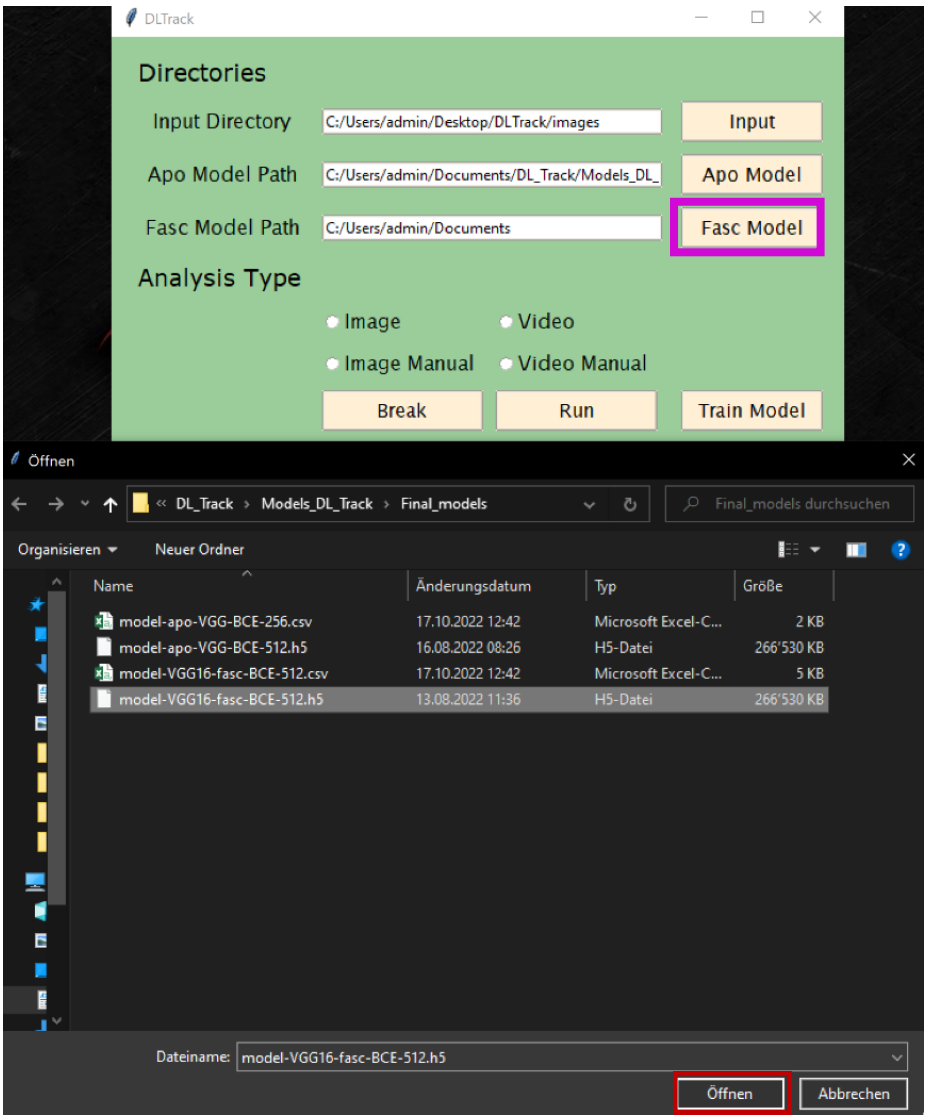
You will begin with specifying the path to the folder containing the **images** to be analysed. Remember this was the folder “DL_Track_US_example/**images**”. By clicking on the **Input** button in the GUI a selection window opens were you need to select the images folder. Click **select folder** to specify the path in the GUI.



Next, you will specify the absolute path to the **aponeurosis neural network**. Remember that you placed it in the "DL_Track_US_example/**models**". By clicking on the **Apo Model** button in the GUI a selection window opens were you need to select the **aponeurosis neural network** in the models folder. Click **open** to specify the path to the **aponeurosis neural network** in the GUI.



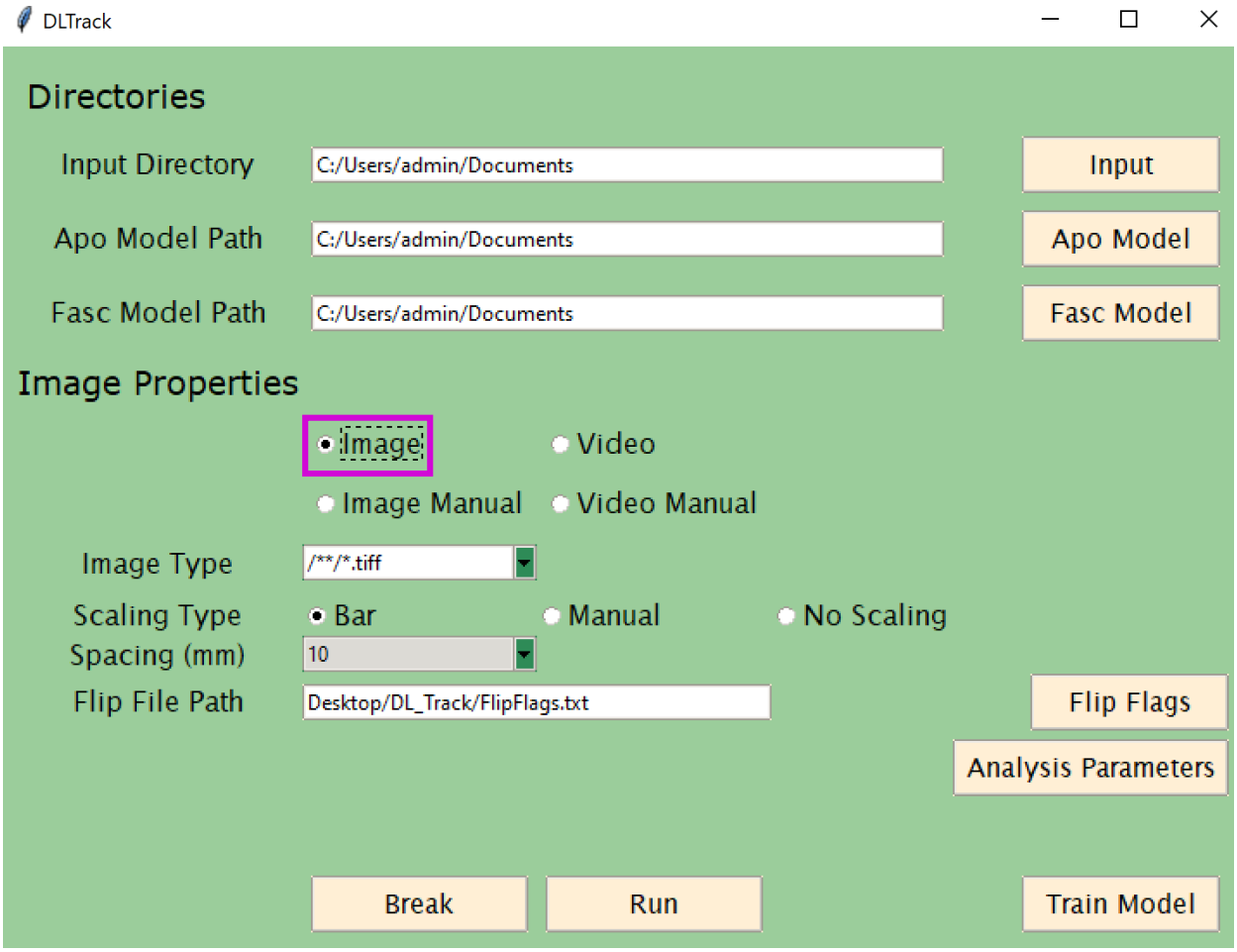
Next, you will specify the absolute path to the **fascicle neural network**. Remember that you placed it in the "DL_Track_US_example/**models**". By clicking on the **Fasc Model** button in the GUI a selection window opens were you need to select the **fascicle neural network** in the models folder. Click **open** to specify the path to the **fascicle neural network** in the GUI.



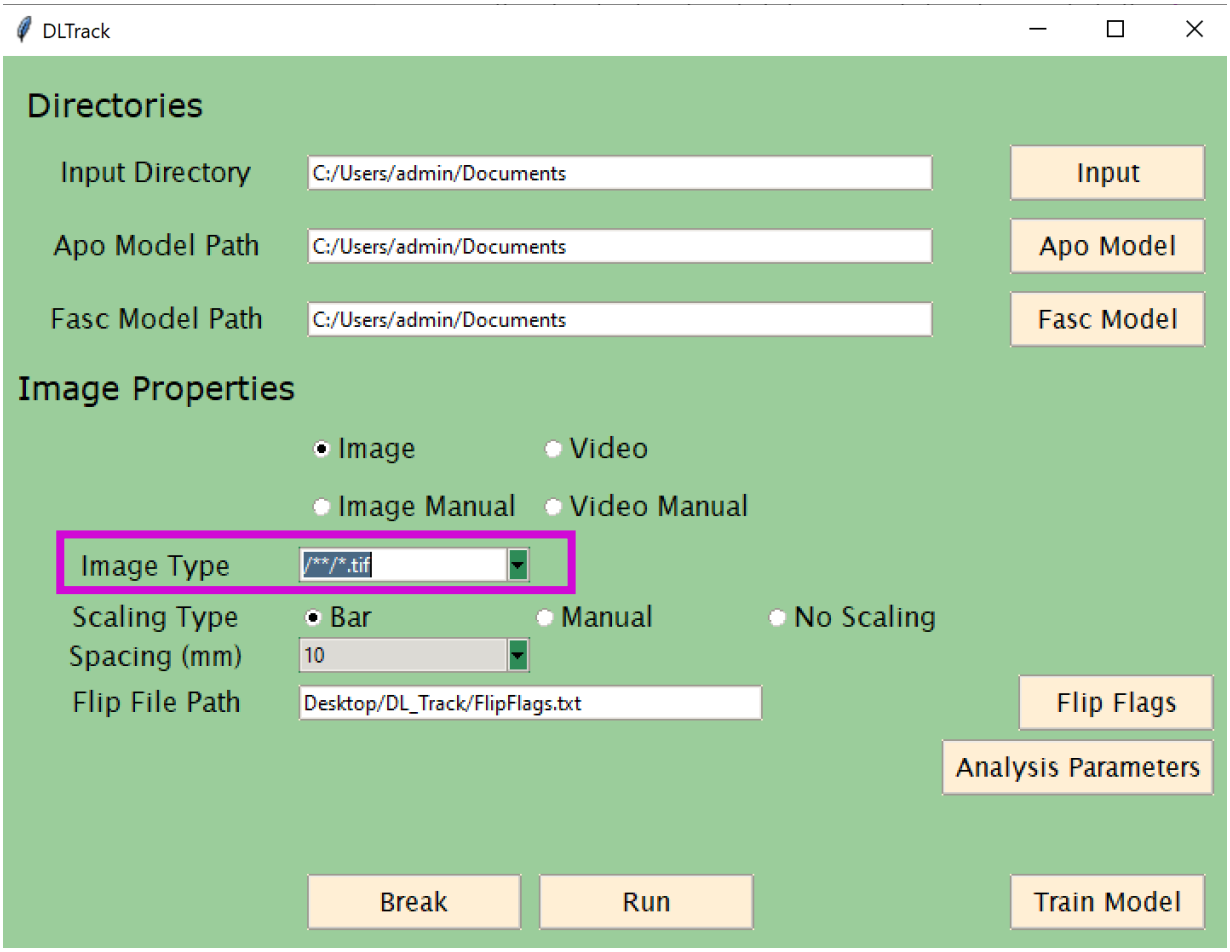
You have now successfully defined all the input directories required for automated image analyses with DL_Track_US. In the next section you will specify all relevant analysis parameters, including the analysis type. We will also explain what each parameter is used for.

4. Specifying Relevant Parameters

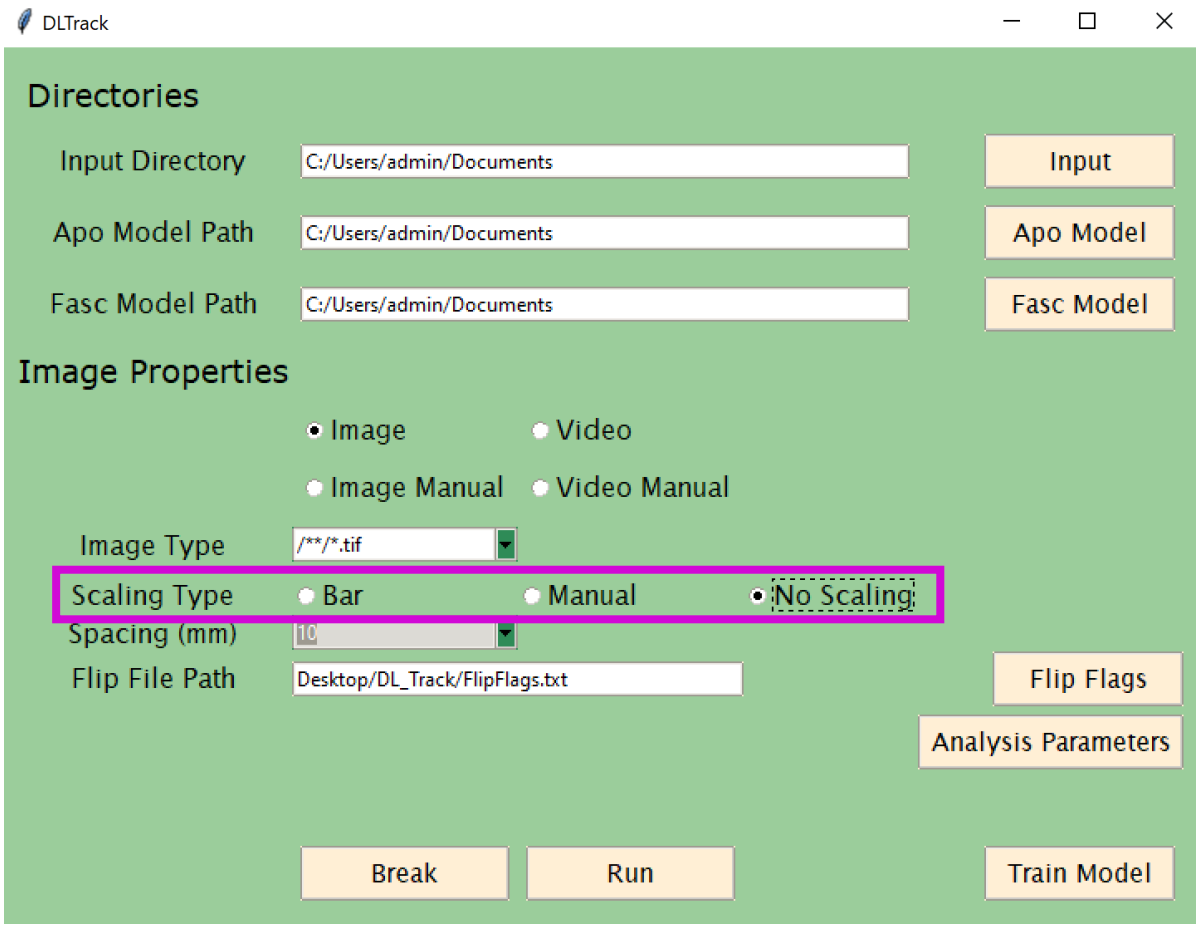
As a first step, you will select the right analysis type in the GUI. Since this section is about automated image analysis, please select the **Image** radiobutton. You can see that the GUI unfolds and several other parameters appear. You will set those in the next steps on the next page.



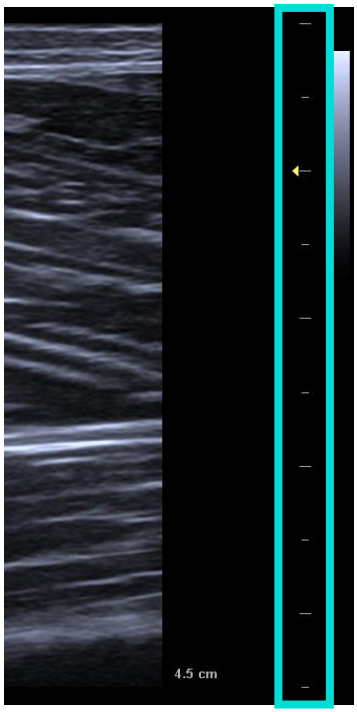
Next, you need to specify the Image Type. The ending of the **Image Type** must match the ending of your **images**, otherwise no files are found by DL_Track_US. You can either select a pre-specified ending from the dropdown list or type your own ending. Please keep the formatting similar to those **Image Types** provided in the dropdown list. All the **images** in the "DL_Track_US_example/**images**" folder are of the **Image Type** ".tif". Thus, you should select the "/**/*.tif" **Image Type**.



Subsequently, you need to specify the image **Scaling Type**. Scaling in general has the huge advantage that the resulting estimated muscle architectural features are in centimetre units rather than pixel units. There are three **Scaling Types** in the DL_Track_US package. For this tutorial however, you will select the **"No Scaling"** option as displayed below. We will explain the other two **Scaling Types** in the next pages.

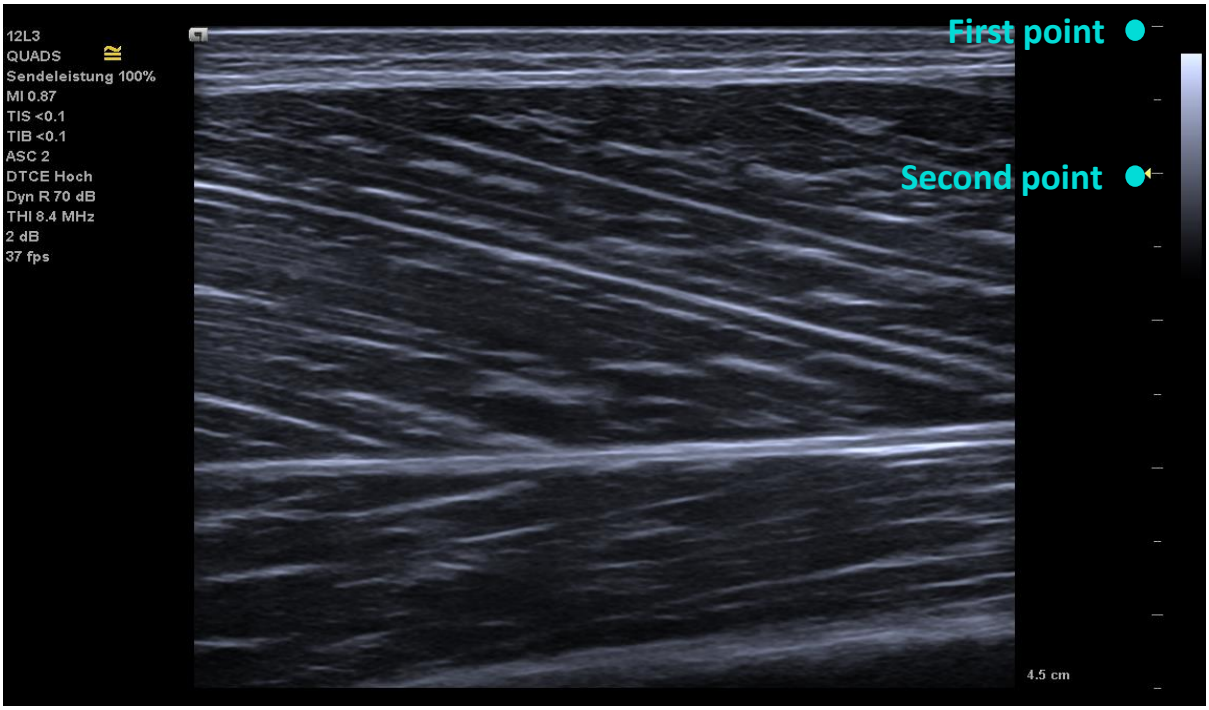


Another **Scaling Type** except **“No Scaling”** is **“Bar”.** This **Scaling Type** is only applicable if there are **scaling bars** in the right side of the ultrasonography image:



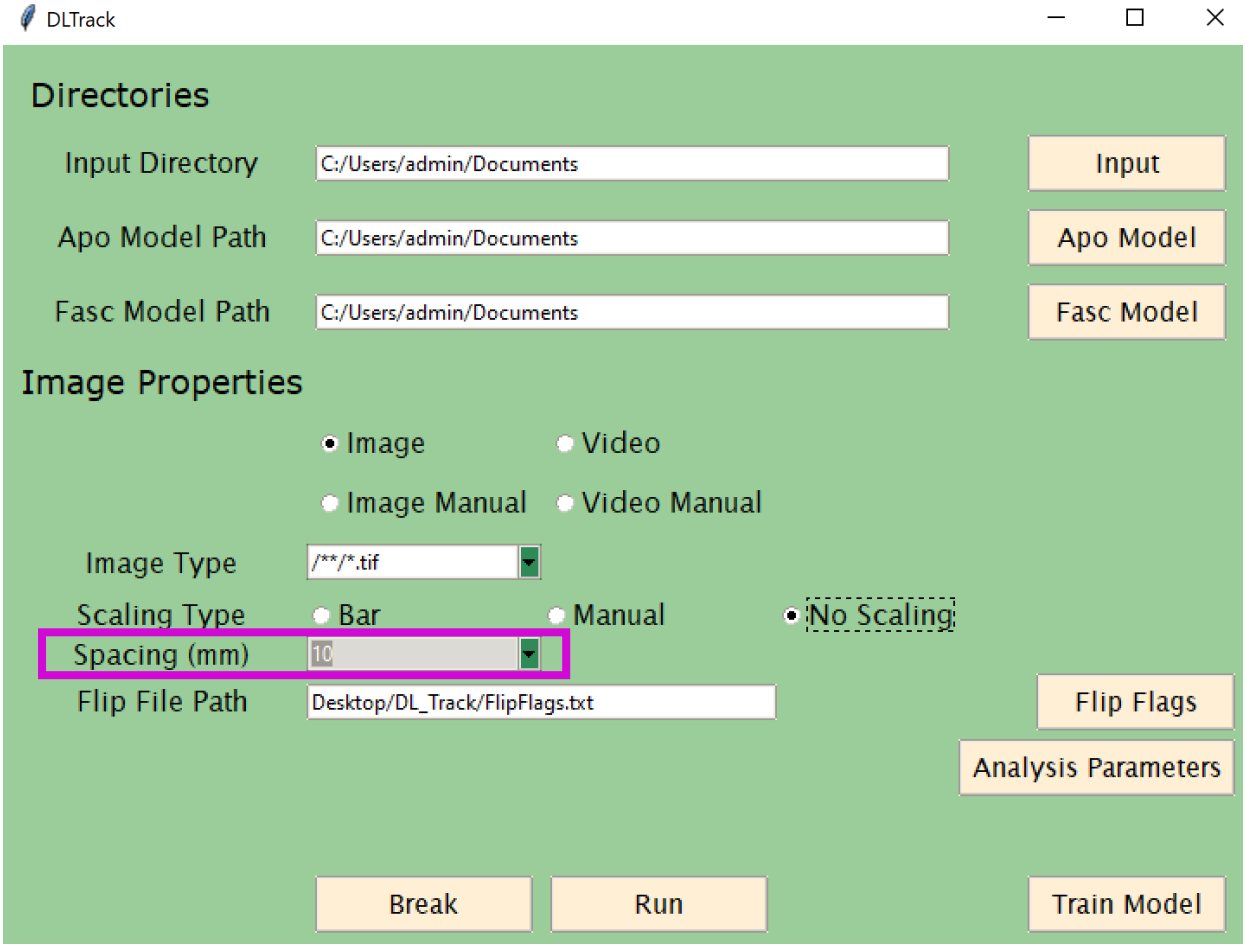
It is not important if the **scaling bars** look exactly like the ones in the above image. They just need to be **next to the image** and **clearly separated** from each other. We advise you to try this **Scaling type** on a few of your images and find out for yourself if it works. Files that cannot be analysed with this **Scaling type** will be recorded in an failed_images.txt file in the image input folder.

The last of the three **Scaling Types** is “Manual”. This **Scaling Type** requires input from the user. When you choose “Manual” as your Scaling type, you need to manually place **two points** on the image using the left mouse button.



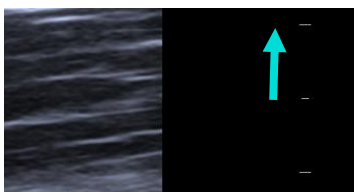
No worries, you do not actually need to draw on the image. Just click one time with your left mouse button to record the **first point** (nothing will be displayed on the images during actual analysis). Place the **second point** at a known distance of either 5, 10, 15 or 20 millimetre. The distance you chose must be represented in the Spacing (see next page) parameter in the GUI.

Whenever you use **“Bar”** or **“Manual”** as your Scaling Type, please make sure that the minimum distance between the scaling bars or the known distance between you manually specified points is represented in the **Spacing** parameter. For the **“No Scaling”** Scaling type, this is not necessary.

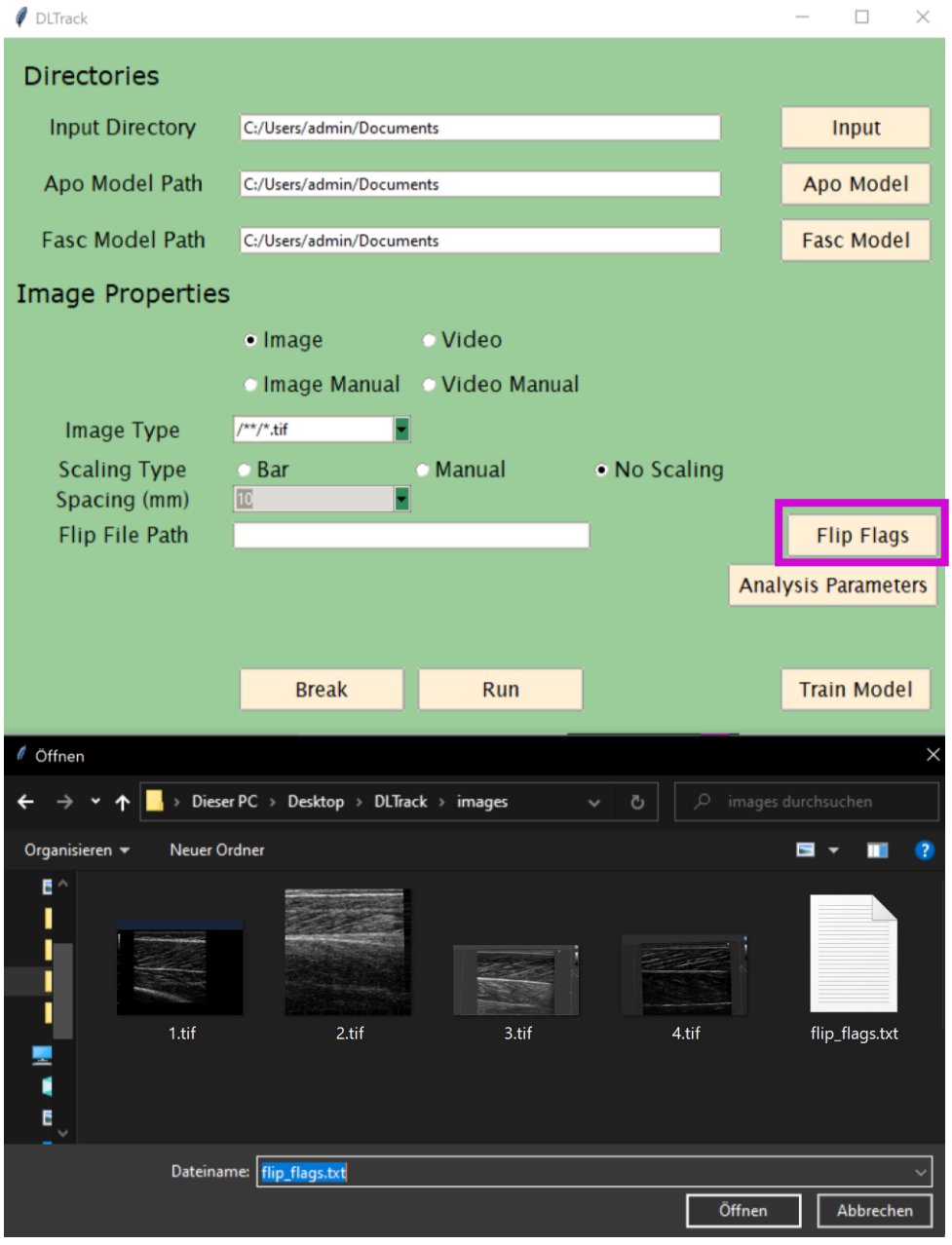


You can select the **Spacing** parameter only from the dropdown list as 5, 10, 15 or 20 millimetre. For this tutorial it is not necessary to select anything, as the Spacing parameter is not used during an analysis with Scaling Type **“No Scaling”**.

So far, we haven’t explained how to determine the minimal **distance** between the scaling bars in an image. This is simply the **distance** in millimeter between the two nearest scaling bars in the image. If you do not know this **distance**, please use **“Manual”** or **“No Scaling”** Scaling Type. For example in the image from before, the **distance** between the nearest bars is 5 millimetre. We know that because the distance between the bigger bars is always 10 millimetre.

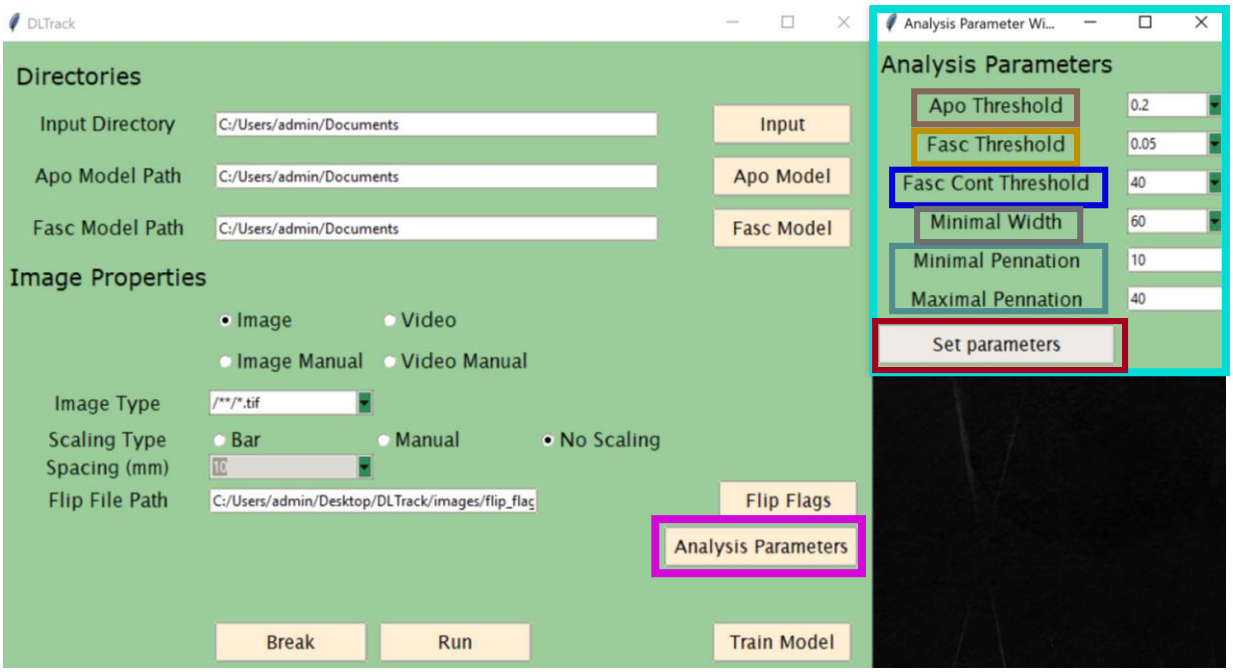


The next step in the image analysis using the DL_Track_US package is to specify the absolute path to the **flip_flag.txt** file. This is actually the same procedure than selecting the absolute to the **aponeurosis and fascicle neural networks** before. By clicking the **Flip Flags** button, a dialogue will pop up and you can select the **flip_flag.txt** file to retrieve it's path. In this example, the **flip_flag.txt** file is located at "DL_Track_US_example/images". Remember, the amount of **flip-flags** in the flip_flag.txt file must equal the amount of **images** in the **images** folder.



