Computer Methods and Programs in Biomedicine 256 (2024) 108398

Contents lists available at ScienceDirect

Computer Methods and Programs in Biomedicine

journal homepage: https://www.sciencedirect.com/journal/computer-methods-and-programs-in-biomedicine

Beyond pixel: Superpixel-based MRI segmentation through traditional machine learning and graph convolutional network ☆

Zakia Khatun [a,b,*], Halldór Jónsson Jr. [c], Mariella Tsirilaki [d], Nicola Maffulli [e,f,g], Francesco Oliva [h], Pauline Daval [i], Francesco Tortorella [a], Paolo Gargiulo [b,j]

[a] Department of Information and Electrical Engineering and Applied Mathematics, University of Salerno, Salerno, Italy
[b] Institute of Biomedical and Neural Engineering, Department of Engineering, Reykjavik University, Reykjavik, Iceland
[c] Department of Orthopaedics, Landspitali University Hospital, Reykjavik, Iceland
[d] Department of Radiology, Landspitali University Hospital, Reykjavik, Iceland
[e] Department of Trauma and Orthopaedic Surgery, Faculty of Medicine and Psychology, University Hospital Sant'Andrea, University La Sapienza, Rome, Italy
[f] School of Pharmacy and Bioengineering, Faculty of Medicine, Keele University, ST4 7QB Stoke on Trent, England
[g] Queen Mary University of London, Barts and the London School of Medicine and Dentistry, Centre for Sports and Exercise Medicine, Mile End Hospital, London, England
[h] Department of Human Sciences and Promotion of the Quality of Life, San Raffaele Roma Open University, Rome, Italy
[i] Biomedical Department, École Polytechnique Universitaire d'Aix-Marseille, Marseille, France
[j] Department of Science, Landspitali University Hospital, Reykjavik, Iceland

ARTICLE INFO

Keywords:
Magnetic resonance imaging
Superpixel
Graph convolutional network
Segmentation via node classification
Achilles tendon

ABSTRACT

Background and Objective: Tendon segmentation is crucial for studying tendon-related pathologies like tendinopathy, tendinosis, etc. This step further enables detailed analysis of specific tendon regions using automated or semi-automated methods. This study specifically aims at the segmentation of Achilles tendon, the largest tendon in the human body.

Methods: This study proposes a comprehensive end-to-end tendon segmentation module composed of a preliminary superpixel-based coarse segmentation preceding the final segmentation task. The final segmentation results are obtained through two distinct approaches. In the first approach, the coarsely generated superpixels are subjected to classification using Random Forest (RF) and Support Vector Machine (SVM) classifiers to classify whether each superpixel belongs to a tendon class or not (resulting in tendon segmentation). In the second approach, the arrangements of superpixels are converted to graphs instead of being treated as conventional image grids. This classification process uses a graph-based convolutional network (GCN) to determine whether each superpixel corresponds to a tendon class or not.

Results: All experiments are conducted on a custom-made ankle MRI dataset. The dataset comprises 76 subjects and is divided into two sets: one for training (Dataset 1, trained and evaluated using leave-one-group-out cross-validation) and the other as unseen test data (Dataset 2). Using our first approach, the final test AUC (Area Under the ROC Curve) scores using RF and SVM classifiers on the test data (Dataset 2) are 0.992 and 0.987, respectively, with sensitivities of 0.904 and 0.966. On the other hand, using our second approach (GCN-based node classification), the AUC score for the test set is 0.933 with a sensitivity of 0.899.

Conclusions: Our proposed pipeline demonstrates the efficacy of employing superpixel generation as a coarse segmentation technique for the final tendon segmentation. Whether utilizing RF, SVM-based superpixel classification, or GCN-based classification for tendon segmentation, our system consistently achieves commendable AUC scores, especially the non-graph-based approach. Given the limited dataset, our graph-based method did not perform as well as non-graph-based superpixel classifications; however, the results obtained provide valuable insights into how well the models can distinguish between tendons and non-tendons. This opens up opportunities for further exploration and improvement.

☆ This work is supported by EU H2020-MSCA-ITN-EJD-2020, Grant Agreement ID: 955685, Project Name: Perspectives For Future Innovation in Tendon Repair (P4-FIT).

* Corresponding author at: Department of Information and Electrical Engineering and Applied Mathematics, University of Salerno, Salerno, Italy.
E-mail address: zkhatun@unisa.it (Z. Khatun).

https://doi.org/10.1016/j.cmpb.2024.108398
Received 6 March 2024; Received in revised form 21 August 2024; Accepted 25 August 2024
Available online 28 August 2024

1. Introduction

Tendons are tough, inelastic bands of connective tissue formed of a dense, highly organized matrix of parallel collagen fibers that is maintained by resident tenocytes. Tendons attach skeletal muscles to bones and other structures and serve to transmit and absorb force [1]. While Achilles tendon is the strongest tendon in the human body, it is prone to injury and is the most commonly ruptured tendon. During activities, the Achilles tendon can bear loads of over 3500 N [2], yet despite its tremendous strength, it is frequently injured [3]. Acute and chronic Achilles tendon pathology is estimated to be responsible for as much as 50% of all sports-related injuries. 75% of Achilles tendon ruptures occur in middle-aged men between the ages of 30 and 49 while participating in sports, and the incidence is rising [4]. As a result, the Achilles tendon stands out as a demanding tendon for research.

Achilles tendonitis (overuse injury) is a common presentation among running and jumping athletes. Chronically, Achilles tendon may develop tendonosis where no inflammatory process exists, but chronic remodeling may occur. Tendinopathy, a widespread and challenging musculoskeletal disorder, affects a considerable amount of our population. It affects up to 30% of medical consultations related to musculoskeletal problems [5]. Regardless of the severity of tendinopathy, its symptoms can profoundly impact the patient's perceived quality of life.

Medical imaging techniques such as Ultrasound (US), Magnetic Resonance Imaging (MRI), and X-rays play a crucial role in the diagnosis and evaluation of tendinopathy. These imaging modalities provide valuable information about structural and pathological changes within tendons, which allows healthcare professionals to make informed clinical decisions. Machine learning, on the other hand, is a subset of artificial intelligence that offers the potential to transform tendinopathy research and diagnosis. It can process large amounts of image data and extract complex features that might not be very clear to the naked human eye. By training machine learning models, it is possible to develop algorithms that can detect, classify, and predict tendinopathy. Khatun et al. investigated various features of the quadriceps muscle and patellar tendon to assess their relationship with cartilage degeneration and tendinopathy [6]. Additionally, the study by [7] examined different features from medical images to explore bone and cartilage. However, research related to tendinopathy presents several challenges due to the complexity of the condition and the lack of knowledge of the exact reasons underlying the pathologies. Before categorizing the pathology, it is crucial to conduct tendon segmentation to identify the region of interest (ROI). Only after this step can further studies be pursued.

In computer vision, image segmentation encompasses a large class of finely related problems. Gupta et al. [8] proposed an automated segmentation of supraspinatus tendons using ultrasound images (US) by image processing techniques. It integrates the curvelet transformation and concepts of logical and morphological operators. An adaptive texture-based Active Shape Model was suggested by Chuang et al. [9] to segment tendon and synovium sheath. Martins et al. [10] proposed a segmentation approach of finger extensor tendon in ultrasound images based on an active contour framework. One drawback of active contour, active shape, and curvelet transform-based segmentation techniques is that edge/gradient information, which is used to guide contour deformation, is not reliable in ultrasound due to the presence of speckle noise and imaging artifacts [11]. [12] performed tendon grading for tear thickness, tear size, etc. by manually extracting features such as non-fluid signal intensity-related signal changes, anteroposterior dimensions, etc., which is known to be costly. Similarly, [13] utilized MRI data, with specialists manually extracting parameters like length, width, and thickness, incurring both time and cost expenses. Recently, superpixel segmentation has attracted a lot of interest in computer vision as it provides a convenient way to compute image features and reduces the complexity of subsequent image-processing tasks. Xu et al. developed a method using a machine learning algorithm based on variable-size superpixel segmentation [14]. In [15], authors proposed a new similarity-based superpixel generation method that was integrated with texton representation to form a spatio-color-texture map of the breast histology image. Zhu et al. [16] proposed a novel lung cancer detection method for CT images based on the superpixels and the level-set segmentation methods. Also, Signoroni et al. and Wan et al. utilized superpixel-based segmentation [17,18]. According to the literature, superpixels can be crucial from several perspectives in the case of medical image segmentation.

In traditional machine learning, unsupervised segmentation techniques rely on the intensity or gradient analysis of the image via various strategies. Such approaches perform well when boundaries are well-defined. In contrast, supervised segmentation methods incorporate prior knowledge about the image-processing task through training samples. On the other hand, deep learning (DL) models have yielded a new generation of image segmentation models with remarkable performance improvements, often achieving the highest accuracy rates on popular benchmarks. By merging semantic segmentation Convolutional Neural Network (CNN), 3D fully connected conditional random field, and 3D simplex deformable modeling, Zhou et al. developed a knee joint segmentation pipeline [19]. Other segmentation methods include techniques such as attention-guided cascaded networks with pixel importance balance loss for segmentation [20], source-free unsupervised domain adaptation for multi-organ segmentation [21], liver segmentation using joint adversarial and self-learning approaches [22], segmentation of knee using source free adaptive technique [23], etc. Martins et al. [10] proposed a segmentation approach of extensor tendon based on active contours, preceded by phase symmetry preprocessing, and with prior knowledge energies. Kuok et al. proposed a unique finger tendon segmentation technique where a hybrid of effective convolutional neural network techniques was applied [24]. However, these types of algorithms are usually considered data-hungry. This means that these algorithms mostly require large amounts of high-quality labeled data to effectively learn and generalize patterns. The performance of many machine learning models, including deep learning neural networks, tends to improve as the volume and diversity of training data increase.

Machine learning in graphs, also known as graph neural networks, is a rapidly growing field within the broader domain of machine learning and artificial intelligence. It focuses on developing algorithms and models to extract meaningful information from structured data represented as graphs or networks. Aiming for automated segmentation, Cai et al. proposed a graph-based decision fusion process combined with deep convolutional neural networks (CNN) [25]. Tian et al. proposed an interactive segmentation method based on a graph convolutional network (GCN) to refine the automatically segmented results [26]. Node classification is perhaps the most popular machine-learning task in graph data, especially in recent years.

Given the existing literature landscape, we propose a comprehensive end-to-end system tailored for Achilles tendon segmentation, eliminating the need for expensive manual feature extraction or an extensive dataset. Our segmentation method involves two distinct approaches. As an initial step for both approaches, our data (MRI) undergoes coarse segmentation, which is based on superpixel generation. Superpixels are characterized by perceptually homogeneous regions. In comparison to pixel representation, superpixel representation decreases the number of image dependencies and offers better support to identify regions depending on image properties [27]. This phase provides a preliminary segmentation that is supposed to reduce the complexity of the final segmentation task. In our first approach, some of the generated superpixels contain tendon region, while others contain different tissues. Using traditional machine learning, the classification of each superpixel as tendon or non-tendon will result in tendon segmentation. Moving on to our second approach, these superpixels are organized into graphs

rather than conventional grids. Graph neural networks, such as GCNs, excel at categorizing graph topologies and yielding a unified categorization of nodes. In our study, this type of graph neural network is employed to distinguish superpixels/nodes as tendons or non-tendons. By primarily relying on superpixels for the initial segmentation (coarse segmentation) and subsequently leveraging graph neural networks, we adopt a strategy that is less data-intensive. To sum up, our system comprises the following elements:

1. Utilization of Simple Linear Iterative Clustering (SLIC) algorithm to generate superpixels from MRI data.
2. Extraction of 94 radiomics features from each superpixel.
3. Classification of each superpixel as tendon or non-tendon, leading to tendon segmentation through two approaches:
    - Superpixel classification using Random Forest and Support Vector Machine classifiers.
    - Superpixel/Node classification based on Graph Convolution Network (GCN).

An overview of our proposed framework to perform Achilles tendon segmentation is shown in Section 2.1. More details about this pipeline are highlighted in the next sections.

2. Materials and methods

2.1. Workflow

The pipeline of our work is illustrated in Fig. 1 which contains several steps. Each of the steps is discussed in detail in the following sections.

2.2. Data

The dataset employed in this study includes a diverse cohort of 76 subjects with a mean age of 45 years and a standard deviation of 19.8 years, including 42 men and 34 women. The participants went through MRI scans at Landspitali University Hospital in Iceland. The study participants are from distinct diagnostic categories. In particular, the Achilles tendon status of a total of 47 participants is listed as healthy. In 11 participants, calcific insertional tendinopathy is identified, characterized by the presence of calcium deposits at the insertion point of the tendon on the calcaneus. Six participants shows typical symptoms of Achilles tendinitis, which is an inflammation of the Achilles tendon. Two participants have a chronic tear in the distal part of the tendon. Lastly, the remaining 10 participants sustain an Achilles tendon injury.

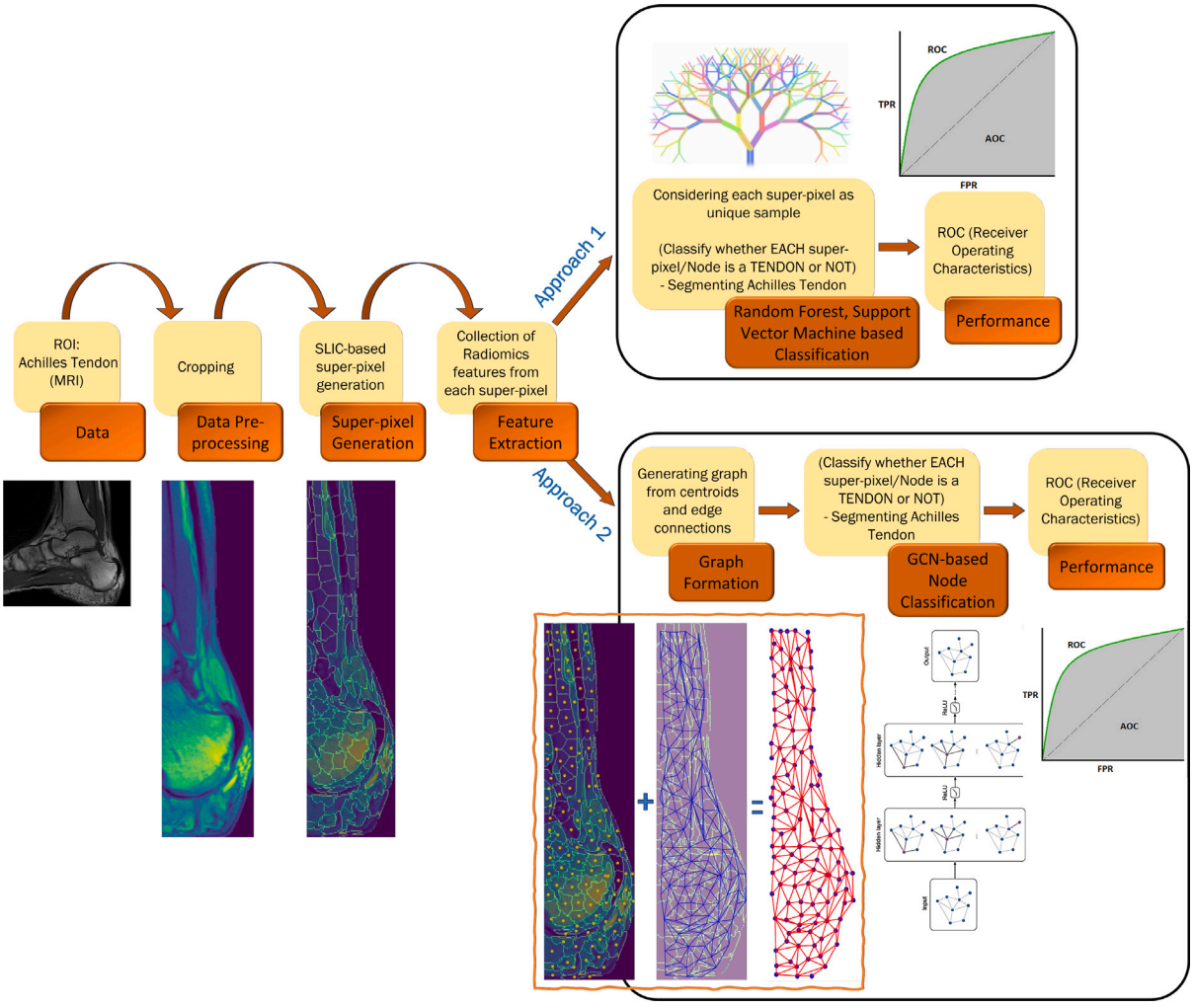


Fig. 1. Graphical Pipeline.

Fig. 2 depicts two randomly chosen MRI slices from different subjects to show what normal Achilles tendons look like (dark band, marked with dotted red color). The slices were chosen to display sagittal views of the Achilles tendon, an imaging plane oriented parallel to the sagittal plane of the body.

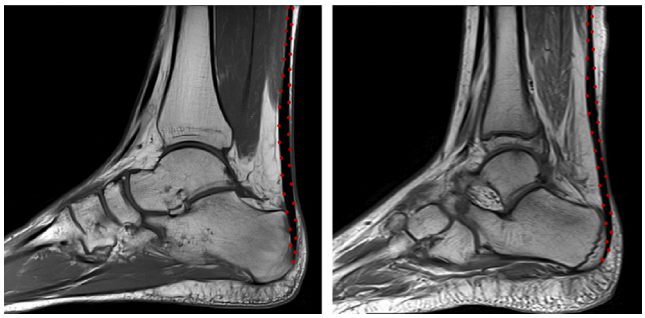


Fig. 2. Healthy Achilles tendons.

Instead, Fig. 3 shows two randomly selected pathological samples. These samples are crucial to our study since they offer in-depth information on deviations from normal subjects and shed light on a range of abnormalities.

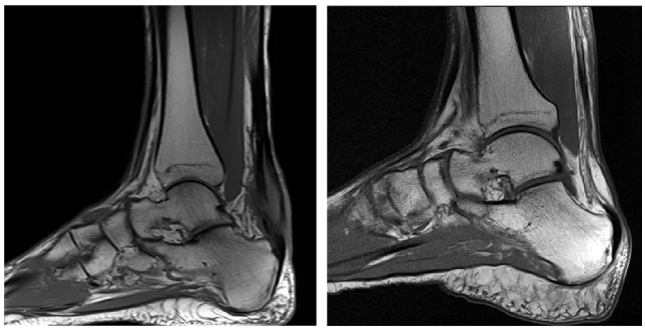


Fig. 3. Pathological Achilles tendons.

Furthermore, the dimensions of the MRIs are variable. Some of the example dimensions are 384 × 384 × 19, 448 × 464 × 26, etc. Where the format X × Y × Z indicates matrix size (or resolution), representing the number of pixels or data points in each direction of a 2D image slice. In this study, a T1-weighted Turbo Spin Echo (T1 TSE) MRI sequence is used. This sequence uses a specific sequence of radiofrequency pulses and magnetic fields to create images. This particular sequence is especially useful for imaging soft tissues, such as tendon, muscle, fat, etc.

2.2.1. Amount of data

In this study, a careful selection process is employed from a cohort of 76 subjects, focusing exclusively on the 2D MRI slices containing the region of interest (ROI). Other slices not containing ROI are excluded. This thorough selection procedure resulted in a substantial dataset, comprising a total of 411 MRI slices. The rationale behind this choice is rooted in the observation that, on average, each subject in our study group provided a notable subset of 4 to 5 sagittal TSE MRI slices featuring Achilles tendon. This selection is made to ensure that the dataset is complete and suitable for subsequent in-depth analysis of the Achilles tendon.

2.3. Data pre-processing

2.3.1. Mask generation

Ground truth masks are an essential component of data-driven work since they serve as the basic block for building, evaluating, and training machine learning models. To generate ground truth masks for our ROI, a software called Materialize Mimics is used, which is an image processing software for 3D design and modeling developed by Materialize NV (https://www.materialise.com/en).

Fig. 4 demonstrates the final Achilles tendon mask (in green) created by the above-mentioned software. It shows a particular example of an original Achilles tendon with pathology and the associated generated mask. Fig. 4, serves as a more metaphorical illustration that emphasizes the complexity of specific occurrences in our database. This specific figure is an example of a situation where creating a ground truth mask required greater attention and assistance from experts. This particular case draws attention to the range of difficulties that can be encountered when making precise masks. To reduce human bias while generating these ground truth masks, two different individuals were involved in the mask generation and correction phases. Different experts and time points provided a rigorous validation method for the consistency and quality of the created masks.

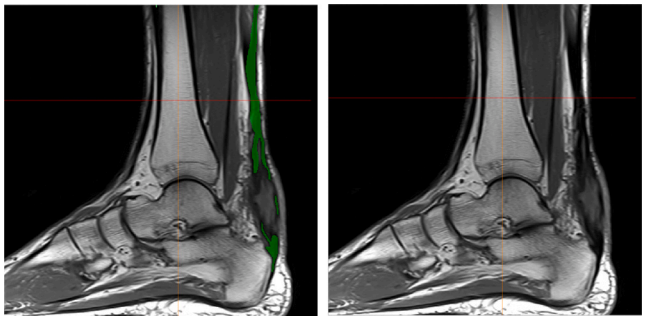


Fig. 4. Generated mask of a pathological subject.

2.3.2. Cropping

One of the fundamental image pre-processing techniques is image cropping, which attempts to eliminate unexpected regions and unnecessary noise from an image by altering its aspect ratio or improving its composition. It is also important, from this point of view, that not all regions of the MRI image are strictly relevant or very informative for the diagnostic purposes of a particular ROI. Additionally, image cropping proves valuable in making data more manageable, especially considering the vast size of full-scale MRI images, which consume significant storage space. Moreover, from a computational standpoint, data cropping serves as a vital step in our data pre-processing.

The following steps are followed to crop both the MRI data and the associated masks:

1. Loading individual MRI slices and corresponding ground truth masks: Each MRI slice is loaded, which typically represents a 2D image depicting a sagittal cross-section of the anatomy of interest. At the same time, the corresponding ground truth mask is loaded, which is another 2D image indicating the ROI highlighted within the MRI slice. As an example, Fig. 5a represents a sagittal MRI slice, and Fig. 5b represents its ground truth mask.
2. Conversion of ground truth masks to binary masks: Our MIMICS-exported ground truth masks are initially in color but are subsequently converted to binary masks for further processing. Subsequently, for each MRI slice, the binary mask is superimposed on the MRI slice to highlight the ROI. This visual overlay is illustrated in Fig. 5c, where all tissues are displayed in their natural appearance, with only the Achilles tendon highlighted in white.
3. Finding contour: This initial cropping phase consists of identifying the contours of the leg. This contour follows the shape of the calcaneus bone and executes the primary cropping step, as illustrated on the right side of Fig. 5d.
4. Recording the full width of the MRI slices: For each subject, the entire width of the MRI slice image is recorded. This width serves as a reference for subsequent cropping operations.

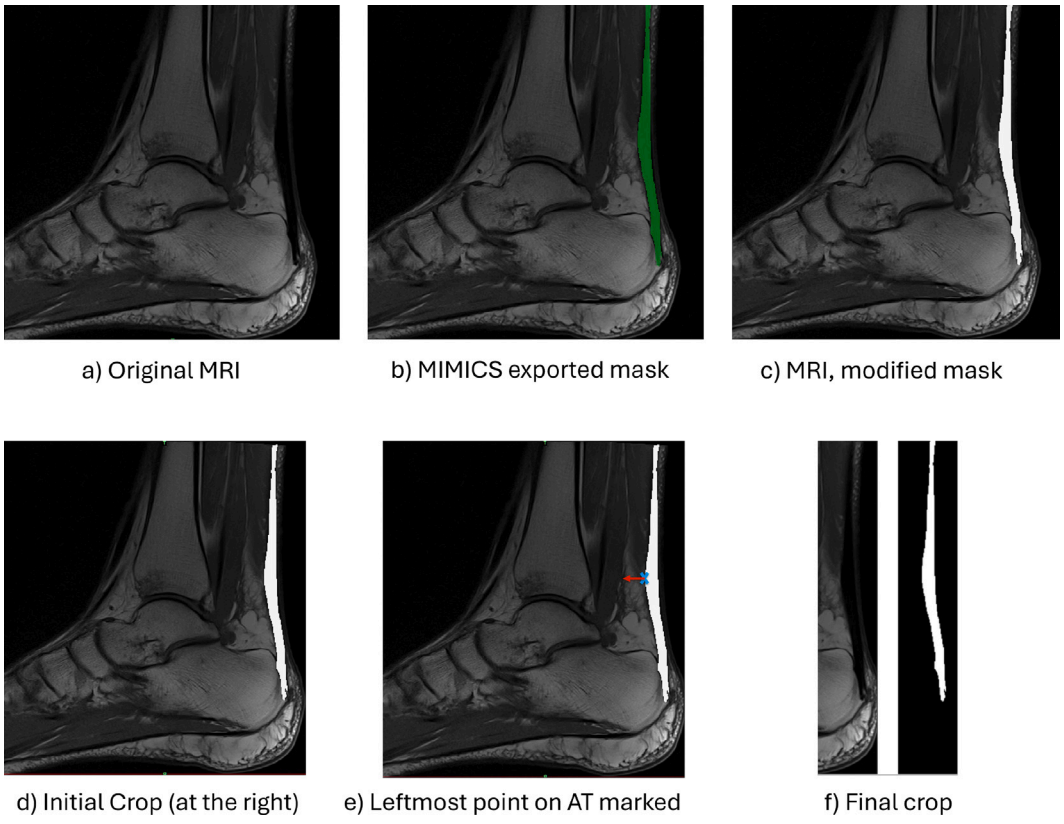


Fig. 5. Cropping Pipeline.

5. Recording tendon's leftmost point for in-depth analysis: For a more detailed analysis of the ROI, the leftmost point within the Achilles tendon mask is recorded. This step is crucial for gaining a comprehensive understanding of the spatial extent of the ROI. The position of the leftmost point in a specific case is depicted in Fig. 5e (in blue with a cross mark).
6. Final crop based on the leftmost coordinates and widths: The final cropped image, followed by a bounding box, is determined using the coordinates of the leftmost point obtained in the previous step. Each image is cropped to include the left portion of the Achilles tendon, starting from the leftmost point. This cropped region has a width equal to 7% of the total image width on the left (starting from the leftmost point) and right sides remaining as usual from step 3, 5d. The cropping operation produced the final representation of the ROI, illustrated in Fig. 5f. This approach ensures precise extraction of the desired region for further analysis and examination.

Some additional examples of cropped images are shown later in Fig. 6.

During testing, it is important to emphasize that the test ground-truth labels were not revealed. Cropping of the test data was conducted based on the leg contour, starting from the rightmost point of the calcaneus bone, with a portion of the slice width cropped from the left side. This approach ensured that the test data encompassed both the region of interest (ROI) and adjacent areas, maintaining consistency in input formatting with the training data, all without relying on the test ground-truth labels for cropping.

2.4. Superpixel generation

The purpose of this step is to generate superpixels on cropped data. Superpixels are perceptual groupings of pixels or over-segmented segments of an image. They capture image redundancy, provide a convenient primitive for computing image features, and significantly reduce the complexity of subsequent image processing tasks [28].

To achieve this, Simple Linear Iterative Clustering (SLIC) algorithm is used. Generally, SLIC algorithm generates superpixels by clustering pixels based on their color similarity and proximity in the image plane, using a five-dimensional [labxy] space, where [lab] represents the pixel color vector in the CIELAB color space and [xy] represents the pixel position. However, for grayscale images, intensity similarity (I) is used instead of color similarity. This is done in a three-dimensional [ixy] space, where [i] represents the pixel intensity and [xy] represents the pixel position. It is necessary to normalize the spatial distances to use the Euclidean distance in this 3D space because the maximum possible distance between two intensity values is fixed, whereas the distance in the XY plane depends on the image size [29]. Further details about this algorithm are outlined below:

1. Initialization:
   - *Grid Placement*: Divide the image into a grid with initial cluster centers placed roughly equal-sized spaced. The spacing S between these centers is determined by the desired number of superpixels K. Typically, $S = \sqrt{N/K}$, where N is the total number of pixels in the image.
   - *Refinement of Centers*: Adjust the initial cluster centers to positions with the lowest gradient in a 3×3 neighborhood to avoid placing centers on edges.
2. Distance Measure:
   - *Intensity and Spatial Proximity*: Define a distance measure D that combines intensity similarity and spatial proximity. For a pixel i with intensity I_i and coordinates (x_i, y_i), and a cluster center k with intensity I_k and coordinates (x_k, y_k),

the distance D is computed as:

$$D = \sqrt{(d_c)^2 + \left(m\frac{d_s}{S}\right)^2}$$

where:

- $d_c = |I_i - I_k|$ is the absolute difference in intensity.
- $d_s = \sqrt{(x_i - x_k)^2 + (y_i - y_k)^2}$ is the Euclidean distance in spatial coordinates.
- m is the compactness parameter that regulates the trade-off between intensity similarity and spatial proximity.
- S is the grid spacing.

The term $\left(m\frac{d_s}{S}\right)^2$ adjusts the spatial distance component based on the compactness parameter m, which influences the relative importance of spatial distance compared to intensity distance.

3. Assignment of Pixels to Nearest Centers:
   - *Local Search*: For each cluster center, consider all pixels within a $2S \times 2S$ region around the center. Assign each pixel to the nearest cluster center based on the distance D.
4. Update Cluster Centers:
   - *Recompute Centers*: After assigning pixels to cluster centers, update each center to be the mean intensity and spatial position of all pixels assigned to it. This adjustment ensures that the center represents the average characteristics of its superpixel.
5. Iterate:
   - *Repeat Assignment and Update*: Repeat the assignment and update steps iteratively until convergence is reached. Convergence typically occurs when cluster centers no longer change significantly between iterations or after a predefined number of iterations.
6. Enforce Connectivity:
   - *Post-processing*: Ensure that each superpixel forms a contiguous region. This is often achieved by reassigning isolated pixels to the nearest large superpixel, ensuring spatial coherence.

The simplicity of this approach makes it extremely easy to use a single parameter that specifies the number of superpixels, and the efficiency of the algorithm makes it very practical [28]. To implement this, the cv.ximgproc.createSuperpixelSLIC() function is used, which is part of the OpenCV library, a widely used library in the computer vision community. The results of the superpixel generation step for both a healthy and pathological case are presented later in the result Section 3. The following parameters are used in our superpixel generation approach:

1. image:
   This parameter searches for the input image of interest on which to perform superpixel generation/segmentation.
2. region_size:
   This parameter controls the compactness and size of the superpixels in the algorithm's initial clustering step. It is a purely computational parameter and is measured in pixels, not real-world units like millimeters or inches. This parameter defines the average size of the superpixels. Smaller values result in smaller superpixels, while larger values result in larger superpixels.
   In many image processing tasks, professionals often opt for a fixed region size for superpixels. However, our study faces a particular challenge due to the size variability within our input images. Therefore, we took a more adaptable approach by tying the region size to a ratio relative to the original image height. This choice is supported by the observation that the image height remains relatively stable even after cropping. Our decision to use $(13/375) \times image\_height$ as the region size is validated through rigorous experimentation. We specifically tested this ratio with both healthy and pathological cases and found that it effectively captured our ROI. One might interpret it as follows: In the context of an image with an approximate height of 375, a superpixel size of 13 is considered an ideal choice for defining the size of regions. Therefore, this dynamic approach to region size calculation offers flexibility between different image sizes, ensuring that superpixel generation remains effective and adaptable. It is worth noting that after applying this ratio-based approach to all cases, we achieved an average region size of approximately 13.8345. This level of consistency underlines the reliability of our initial approach.
3. Compactness Factor:
   It is one of the other important parameters in our study that plays a significant role in shaping the properties of the generated superpixels. When the compactness factor is set to a very high value, the resulting superpixels tend to be extremely compact, meaning they may not conform well to natural boundaries within the image. On the other hand, when this value is small, the superpixels are more closely aligned to the edges of the image, but their sizes and shapes may become less regular and uniform. In this specific study, we carefully selected a compactness factor value of 10. This choice was made after our careful experimentation and analysis. By setting the compactness factor to 10, there is a balance between these two extremes. It allows the generation of superpixels, which are not very compact, to still maintain a strong adherence to the boundaries of the underlying image. This careful selection allowed us to obtain a segmentation result that aligns well with the natural form of the image, achieving a balance between compactness and adherence to boundaries.

Several superpixel generation algorithms exist including Normalized cuts, Entropy Rate Superpixels (ERS), turbopixels method [30], mean-shift [31], watersheds in digital space [32], Simple Linear Iterative Clustering (SLIC) and g-SLIC algorithms, which address many requirements and outperform other state-of-the-art algorithms. In our study, we have chosen the SLIC algorithm for superpixel generation due to its efficiency in producing compact and uniform superpixels with linear time complexity by limiting the search space around each cluster center. SLIC balances color similarity and spatial proximity through a controllable compactness parameter, resulting in contiguous and noise-insensitive superpixels. Compared to other methods, SLIC is simpler to implement and equally robust, making it ideal for various image-processing tasks. Our choice of SLIC over other algorithms is also supported by the literature [33–38], etc.

2.4.1. Background removal on SLIC-based coarse segmentation

Background removal is a widely used technique in the field of image processing. Specifically, it is a critical preprocessing step in SLIC-based medical image segmentation. It not only improves segmentation accuracy but also improves computational efficiency, reduces errors, and facilitates consistent and interpretable results. The quality of background removal can also be affected by the complexity of the background, including any patterns, textures, and overlapping elements. Complex backgrounds can be more difficult to remove and can produce an uneven or inconsistent final product. In our study, this step is a challenge due to the distinctive characteristics of our ROI, namely the Achilles tendon located close to the contour of the leg or background. The adjacency of this region to the image boundary introduced greater complexity to the task.

Furthermore, as illustrated in Fig. 6, different ankle MRI images show a wide range of attributes at their edges. In some cases, a thin layer of fatty tissue is present. Additionally, there were scenarios where the boundary comprised predominantly fat, and in some other cases, the boundary exhibited a lack of clear delineation. These multifaceted scenarios made the simple application of conventional background removal techniques impractical. Given these distinct conditions, it was necessary to devise a more specialized and personalized approach to effectively address this issue of background removal in the context of our study. The steps involved in the background removal process of this study are shown below:

1. Gaussian Blur: This step reduces noise and enhances the overall image quality.
2. Grayscale Conversion: The image data is then converted to grayscale. The data type is changed from int16 to uint8 to ensure compatibility and appropriate representation for subsequent steps.
3. Contrast Enhancement: Next, Contrast Limited Adaptive Histogram Equalization (CLAHE) is applied to enhance the contrast of the grayscale images. This step helps in making the image details more prominent.
4. Thresholding: OpenCV's binary thresholding technique, following the OTSU method, is applied to contrast-enhanced images. This operation is used to create a binary mask, which helps to distinguish foreground from background regions.
5. Bitwise Operation: The binary mask obtained in the previous step is used for a bitwise operation with the contrast-enhanced image. This step effectively isolates the regions of interest while suppressing the background.
6. Morphological Closing: Later, a morphological closing operation is performed using a morphological ellipse. This step helps in closing small gaps or holes in the foreground regions.
7. Morphological Dilation and Erosion: To further refine the binary mask and ensure a clean separation between the foreground and background, morphological dilation (iteration = 1) followed by morphological erosion (iteration = 7) is applied. These operations help to smooth the edges and eliminate any remaining artifacts.

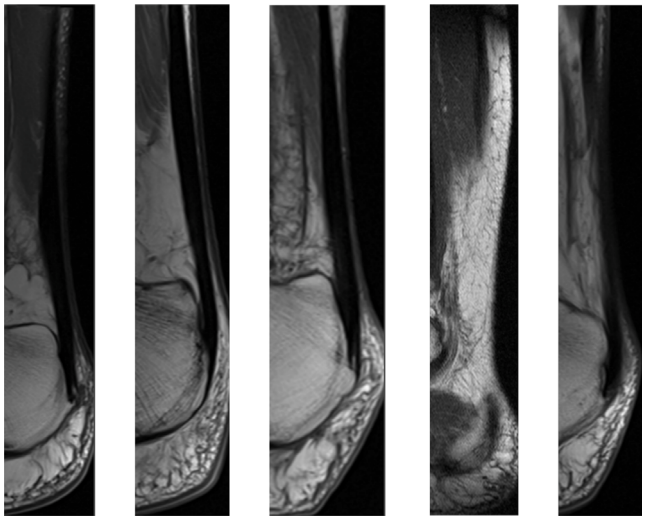


Fig. 6. Cropped Images (Variations in boundaries).

The outcome of this step, displaying superpixels without background, for both a healthy case and a pathological case is presented in the results Section 3, particularly in Figs. 12 and 11(a).

2.5. Feature extraction

As illustrated in our workflow, our methodology involves extracting radiomic features from the generated superpixels. Radiomics is an emerging field of research that addresses the extraction of mineable high-dimensional data from medical images. Radiomics enables the extraction of a significant number of quantitative characteristics from standard-of-care images from modalities like CT, MRI, and PE [39]. Leveraging the feature extractor provided by PyRadiomics (an open-source Python package), a total of 94 radiomics features are extracted [40] which are categorized into the following groups:

- First Order Statistics (19 Features): First-order radiomics features are statistical measures that describe the distribution of pixel intensities within a region of interest in an image. They include metrics such as mean, median, standard deviation, skewness, kurtosis, entropy, range, interquartile range etc.
- Gray Level Co-occurrence Matrix (24 Features): This is a method used to analyze the spatial relationships between pixel intensities in an image. It generates a matrix that records how frequently different pairs of pixel with specific values occur in a specified spatial relationship (*e.g.*, horizontally, vertically) within a region of interest. From this matrix, 24 features are extracted, including measures of contrast, correlation, energy, homogeneity, etc.
- Gray Level Run Length Matrix (16 Features): These features are used to analyze texture by evaluating the lengths of consecutive runs of pixels with the same intensity. It produces 16 features including measures like Short Run Emphasis which highlights the prevalence of short runs, and Long Run Emphasis which focuses on longer runs. Other features include Gray Level Non-Uniformity, Run Length Non-Uniformity, etc.
- Gray Level Size Zone Matrix (16 Features): This is a radiomics method used to describe the texture of an image by analyzing the size of zones with uniform gray levels. It generates 16 features that include metrics such as Zone Percentage which measures the proportion of zones of a certain size, and Gray Level Non-Uniformity, which quantifies the variability in zone sizes across different gray levels. Other features like Zone Size Non-Uniformity and Small Zone Emphasis capture the distribution and emphasis of various zone sizes.
- Neighboring Gray Tone Difference Matrix (5 Features): This radiomics method is used to characterize texture by assessing the differences in gray tone values between neighboring pixels. It calculates five key features: Coarseness, which measures the average size of the texture patterns; Contrast, which reflects the variation in gray levels between neighboring pixels; Busyness, indicating the frequency of gray tone changes; Complexity, which captures the texture's irregularity; and Strength, which quantifies the intensity of texture patterns.
- Gray Level Dependence Matrix (14 Features): These features are used in texture analysis to describe the spatial relationship between pixels in an image. It captures how frequently certain gray levels occur at specific distances and orientations relative to each other.

2.5.1. Data standardization

Prior to feeding the features into classifiers, they are standardized using *StandardScaler* from scikit-learn library [41]. First, we did fit and transform the training data with *StandardScaler* class which calculated and stored the mean and standard deviation. These computed statistics are subsequently applied to the test data ensuring that it is standardized on the same scale as the training data. This approach maintains consistency and prevents data leakage.

2.5.2. Motivation behind superpixel-based feature extraction

While convolutional neural networks (CNNs) are a common choice for such segmentation tasks, our approach takes a different approach by emphasizing the extraction of radiomic features from superpixels. This choice is in line with our hypothesis, which is:

- Robustness: Superpixels group pixels with similar features, which can make radiomic features more resistant to noise and small variations in the image. CNN may be more sensitive to fine-grained variations, which can lead to overfitting when dealing with limited training data or noisy medical images.
- Data Availability: Another big motivation was the amount of data available. With many medical imaging applications, data is limited, and acquiring labeled data to train deep learning models can be difficult. Radiomics, with its reduced dimensionality and potentially more robust features, may require fewer labeled samples for effective classification.

2.6. Approach 1: Superpixel-based features and random forest, support vector machine classifiers

Our first approach for Achilles tendon segmentation is illustrated in Fig. 7. After successfully performing the previous steps (creating superpixels on cropped images and extracting radiomic features from these superpixels), a classification task is performed using two different classifiers (a Random Forest classifier and a Support Vector Machine) to classify each superpixel. In this scenario, the input data for our classifiers comes from the steps mentioned above, where the inputs are radiomic features of these superpixels and their class labels. Any superpixel/node belonging to our ground truth mask is labeled as a tendon node, while those representing surrounding tissues are labeled as a non-tendon node. These details are organized as a .csv file for each image. As a result, for the total of 411 images (previously discussed in 2.2.1), there are a total of 411 corresponding .csv files. Therefore, by predicting whether a given superpixel or node falls into the label category of the tendon class or not, the segmentation process for the tendon region can be performed effectively.

2.6.1. Model training

During the implementation phase, the IDs of our 76 subjects are shuffled and split, resulting in an approximate 80%–20% split. As a result, there are 60 subjects in one group, called Dataset 1, and the remaining 16 subjects in another group, called Dataset 2. All associated MRI slices for each subject are then aggregated within their respective groups. It produced 320 MRI slices/images in Dataset 1 and 91 images in Dataset 2. This initial split is for training, validation, and final testing. Dataset 1 is used exclusively for training and validation processes. Dataset 1 contains a total of 30,219 superpixels/nodes while Dataset 2 contains a total of 8423 superpixels.

Dataset 1 is then used for training and validation via a leave-one-group-out cross-validation method that employs both Random Forest (RF) and Support Vector Machine (SVM) classifiers. Cross-validation is a resampling technique utilized for the assessment of machine learning models on a limited data sample. Leave-one-group-out cross-validation (LOGOCV) is a variation of the k-fold cross-validation technique used in machine learning and statistical modeling. In LOGOCV, instead of splitting the data into k-folds as in traditional k-fold cross-validation, the data is divided into groups or clusters, and each time one group is left out as the validation set while the model is trained on the remaining groups. RF classification, as an ensemble learning method tailored for classification tasks, builds numerous decision trees during training and produces a class prediction based on the mode of the classes determined by the individual trees. On the other hand, SVM is a powerful supervised machine learning algorithm used for classification tasks. In the context of classification, SVM aims to find the optimal hyperplane that separates data points of different classes with the maximum margin. The hyperplane is defined as the decision boundary that best separates the classes in the feature space.

Next, Dataset 2 is used for final testing using the 10 best-performing models trained and validated in Dataset 1. During training, special emphasis is placed on ensuring that superpixels/nodes derived from an MRI slice and the same subject are exclusively assigned to the training or validation phase, avoiding any partial assignment to both. This means that each subject (all superpixels of all associated slices) is used either for training or validation.

To implement our RF and SVM classifiers, the Scikit-learn library is used, more specifically sklearn.ensemble.RandomForestClassifier and sklearn.svm.SVC are used respectively where in both cases a balanced class weight is used since we have data imbalance (most superpixels fall into the non-tendon category rather than tendon superpixels). The balanced mode uses the values of y to automatically adjust weights inversely proportional to class frequencies in the input data as n_samples / (n_classes * np.bincount(y)), and all other parameters are set to the default values provided by the Scikit-learn library's built-in packages mentioned earlier.

2.7. Approach 2: Superpixel-based features and GCN-based node classification

The workflow of our second approach for Achilles tendon segmentation is depicted in Fig. 8, which is based on graph-based learning. In graph-based datasets, entities (nodes) are connected by relationships (edges), making them a natural representation of various real-world scenarios. One of the most popular baseline graph neural network models, the graph convolutional network (GCN), employs symmetric-normalized aggregation as well as the self-loop update approach. This approach was first outlined by [42] and has proved to be one of the

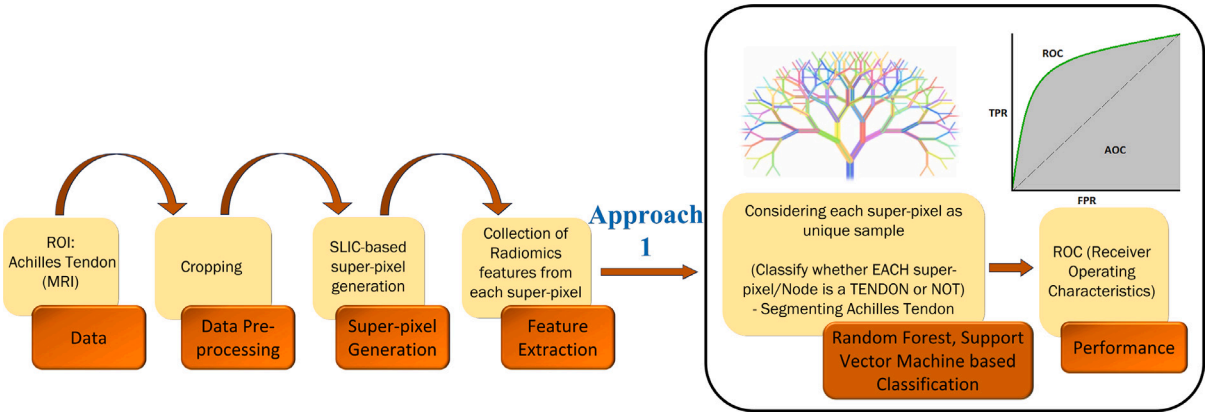


Fig. 7. Approach 1, Superpixel-based features and RF, SVM classifiers.

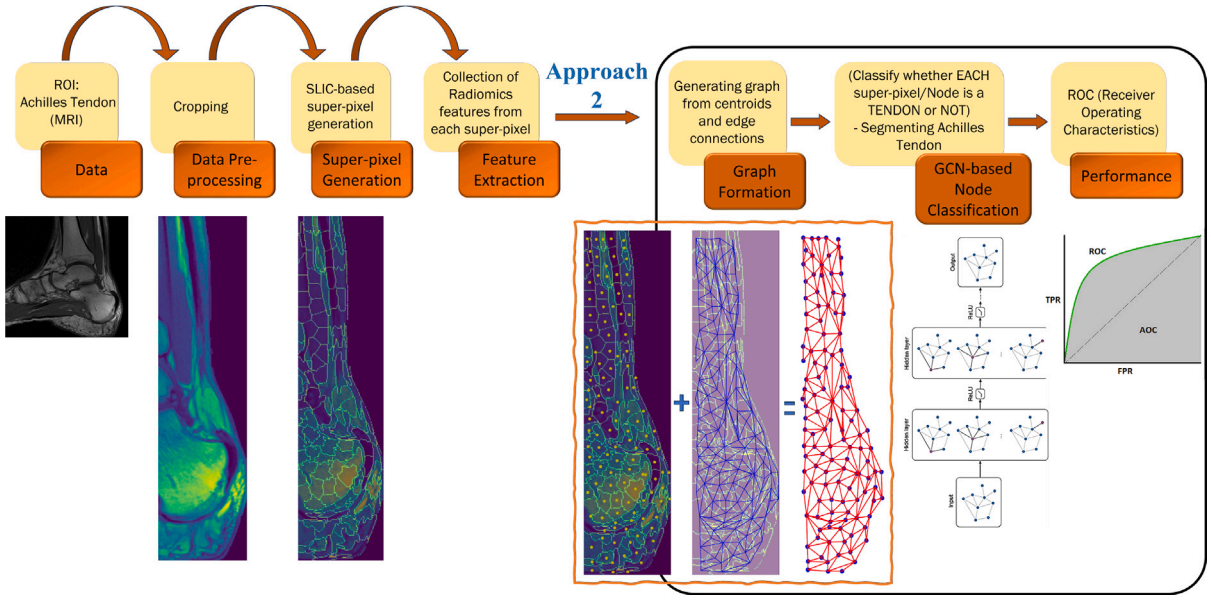


Fig. 8. Approach 2, Superpixel-based features GCN-Based Node classifier.

most popular and effective baseline GNN architectures. Our approach is also GCN-based, with the primary objective of node classification ultimately leading to tendon segmentation.

2.7.1. Graph formation

Graph construction is the preliminary step of our second approach. A graph $G = (V, E)$ is defined by a set of nodes V and a set of edges E between these nodes. It is denoted that an edge going from node $u \in V$ to node $v \in V$ as $(u, v) \in E$. A convenient way to represent graphs is through an adjacency matrix $A \in \mathbb{R}^{|V|\times|V|}$. To represent a graph with an adjacency matrix, the nodes in the graph are ordered so that every node indexes a particular row and column in the adjacency matrix. The presence of edges is represented as entries in this matrix: $A[u, v] = 1$ if $(u, v) \in E$ and $A[u, v] = 0$ otherwise. If the graph contains only undirected edges, then A will be a symmetric matrix, but if the graph is directed (*i.e.*, edge direction matters), then A will not necessarily be symmetric.

In our methodology, the process of creating these nodes and edges is based on the coarse segmentation results obtained via the SLIC (Simple Linear Iterative Clustering) algorithm. Each distinct superpixel generated by SLIC is treated as an independent node within the graph, while edges are formed by connecting vertices associated with neighboring superpixels, forming unweighted edges. After forming a graph for each slice, the structure of the resulting graph resembles the representation illustrated in Fig. 9.

As highlighted in the previous section on feature extraction, these extracted radiomic features for each node within each image are systematically stored in a .csv file. Similarly, edge connections between nodes are saved in a separate .csv file. This file containing the list of edge connections is subsequently interpreted as an adjacency matrix. In this format, the first dimension keeps track of the source node, while the second dimension keeps track of the corresponding destination node. In summary, for each MRI slice, the following information is extracted and stored: a .csv file containing radiomic features per node/superpixel, another .csv file containing class labels per node, and a separate .csv file including the list of edge connections between the nodes.

2.7.2. Model training

After the graphs are generated, it resulted in a total of 411 graphs from 76 subjects. They are divided in the same way as our previous approach (discussed in 2.6.1) which produced 60 subjects in Dataset 1 and their 320 graphs (from 320 MRI images) and the remaining 16 subjects in Dataset 2, which formed 91 graphs (from 92 MRI images).

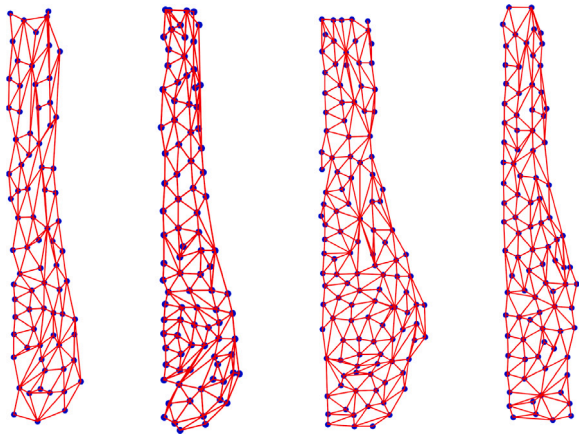


Fig. 9. Samples of Graphs.

In this phase, our node classification task is carried out through the utilization of a two-layer GCN architecture, as shown in Fig. 10. The adoption of this neural network architecture is motivated by the graph-structured nature of our input data. As illustrated in Fig. 10, a graph convolution operation is executed in the first layer, leveraging the node features and the edge connections among the nodes. This operation is implemented using the GCNConv class from PyTorch Geometric (PyG), a foundational component for Graph Convolutional Networks. GCNConv receives two primary inputs: (1) The number of input features for each node in the graph (input dim) and (2) The number of output features for each node in the graph (output dim). The input to this layer is a node feature matrix, and the output is also a node feature matrix, essentially mapping features from the input space to a new feature space. In the process of performing message passing between neighboring nodes in the graph, GCNConv aggregates information from these neighboring nodes to compute new features for each node. This aggregation is carried out by calculating a weighted sum of the features associated with the neighboring nodes, and the weights are learned during the training process. Notably, GCNConv

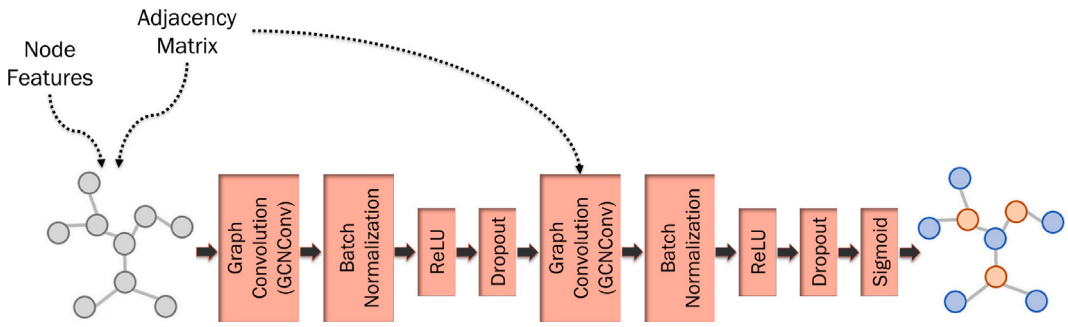


Fig. 10. Model Architecture.

incorporates a normalization step as part of the aggregation procedure, scaling the aggregated information in proportion to the inverse of each node's degree. This normalization aids in ensuring that nodes with varying degrees contribute equally. Subsequently, a linear transformation is applied to the aggregated information, with GCNConv featuring learnable parameters, including weight matrices for linear transformations and normalization coefficients. These parameters are adjusted and optimized during the model's training phase.

Here, node features are the radiomics features extracted from each superpixel. A is the adjacency matrix of the graph.

$$\hat{A} = \tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}}$$

is the normalized adjacency matrix, where $\widetilde{A} = A + I$, I is the identity matrix, $\tilde{D}_{ii} = \sum_j \tilde{A}_{ij}$ represents the degree of the node i.

To further enhance the model's stability and expedite the training process, a layer of batch normalization is employed, which acts as a standard regularization technique frequently utilized in deep neural networks. Following batch normalization, the Rectified Linear Unit (ReLU) is employed as the activation function. ReLU is a widely used activation function in deep learning models, defined as ReLU(x) = max(0, x), effectively replacing negative values with zeros while preserving positive values. This introduces non-linearity into the model, enabling it to capture complex, non-linear relationships within the data. This non-linearity aids in the model's ability to identify and propagate information relevant to the node classification task while suppressing extraneous data. Additionally, the utilization of ReLU helps in mitigating the vanishing gradient problem and contributes to faster convergence during training.

Furthermore, a dropout layer is introduced as a form of regularization within our network. The torch.nn.Dropout module is utilized, which automatically applies dropout during training, but refrains from doing so during the model's inference phase. This behavior is governed by the self.training attribute. In our training regimen, a dropout rate of 0.5 is specified.

The previously described block is reused, constituting what is referred to as the 2-layer GCN mode. Finally, a sigmoid layer is employed in the final layer, which is particularly well-suited for binary classification tasks requiring a decision between two classes, such as 0 or 1. The sigmoid layer compresses the model's output into a range between 0 and 1, representing the probability of belonging to the positive class.

Once the model is defined, it is trained and validated in a leave-one-group-out cross-validation fashion (similar to approach 1). It means that in our setup, in each run, training on all graphs except one subject's graphs is discarded for validation. Through this iterative process, the model is systematically assessed on distinct data subsets, facilitating a more dependable performance estimation and mitigating the risk of overfitting.

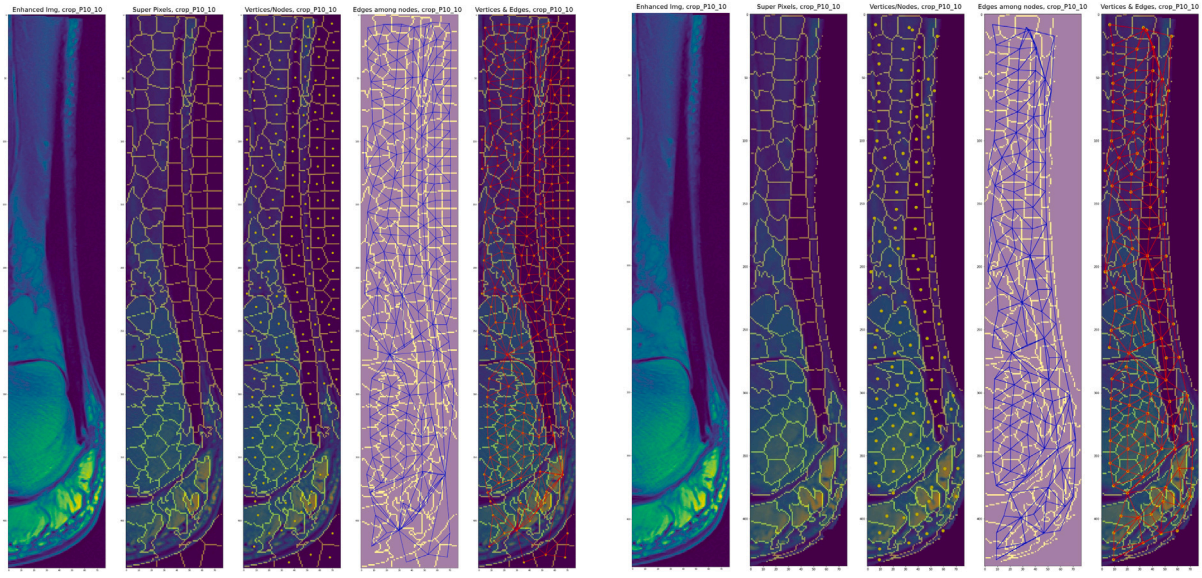
For the sake of the model optimization process, the Adam optimizer is used, which is a widely adopted algorithm for training neural networks. Adam is known for its adaptability in adjusting the learning rate during training, drawing from the strengths of both the AdaGrad and RMSprop optimizers. An optimizer is a function or an algorithm that adjusts the attributes of the neural network, such as weights and learning rates. Thus, it helps in reducing the overall loss and improving accuracy. Two critical parameters are provided to the optimizer: (1) learning rate, a hyperparameter, that regulates the size of the optimization steps taken throughout the training process. It influences the pace at which the optimizer refines the model's parameters based on the gradients of the loss function, (2) Another input parameter is weight decay, which serves as a regularization hyperparameter. This feature introduces L2 regularization, commonly referred to as weight decay, into the optimization procedure. Weight decay introduces a penalty term in the loss function, depending on the magnitude of the model's parameters. This penalty encourages smaller parameter values and serves as a preventive measure against overfitting.

The loss function used in our approach is called BCEWithLogitsLoss (imported from torch.nn), which stands for Binary Cross-Entropy Loss with Logits, and it is used in binary classification tasks. BCEWithLogitsLoss allows different weights for positive and negative examples. This feature is especially useful when dealing with imbalanced datasets, where one class contains significantly more samples than the other. Since our problem contains node imbalance (most nodes are non-tendon nodes, while a smaller amount of tendon nodes). In our implementation, assigning distinct weights to positive and negative examples in binary classification tasks is achieved using a parameter known as pos_weight. This parameter represents a scalar value that determines the relative significance of the positive class (class 1) compared to the negative class (class 0) when calculating the loss. The following hyperparameters are used in our model training: epochs = 100, batch_size = 32, dropout_rate = 0.5, optimier = ADAM, learning_rate = 0.01, weight_decay = 5e−4, loss_function = BCEWithLogitsLoss. All other parameters are set to their default values as provided by the built-in packages of the associated libraries mentioned earlier.

3. Results

Fig. 11(a) provides a visual representation of the generated superpixels, incorporating background for a comprehensive understanding. The importance of the final superpixel segmentation (after background removal) becomes clear in the later stages of our analysis, as demonstrated in Figs. 12 and 11(b). This critical phase plays a fundamental role in the overall process, serving as the basis for subsequent feature extraction. Carefully delineating superpixels containing only foreground regions sets the stage for training our model, as subsequent features are extracted from them. These extracted features are key to training our model to distinguish between tendon and non-tendon attributes.

To evaluate the performance of our tendon segmentation (via node classification), the Receiver Operating Characteristics: Area Under the Curve (ROC-AUC) is used as a performance metric. It is a valuable evaluation metric in certain situations due to its ability to provide



(a) Superpixels with image background (b) Superpixels without image background

Fig. 11. Comparison of superpixels with and without image background.

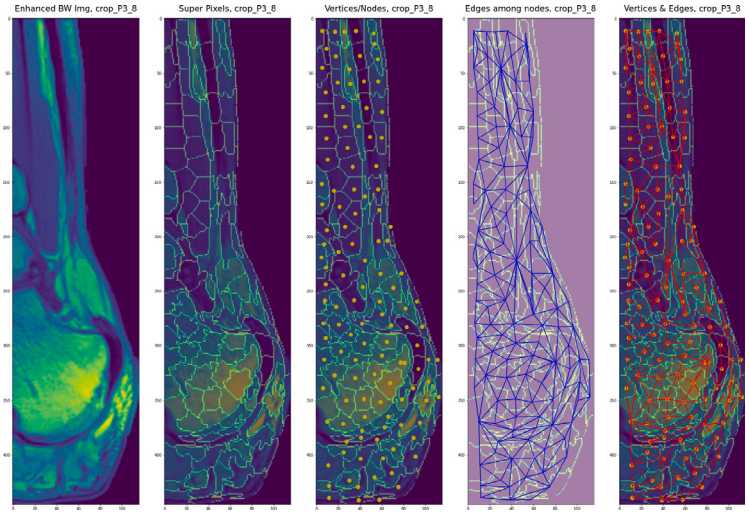


Fig. 12. Superpixels of a pathological subject.

insights into the performance of a classification model, especially in binary classification problems. AUC represents the degree or measure of separability. It tells how much the model is capable of distinguishing between classes. The higher the AUC, the better the model is at predicting 0 as 0 and 1 as 1. Another advantage is the data imbalance, which is the biggest motivation in our study for choosing this evaluation metric. When dealing with imbalanced datasets where one class significantly outperforms the other, precision can be misleading. ROC-AUC takes into account the trade-off between true positive rate and false positive rate and is less affected by class imbalance. Being a threshold-independent evaluation metric, AUC offers a more balanced assessment of model performance. On the other hand, sensitivity measures the proportion of true positive cases correctly identified by a classification model. These two metrics are also widely used as evaluation metrics for binary classification tasks with data imbalances. Additionally, we used specificity and balanced accuracy as other metrics. Specificity assesses the proportion of true negatives correctly identified by the model, while balanced accuracy is the simple average of sensitivity and specificity which offers a more comprehensive evaluation by considering both true positives and true negatives.

3.0.1. Performance of approach 1 (Superpixel-based features and RF, SVM classifiers)

Since our segmentation task is achieved through superpixel classification, we have selected the Area Under the ROC Curve (AUC) as our evaluation metric to assess classification performance. This metric ultimately reflects the effectiveness of our tendon segmentation module. AUC score serves as a vital evaluation metric to evaluate the performance of our RF and SVM classifiers, which are built on our superpixel-based coarse segmentation method. This metric provides a comprehensive view of the classifier's ability to discriminate between positive (tendon) and negative (non-tendon) instances within our dataset. Additionally, sensitivity, specificity, and balanced accuracy (the average of sensitivity and specificity) are emphasized as alternative evaluation metrics.

To ensure the robustness and reliability of our evaluation, a leave-one-group-out cross-validation procedure is used to train and evaluate the model on Dataset 1. The average scores of all evaluation metrics are shown in Table 1. In addition to the leave-one-group-out cross-validation performance, the performance of the classifiers on the test

Table 1
Evaluation metrics for superpixel classification/tendon segmentation.

Model name	Training and Validation (on Dataset 1)				Test (on Dataset 2)			
	AUC	Sensitivity	Specificity	Balanced Acc.	AUC	Sensitivity	Specificity	Balanced Acc.
Random forest	0.984	0.871	0.983	0.927	0.992	0.904	0.975	0.939
Support vector machine	0.983	0.943	0.940	0.942	0.987	0.966	0.941	0.953
GCN	0.922	0.879	0.896	0.888	0.933	0.899	0.902	0.901

data (Dataset 2) using the top 10 models from the training phase, the average scores of evaluation metrics are reported in Table 1. This provides insight into the model's performance on unseen and out-of-sample data, which is valuable for evaluating its real-world applicability and generalization beyond the training dataset. Combining the cross-validation results and performance of test data offers a comprehensive evaluation of the effectiveness of the RF and SVM classifier-based node classification in our Achilles tendon segmentation task.

3.0.2. Performance of approach 2 (Superpixel-based features and GCN-based node classification)

To conduct a performance analysis comparable to Approach 1, the AUC score is used as the evaluation metric for the model. Both the AUC score, sensitivity, specificity, and balanced accurracy on Dataset 1 are shown in Table 1. Using a similar setup, the top 10 best models (based on AUC score) from the training phase are used for the final prediction of the test data (Dataset 2). The mean scores of all the evaluation metrics are recorded in Table 1.

4. Discussion

The results outlined in the performance evaluation underscore the effectiveness of our both approaches in tendon segmentation, especially in the area of node classification.

As demonstrated in Figs. 12 and 11(b), for superpixel generation, our technique consistently shows good performance in accurately delineating regions of interest in both healthy and pathological cases. In particular, our configured superpixel generation parameters produce segmentation results that align perfectly with the intrinsic structure of the image. This result attests to the effectiveness of our approach in finding a harmonious balance between compactness and adherence to boundaries. This optimized balance is especially crucial when dealing with diverse datasets that include both healthy and pathological instances, demonstrating the versatility and robustness of our methodology in capturing different ROIs.

Our first approach which uses Random Forest (RF) and Support Vector Machine (SVM) classifiers for node classification while exploiting superpixel-derived radiomic features shows notable levels of performance. Both RF and SVM classifiers are well established for their effectiveness and have been widely applied in various classification tasks. Recent studies have leveraged these classifiers in several contexts, including using RF with GLRLMS feature extraction to achieve maximum classification accuracy in identifying the severity of COVID-19 [43], its application in the classification of MRI-based brain tumors with superior accuracy [44], use of RF and SVM classifiers in breast cancer detection [45], among other notable applications. Being inspired by the continued success of these methodologies, our approach incorporated both RF and SVM-based classifiers paired with superpixel generation. The results (see Table 1) highlight the robust capabilities of RF-based and SVM-based classifiers when coupled with superpixel generation, shedding light on the potential effectiveness of such segmentation approaches within our research domain. Notably, the AUC scores obtained from both classification configurations on our test data are 0.992 and 0.987 for RF and SVM, respectively. While the RF classifier has a slightly higher AUC, SVM demonstrates superior sensitivity performance. Another observation is that SVM shows the highest sensitivity across both datasets, which highlights its effectiveness in accurately identifying positive instances (the tendon class). This high sensitivity is particularly crucial for our study, as accurately detecting the tendon class is more important than achieving the highest specificity, which is the strength of the Random Forest classifier. Additionally, SVM outperforms the Random Forest classifier in terms of balanced accuracy on both datasets. This further underscores the superior performance of the SVM classifier over RF. When evaluating AUC and balanced accuracy, it is crucial to understand that they measure different aspects of model performance and are not directly comparable. Balanced accuracy is determined at a specific threshold, which may not be optimal for every model. Therefore, when the chosen threshold (the default of 0.5) is not ideal, balanced accuracy may be lower than AUC. This seems to be the case here, where the default threshold resulted in a lower balanced accuracy. In contrast, AUC assesses performance across all possible thresholds, providing a more comprehensive view of the model's ability to distinguish between tendon and non-tendon classes. This broader perspective explains why AUC shows slightly higher value and aligns better with our metrics of interest. Overall, depending on the specific point of interest, both RF and SVM performed equally well in our study.

In contrast, our second approach (graph-based approach) also demonstrates promising performance. This approach leverages a versatile graph structure, coupled with the generation of superpixels. On our test data, this approach achieved an AUC of 0.933, with a sensitivity of 0.899, a specificity of 0.902, and a balanced accuracy of 0.901. While these metrics are slightly lower compared to the performance of RF and SVM-based experiments, they still indicate potent performance. These results lead to two key observations. First, as a standalone model the graph-based approach performs well. However, when compared to non-graph-based models, its performance appears relatively lower. This discrepancy seems to be attributed to the limited amount of available data. Deep learning-based models typically require a substantial amount of data for effective training, and with only 76 unique subjects in our dataset, the potential of the graph-based approach may not have been fully realized. Initially, we hypothesized that the incorporation of coarse segmentation would enhance the performance of the graph-based final classification. However, this approach did not outperform the non-graph-based models, suggesting that further exploration is needed. This is addressed in the future work section.

To our knowledge, our study is among the first in the field of Achilles tendon segmentation which makes direct comparisons with other state-of-the-art methods challenging. Alzyadat et al. [46] conducted automatic segmentation of Achilles tendon using deep CNN. They used datasets of 3708 for training and 2472 for validation. Their approach involved ensembling different networks for the final segmentation. Due to the differences in their methodology and the amount of data used, direct performance comparison with our study is not feasible. A very recent study [47] conducted tendon segmentation on ultrasound images using gray-level co-occurrence matrix features and hidden Gaussian Markov random fields. The primary aim was to provide a quantitative and automated method for detecting potential structural changes in tendinopathy. As with previous studies, the objectives, methodologies, and data differ from those used in our

research. Within the domain of medical image segmentation, there is a very recent study that has focused on the segmentation of diverse anatomical structures in medical images. Known as MedSAM [48], it is a deep learning-powered foundation model. It is trained on a large-scale dataset of over one million image-mask pairs. The network utilized in MedSAM was built on a transformer architecture. Notably, the study underscores the significance of the training dataset size in determining final performance metrics. Moreover, considerations such as data modality, data quality, and regions of interest are also some of the crucial factors when comparing the performances of different segmentation models.

A key limitation of this study is the small dataset, which may have constrained the performance of our GCN-based approach compared to traditional machine learning models, as deep learning methods generally require larger datasets. Additionally, the diversity within the dataset made it challenging to optimize background removal, possibly necessitating further adjustments for varying data. Another limitation is that our pipeline is trained and tested on cropped data, which may not fully address the challenges of applying the approach to full-scale images, highlighting an area for future research. Moreover, since superpixels define the ROI, different images or ROIs will require adjustments to the superpixel generation parameters. Finally, being one of the pioneering efforts in Achilles tendon segmentation, our study had limited opportunities for direct comparison with state-of-the-art methods.

5. Conclusion and future work

Our proposed module for tendon segmentation has demonstrated its effectiveness through the utilization of superpixel generation as a coarse segmentation step preceding the final segmentation task. This approach formulates the segmentation task as a superpixel classification problem, aiming to classify each superpixel as either tendon or non-tendon. The primary motivation behind using superpixel-based coarse segmentation was to address the fact of traditional neural networks, which must simultaneously learn both lower-level and higher-level information within the same architecture. By grouping similar pixels, this method simplifies the image, reduces complexity, preserves boundary information, and integrates higher-level features, leading to more accurate and robust segmentation results.

In terms of contribution, our proposed tendon segmentation module delivered impressive results across several key metrics. The good performance is strengthened by computationally efficient superpixel generation, which streamlines image processing without compromising precision. Additionally, the versatile nature of our module implies this through its multiple approaches (both traditional machine learning and GCN-based approaches). This flexibility empowers users to choose the method that best aligns with the computational resources, overall setups, and data properties. Finally, our generalizable framework which is built on robust principles and modular design, holds exciting potential for adaptation to various medical imaging tasks beyond tendon segmentation. Overall, this module's combination of high performance, versatility, and generalizability positions it as a valuable tool for advancing medical image analysis.

To further refine our module, future work will involve exploring its performance on larger and more diverse datasets to solidify its generalizability and robustness. Another area for exploration could include working with full-scale images instead of cropped data. Incorporation of additional features such as demographic-related or pain-level data can be another edition. Additionally, investigating more advanced graph network architecture holds promise for potentially boosting performance and capturing intricate tendon-related features. Ultimately, integrating this segmentation module into clinical workflows for tendon pathology diagnosis, treatment planning, and outcome assessment represents a crucial step toward its real-world impact on improving patient care. In conclusion, our research introduces a new methodology and valuable insights into segmentation, paving the way for improved understanding and diagnosis of tendon-related pathologies.

CRediT authorship contribution statement

Zakia Khatun: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Data curation, Conceptualization. **Halldór Jónsson Jr.:** Writing – review & editing, Resources, Project administration. **Mariella Tsirilaki:** Data curation. **Nicola Maffulli:** Project administration, Funding acquisition. **Francesco Oliva:** Project administration. **Pauline Daval:** Data curation. **Francesco Tortorella:** Writing – review & editing, Supervision, Project administration, Methodology, Conceptualization. **Paolo Gargiulo:** Writing – review & editing, Supervision, Resources, Project administration, Conceptualization.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Zakia Khatun reports financial support was provided by European Commission. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Funding statement

This work is supported by EU H2020-MSCA-ITN-EJD-2020, Grant Agreement ID: 955685, Project Name: Perspectives For Future Innovation in Tendon Repair (P4-FIT).

References

[1] H.L. Birch, Tendon matrix composition and turnover in relation to functional requirements: Tendon matrix composition and turnover, Int. J. Exp. Pathol. 88 (4) (2007) 241–248.
[2] S. Fukashiro, P.V. Komi, M. Jarvinen, M. Miyashita, In vivo achilles tendon loading during jumping in humans, Eur. J. Appl. Physiol. 71 (5) (1995) 453–458.
[3] B.R. Freedman, J.A. Gordon, L.J. Soslowsky, The Achilles Tendon: Fundamental Properties and Mechanisms Governing Healing, arXiv:PMID:25332943.
[4] T.A.H. Järvinen, P. Kannus, N. Maffulli, K.M. Khan, Achilles tendon disorders: Etiology and epidemiology, Foot Ankle Clin. 10 (2) (2005) 255–266.
[5] D. Morrissey, Guidelines and pathways for clinical practice in tendinopathy: Their role and development, J. Orthop. Sports Phys. Ther. 45 (11) (2015) 819–822.
[6] Z. Khatun, M. Tsirilaki, A. Lindemann, F. Tortorella, N. Maffulli, H. Jonsson, P. Gargiulo, The role of muscle and tendon in predicting cartilage degeneration and tendinopathy, in: 2022 IEEE International Conference on Metrology for Extended Reality, Artificial Intelligence and Neural Engineering, MetroXRAINE, Rome, Italy, 2022, pp. 289–294.
[7] R. Aubonnet, J. Ramos, M. Recenti, et al., Toward new assessment of knee cartilage degeneration, Cartilage 14 (3) (2023) 351–374.
[8] R. Gupta, et al., Curvelet based automatic segmentation of supraspinatus tendon from ultrasound image: A focused assistive diagnostic method, BioMed. Eng. OnLine 13 (1) (2014) 157.
[9] B.-I. