Sparsh Kudrimoti
GTMI 455, Prof. Sawicki
PURA Final Report
06 May 2022

Development of an Automated Tool to Process Ultrasound Images in Leg Muscles using Machine Learning

Research Topic Introduction

Ultrasound provides a non-invasive, low-cost approach for visualizing muscle-tendon anatomy. However, rapid and accurate analysis of ultrasound images has been an ongoing challenge. Until recently, each frame of an ultrasound video needed to be tracked manually [1]. The primary reasons for this imaging challenge include low image quality, the lack of consistency between different users, and the variation of muscles that can be imaged when using ultrasound. New automated or semiautomated ultrasound image processing approaches rely on rules-based examinations, which tend to fall short if the images in a data set vary greatly from those that were used to develop the method [1] – [4]. Thus, future approaches to automated ultrasound image processing should allow users to rapidly train their own study-specific models. In this report I present an automated machine learning-based tool for calculating muscle architectural objects from ultrasound images that can be rapidly retrained for different studies.

Related Work and Background Information

Ultrasound has been widely used in healthcare for the last 70 years, primarily in obstetrics. Recently, the use of ultrasound has been expanded to the musculoskeletal field, where muscle and tendon function can be studied in a dynamic setting. The key structures within an ultrasound image include the aponeuroses (muscle borders) and the muscle fascicles which lie in between the aponeuroses. Using an ultrasound image, muscle fascicle length, pennation angle, and muscle thickness can be computed. Specifically, pennation angle is angle at which muscular force is produced in relation to its tendon. This parameter is important in determining a fascicle's force contribution to musculoskeletal movement. Using these parameters, the ultrasound method can be applied to many studies, such as examining the effects of ageing on muscle architecture. This highlights ultrasound's broad scope and the need for proper analytical processes to examine its data.

Until recently, the principal procedure to analyze ultrasound images was to manually label structures. The main drawback of this technique is the variability between different researchers to mark pieces of the muscle architecture [5]. Even the gold standard in the field, Ultratrack, requires user input to address drift over long time scales, and cannot extrapolate long fascicles of thigh musculature [1].

Machine learning is a method of data analysis that can recognize patterns in a dataset. A popular machine learning tool is a neural network, where nodes are interconnected between the input layers, hidden layers, and output layers. While training, data is inputted to the network, and the hidden layers "learn" recurring characteristics of the training images. Then, the "knowledge" of the hidden layers is outputted when the model attempts to predict the same characteristics in a new set of images. A specific type of neural network that is advantageous when using image data is a convolutional neural network (CNN). In a CNN, all the nodes are not fully connected and instead filters containing weights are arranged upon the inputs. Models developed using CNNs have persistently been shown to perform well in feature recognition projects, including instances where the inputted images are not well standardized or have clearly perceptible structures. Therefore, the goal of this project was to use machine learning and CNNs to identify the top and bottom borders of a muscle from ultrasound images.

Methods and Techniques

For the present project, I chose to implement a U-net architecture from Cronin's Aponeurosis training model [2]. All the code for this project was written in MATLAB and the Deep Learning Toolbox was used to design and implement the model. To begin the project, I manually labeled the top and bottom aponeuroses on pre-collected raw ultrasound data of the rectus femoris leg muscle. Using an image processing program called ImageJ, I traced each aponeurosis and combined them into a single binary mask. This was done for 490 raw images (70 images from each of 7 participants) and each mask was saved as a Tagged Image File (.tif). After saving all the masks and raw images in separate directories, I preprocessed the images so that they were all cropped and resized. The images were cropped at specific dimensions so that the raw images had no white border. Then they were resized to 512x512 pixels to maintain homogeneity between the masks and raw images. The masks were separated into top and bottom aponeuroses and all the images were converted to the unit8 data type. Two different networks were developed: one for the top aponeruosis and one for the bottom aponeurosis training. After shuffling the loaded data, I used an 80% to 20% split between the training and validation data, respectively. To train the networks, I implemented Cronin's aponeurosis training model from Python using MATLAB's *importKerasNetwork* function. Cronin's U-net architecture was slightly modified by removing the input layer and changing the output to a pixel classification layer instead of binary-cross entropy. Training was performed using an RTX2060 GPU and took around one hour per model using 60 epochs and a batch size of 2, with the Adam optimizer. Using the *trainNetwork* function I was able to create two models to identify each aponeurosis for every image. After training each network, I verified the results by overlaying the generated aponeuroses on the raw image. Figure 1 illustrates the flow of the work completed to build such a model.

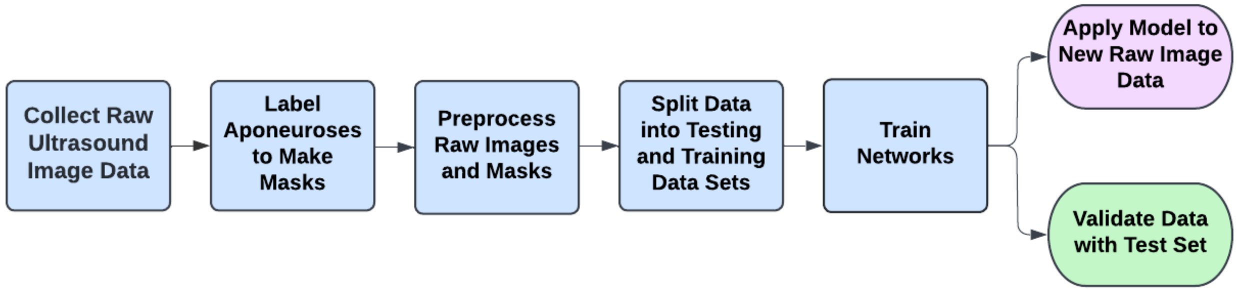

Figure 1: Model Development Flowchart

Results

For the post-processing and validation stage, I outputted an overlay of the top and bottom aponeurosis on a representative ultrasound raw image. As seen in Figure 2, the top left graphic is a single raw image and the top right graphic is the model prediction of its top aponeurosis. Similarly, the bottom left graphic is the model prediction for its bottom aponeurosis. In this example, the aponeuroses are almost perfectly marked over the raw image from a visible perspective.

To quantify the data set metrics of the model, Global Accuracy, Mean Accuracy, Mean IoU (intersection over union), Weighted IoU, and Mean Boundary F1 (BF) Score were computed as presented in Table 1. The Weighted IoU indicates the average IoU weighted by the number of pixels for all raw images. The Weighted IoU value for the algorithm-generated aponeurosis versus the manually traced masks was greater than 0.9 in both the bottom and top aponeurosis predictions. The Global Accuracy measures the ratio of the correctly classified pixels to total pixels, regardless of the image the model is predicting. The Global Accuracy for both top and bottom aponeurosis models were above 99%.

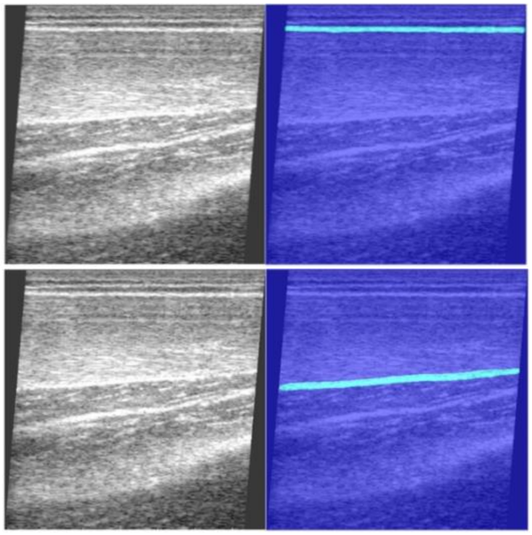

Figure 2: Predictions for a single raw image

Table 1: Model performance on validation set

Aponeurosis	Global Accuracy	Mean Accuracy	Mean IoU	Weighted IoU	Mean BF Score
Top	0.99516	0.95242	0.9157	0.99071	0.99261
Bottom	0.99071	0.89438	0.84653	0.98253	0.94472

Conclusions

In this report I present a machine learning approach to automate the analysis of muscle architecture in ultrasound images. The results generated by the model prove to be more promising than those from a manual approach or other existing algorithmic methods. The new model produces quick and accurate results that will be used to identify muscle architectural features in future test studies. Overall, these models will become a new asset for evaluating ultrasound data through MATLAB, so that other data analysis scripts can be used in conjunction with them.

References

[1] D. J. Farris and G. A. Lichtwark, "UltraTrack: Software for semi-automated tracking of muscle fascicles in sequences of B-mode ultrasound images," *Computer Methods and Programs in Biomedicine*, vol. 128, pp. 111–118, 2016, doi: 10.1016/j.cmpb.2016.02.016.

[2] N. Cronin, "Automated Analysis of Musculoskeletal Ultrasound Images Using Deep Learning," 2020.

[3] R. Marzilger, K. Legerlotz, C. Panteli, S. Bohm, and A. Arampatzis, "Reliability of a semi-automated algorithm for the vastus lateralis muscle architecture measurement based on ultrasound images," *European Journal of Applied Physiology*, vol. 118, no. 2, pp. 291–301, 2018, doi: 10.1007/s00421-017-3769-8.

[4] T. J. van der Zee and A. D. Kuo, "Fully automated algorithm estimates muscle fascicle length from ultrasound image," *bioRxiv*, 2020, doi: 10.1101/2020.08.23.263574.

[5] J. Aeles, G. A. Lichtwark, S. Lenchant, L. Vanlommel, T. Delabastita, and B. Vanwanseele, "Information from dynamic length changes improves reliability of static ultrasound fascicle length measurements," *PeerJ*, vol. 2017, no. 12, pp. 1–18, 2017, doi: 10.7717/peerj.4164.