5. Specifying Analysis Parameters

Awesome, you have now successfully selected all relevant parameters in the main GUI window. As a LAST step, you need to specify the analysis parameters for the **aponeurosis and fascicle neural networks.** When you press the **Analysis Parameters** button, a separate window will pop-up.

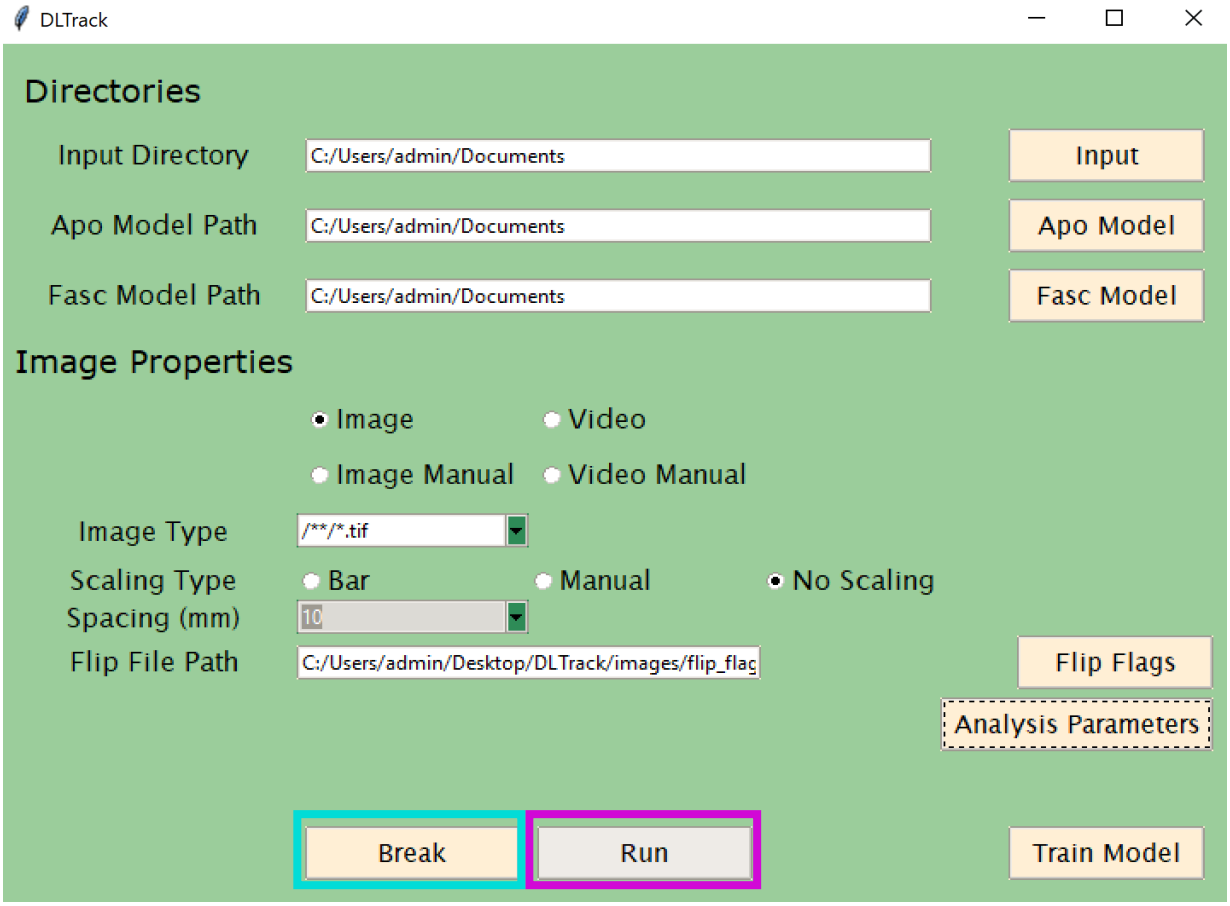


In the **Analysis Parameter window**, all parameters used by the **aponeurosis and fascicles neural networks** during inference are specified. We will give a short explanation to each of those parameters. The **Apo Threshold** parameters determines the threshold of the minimal acceptable probability by which a pixel is predicted as aponeurosis. The lower, the more pixels will be classified as aponeurosis. The **Fasc Threshold** is the same thing just for fascicle segments. The lower the **Fasc Cont Threshold**, the shorted the minimal acceptable length of detected fascicle segments to be included in the results. The **Minimal Width** determined the minimal acceptable distance between superficial and deep aponeurosis. **Minimal and Maximal Pennation** describe the respective minimal and maximal pennation angle that is physiologically possible in the analysed image/muscle.

For this example, you can leave all parameters the way they are. You can set the parameters by clicking the **Set parameters** button, the **Analysis Parameter window** will then close automatically. Please make sure to adapt these parameters according to your images in analyses outside of this example. For future analyses, it's best you test the ideal parameter configuration to get the best prediction results on your images in a small sample prior to the actual analysis.

6. Running / Breaking DL_Track_US

Now its time to start the actual analysis of the example images in the "DL_Track_US_example/images" folder. You can do so by clicking the **Run** button in the main GUI window.

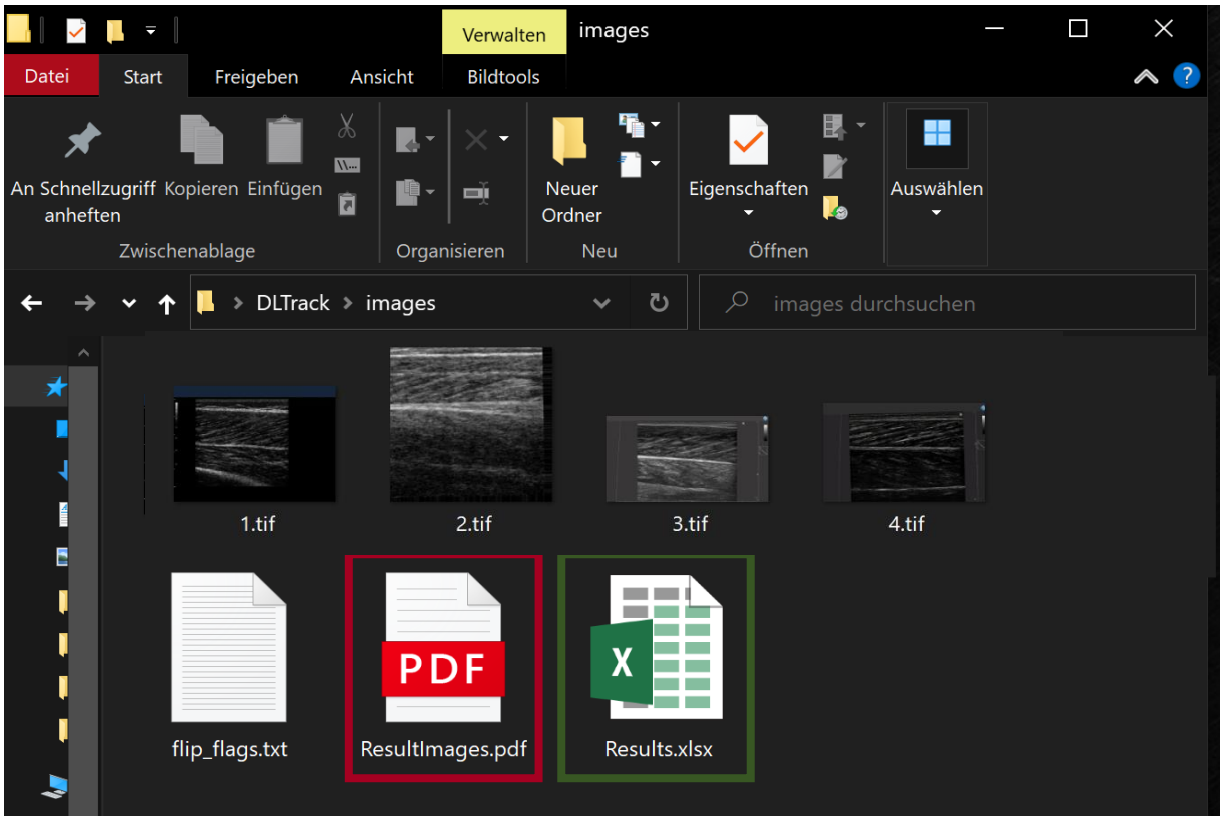


When you started the GUI using the command prompt in any way, you will see that the analysis is started by the statements printed in the prompt. When you started the GUI using the executable, you will just have to believe us that the analysis is started.

Moreover, you can see that there is a **Break** button placed in the GUI as well. Clicking the Break button allows you to stop the analysis at any point. The currently evaluated image will be processed and then the analysis is terminated.

Take a look at the next page to see what happens when the analysis is finished.

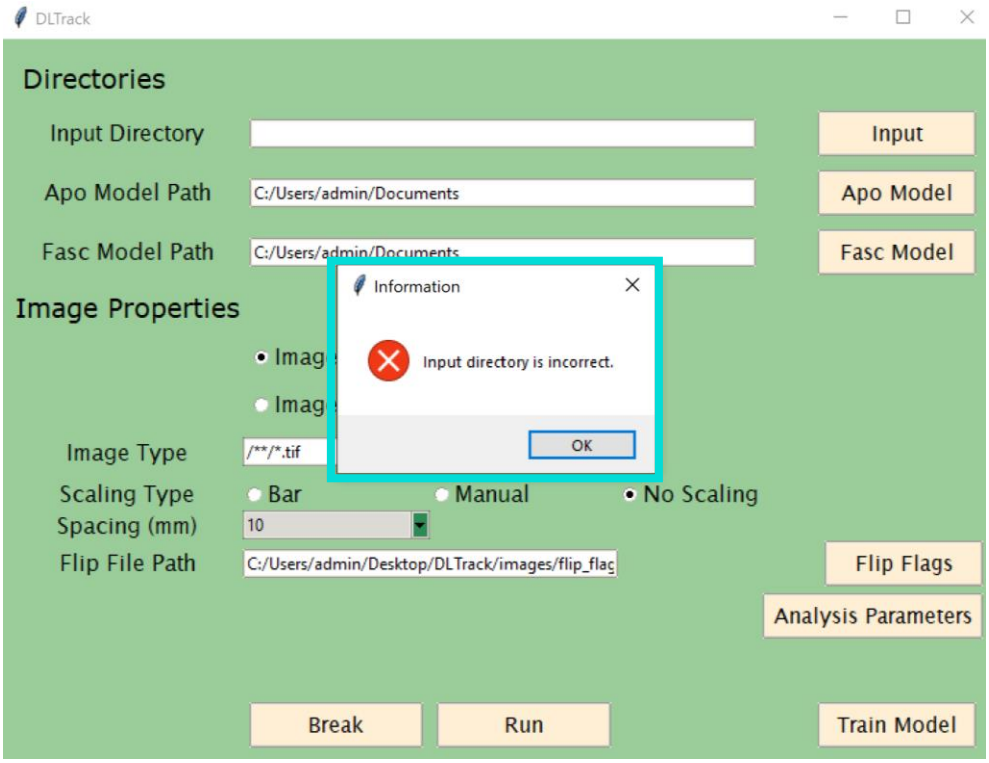
Subsequently to clicking the Run button in the main GUI, navigate again to the "DL_Track_US_example/images" folder in your explorer. You will see that two files will be / have been created, **ResultImages.pdf** and **Results.xlsx**. The **ResultImages.pdf** file contains each original input image and concomitant prediction results with fascicles and aponeurosis displayed. This file allows you to visually inspect the model outputs. In your future analysis outside of this tutorial, you should always visually inspect the **ResultImages.pdf** file. The **Results.xlsx** file contains the actual architectural parameter estimates for each input image. There, the median value of all detected muscle fascicle length and pennation angles as well a the calculated muscle thickness will be displayed. Each input image is displayed in a separate row. Note that the **ResultImages.pdf** file can be opened only after the **Results.xlsx** was created.



When both files can be opened and you can see the analysis results, original image and the prediction result, we must congratulate you! You have now officially and successfully completed the DL_Track_US tutorial for automated image analysis! There is one more thing though, error handling. Take a look at the next section to get more information.

7. Error handling

Whenever an error occurs during the analysis process, the DL_Track_US GUI will open a **messagebox**. This looks always similar to this:



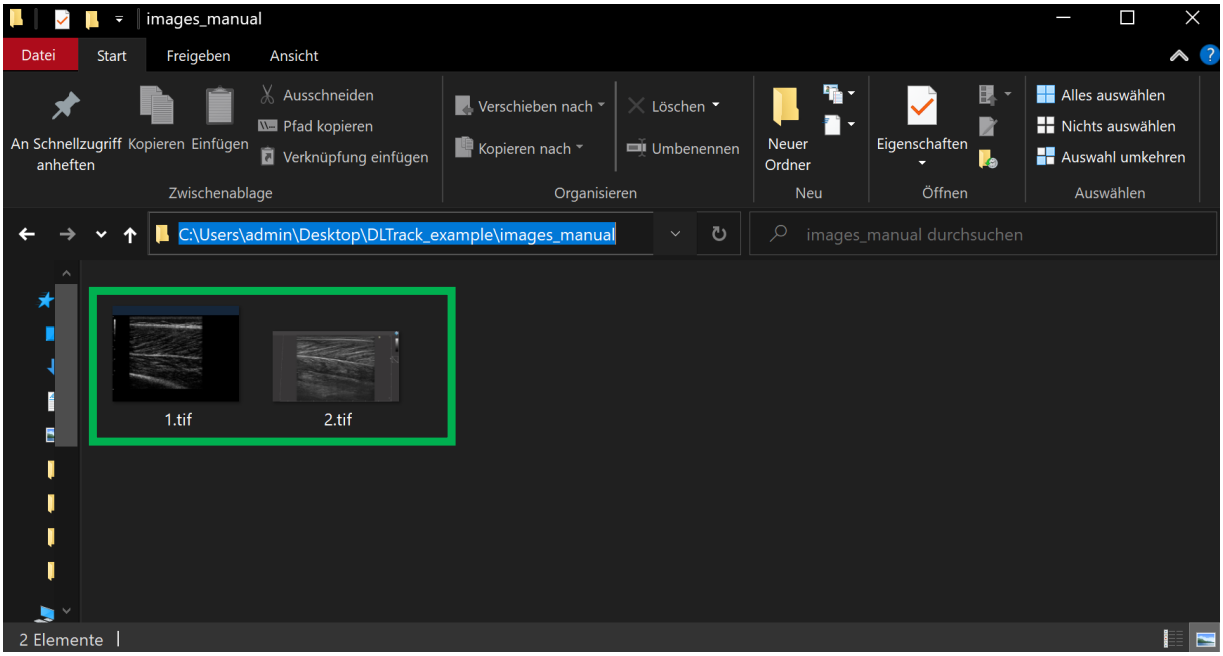
We tried to formulate these **messageboxes** as concise as possible. Just follow their instructions to fix the error and run the analysis anew. In case an error occurs that is not caught by an error **messagebox,** don't hesitate to report this in the Q&A section in the DL_Track_US discussion forum. Please take a look here how do best do this. Otherwise, you can contact us by email at paul.ritsche@unibas.ch, but we would prefer the other way.

Manual Image Analysis

The DL_Track_US python software package offers several different analysis types for analysis of human lower limb longitudinal ultrasonography images. The next analysis type this tutorial covers is the manual image analysis. The images are evaluated manually by drawing the muscle thickness, fascicle length and pennation angles directly on the Image. Scaling the images will ensure estimated muscle architectural parameters are converted to centimetre units. For this type of analysis, single images (not videos) are a prerequisite. These images should be contained in a single folder, like in the "DL_Track_US_example/images_manual" folder. If you haven't downloaded this folder, please do so now (link: DL_Track_US - Examples & Models | Zenodo). Unzip the folder and put it somewhere accessible, for example on your desktop. We will make use of the included example files extensively during this tutorial. In the next few pages, we will look at every required step to successfully perform automated image analysis with DL_Track_US.

1. Creating Image Directory

In order for DL_Track_US to recognize your images, they should best be in a single folder (though one subfolder structure is acceptable as well). Take a look how you might structure this:

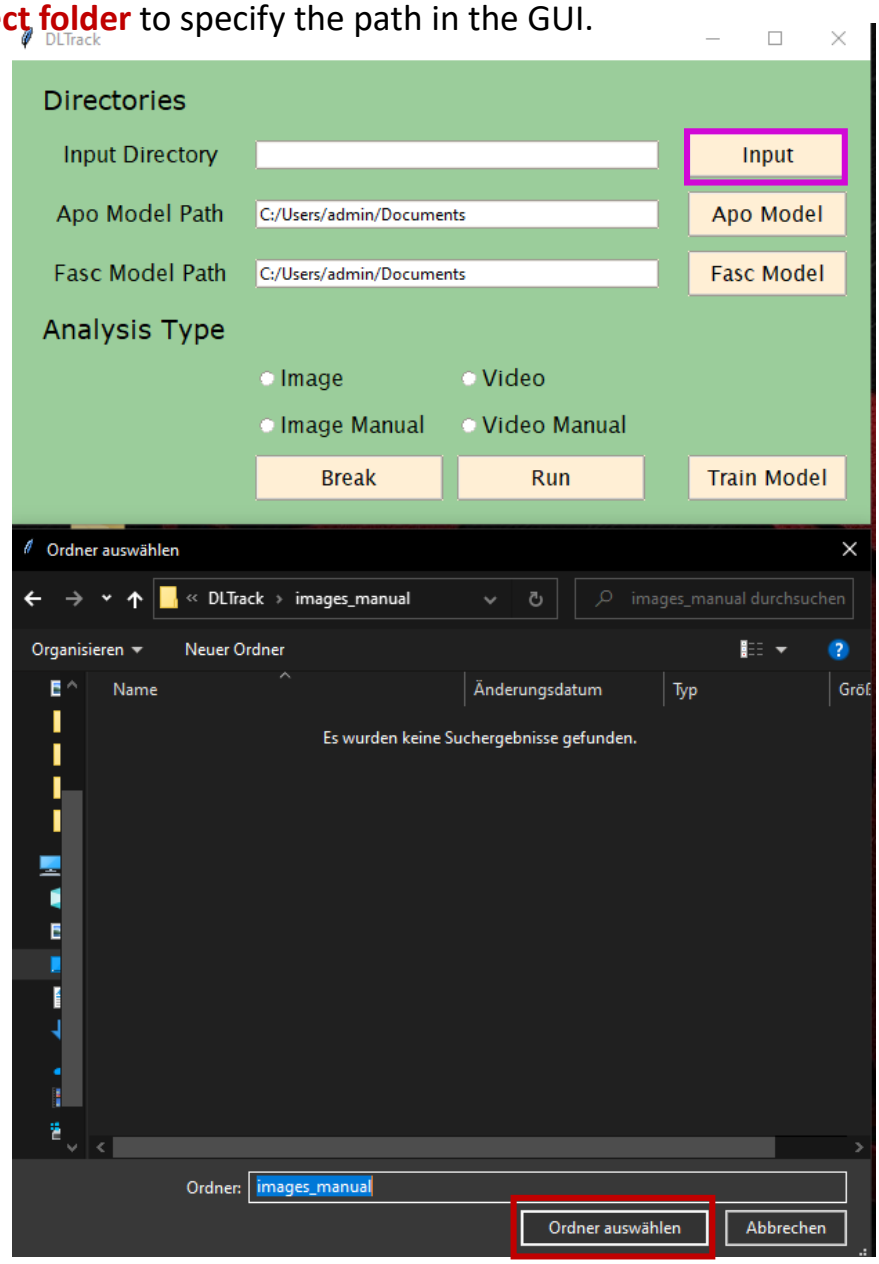


You can see in the picture above that the folder contains **2 images** and is **located on the desktop**. This structure is already included in the "DL_Track_US_example" folder. In contrast to automated image analysis, you do not need a flip_flag.txt file nor do you need neural networks that do predictions. Here, you are the neural network. So, the next step is to specify the input directory in the GUI.

2. Specifying Input Directories in the GUI

You can start with the actual analysis! The first step of every analysis type in DL_Track_US is to specify the input directories in the graphical user interface (GUI). We assume that you have already opened the GUI. (If not, take a look at the first chapter of this document “Starting the GUI”.).

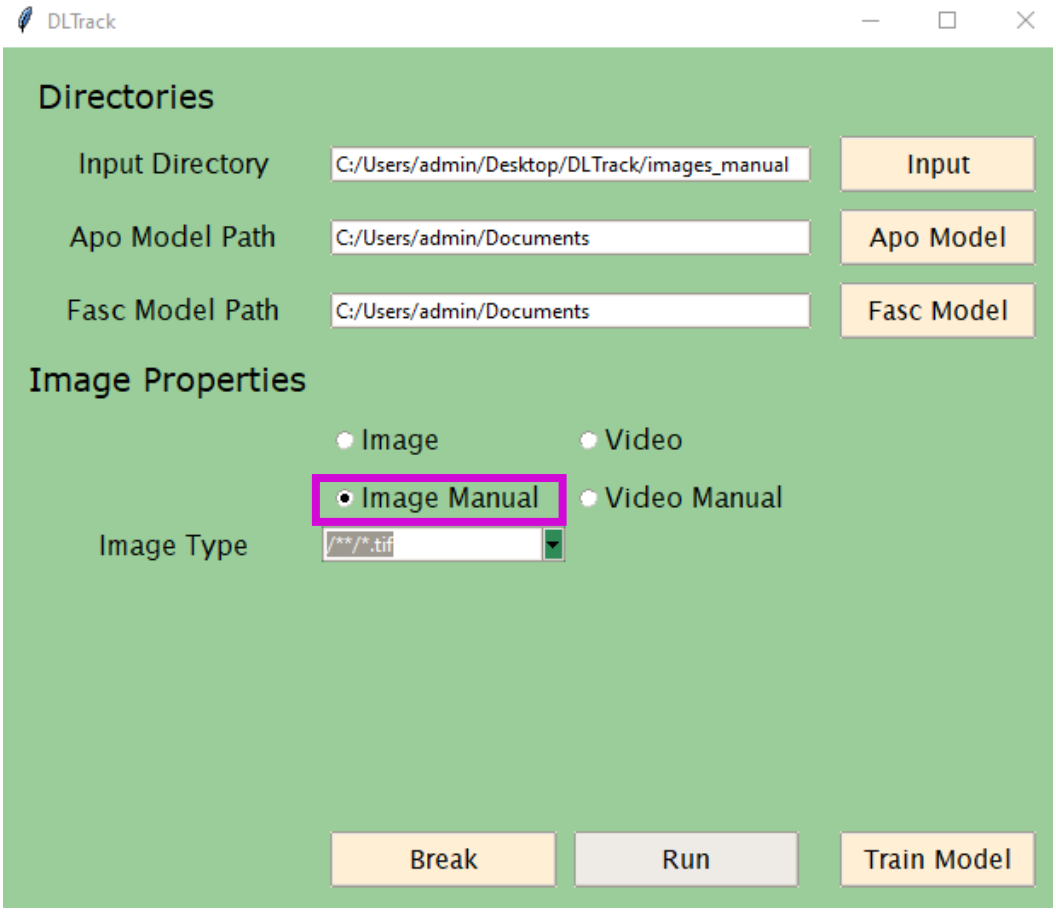
You will begin with specifying the path to the folder containing the **images** to be analysed. Remember this is the folder “DL_Track_US_example/images_manual”. By clicking on the **Input** button in the GUI a selection window opens were you need to select the images folder. Click **select folder** to specify the path in the GUI.



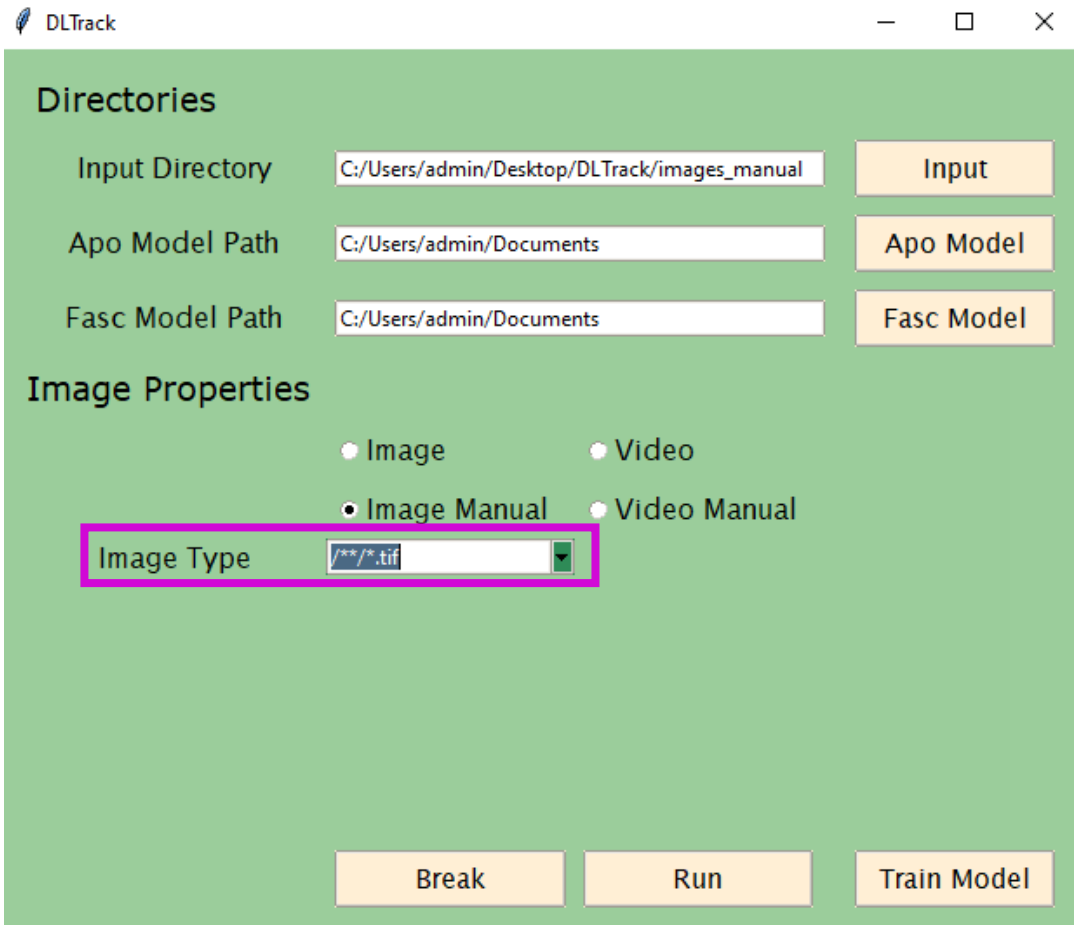
Once that is done, the path will be displayed in the entry filed and you can start to specify the relevant parameters for the analysis.

3. Specifying Relevant Parameters

As a first step, you will select the right analysis type in the GUI. Since this section is about manual image analysis, please select the **Image Manual** radiobutton. You can see that the GUI unfolds and another parameter appear. You will set this one in the next step on the next page.

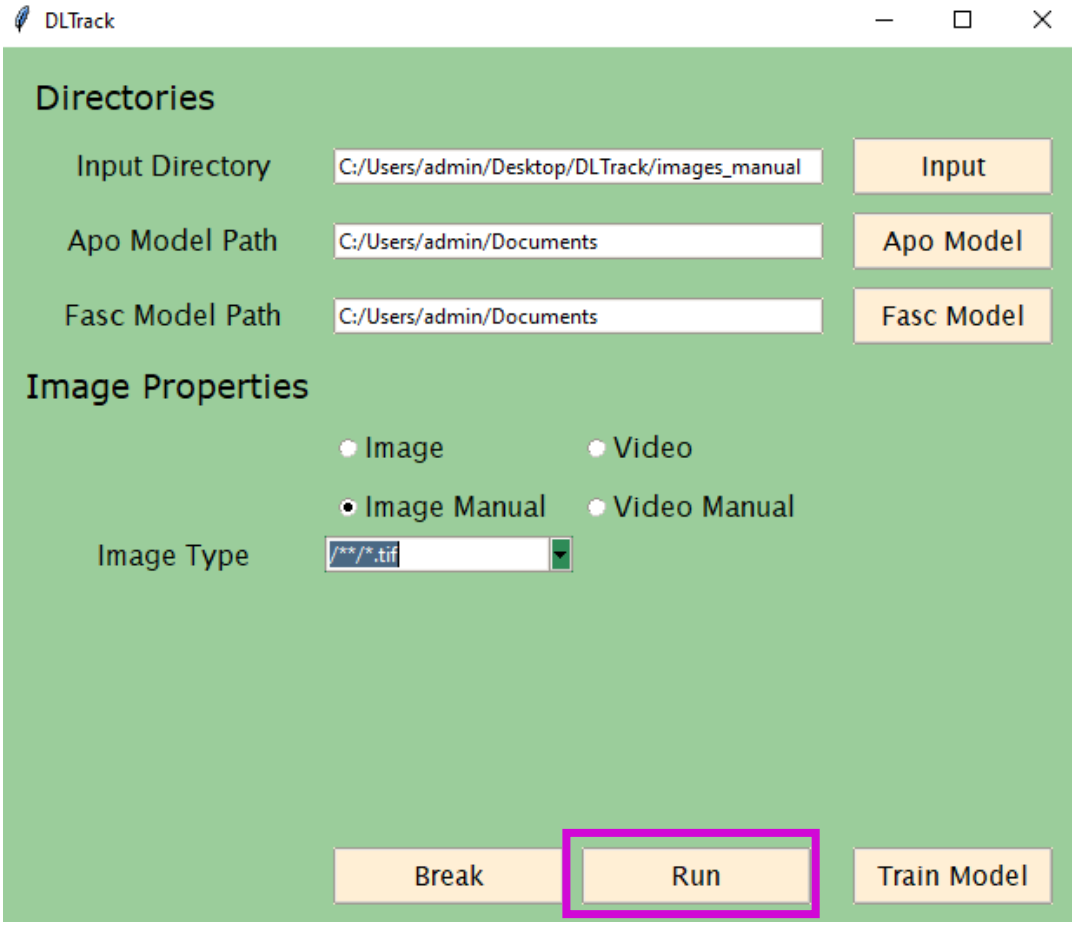


Next, you need to specify the Image Type. The ending of the **Image Type** must match the ending of your **images**, otherwise no files are found by DL_Track_US. You can either select a pre-specified ending from the dropdown list or type your own ending. Please keep the formatting similar to those **Image Types** provided in the dropdown list. All the **images** in the "DL_Track_US_example/images_manual" folder are of the **Image Type** ".tif". Thus, you should select the "/**/*.tif" **Image Type**.



Allright, once you have specified the Image Type, you can start with the analysis of the images contained I the "DL_Track_US_example/images_manual" folder.

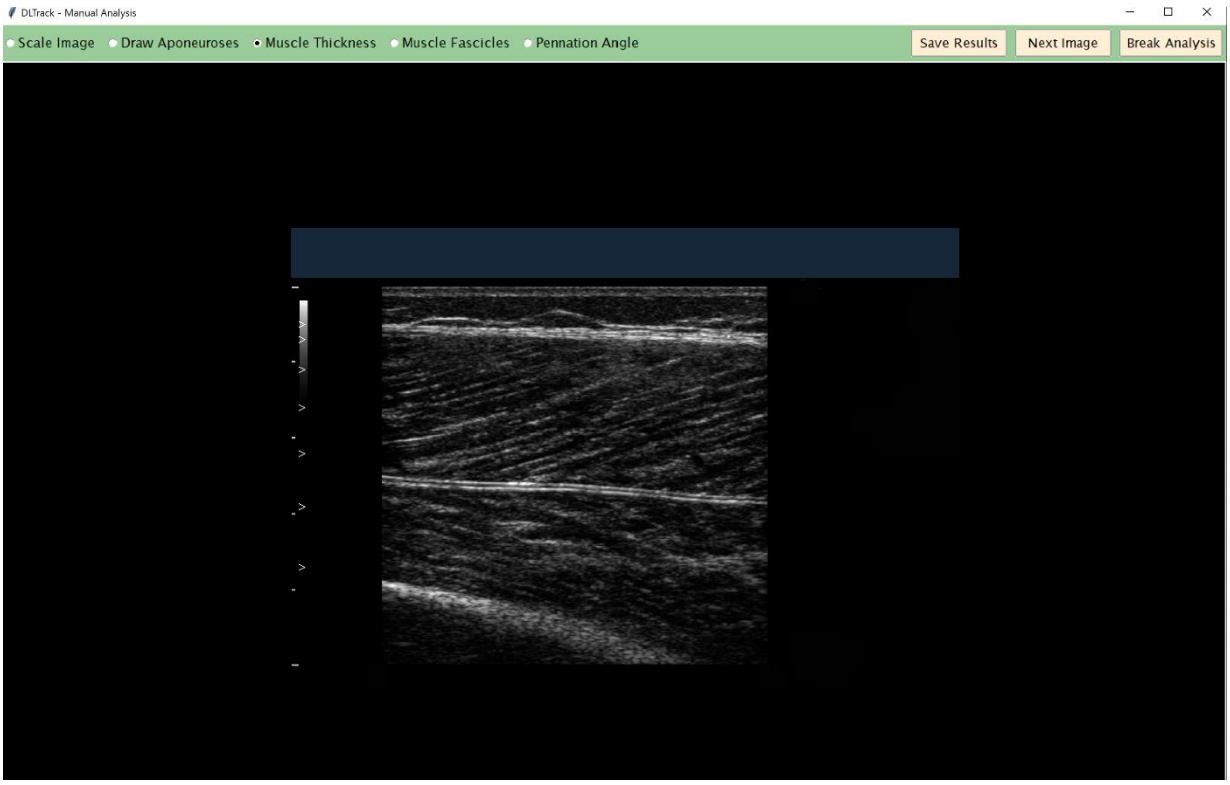
Allright, once you have specified the Image Type, you can start with the analysis of the images contained I the "DL_Track_US_example/images_manual" folder. You can start the analysis by clicking the **Run** button in the main GUI.



Take a look at the next page to see how to continue in the "Manual Analysis window" that pops up.

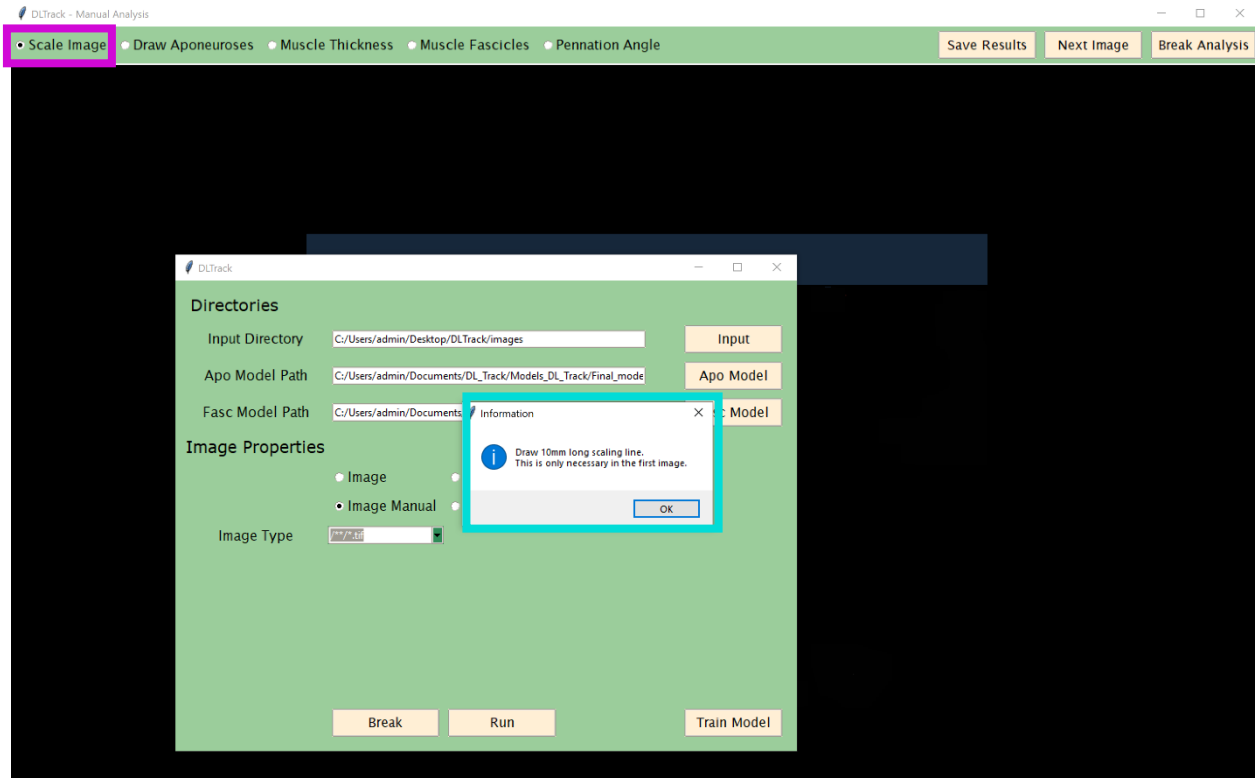
4. Manual Analysis of Images

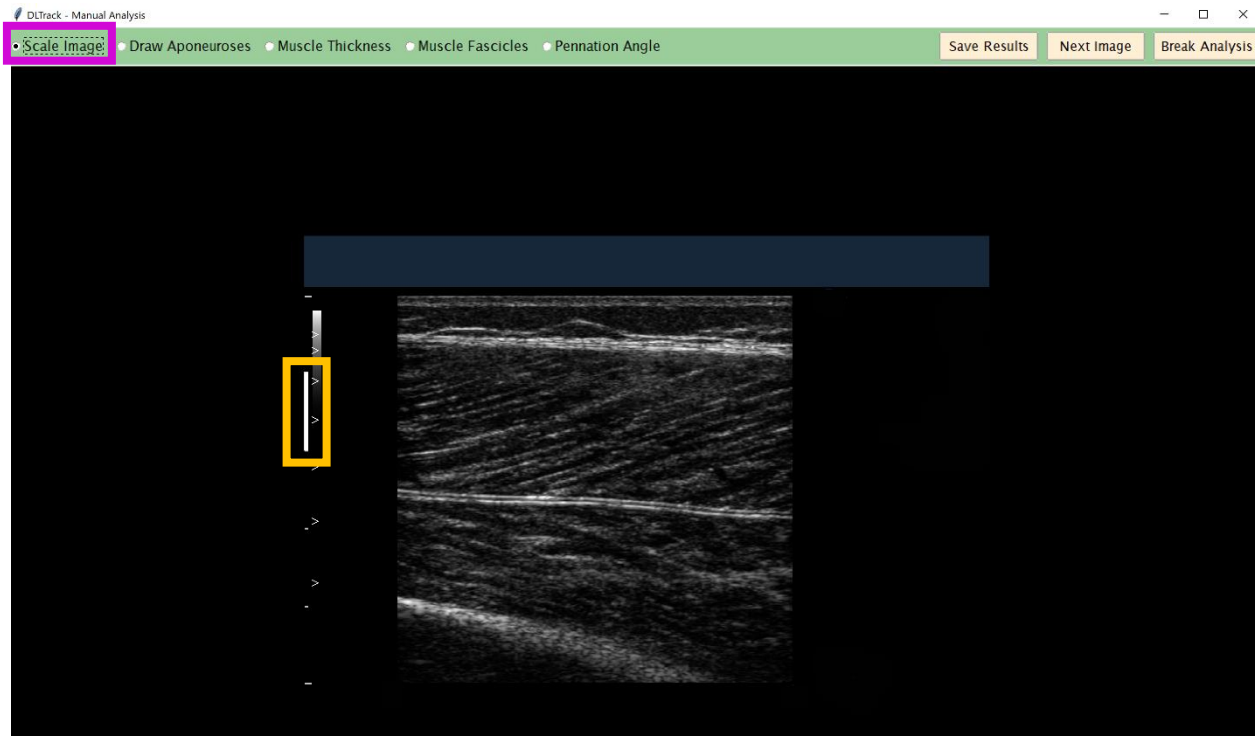
Subsequent to clicking the Run button in the main GUI, the “Manual Analysis window” opens. Here you can analysis the image by marking the respective architectural parameter you want to analyse directly on the image. We will guide you through that on the next pages. But first of all, here is how the “Manual Analysis window” looks like:



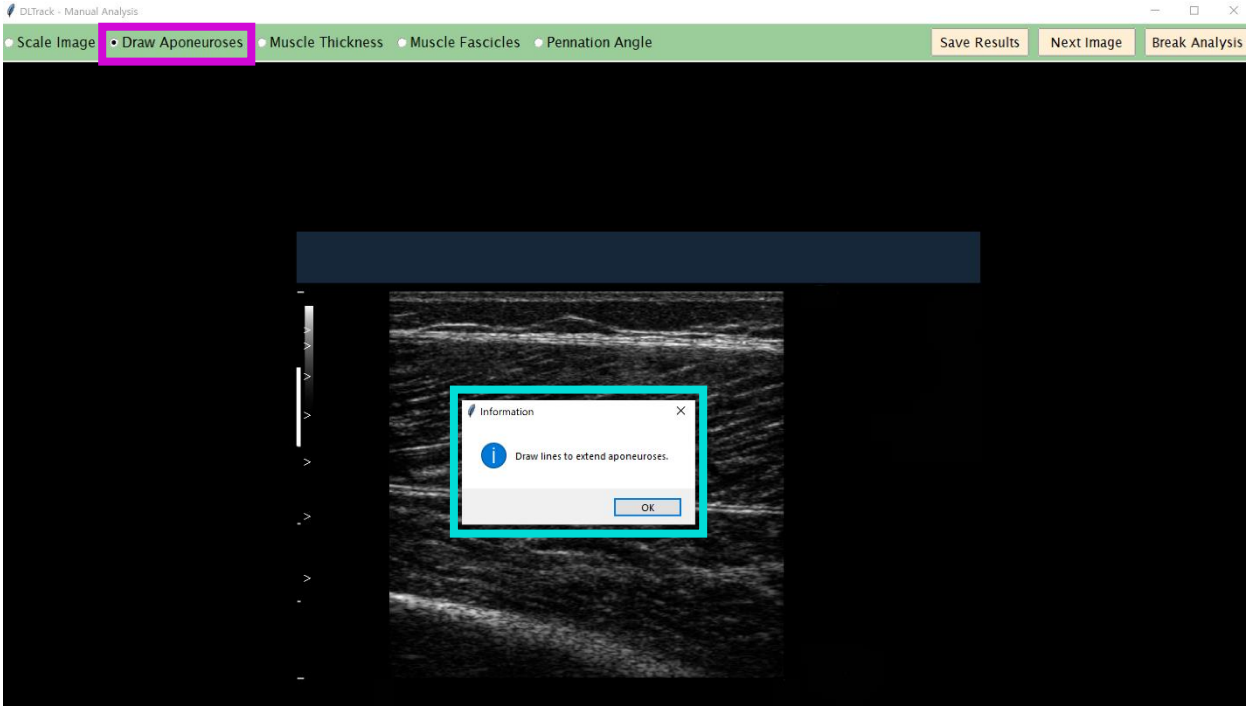
It is of utmost importance to keep in mind that the lines you are about to draw on the image are only there to help you remember what it did. The actual lines are not used during the computation of the architectural parameters. The start- and endpoints of each line are relevant. The start point is defined as the point where you clicked the left mouse button to start drawing the line. The endpoint is defined as the point where you released the left mouse button to stop drawing the line. The line follows the cursor as long as the left mouse button is pressed. The calculations of the scaling line length, muscle thickness, fascicle length and pennation angle are dependent on the number of specified lines/segments. Do NOT click somewhere random on the image during the analysis of a parameter and exactly follow the instructions. If additional clicks happened, start the analysis anew by selecting the radiobutton representing the parameter again. If you do not follow the instructions presented in this tutorial, we cannot guarantee the correctness of the analysis results.

First of all, you will scale the images manually so that the calculated architectural parameters are returned in centimetre rather than pixel units. For the scaling, please draw a **one centimetre long straight line in the image**. The distance of one centimetre is usually recognizable in the scaling bars in the image. You can initiate the scaling process by selecting the **Scale Image** radiobutton in the "Manual Analysis window". A **messagebox** will appear advising you what to do. The **drawn line** should look like this.

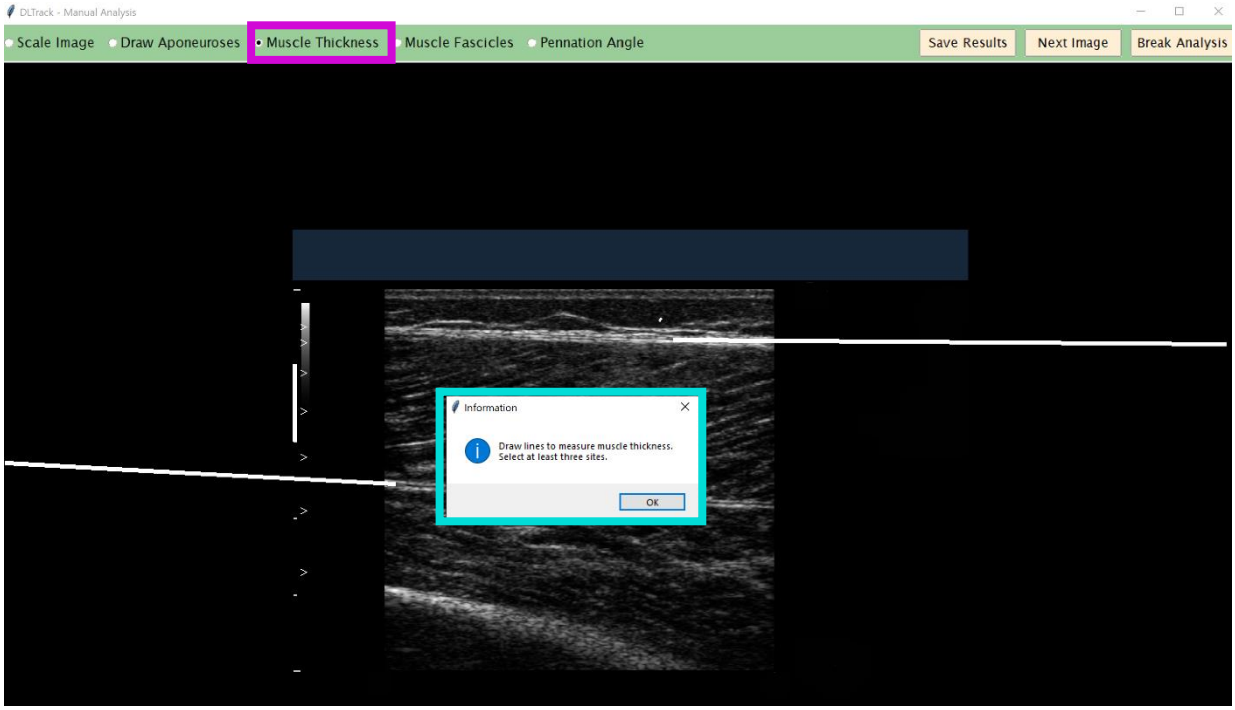


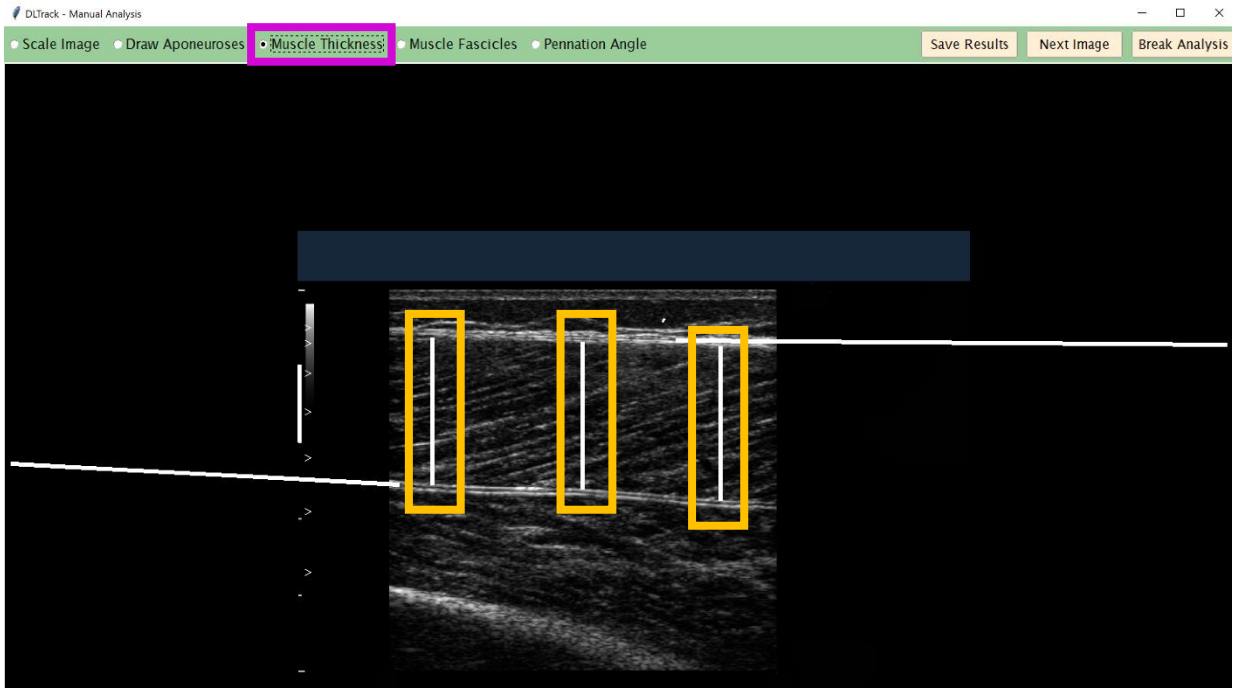


As a next step you have the option to extend the muscle aponeuroses to ease the extrapolation of fascicles extending outside of the image. This however not required, merely an option. You can do by selecting the **Draw Aponeurosis** button in the “Manual Analysis window” and draw the **aponeurosis lines** on the image as shown below. A **messagebox** will appear advising you what to do.

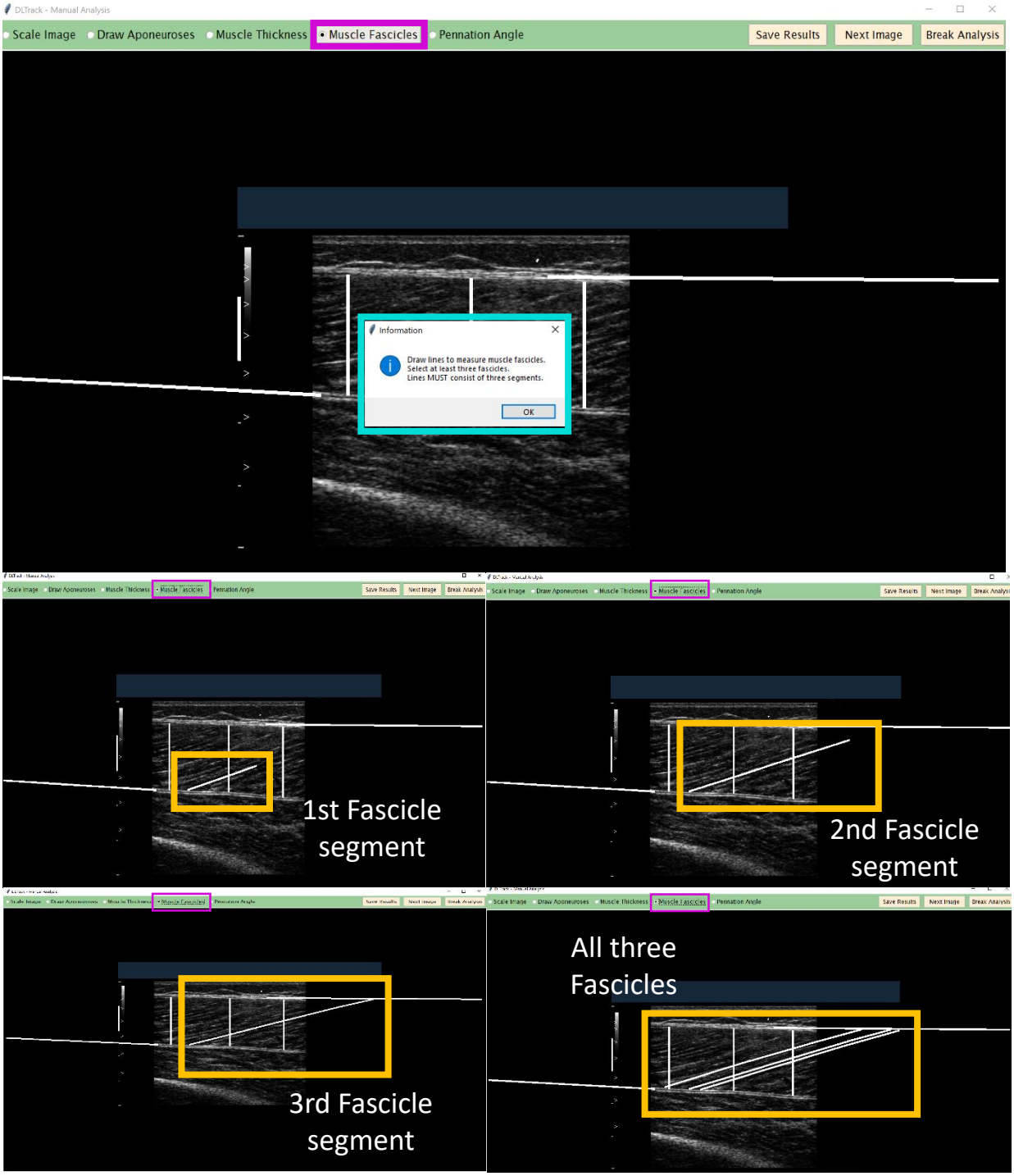


Once the image is scaled and aponeuroses structures extended, you can start analysing the muscle architectural parameters. You will start with the muscle thickness. Therefore, select the **Muscle Thickness** radiobutton in the “Manual Analysis window”. A **messagebox** will appear advising you what to do. We advise to you to now draw **three straight lines** reaching from the superficial to the deep aponeurosis in the middle, right and left portion of the muscle image.

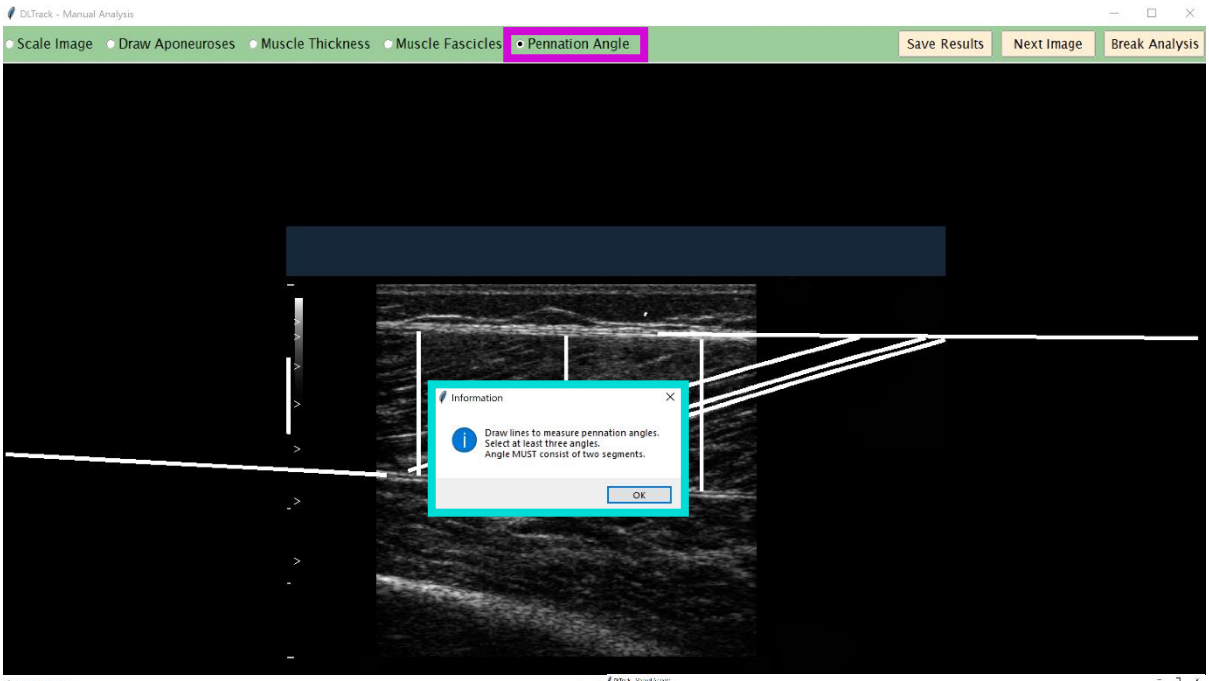


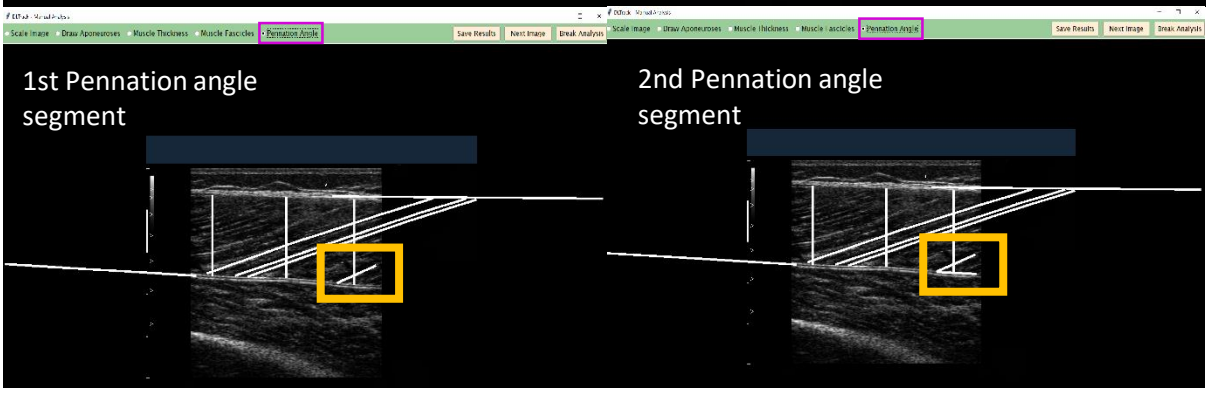


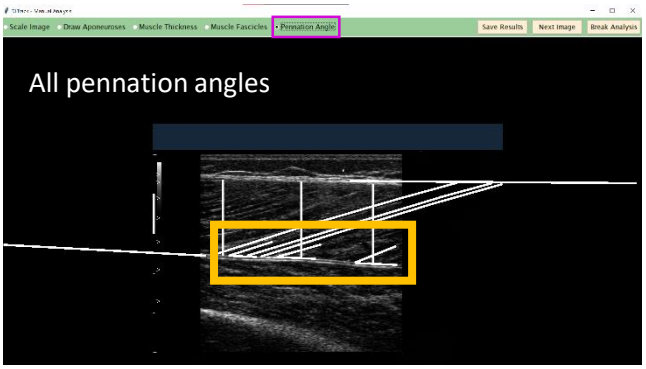
Next you can mark single fascicles on the image by selecting the **Muscle Fascicles** radiobutton in the “Manual Analysis window”. A **messagebox** will appear advising you what to do. We advise to draw at least three **fascicles** per image in different regions of the image. Please consider that these fascicles must be clearly visible and not guessed. It is possible to extrapolate the **fascicles** outside of the image region. Each **fascicle** MUST consist of three segments. Do not draw more or less segments per **fascicle** and pay attention to avoid any extra unwanted mouse clicks. One segment MUST start where the previous segment ended. Take a look at the image sequence below to see how it is done:



The last architectural parameter that you can manually analyse using the DL_Track_US package is the muscle pennation angle. Please select the radiobutton **Pennation Angle**. A **messagebox** will appear advising you what to do. We advise you to draw at least three **pennation angles** per image at different regions of the image. As with the fascicles, these should be clearly visible. Each drawn **pennation angle** MUST consist of two segments. The first segment should follow the orientation of the fascicle, the second segment should follow the orientation of the deep aponeurosis. The segments should both originate at the insertion of the fascicle in the deep aponeurosis. Please pay attention to avoid unwanted clicks on the image. Here is how its done:

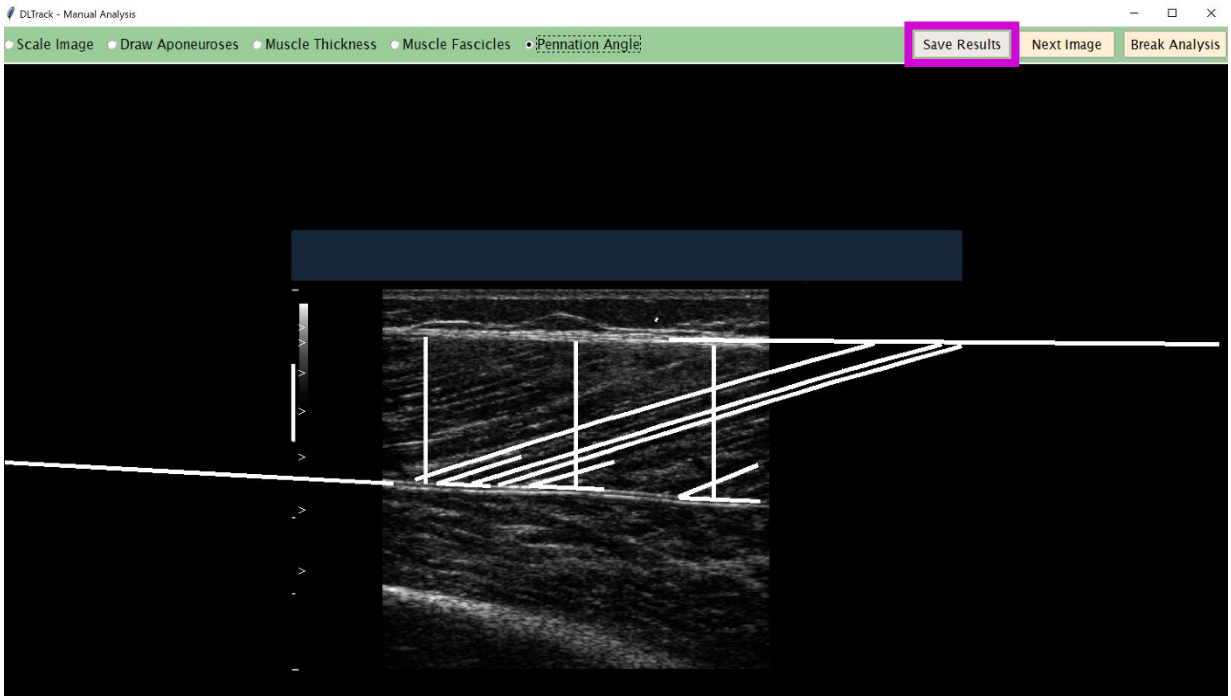




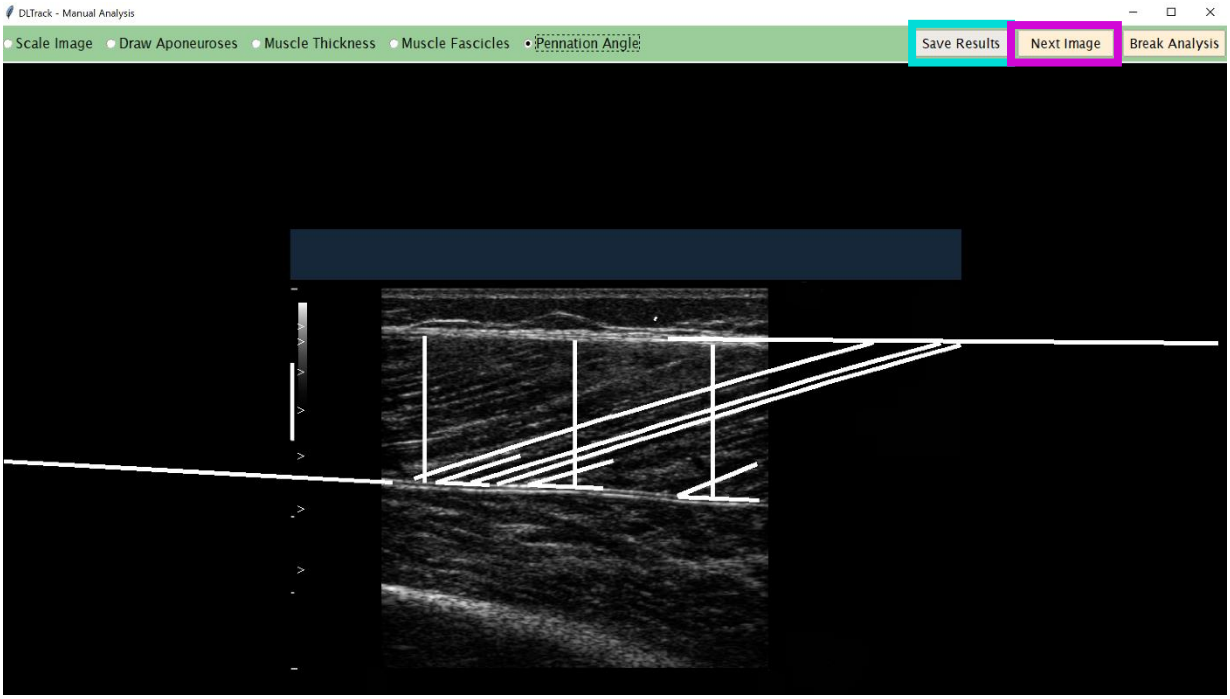


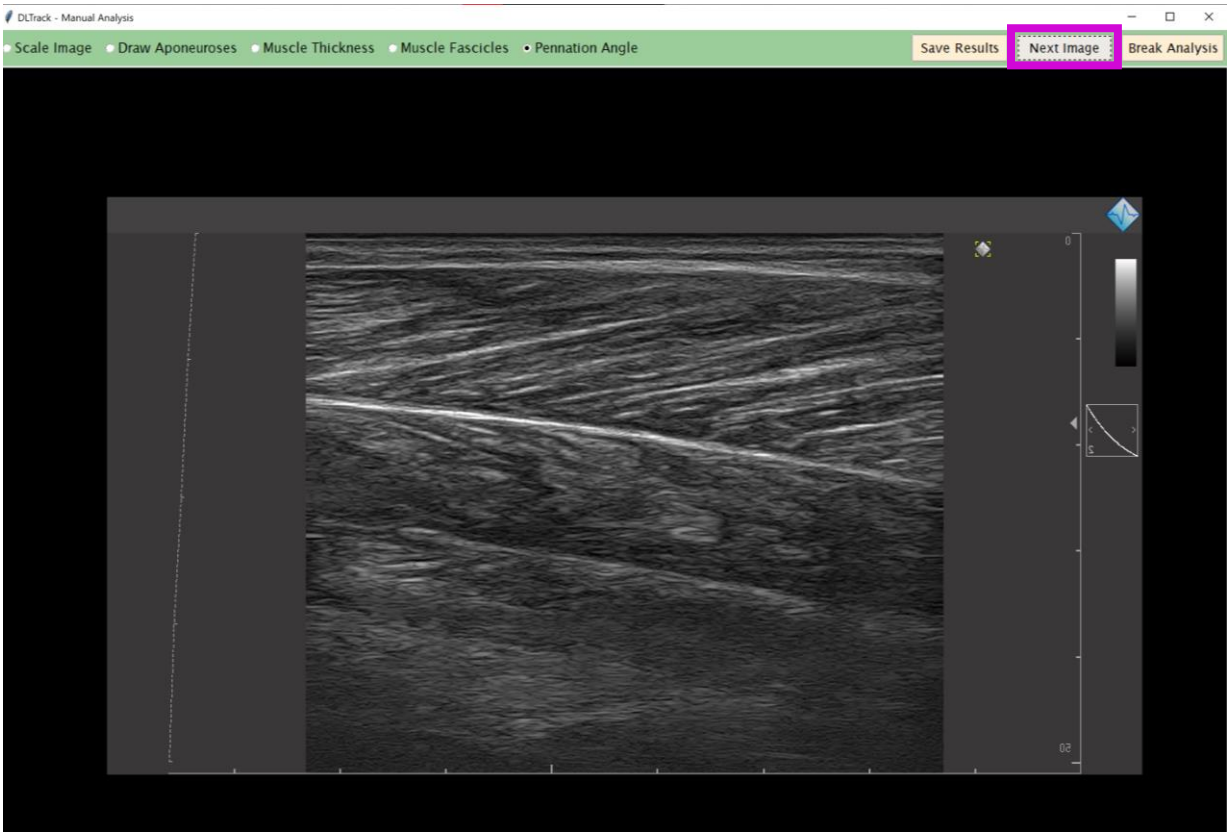
5. Saving / Breaking / Next Image

Great, you have now successfully analysed all muscle architectural parameters using the DL_Track_US manual image analysis tool. However, there are three buttons in the “Manual Analysis window” left to explain. The first button is the **Save Results** button. The **Save Results** button is a very important button! Before you have pressed this button, none of your analysis results are saved. Therefore, please press the **Save Results** button once you have analyzed all parameters that you wanted to analyze and **before** continuing with the next image. An excel file with the name Manual_Results.xlsx is saved in the directory of the input images upon pressing the **Save Results** button. Therein, all analysis results are stored. In your case the file is saved in the “DL_Track_US_example/images_manual” folder.

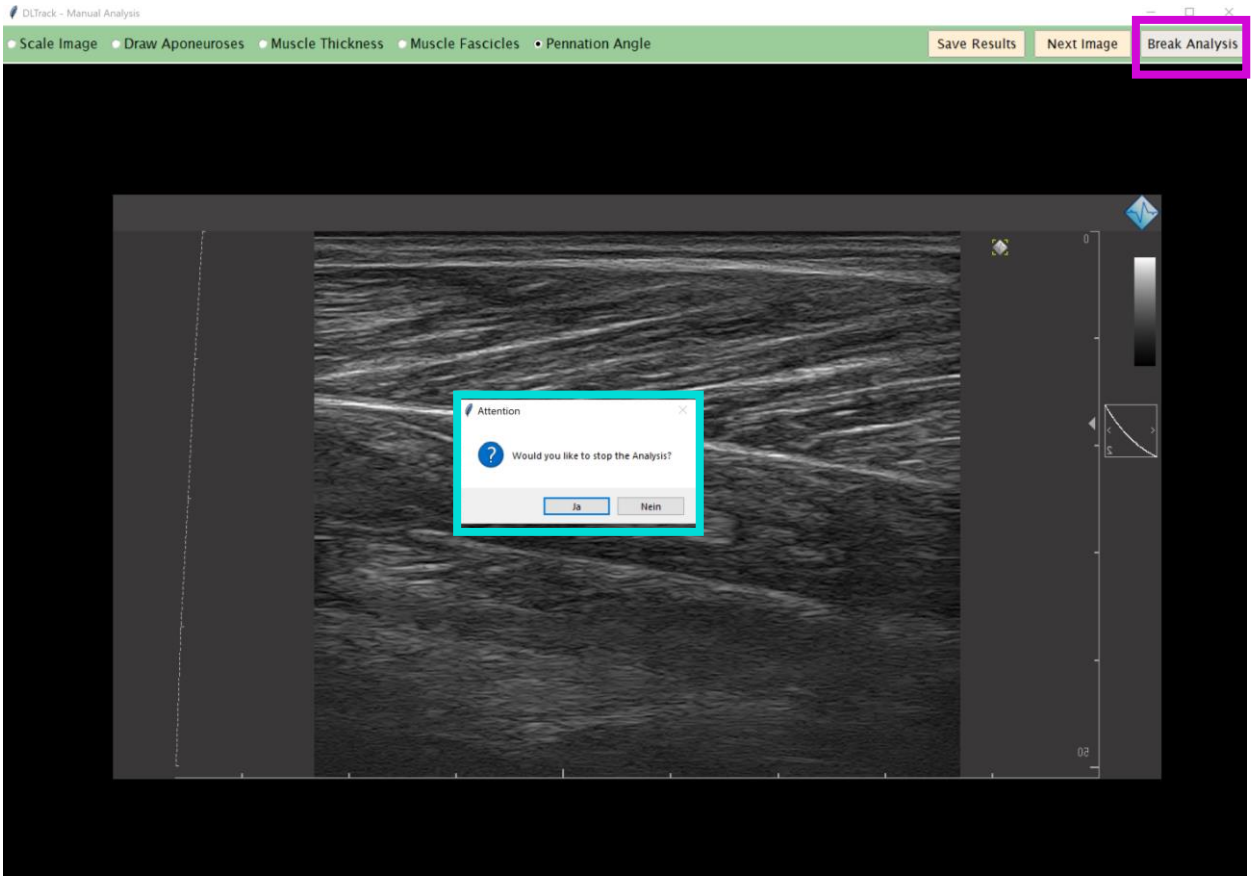


The second button we haven't explained yet is the **Next Image** button. By clicking this button, you can proceed to the next image in the input folder (in your case the "DL_Track_US_example/images_manual" folder). Please remember to press the **Save Results** button prior to proceeding to the next images, otherwise you analysis results for this image will be lost. When the **Next Image** button is pressed, the displayed image is updated.

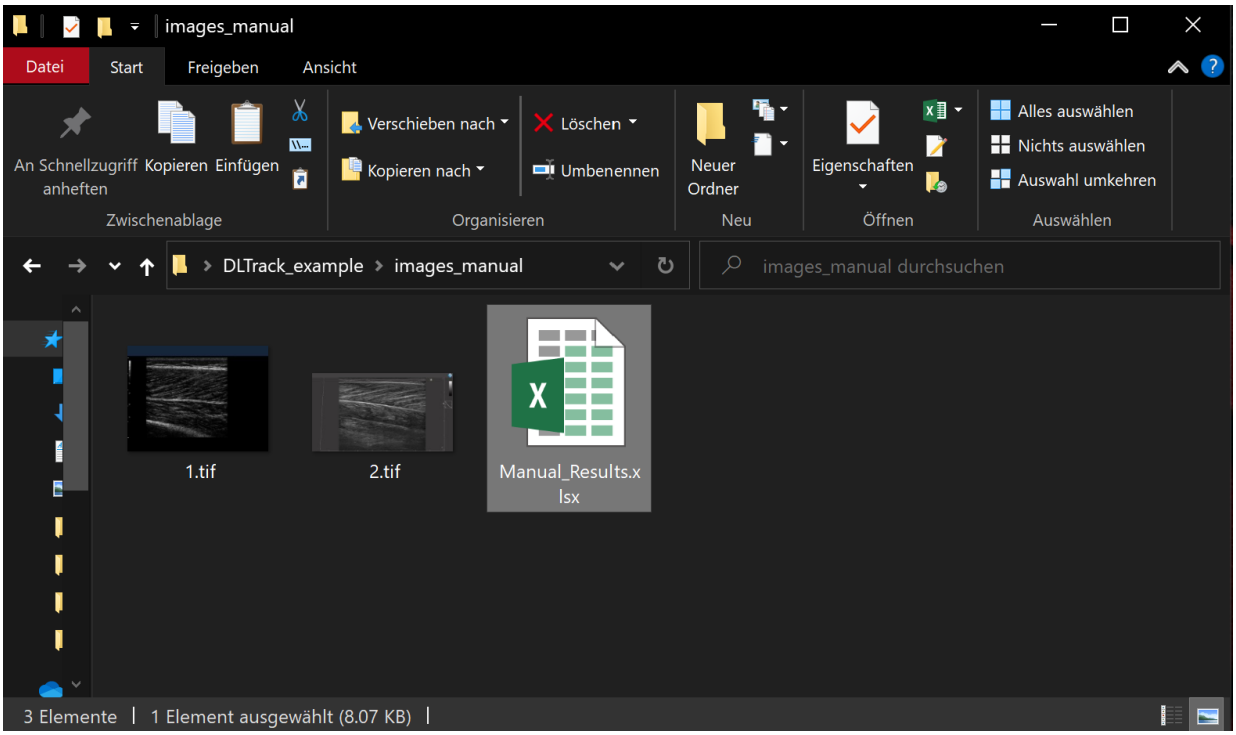




The last button we need to explain is the **Break Analysis** button. Pressing this button allows you to terminate the analysis and return to the main GUI window. A **messagebox** will appear asking you if you really want to stop the analysis. Once the **Break Analysis** button is pressed and you answered the messagebox with “YES”, the “Manual Analysis window” will be automatically closed. Remember to save your analysis results before doing so. We think you know which very important button can do just that!



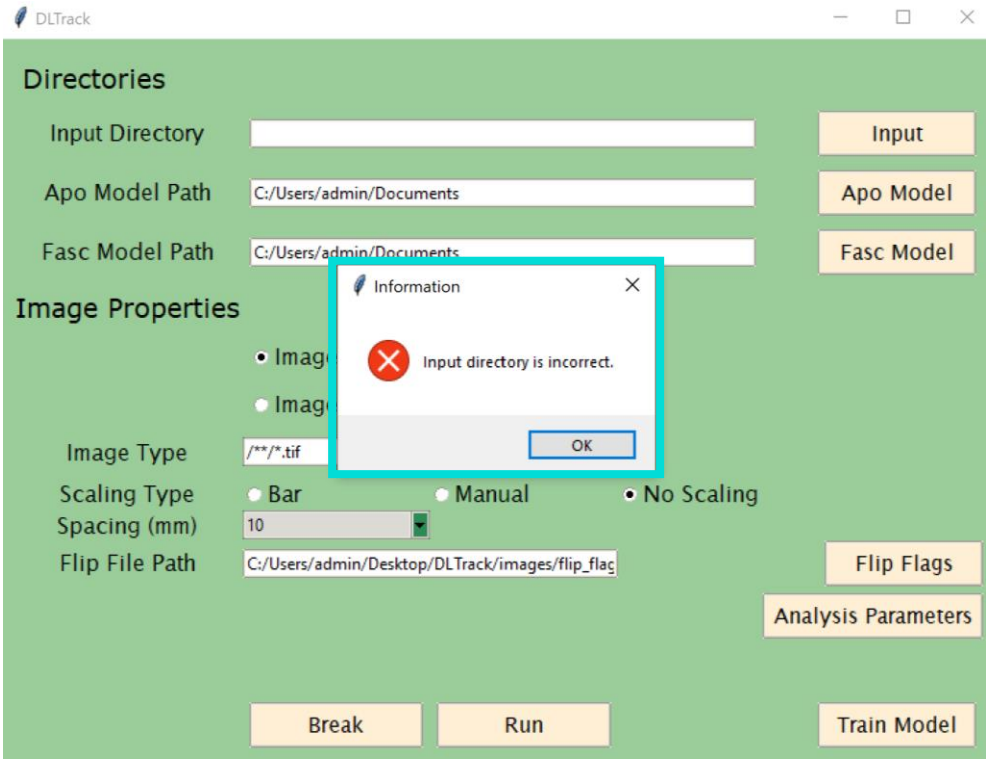
When you have saved your results clicking the very important button and followed our instructions during this tutorial, your input directory “DL_Track_US_example/images_manual” should look like this. It should contain the images as well as the Manual_Results.xlsx file.



Congrats are in order! You have now officially and successfully completed the DL_Track_US tutorial for manual image analysis! There is one more thing though, error handling. Take a look at the next section to get more information.

6. Error handling

Whenever an error occurs during the manual image analysis process, the DL_Track_US GUI will open a **messagebox**. This looks always similar to this:



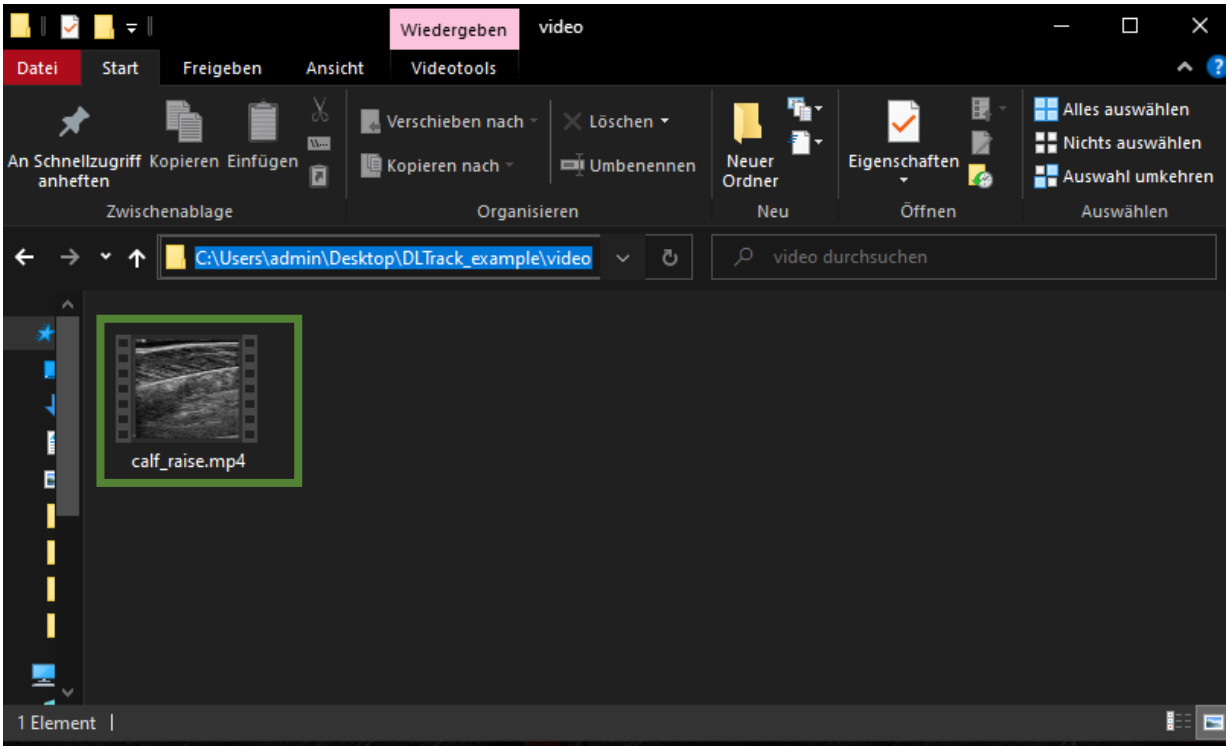
We tried to formulate these **messageboxes** as concise as possible. Just follow their instructions to fix the error and run the analysis anew. In case an error occurs that is not caught by an error **messagebox,** don't hesitate to report this in the Q&A section in the DL_Track_US discussion forum. Please take a look here how do best do this. Otherwise, you can contact us by email at paul.ritsche@unibas.ch, but we would prefer the other way.

Automated Video Analysis

The DL_Track_US python software package offers several different analysis types for analysis of human lower limb longitudinal ultrasonography images. This section of the tutorial covers the automated video analysis. The videos are evaluated without user input and may be scaled. Scaling the images will ensure estimated muscle architectural parameters are converted to centimetre units. For this type of analysis, videos are a prerequisite. These videos should be contained in a single folder, like in the "DL_Track_US_example/videos" folder. If you haven't downloaded this folder, please do so now (link: DL Track US - Examples & Models | Zenodo). Unzip the folder and put it somewhere accessible, for example on your desktop. We will make use of the included example files extensively during this tutorial. The automated video analysis is very similar to the automated image analysis. In fact, the inputted video is analysed frame by frame and each frame is therefore treated like an independent image. Moreover, only few analysis parameters are different between both analysis types. In the next few pages, we will look at every required step to successfully perform automated video analysis with DL_Track_US.

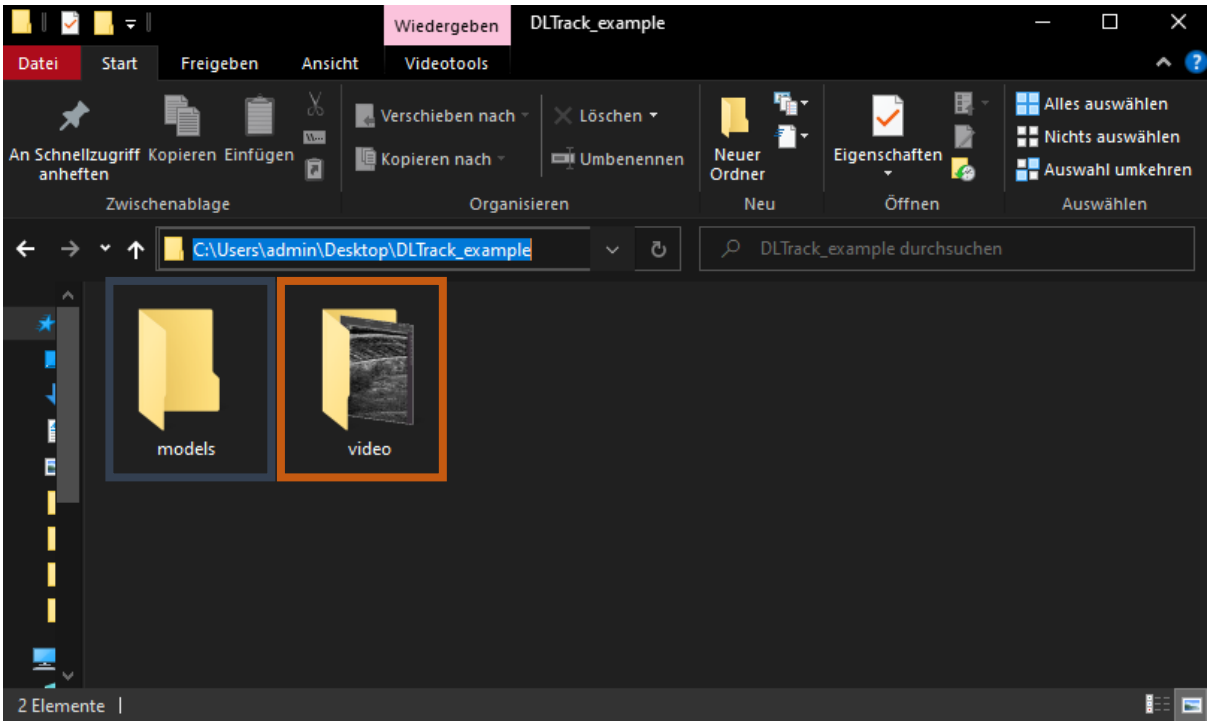
1. Creating Video and Network Directories

In order for DL_Track_US to recognize your videos, they should best be in a single folder (though one subfolder structure is acceptable as well). Take a look how you might structure this:



You can see in the picture above that the folder contains **one video** and is **located on the desktop**. This structure is already included in the "DL_Track_US_example" folder. We will continue with demonstrating how to create folders for the **aponeurosis and fascicle neural networks** on the next page.

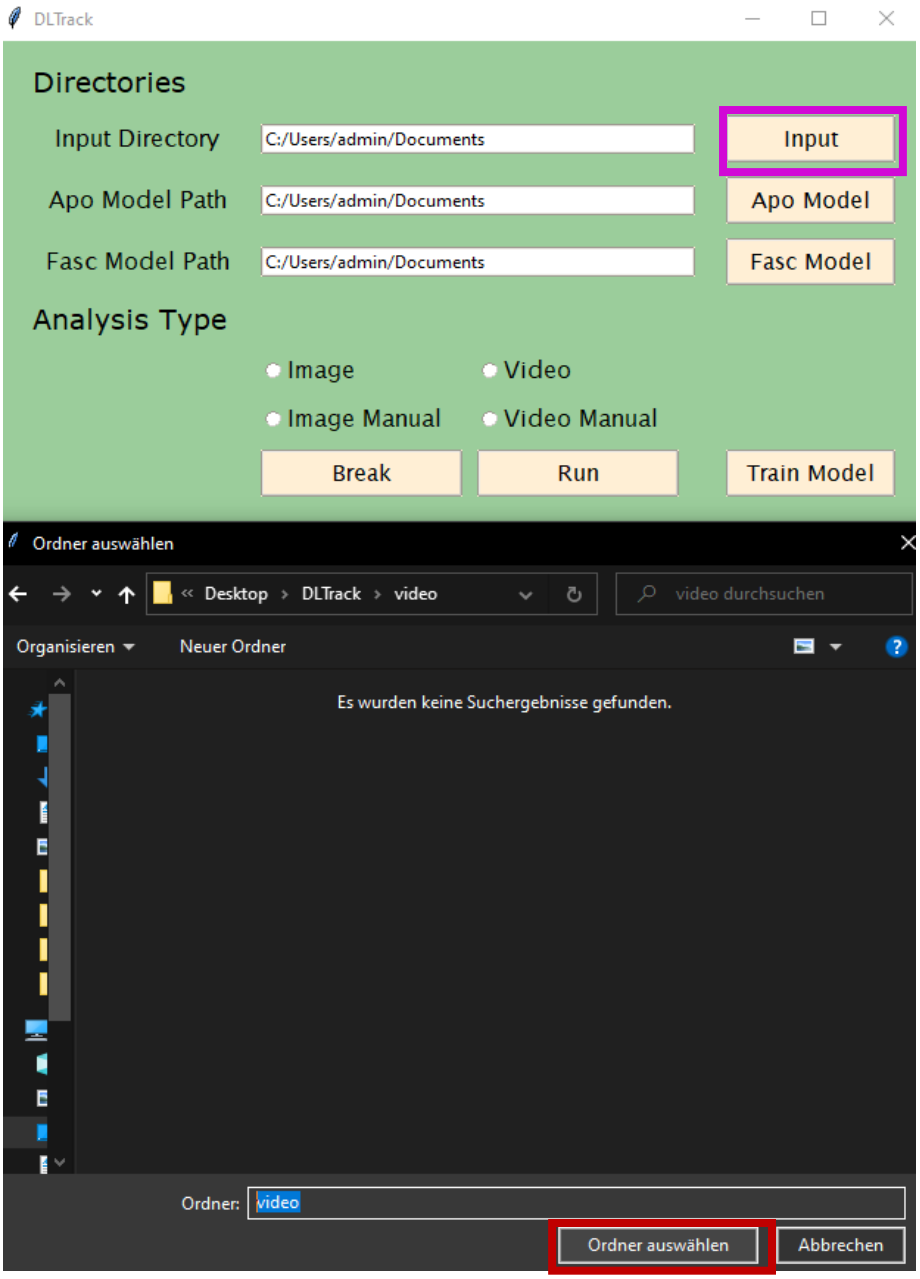
The **folder containing the video** (in this case the "DL_Track_US_example/video" folder) is already included in the "DL_Track_US_example" folder. We will now create a separate folder for the pre-trained **aponeurosis and fascicle neural networks**. In case you have not downloaded the **models**, please do so now (link:). Place them in a subfolder of the "DL_Track_US_example" folder for the moment, like "DL_Track_US_example/**models**". You will make use of these **neural networks** later as well, when you analyse your own images outside of this tutorial. Of course you can move them to a different folder then. Your folder structure inside the "DL_Track_US_example" folder should now look something like this with **the neural networks** placed in the **models** folder.



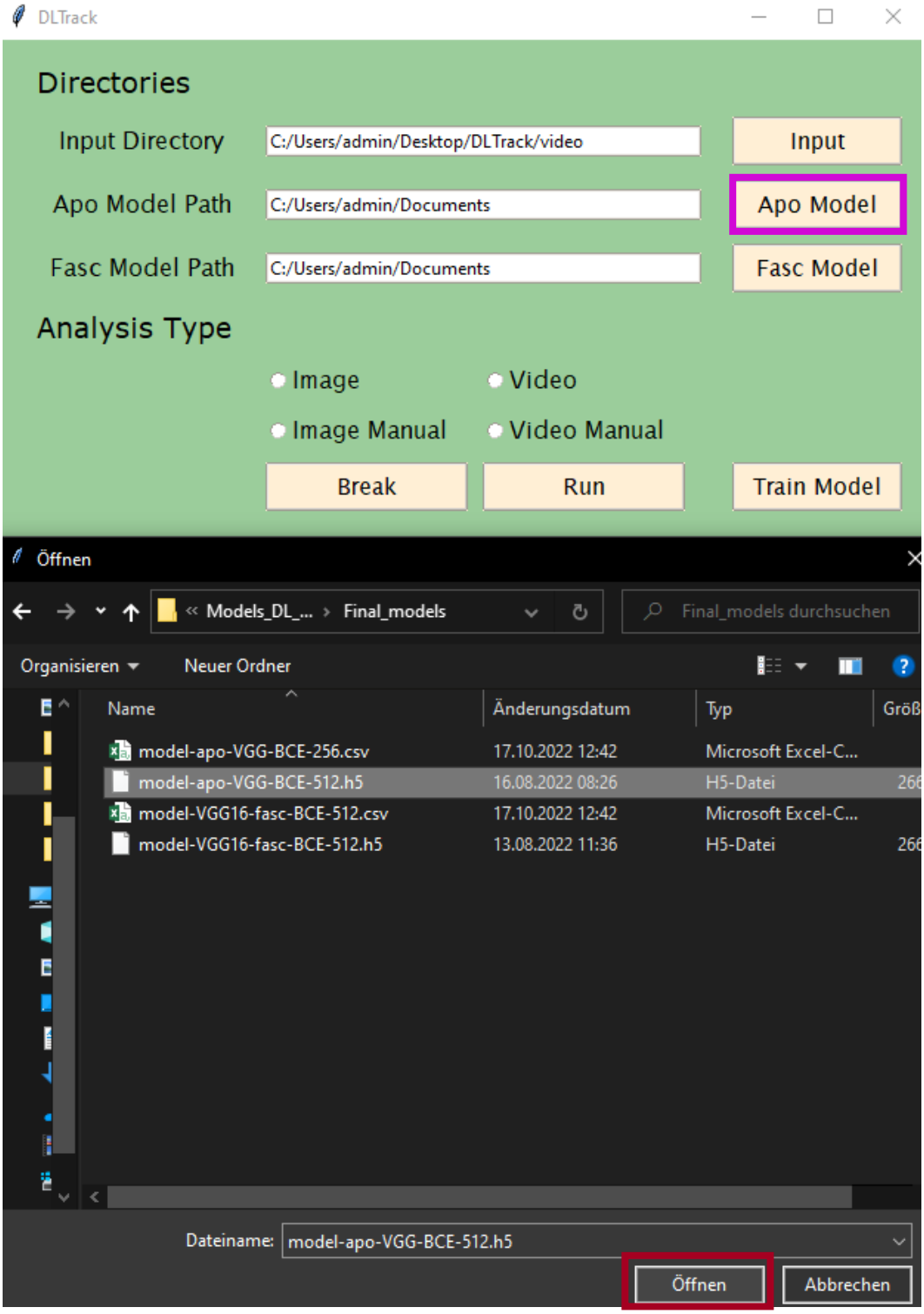
2. Specifying Input Directories in the GUI

Finally you can start with the actual analysis! The first step of every analysis type in DL_Track_US is to specify the input directories in the graphical user interface (GUI). We assume that you have already opened the GUI. (If not, take a look at the first chapter of this document “Starting the GUI”.).

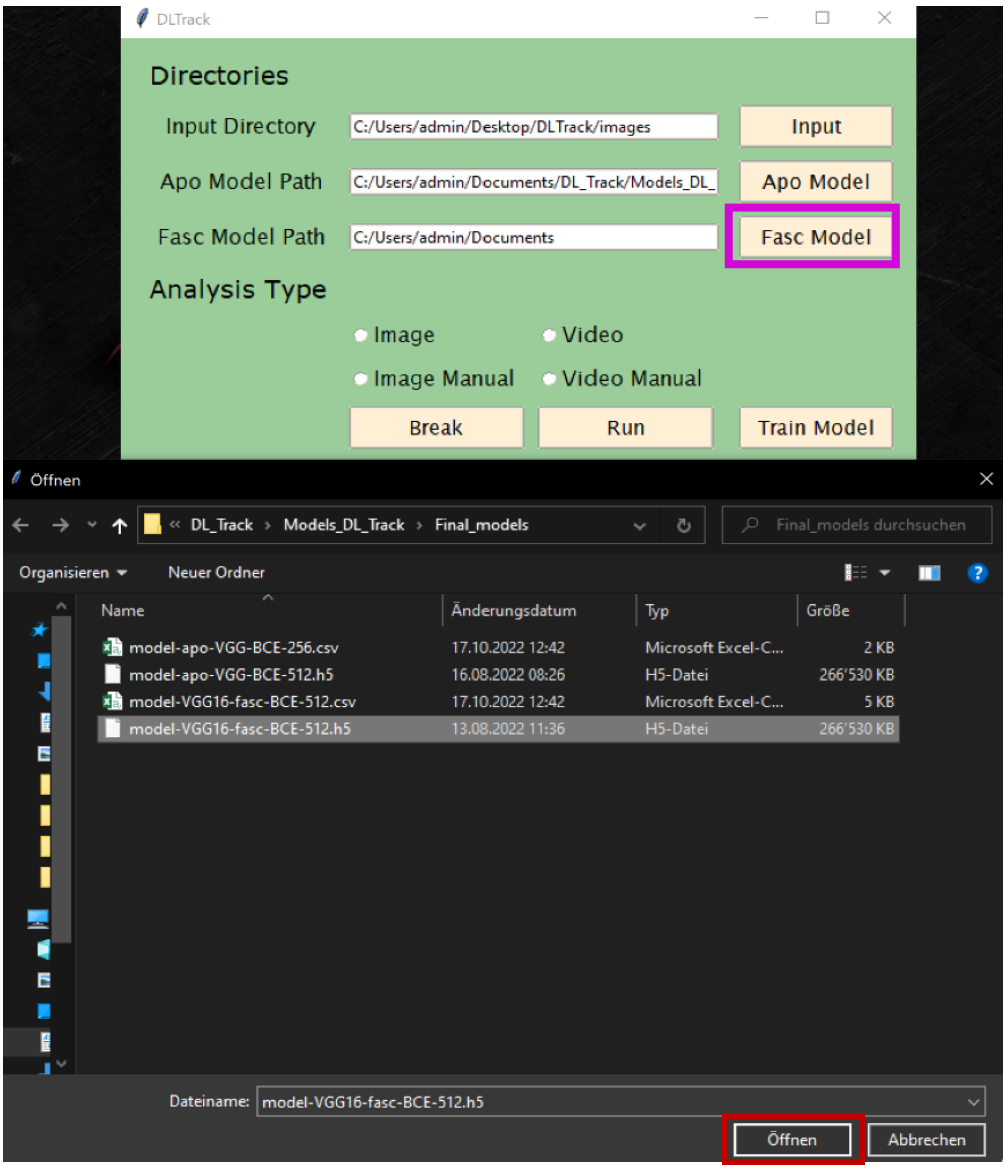
You will begin with specifying the path to the folder containing the **video** to be analysed. Remember this was the folder “DL_Track_US_example/**video**”. By clicking on the **Input** button in the GUI a selection window opens were you need to select the images folder. Click **select folder** to specify the path in the GUI.



Now, you will specify the absolute path to the **aponeurosis neural network**. Remember that you placed it in the "DL_Track_US_example/**models**". By clicking on the **Apo Model** button in the GUI a selection window opens were you need to select the **aponeurosis neural network** in the models folder. Click **open** to specify the path to the **aponeurosis neural network** in the GUI.



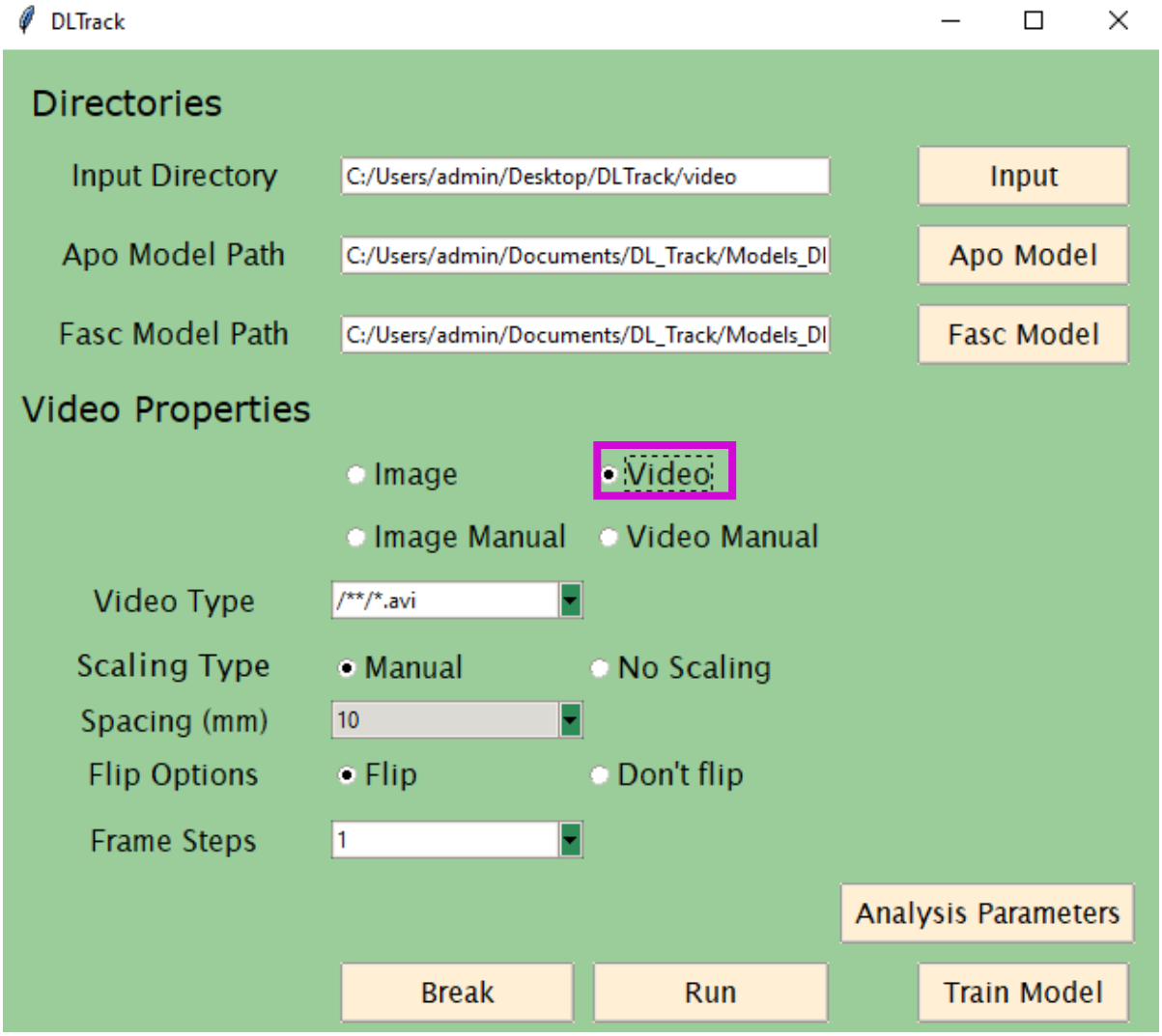
Next, you will specify the absolute path to the **fascicle neural network**. Remember that you placed it in the "DL_Track_US_example/**models**". By clicking on the **Fasc Model** button in the GUI a selection window opens were you need to select the **fascicle neural network** in the models folder. Click **open** to specify the path to the **fascicle neural network** in the GUI.



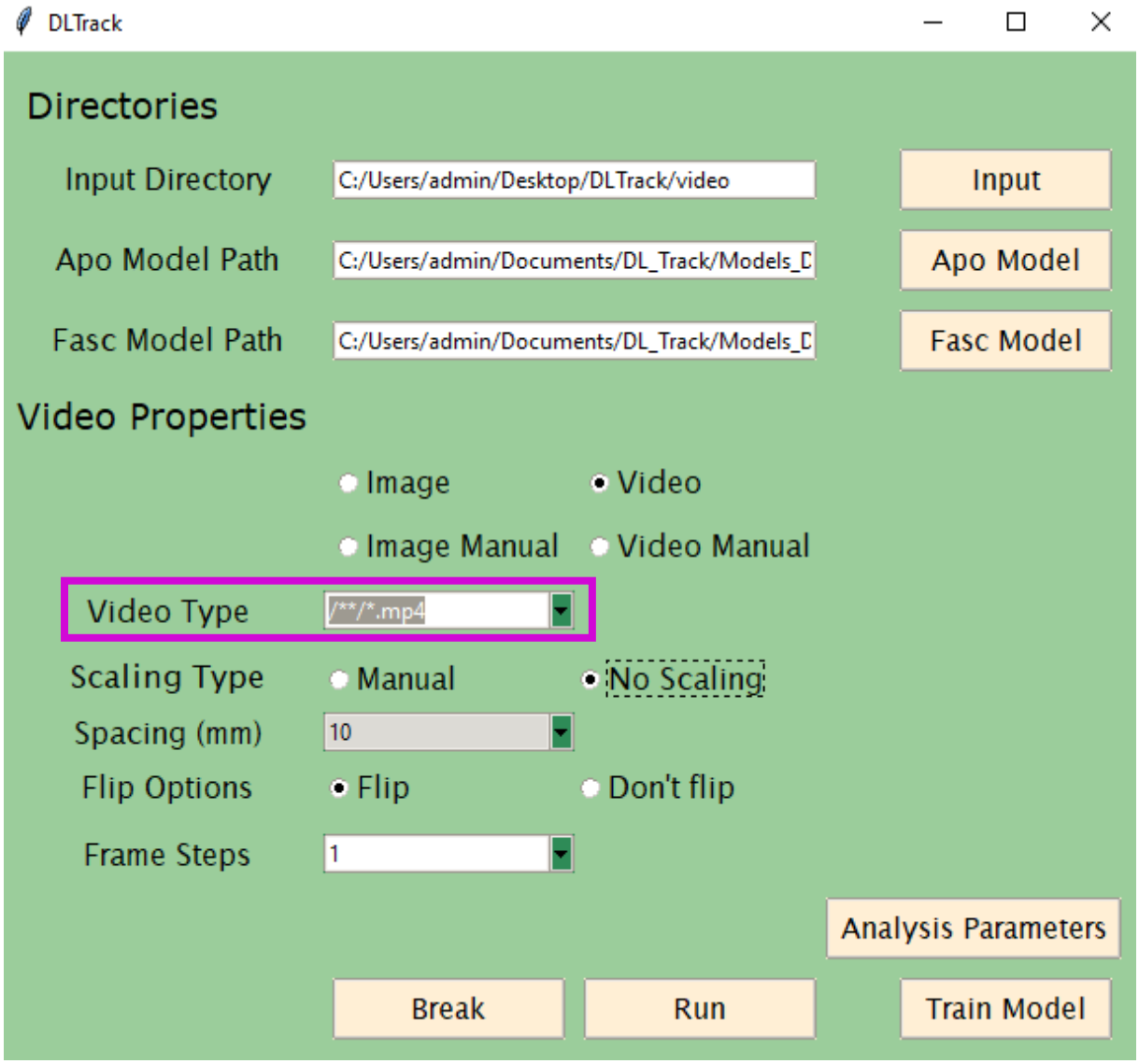
You have now successfully defined all the input directories required for automated image analyses with DL_Track_US. In the next section you will specify all relevant analysis parameters, including the analysis type. We will also explain what each parameter is used for.

3. Specifying Relevant Parameters

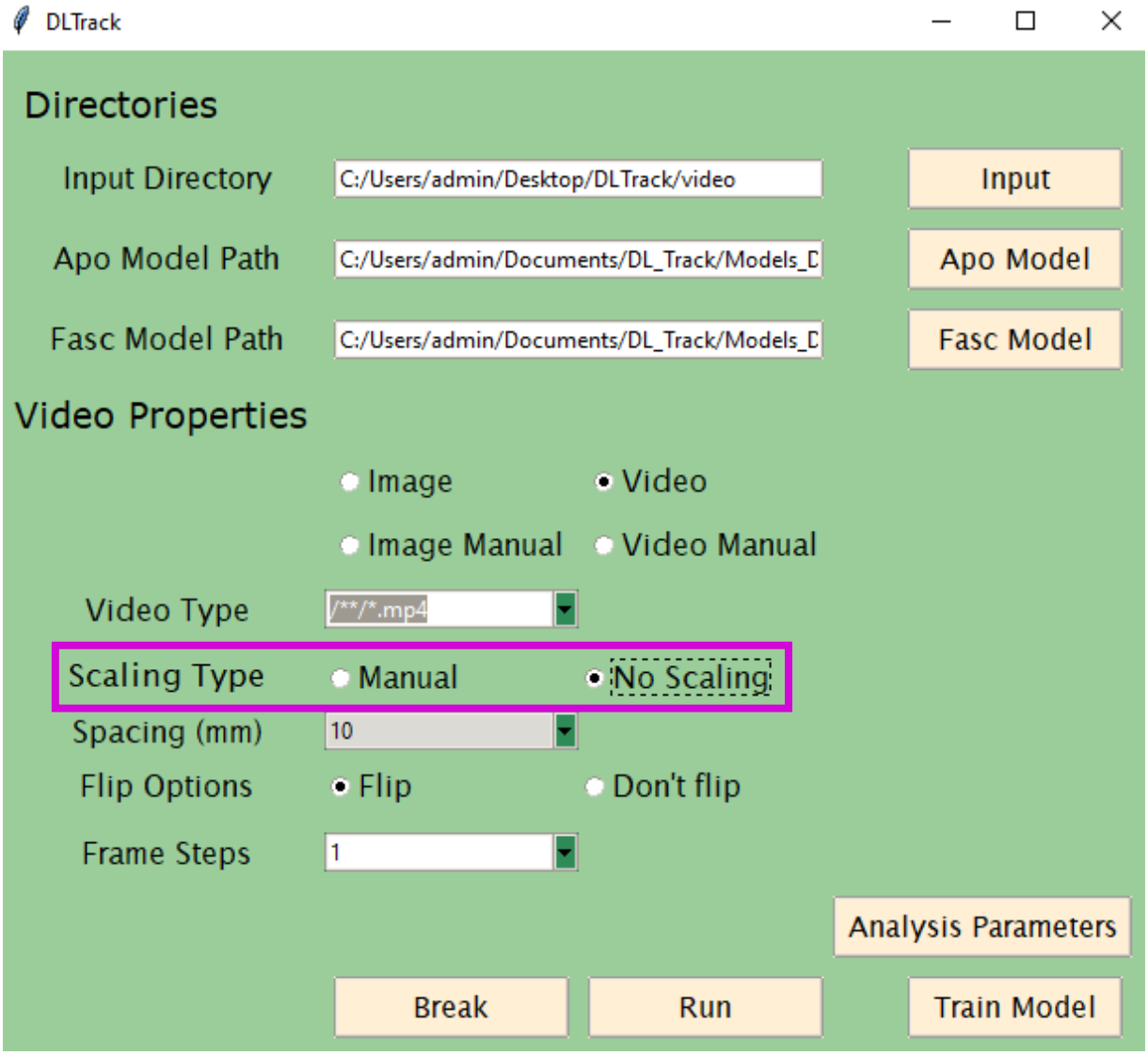
As a first step, you will select the right analysis type in the GUI. Since this section is about automated video analysis, please select the **Video** radiobutton. You can see that the GUI unfolds and several other parameters appear. You will set those in the next steps on the next page.



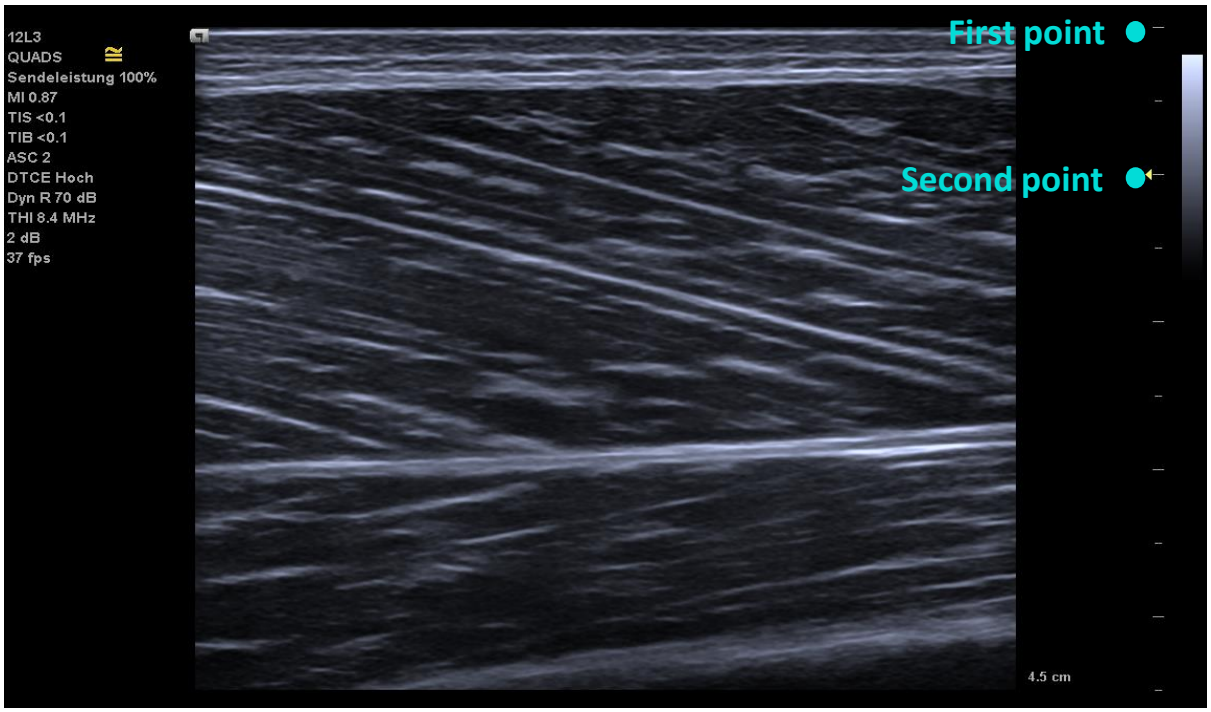
You now need to specify the **Video Type**. The ending of the **Video Type** must match the ending of your **videos**, otherwise no files are found by DL_Track_US. You can either select a pre-specified ending from the dropdown list or type your own ending. Please keep the formatting similar to those **Video Type** provided in the dropdown list. The **video** in the "DL_Track_US_example/**video**" folder are of the **Video Type** ".mp4". Thus, you should select the "/**/*.mp4" **Video Type.**



Subsequently, you need to specify the video **Scaling Type**. Scaling in general has the huge advantage that the resulting estimated muscle architectural features are in centimetre units rather than pixel units. There are two **Scaling Types** in the DL_Track_US package. For this tutorial however, you will select the **"No Scaling"** option as displayed below. We will explain the other **Scaling Type** on the next.

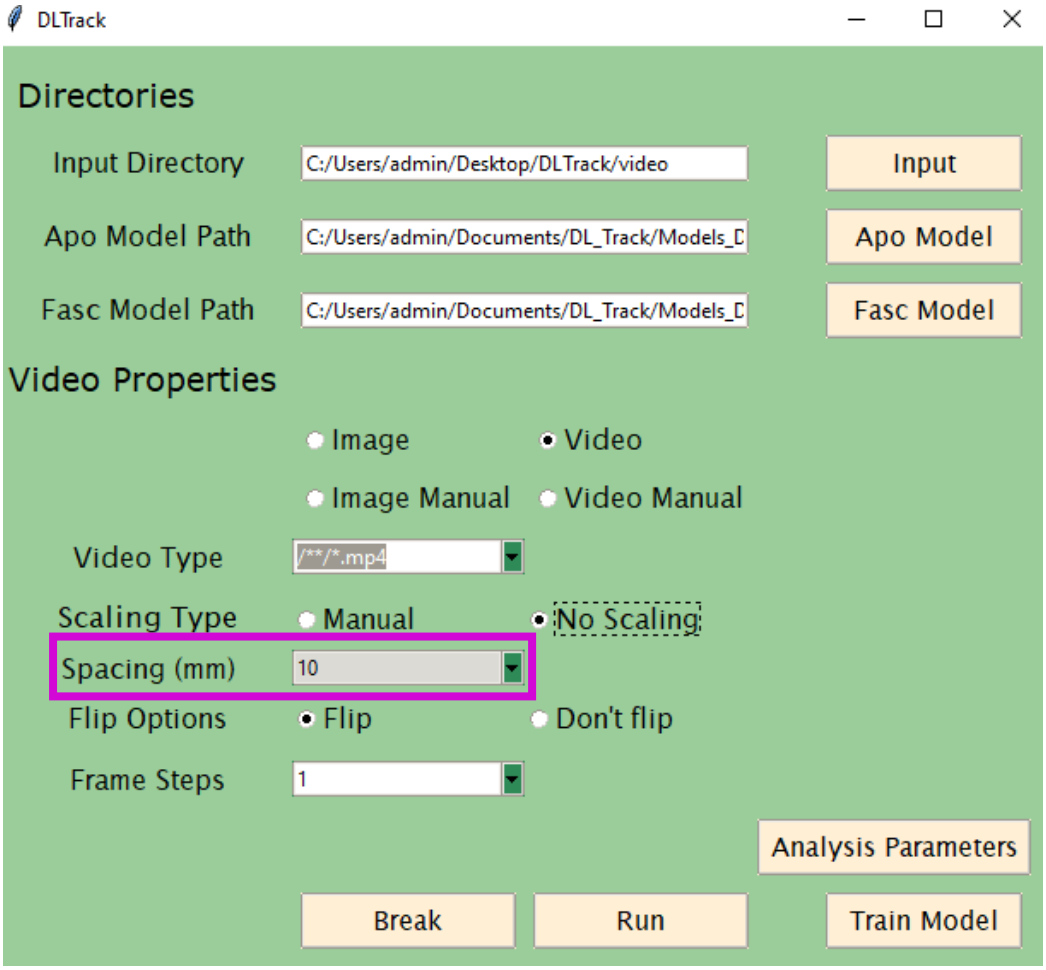


The other **Scaling Types** is “Manual”. This **Scaling Type** requires input from the user. When you choose “Manual” as your Scaling type, you need to manually place **two points** on the first video frame using the left mouse button. This step is similar to the “Manual” scaling option for automated and manual image analysis.



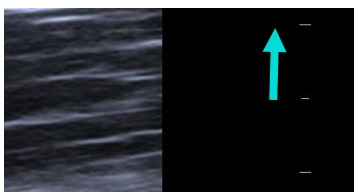
No worries, you do not actually need to draw on the video frame. Just click one time with your left mouse button to record the **first point** (nothing will be displayed on the video frames during actual analysis). Place the **second point** at a known distance of either 5, 10, 15 or 20 millimetre. The distance you chose must be represented in the Scaling (see next page) parameter in the GUI.

Whenever you use **“Manual”** as your Scaling Type, please make sure that the minimum distance between the scaling bars or the known distance between your manually specified points is represented in the **Spacing** parameter. For the **“No Scaling”** Scaling type, this is not necessary.

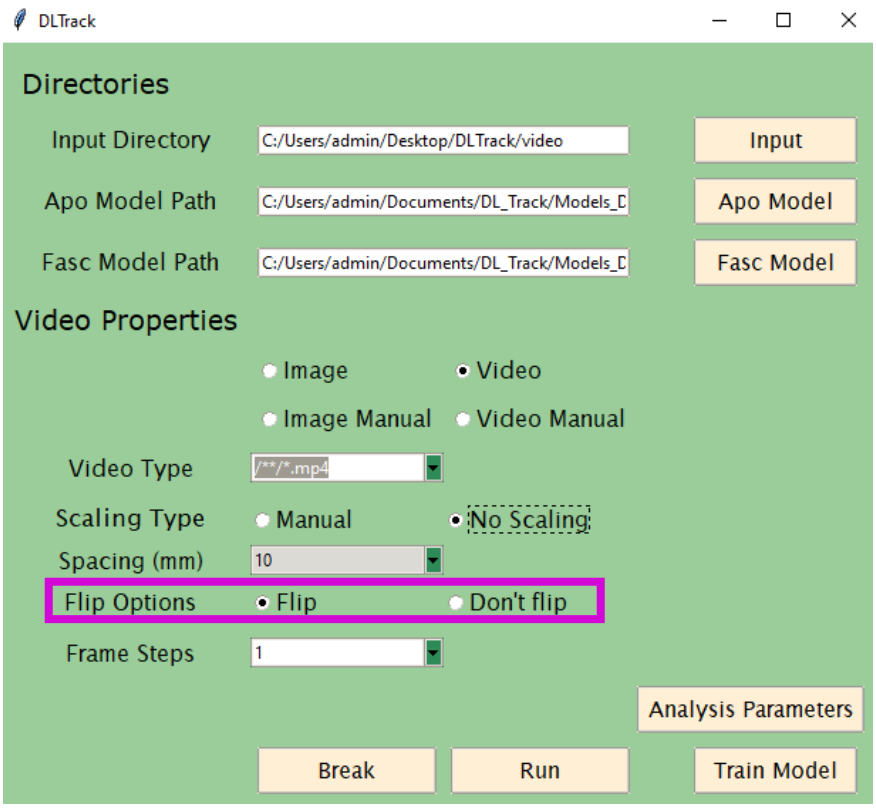


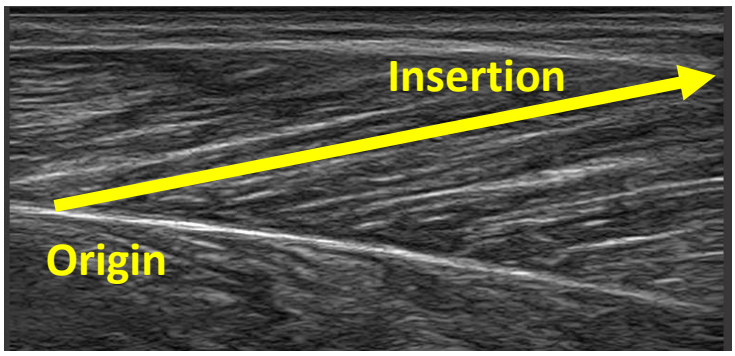
You can select the **Spacing** parameter only from the dropdown list as 5, 10, 15 or 20 millimetre. For this tutorial it is not necessary to select anything, as the **Spacing** parameter is not used during an analysis with Scaling Type **“No Scaling”**.

So far, we haven’t explained how to determine the minimal **distance** between the scaling bars in a video frame. This is simply the **distance** in millimeter between the two nearest scaling bars in the frame. If you do not know this **distance**, please use **“Manual”** or **“No Scaling”** Scaling Type. For example in the frame from before, the **distance** between the nearest bars is 5 millimetre. We know that because the distance between the bigger bars is always 10 millimetre.

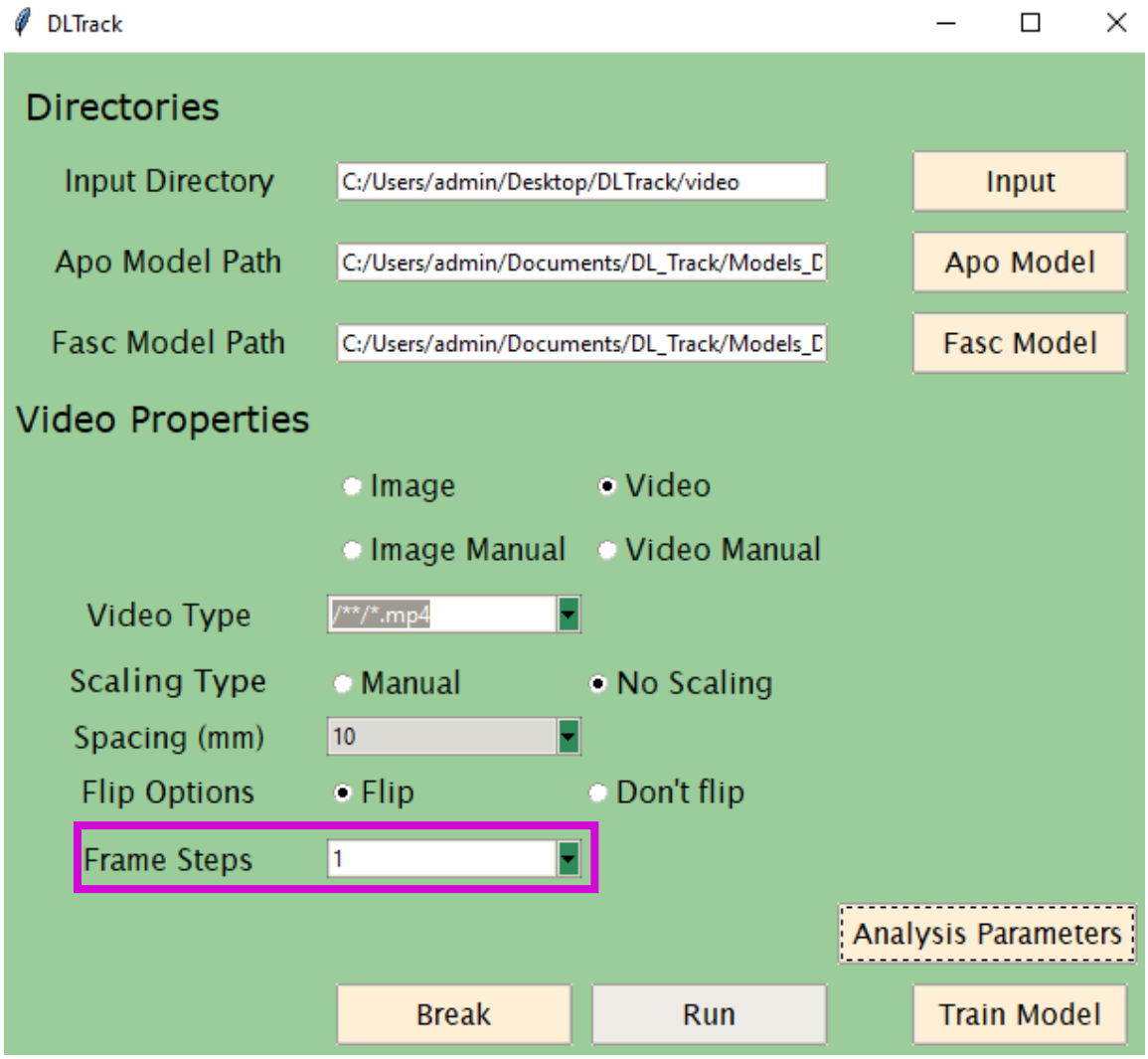


Another parameter that you need to specify is the **Flip Options** parameters. The **Flip Options** parameter determines if the whole **video** is flipped along the vertical axis. **"Flip"** stands for flipping the video, whereas **"Don't Flip"** means please do not flip the video. The example **video** must be flipped. Its fascicle orientation is **incorrect**, with fascicles originating at the bottom right and inserting on the top left. This would confuse our models. Below is a visual representation of a **correct** fascicle orientation. The fascicles are originating at the bottom left and are inserting on the top right. If the fascicles in your **video** outside of this tutorial are orientated differently, please specify "Flip" in the **Flip Options** parameter for those **videos**. Note that all videos in the specified input folder, in this case the DL_Track_US_example/**video**" folder, MUST have the same fascicle orientation, since the **Flip Option** is applied to all of them. It does however not matter if this is the correct or incorrect fascicle orientation.



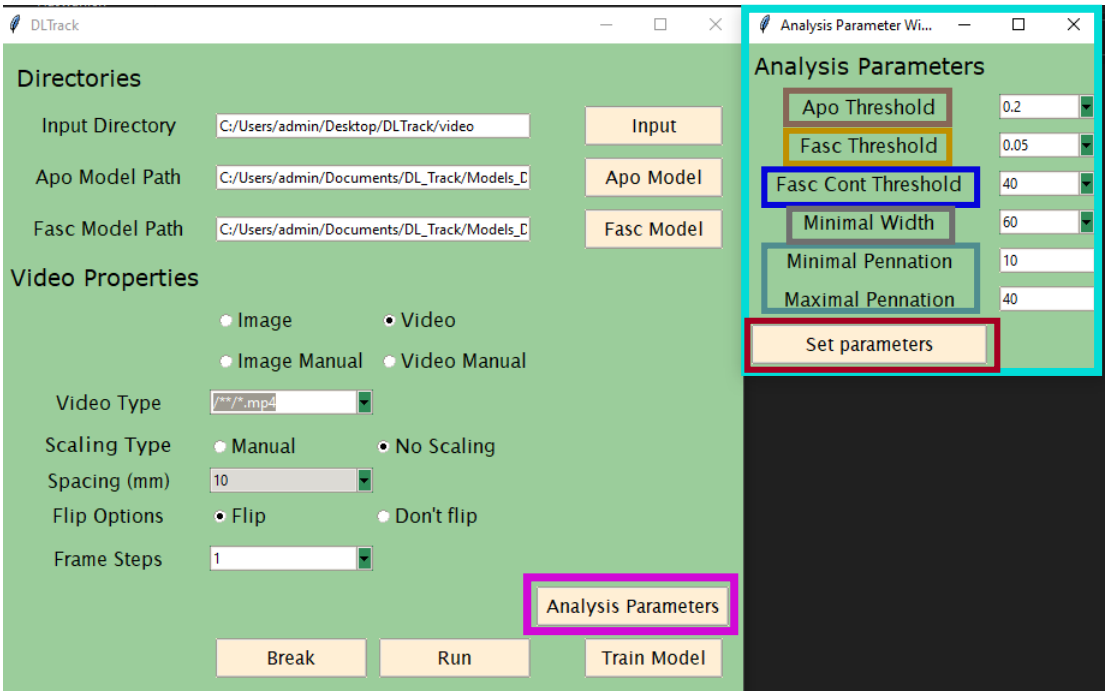


The next step in the automated video analysis using the DL_Track_US package is to specify the **Frame Steps**. You can either select a pre-specified **Frame Step** from the dropdown list or type your **Frame Step**. The **Frame Step** is used during the analysis as a step size while iterating through all the frames in a video. In this tutorial you should specify a **Frame Step** of 1. This means that every video frame is analysed. With a **Frame Step** of 3, every 3rd frame is analysed. With a **Frame Step** of 10, every 10th frame an so on. Although **information is lost** when you skip frames during the analysis, it also **reduces the overall analysis time**. IMPORTANT: when you skip frames in videos outside of this tutorial, the frame rate with which the video was acquired should be high a enough and the captured motion a slow one. Otherwise, too much relevant information is lost.



4. Specifying Analysis Parameters

Awesome, you have now successfully selected all relevant parameters in the main GUI window. As a LAST step, you need to specify the analysis parameters for the **aponeurosis and fascicle neural networks.** When you press the **Analysis Parameters** button, a separate window will pop-up.

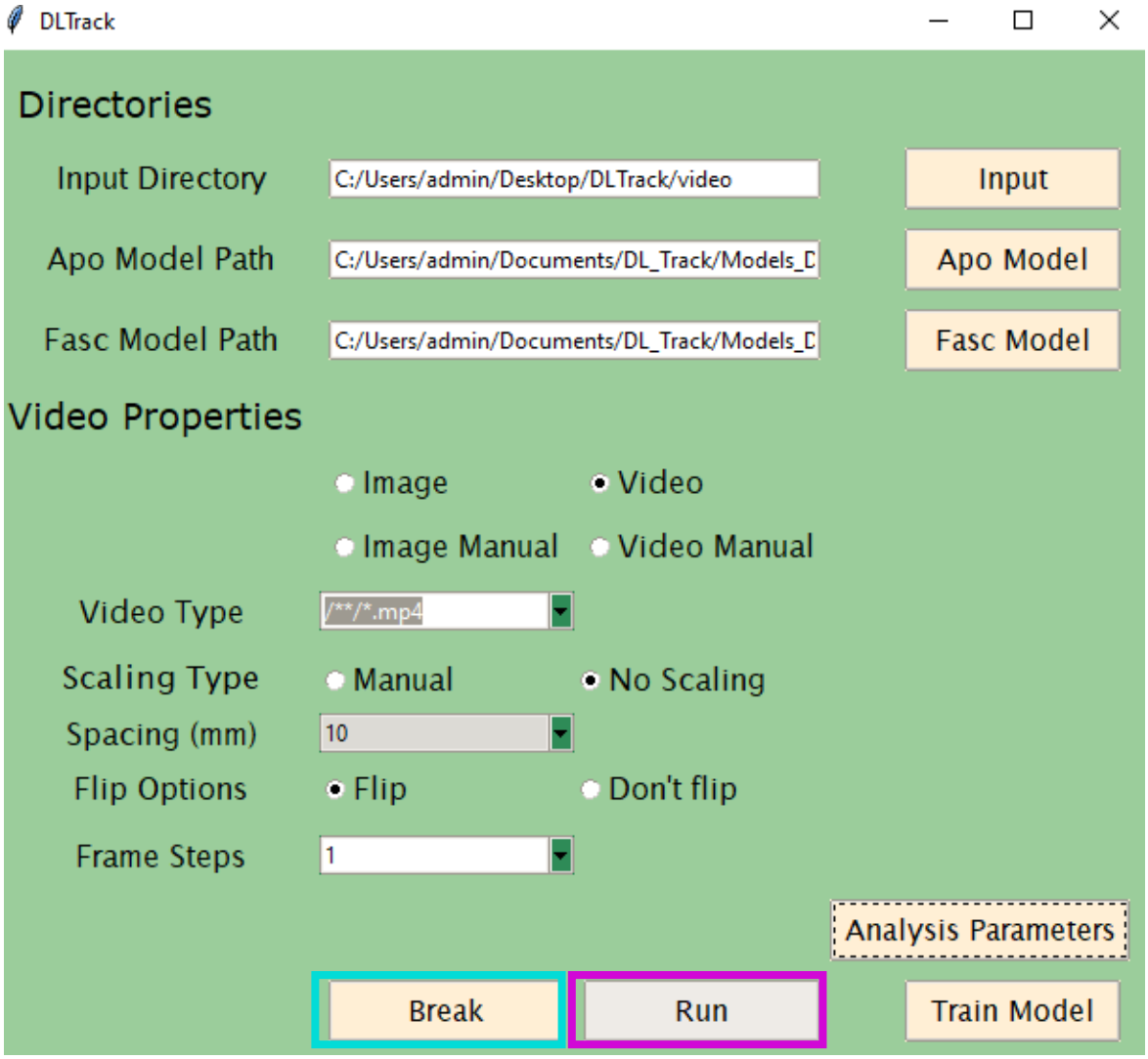


In the **Analysis Parameter window**, all parameters used by the **aponeurosis and fascicles neural networks** during inference are specified. We will give a short explanation to each of those parameters. The **Apo Threshold** parameters determines the threshold of the minimal acceptable probability by which a pixel is predicted as aponeurosis. The lower, the more pixels will be classified as aponeurosis. The **Fasc Threshold** is the same thing just for fascicle segments. The lower the **Fasc Cont Threshold**, the shorted the minimal acceptable length of detected fascicle segments to be included in the results. The **Minimal Width** determined the minimal acceptable distance between superficial and deep aponeurosis. **Minimal and Maximal Pennation** describe the respective minimal and maximal pennation angle that is physiologically possible in the analysed video frame/muscle.

For this example, you can leave all parameters the way they are. You can set the parameters by clicking the **Set parameters** button, the **Analysis Parameter window** will then close automatically. Please make sure to adapt these parameters according to your images in analyses outside of this example. For future analyses, it's best you test the ideal parameter configuration to get the best prediction results on your images in a small sample prior to the actual analysis.

5. Running / Breaking DL_Track_US

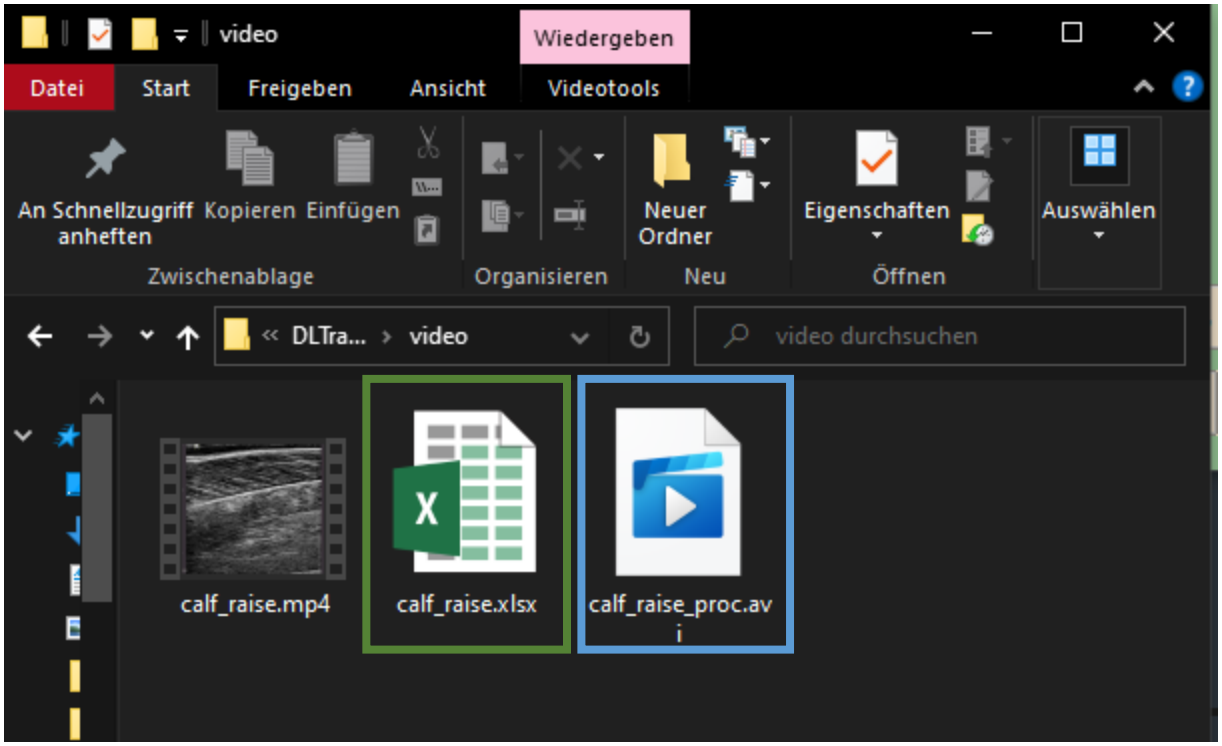
Now its time to start the actual analysis of the example video in the "DL_Track_US_example/video" folder. You can do so by clicking the **Run** button in the main GUI window.



When you started the GUI using the command prompt in any way, you will see that the analysis is started by the statements printed in the prompt. Moreover the currently analysed frame with the segmentation results will pop up. When you started the GUI using the executable, you can see only the current frame with the segmentation results pop up. Moreover, you can see that there is a **Break** button placed in the GUI as well. Clicking the Break button allows you to stop the analysis at any point. The currently evaluated frame will be processed and then the analysis is terminated.

Take a look at the next pages to see what happens during the analysis and once the analysis is finished.

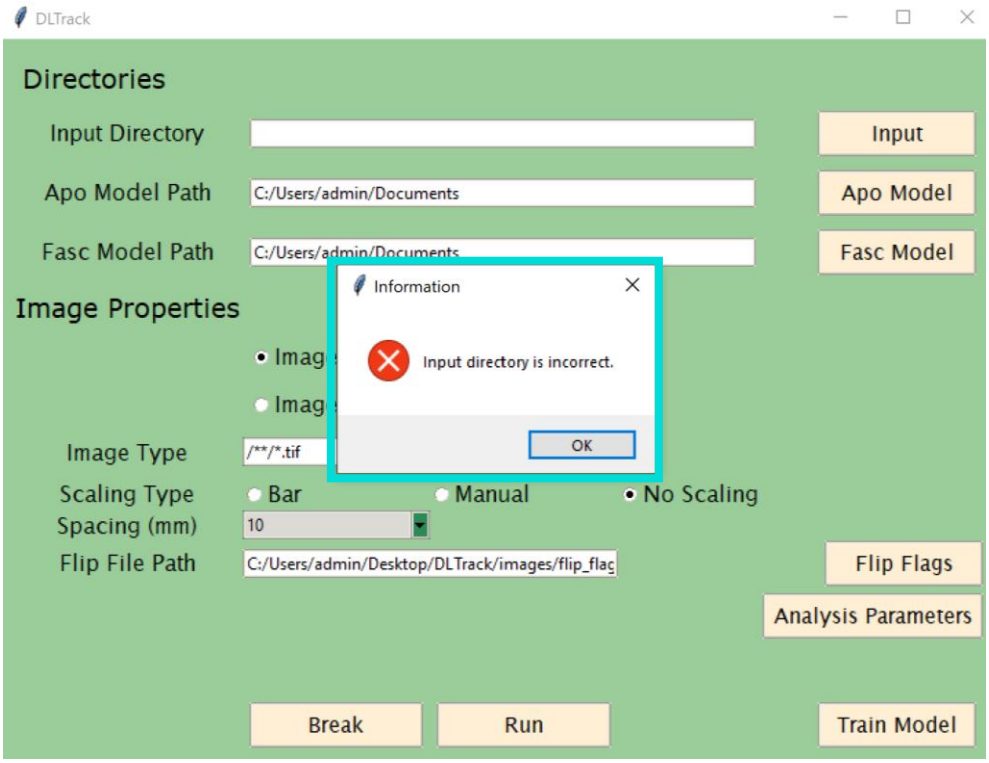
Subsequently to clicking the Run button in the main GUI, navigate again to the "DL_Track_US_example/video" folder in your explorer. You will see that two files will be / have been created, **calf_raise_proc.avi** and **calf_raise.xlsx**. The **calf_raise_proc.avi** file contains each the input video with overlaid segmented fascicles and aponeurosis. This file allows you to visually inspect the model outputs. In your future analysis outside of this tutorial, you should always visually inspect the **calf_raise_proc.avi** file. The **calf_raise.xlsx** file contains the actual architectural parameter estimates for each video frame. There, all detected muscle fascicle lengths and pennation angles as well a the calculated muscle thickness will be displayed. Each video frame is displayed in a separate row. Note that the **calf_raise_proc.avi** file can be opened only after the **calf_raise.xlsx**. was created.



When both files can be opened and you can see the analysis results, original image and the prediction result, we must congratulate you! You have now officially and successfully completed the DL_Track_US tutorial for automated video analysis! There is one more thing though, error handling. Take a look at the next section to get more information.

6. Error handling

Whenever an error occurs during the analysis process, the DL_Track_US GUI will open a **messagebox**. This looks always similar to this:



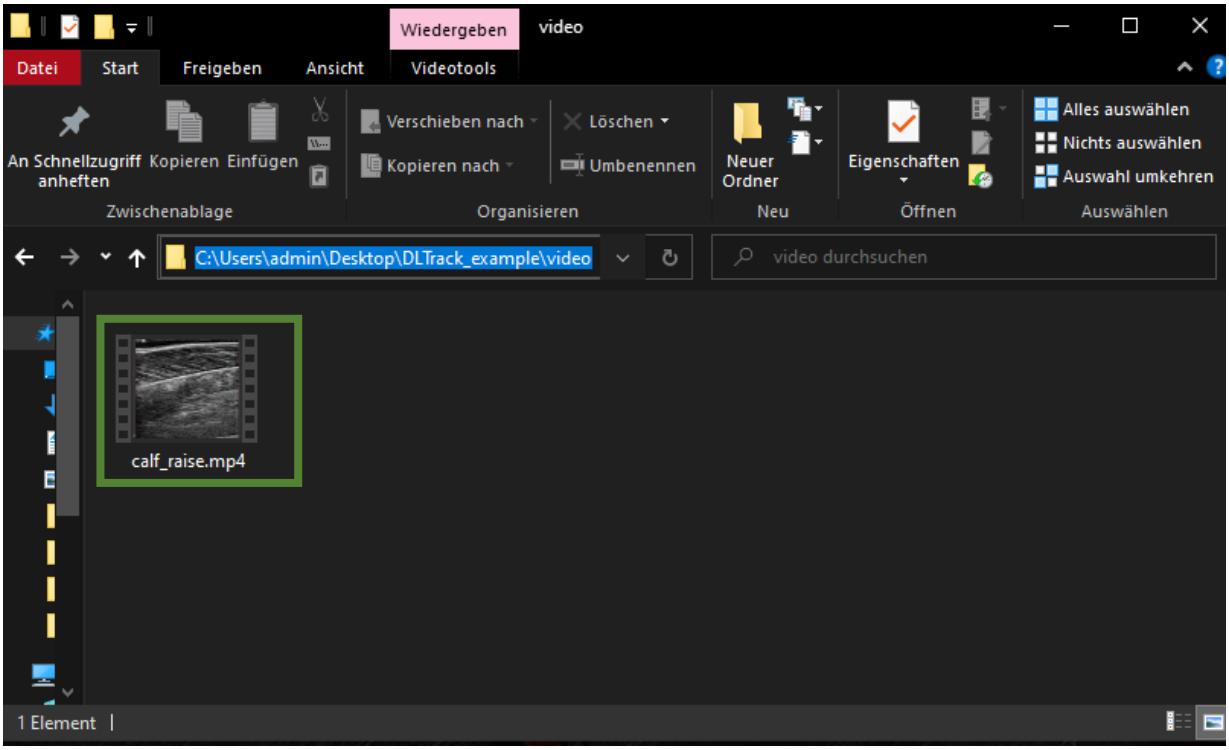
We tried to formulate these **messageboxes** as concise as possible. Just follow their instructions to fix the error and run the analysis anew. In case an error occurs that is not caught by an error **messagebox,** don't hesitate to report this in the Q&A section in the DL_Track_US discussion forum. Please take a look here how do best do this. Otherwise, you can contact us by email at paul.ritsche@unibas.ch, but we would prefer the other way.

Manual Video Analysis

The DL_Track_US python software package offers several different analysis types for analysis of human lower limb longitudinal ultrasonography images. The next and last analysis type this tutorial covers is the manual video analysis. The images are evaluated manually by drawing the muscle thickness, fascicle length and pennation angles directly on the Image. Scaling the images will ensure estimated muscle architectural parameters are converted to centimetre units. For this type of analysis, single videos are a prerequisite. These videos should be contained in a single folder, like in the "DL_Track_US_example/videos_manual" folder. If you haven't downloaded this folder, please do so now (link: DL_Track_US - Examples & Models | Zenodo). Unzip the folder and put it somewhere accessible, for example on your desktop. We will make use of the included example files extensively during this tutorial. The manual video analysis type is identical to the manual image analysis type. The only difference is that the absolute video path must be specified instead of the File Type. The video is first converted and all the contained frames are separately stored as single images. Then, each frame image is analysed separately. In the next few pages, we will look at every required step to successfully perform manual video analysis with DL_Track_US.

1. Creating a Video Directory

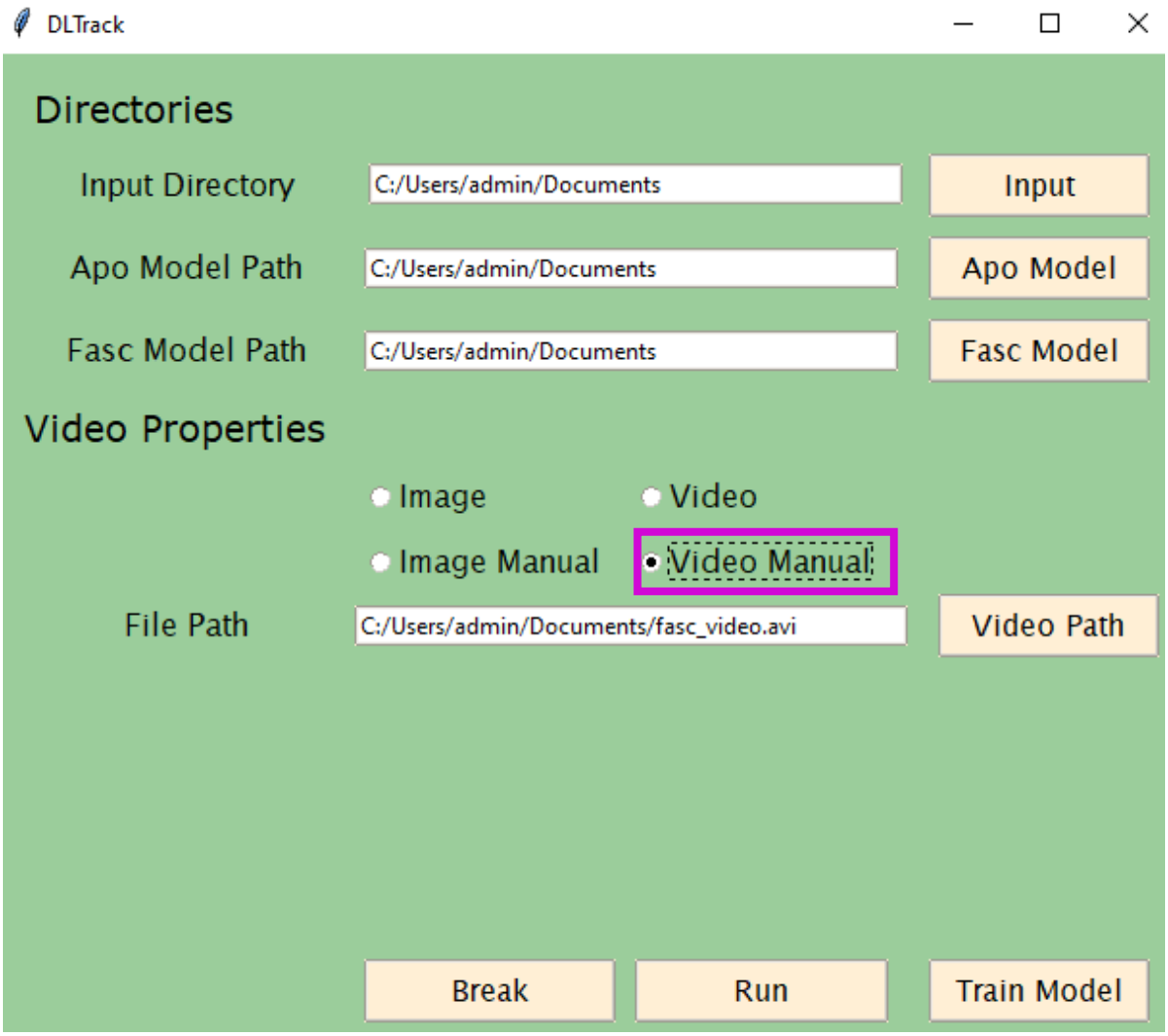
In order for DL_Track_US to recognize your videos, they should best be in a single folder (though one subfolder structure is acceptable as well). Take a look how you might structure this:



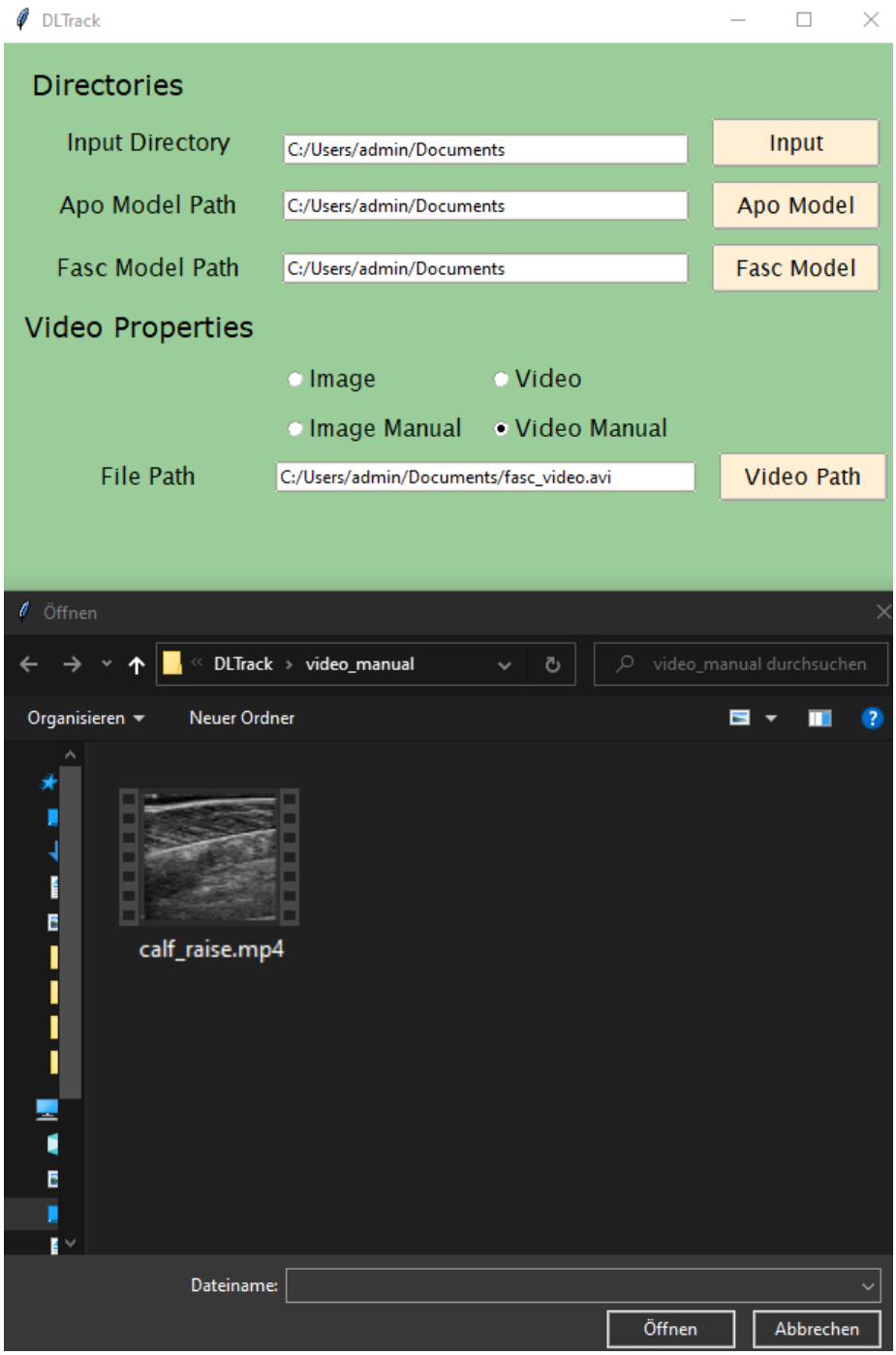
You can see in the picture above that the folder contains **one video** and is **located on the desktop**. This structure is already included in the "DL_Track_US_example" folder. We will continue with demonstrating how to create folders for the **aponeurosis and fascicle neural networks** on the next page.

2. Specifying Relevant Parameters

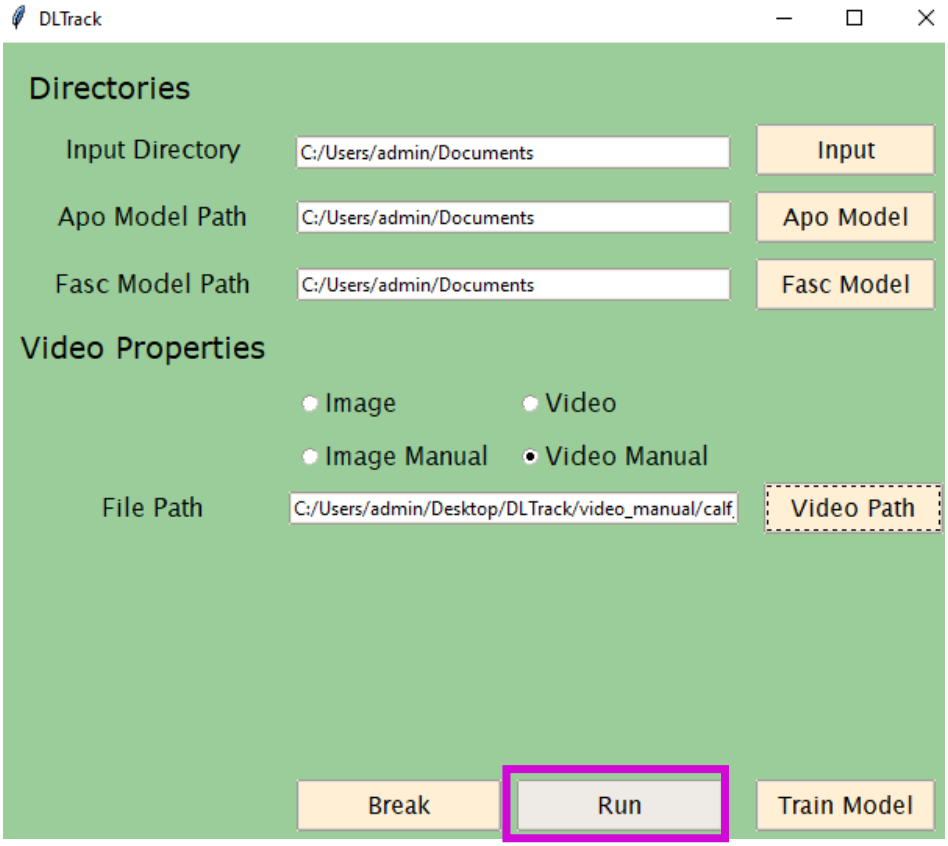
For the manual video analysis DL_Track_US analysis type, you do not need to specify any directories. Therefore, as a first step already, you will select the right analysis type in the GUI. Since this section is about manual video analysis, please select the **Video Manual** radiobutton. You can see that the GUI unfolds and another parameter appears. You will set this one in the next step on the next page.



Next, you need to specify the absolute **File Path** of the **video file** to be analysed. Remember that the example **video file** is placed in the "DL_Track_US_example/**video_manual**" folder. By clicking on the **Video Path** button in the GUI, a selection window opens were you need to select the example **video file** in the **video_manual**. Click **open** to specify the path to the **video file** in the GUI.



Allright, once you have specified the video file path, you can start with the analysis of the **example video** contained in the "DL_Track_US_example/**video_manual**" folder. You can start the analysis by clicking the **Run** button in the main GUI.



Once you clicked the Run button, the "Manual Analysis window" will pop up. If this happens, congrats to you! You have entered all relevant parameters for the manual video analysis correctly!

From here, all further steps are identical with the manual image analysis. The only difference though is that in the folder of the inputted video, a new folder is created containing all the single image frames.

The scaling of the image, extending of the aponeuroses, single segment muscle thickness measurements, three segment muscle fascicle measurement and two segment pennation angle measurement are identical. Saving the results (with the very important button), continuing to the next image frame, terminating the analysis process and error handling is identical. **Therefore, we kindly refer you to section 4 of this tutorial Manual Image Analysis** (because we don't want to repeat ourselves) **to see how the all the architectural parameters are analysed.** We are confident that you can do this on your own now!

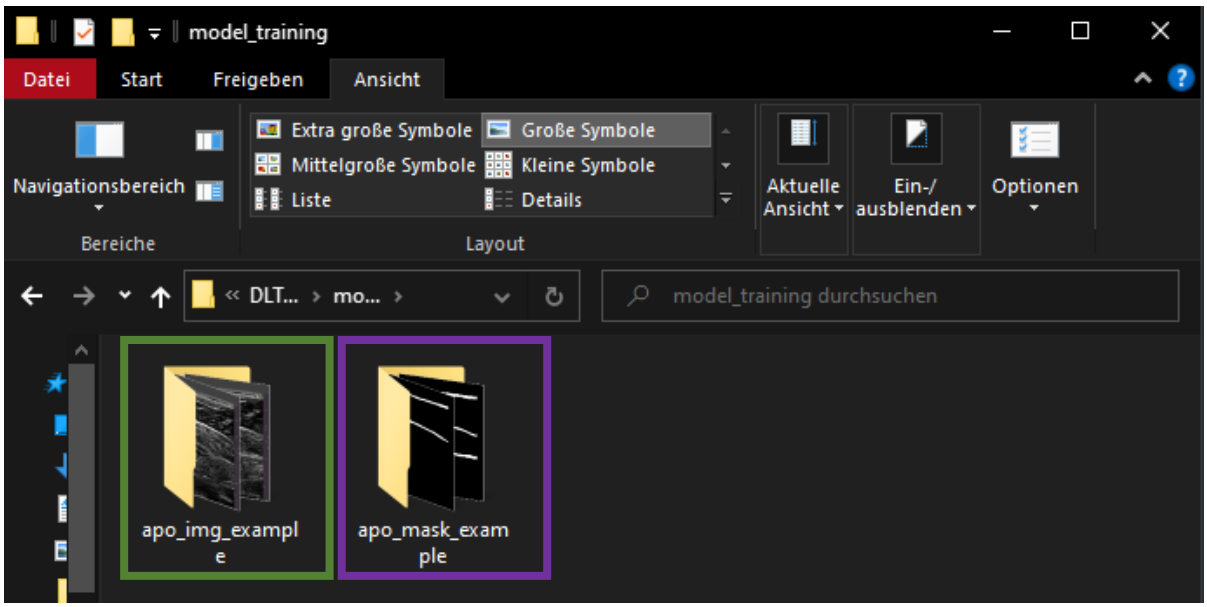
Training Your Own Networks

Not only does the DL_Track_US 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 to have a working GPU setup, otherwise model training will take much longer. How to setup you GUI for DL_Track_US is described in the installation guidelines of our Github repository. In the next few pages, you will learn to train your own neural to train your own neural networks. This is also the last part of this tutorial. After completion of this chapter, you will know the DL_Track_US GUI as well as the back of your hand. If you don't have any experience with training deep neural networks, we strongly advise to work with the pre-defined settings. Otherwise, you are of course free to choose. For this introductory tutorial, we would however like you to use the pre-defined settings no matter what your experience level is. Although you can generally adapt a number of parameters during training, you cannot change the neural network architecture from the GUI (of course you could modify source code to do so). This is because during experimenting with different model architectures, we found a combination of a on imagenet pre-trained VGG16 encoder and a standard U-net decoder to be the best performing model. Thus, all the models trained using the GUI will have this architecture. To explain you the parameters used during model training that are adaptable from the GUI is out of the scope of this tutorial. However, we would like to refer you to this excellent introductory course in case you are a deep learning beginner. Training your own networks for muscle architecture analysis requires pairs of original images and manually labelled masks. Examples are provided for you in the "DL_Track_US_example/model_training" folder. If you haven't downloaded this folder, please do so now (link:). Unzip the folder and put it somewhere accessible, for example on your desktop. We will make use of the included example files extensively during this tutorial. In this tutorial, you will only learn to train model that segment the muscle aponeuroses. Yes that's right, segment the fascicles requires a separate model because the task is so different. However, everything works the same, you just need to use fascicle images and masks instead of aponeurosis images and masks. Examples of those are provided in the "DL_Track_US_example/model_training" folder as well.

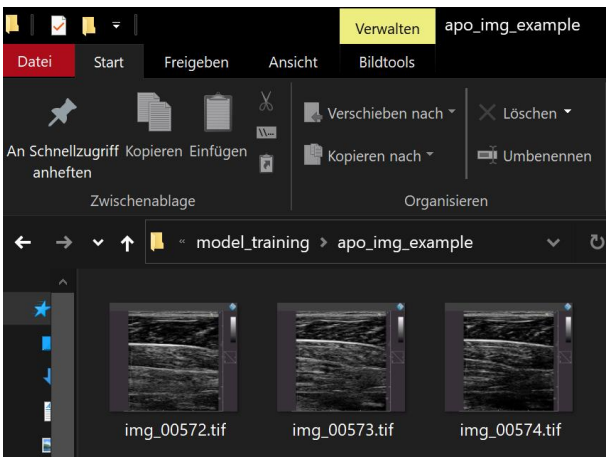
Anyway, enough said. Let's get to the model training part. The most important part is data preparation and labelling. This is where you will start.

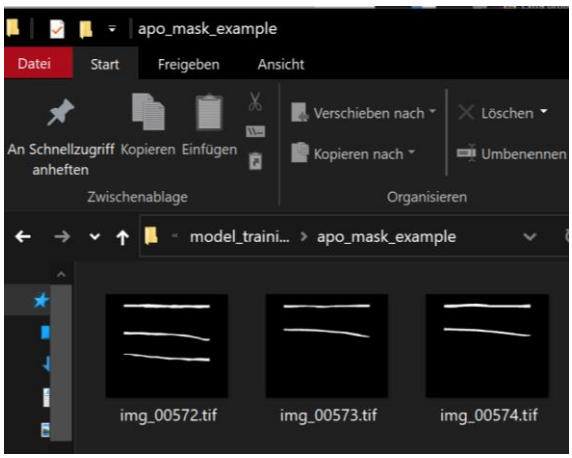
1. Data Preparation and Image Labeling

In order for DL_Track_US to recognize your **aponeurosis images** and **aponeurosis masks**, they should best be in different single folders. Take a look how you might structure this:



You can see in the picture above that the "DL_Track_US_example/model_training" folder contains two subfolders, **"apo_img_example"** and **"apo_mask_example"**. The original images are located in the **"apo_img_example"** folder whereas the corresponding masks are located in the **"apo_maks_example"** folder. We advise you to keep a similar folder structure when you train your own models outside of this tutorial. When you take a look below, you can see that the **original image** and the **corresponding masks** have exactly the same name. This is **SUPER MEGA** important. Otherwise, the model is trained using the wrong masks for the images. You do not want this!

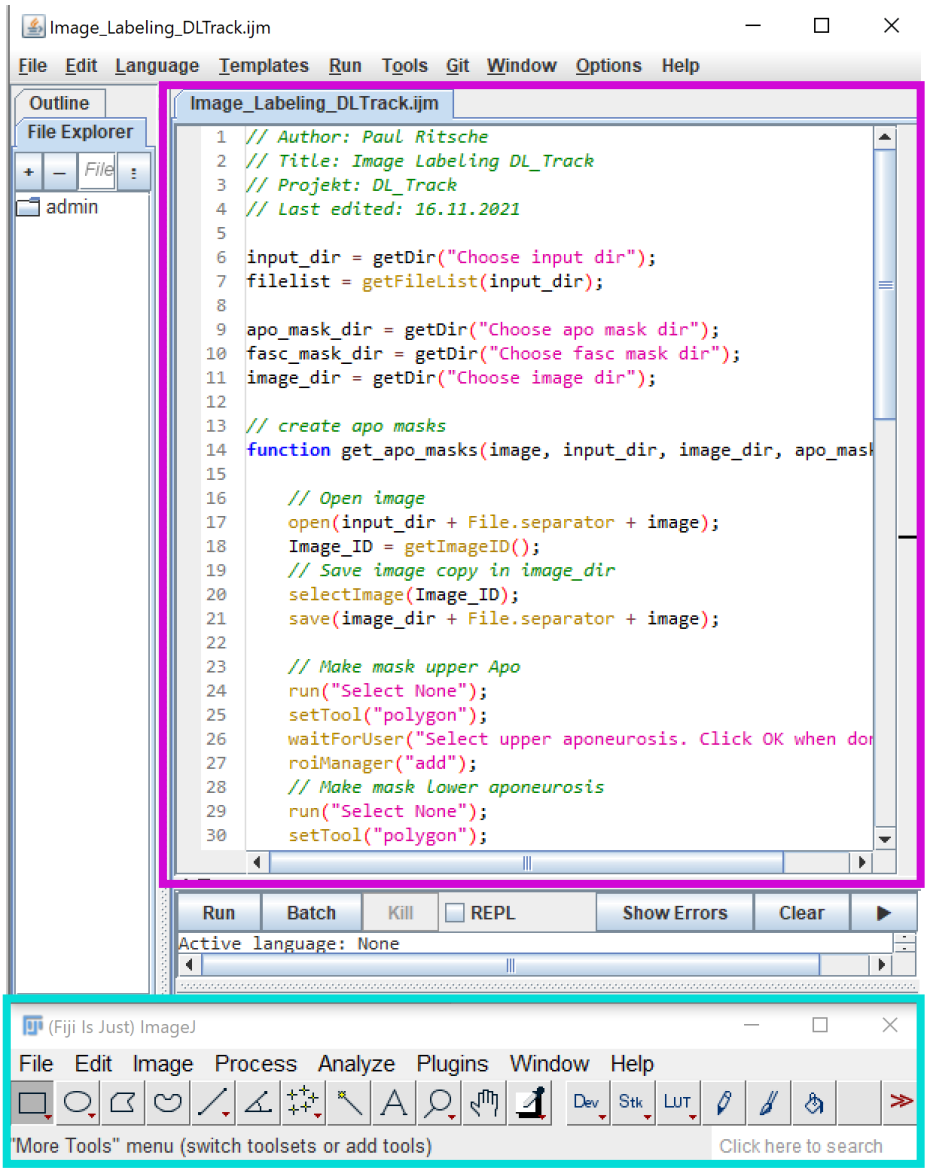




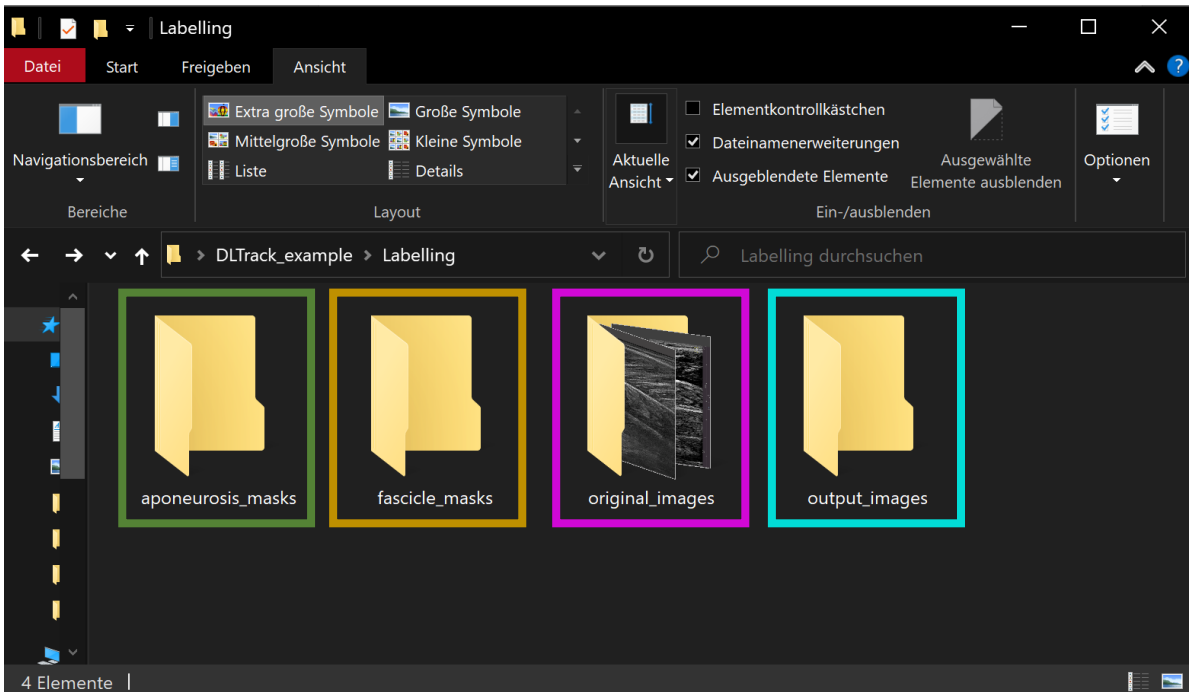
Caution, this section is not part of the tutorial. It is merely additional information for you that is not required to run the GUI and follow the example model training tutorial. The actual tutorial continues on page 72.

When you want to train your own networks outside of this tutorial, you need to label your original ultrasonography images of the lower limbs. We have prepared you the instructions how to use the **automated script** that we provide (this script does not automatically label the images, but automates the selection processes and image / mask saving). The software you will perform the labelling in is called **ImageJ / Fiji**. You can download the software here. The **automated script "Image_Labeling_DL_Track_US.ijm"** is located in the folder "DL_Track_US/docs/labeling/" in our Github repository.

The easiest way to run the **"Image_Labeling_DL_Track_US.ijm"** script is by simply drag and drop it in the **Fiji / ImageJ** window. Therefore, **Fiji / ImageJ** must be already running. As a result, the script will opened.

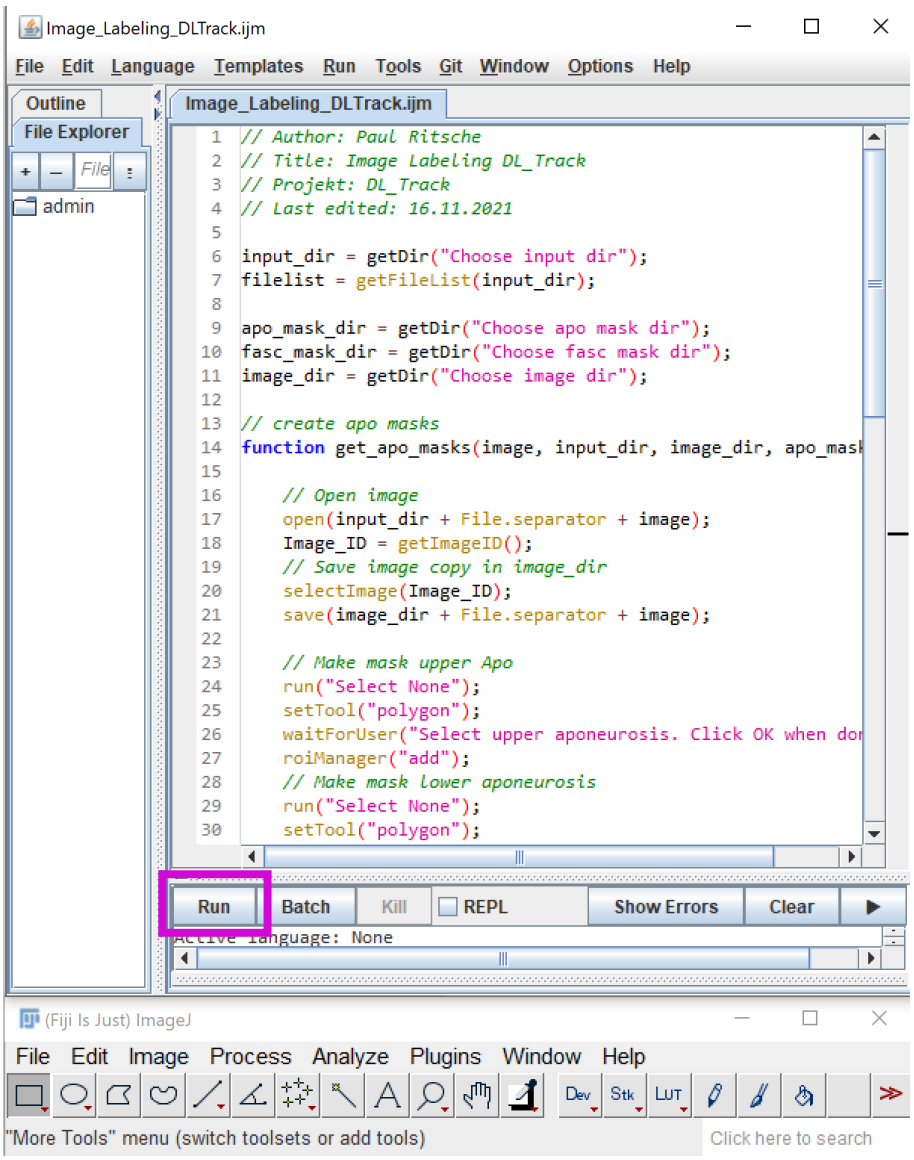


Before you can start the labelling process however, you need to create four folders in an easily accessible place (perhaps your desktop?). One folder containing the **original images** you want to label. Then create three more folders, one named **“output_images”**, the second called **“fascicle_masks”** and the third called **“aponeurosis_masks”**. In the **“output_images”** the original images are saved with an adapted name. In the **“fascicle_masks”** and **“aponeurosis_masks”** folder the respective masks are saved with the same name as the corresponding image in **“output_images”**.



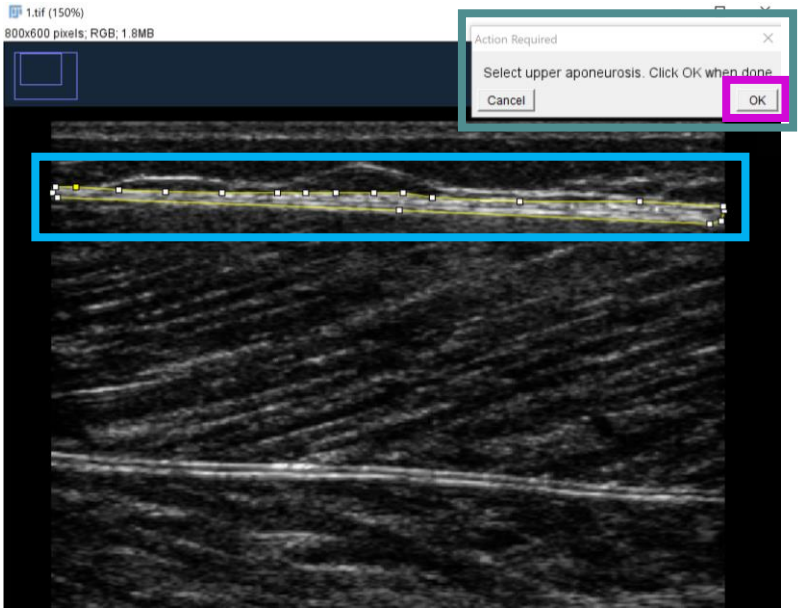
Take a look at the next page on how to continue once you created the above demonstrated folder structure.

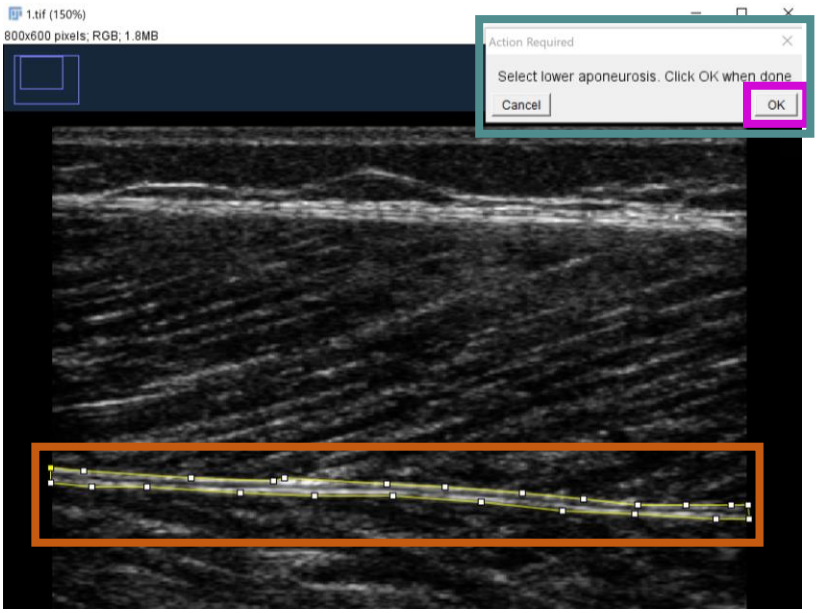
Once you created the above demonstrated folder structure, simply press the **Run** button in the Fiji / ImageJ API to start the "Image_labelling_DL_Track_US.ijm" script.



Please follow the instructions appearing in the messageboxes. To begin with, you need to specify the four directories. The first directory you need to select is the original image folder (called input dir). The second folder is the "aponeurosis_masks" folder (called apo mask dir). The third is the "fascicle_masks" folder (called fasc mask dir). The last folder you need to specify is the "output_images" folder (called image dir). Subsequent to specifying the directories, you are required to create the masks. First the aponeurosis mask, then the fascicle mask. How to do this is demonstrated on the next page.

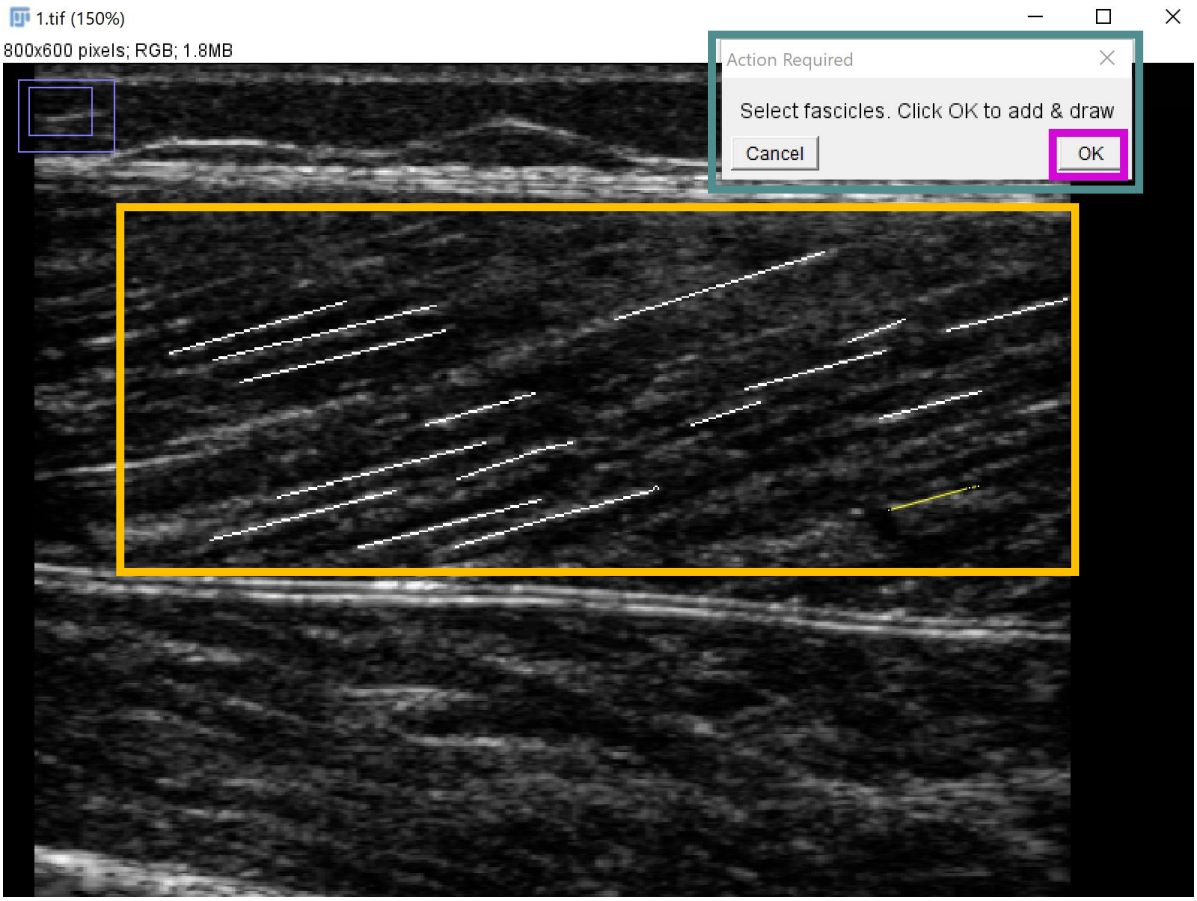
The polygon tool is selected automatically for you to draw around the **superficial aponeurosis**. Again, follow the instructions in the **messagebox**. Draw around the **superficial aponeurosis** (double click to start drawing, click to add a segment, double click do stop drawing) and once you are finished, click the **OK** button in the **messagebox** to proceed to the selection of the **lower aponeurosis** Please be careful to only include aponeurosis tissue in your selection and no surrounding tissue. The result should look like this for the **upper** and **lower aponeurosis**:





Once you have selected the lower aponeurosis, click the **OK** button in the **messagebox** to proceed to the fascicle labelling. Take a look on the next page to see how this is done.

The segmented line tool is selected automatically for you to follow the visible fascicle segments. Again, simply follow the instructions in the **messagebox.** It is of **utmost importance** that you draw **only over the actually visible parts** of the **fascicle** segment. Make sure that you only label bright **fascicle** tissue that is clearly visible. Once you drew one **fascicle** with segmented line tool (double click to start drawing, click to add a segment, double click do stop drawing) click the **OK** button in the **messagebox** to proceed to the next **fascicle** segment. Draw as many segments as are clearly visible on the image. When you press the **OK** button in the **messagebox** without making a further selection, you will proceed to the next image in the original image folder and start again with the aponeurosis labelling. The result of you labelling should look something like this:

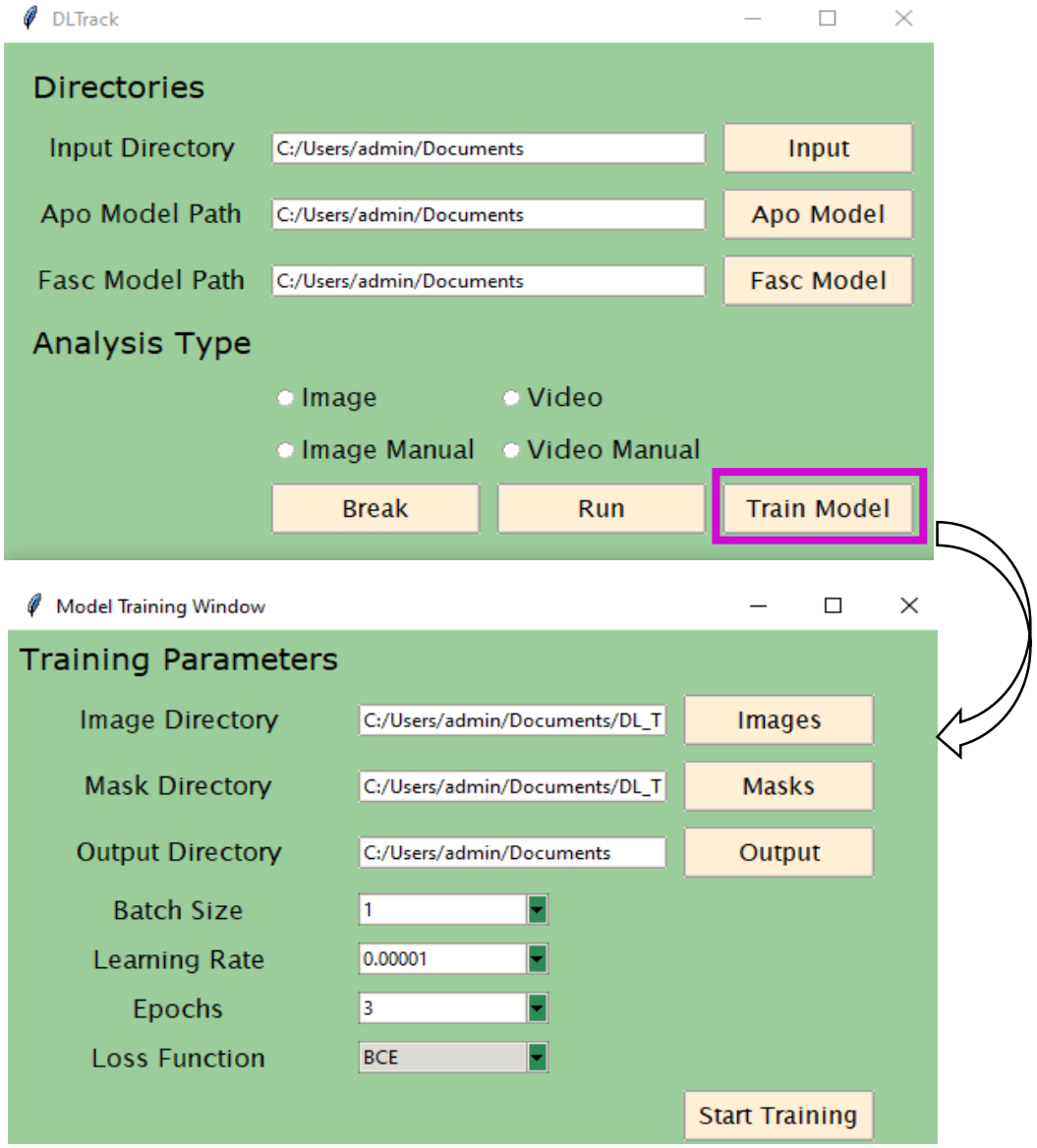


When you read this, you now how to label images to train your own neural networks with your own lower limb ultrasonography images! Congrats! We will now continue with the example tutorial on how to train your own networks with the example images and masks provided.

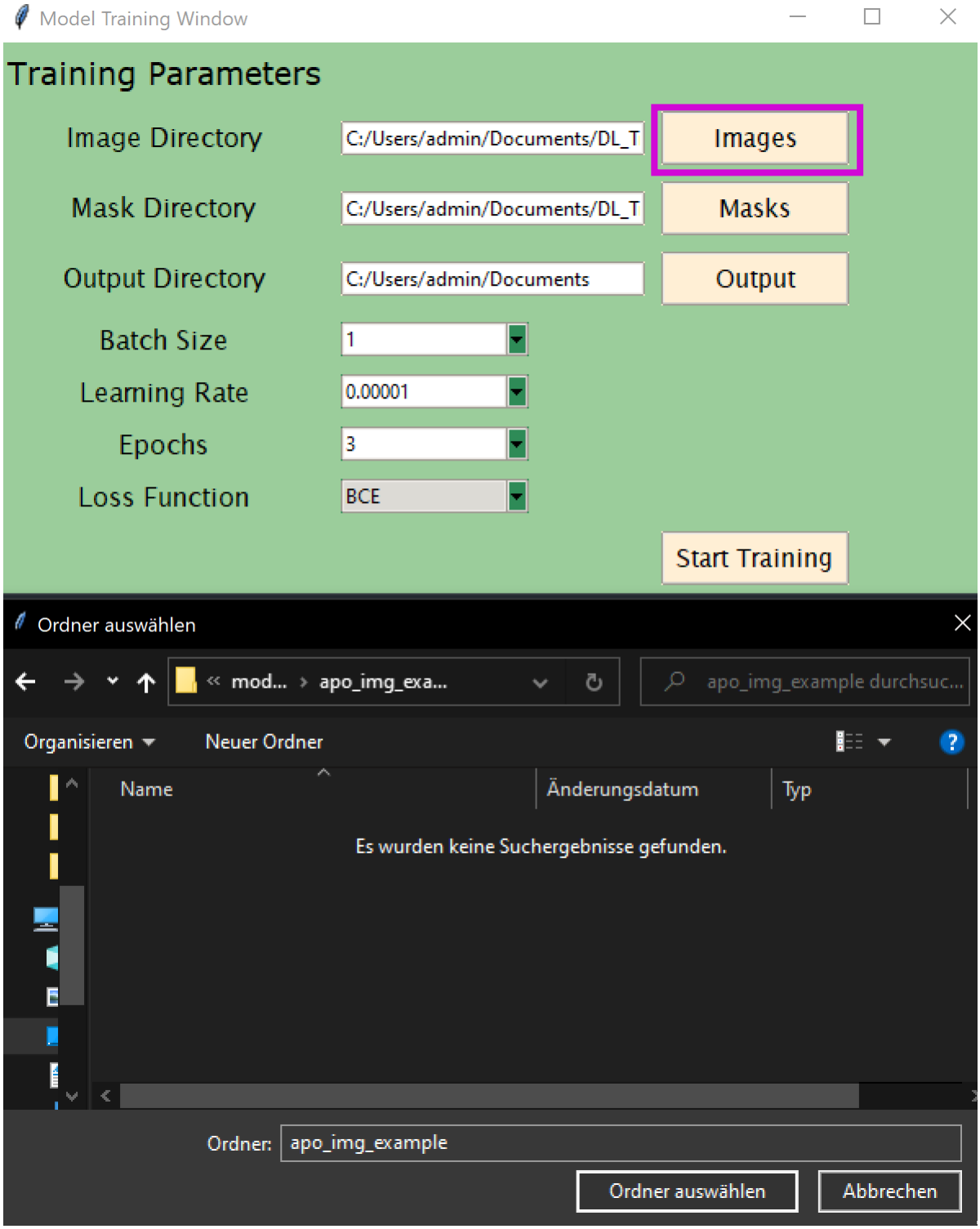
2. Specifying Relevant Directories

Caution, the actual tutorial continues here!
Please keep in mind that the model training process will be illustrated by training a model for aponeurosis segmentation. The process is exactly the same for training a fascicle segmentation model. Solely the images and masks should then contain fascicles and fascicle labels.

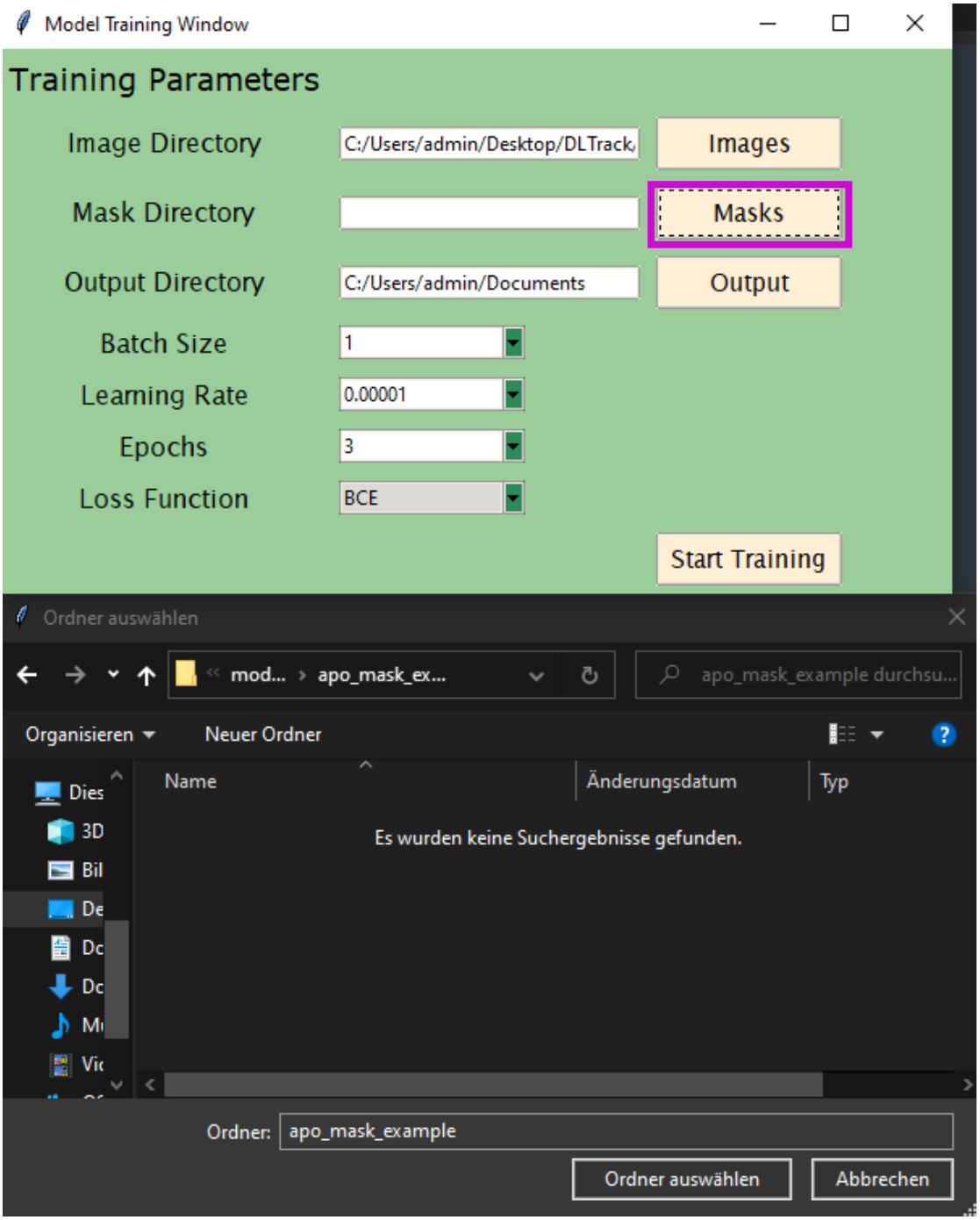
As a next step, you can start the GUI. If you do not know how to do this, please take a look at section two of this tutorial, Starting the GUI.
Once you started the GUI and the main GUI window opened, click on the **Train Model** button to select the relevant directories and model training parameters. The separate "Model Training window" will pop up. We will explain this window on the next page.



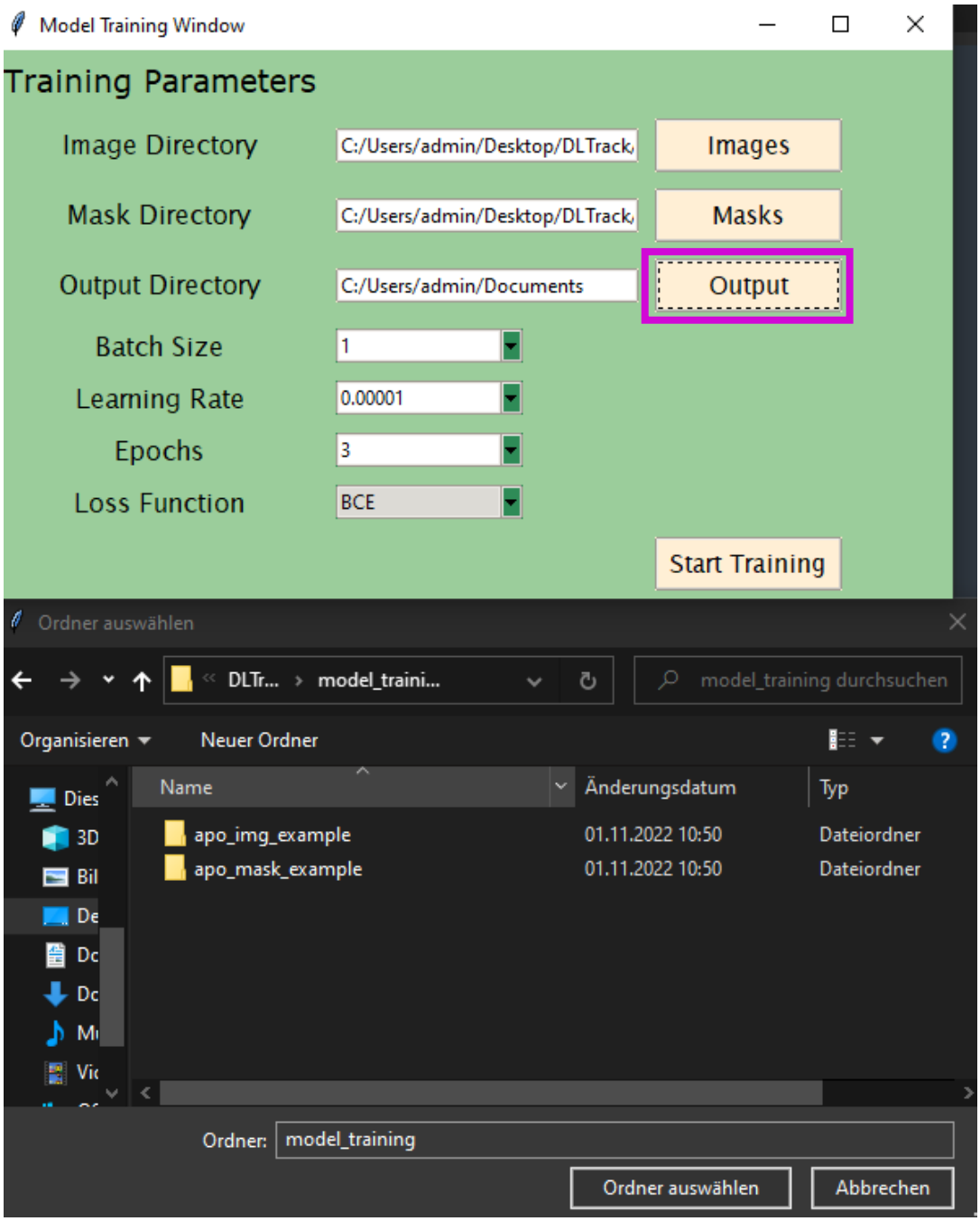
Once the “Model Training window” opened up, you first step is to select the “Image Directory” by clicking the button **Images**. A selection window will appear and you can select the folder containing the **original images**. In this tutorial, select the “DL_Track_US_example/model_training/**apo_img_example**” folder.



Your next step is to select the “Mask Directory” by clicking the button **Masks**. A selection window will appear and, you guessed correctly, you can select the folder containing the **mask images**. In this tutorial, select the “DL_Track_US_example/model_training/**apo_mask_example**” folder.



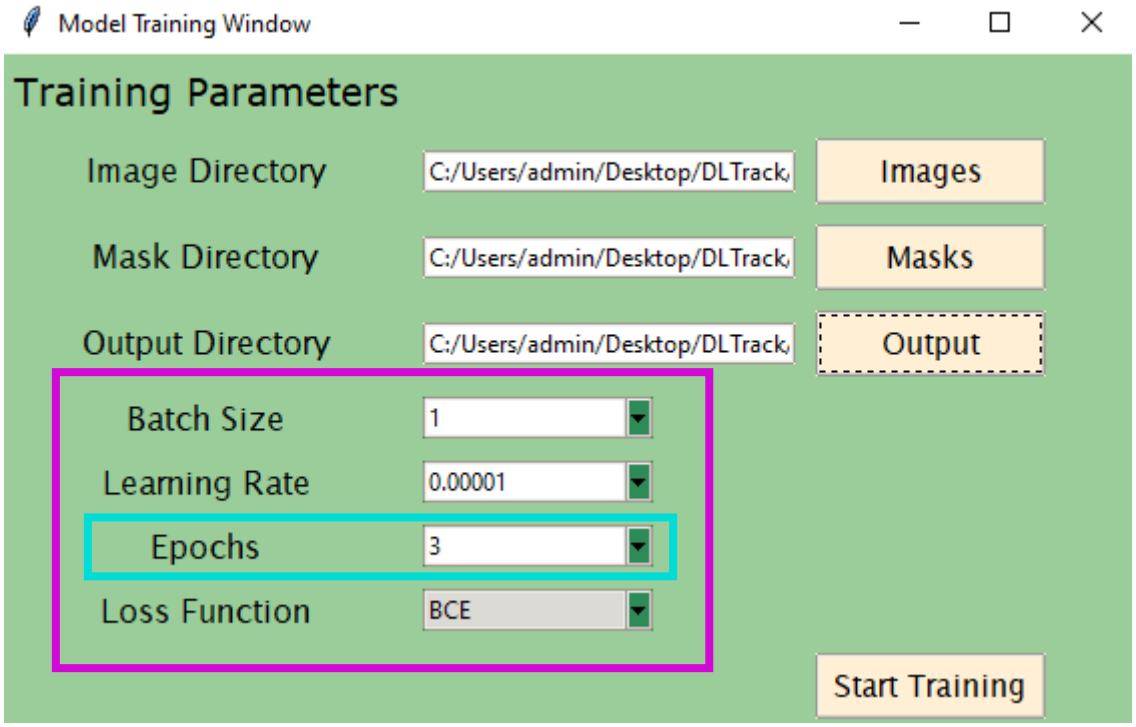
The last directory you need to select for training your own network is the "Output Directory" by clicking the button **Output**. In the **Output** directory, the trained model, the corresponding loss calculation results and a graphic displaying plotting the training epochs against the loss values will be saved. A selection window will appear and you can select any folder you like. In this tutorial, for simplicity reasons, please select the "DL_Track_US_example/model_training" folder.



Great, you have selected all relevant directories for model training! Now we shall take a look at the training parameters on the next page.

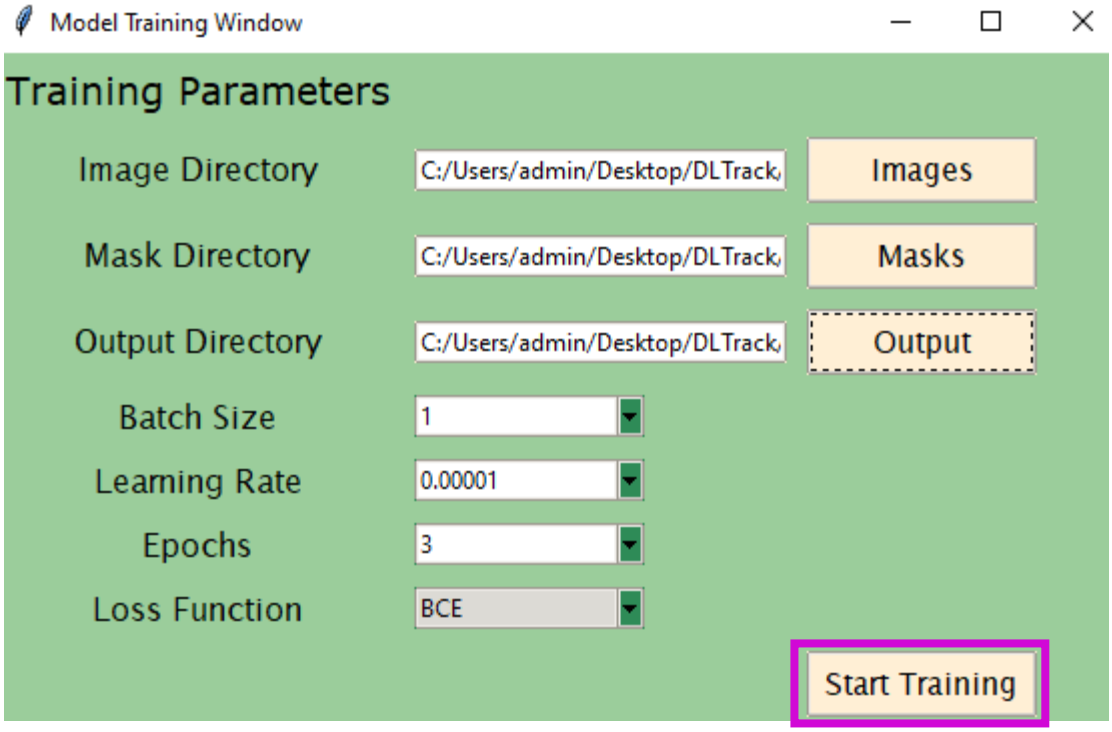
3. Specifying Training Parameters

Now to specifying the **training parameters**. You actually don't have to do anything at this point, just leave the pre-specified selections as they are. If you do not know what these **training parameters** mean, take a look at the videos we mentioned in the introductory text of this chapter. The only thing we have to say is that you must **NEVER** use only three **Epochs** for actual model training. Such a small number of training **Epochs** is only acceptable for demonstration and testing purposes. For actual training of your own neural networks, go with at least 60 **Epochs** (or maybe 42 is the better choice?).



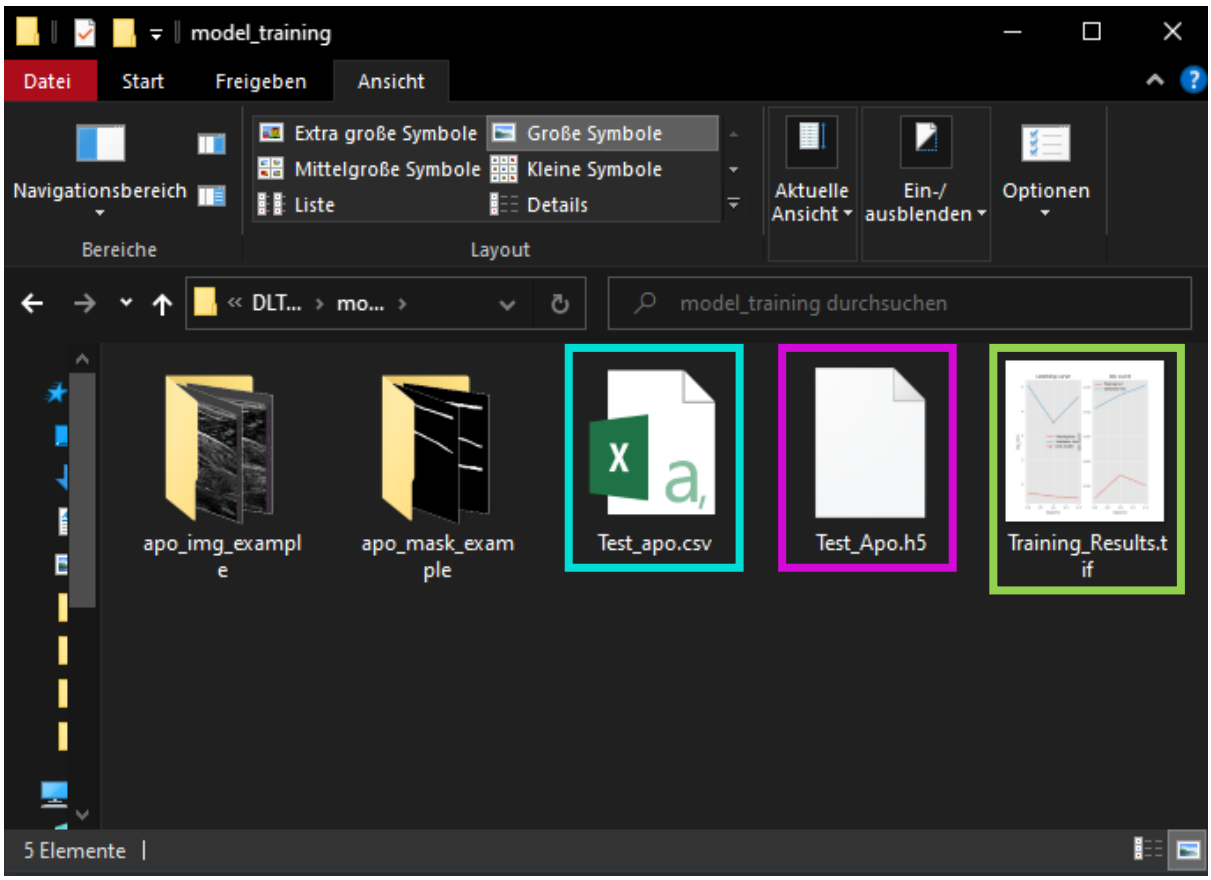
Take a look at the next page to learn how to continue from here.

The only thing you have left to do for the training process to start is to click the **Start Training** button.



During the training process, three messageboxes will pop up. The first one will tell you that the images and masks were successfully loaded for further processing. The second one will tell you that the model was successfully compiled and can now be trained. The last one will tell you that the training process was completed. You do have a choice in each messagebox of clicking "OK" or "Cancel". Clicking "OK" will continue the training process, whereas clicking "Cancel" will be cancelling the ongoing training process.

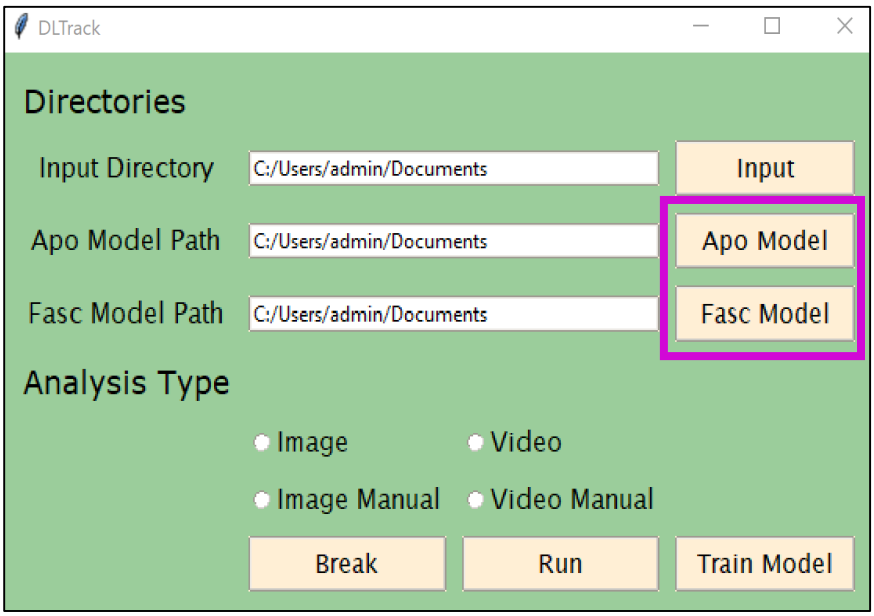
Once the model training process is completed, there will three new files in your selected output directory. Remember, you specified the "DL_Track_US_example/model_training" folder as your output directory. These new files are the **trained model** as a Test_Apo.h5 file, the **corresponding loss values** for each epoch as Test_apo.csv file and a **graphical representation** of the training process as Training_Results.tif file.



In case you can open the loss values and the graphical training representation, we congratulate you! You have successfully trained your first own neural network using the DL_Track_US package. Awesome! This is also the end of this tutorial, as Training Your Own Networks was the final chapter. Have a look at the closing remarks for a few more things. And, before it is forgotten, there is one more section on the next page. Error handling.

4. Using Your Own Networks

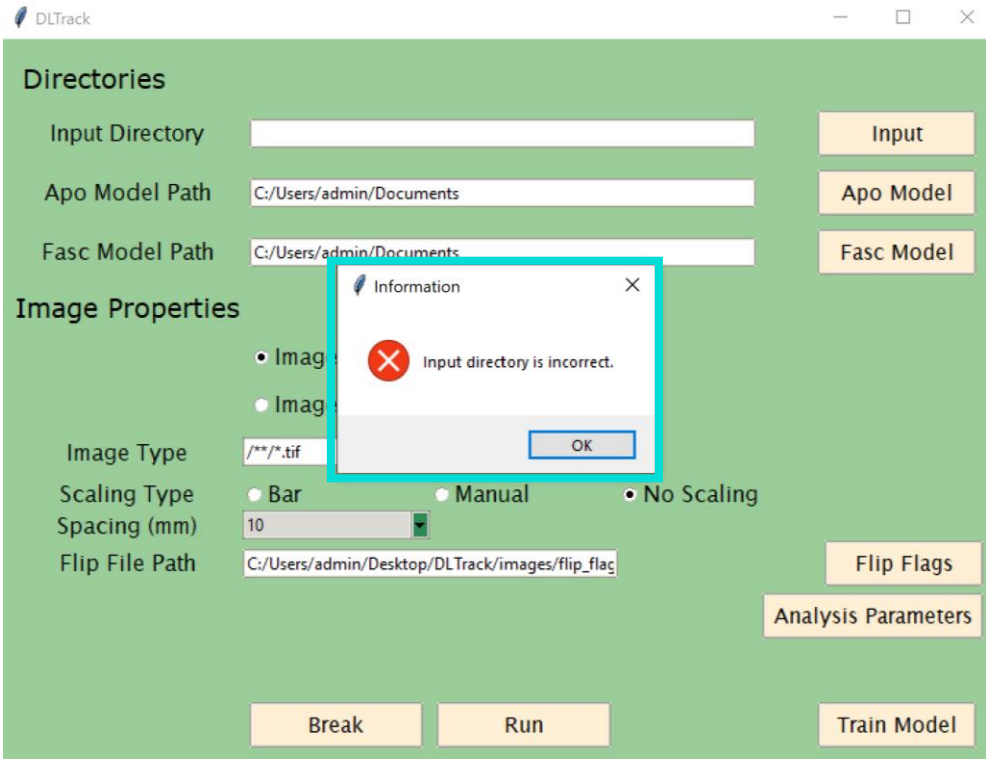
Once you have trained your aponeurosis or fascicle segmentation model (in case of this tutorial to model was called “Test_Apo.h5”), it is time to start using it. You have previously learned how to do that, in section 3 “Automated Image Analysis” on page 14 and 15 of this tutorial to be precise. Simply select the path to your model (or the “Test_Apo.h5” model) by clicking the **Apo Model** or **Fasc Model** buttons in the GUI, depending which kind of model you have trained. In our case, we would click the **Apo Model** button, since we trained the “Test_Apo.h5” model on aponeurosis images and labels. Subsequently to specifying all other relevant parameters for your analysis in the GUI (as you have learned a couple pages ago), DL_Track_US will analyse your data using your own model (or the “Test_Apo.h5” model we have trained together).



Lastly, a short disclaimer when training your own model. It is called bad practice when using the same images for model training and inference. At least during the model evaluation phase (checking how good your model actually is). Without going into details (see the introductory course we mentioned at the beginning), the model should not be used for analysing images it was trained on because it already knows the characteristics of these images. Since you should **ALWAYS** compare the results of your model to a manual evaluation on a few of your own images, use different images (best from different individuals) for model training and comparison to manual analysis. If this seems strange to you, don’t hesitate to ask for further clarification in the DL_Track_US discussion forum.

5. Error handling

Whenever an error occurs during the analysis process, the DL_Track_US GUI will open a **messagebox**. This looks always similar to this:



We tried to formulate these **messageboxes** as concise as possible. Just follow their instructions to fix the error and run the analysis anew. In case an error occurs that is not caught by an error **messagebox,** don't hesitate to report this in the Q&A section in the DL_Track_US discussion forum. Please take a look here how do best do this. Otherwise, you can contact us by email at paul.ritsche@unibas.ch, but we would prefer the other way.

Closing remarks

Thanks for checking out the DL_Track_US python package tutorial. We hope you were able to enjoy it a bit. Moreover, we hope it was clear, concise and easy to follow. We tried to put our biases aside and to start from scratch. In case we failed to do so at some point and something was not clearly illustrated, please let us know. Don't hesitate to report this in the Q&A section in the DL_Track_US discussion forum. Otherwise, you can contact us by email at paul.ritsche@unibas.ch, but we would prefer the other way.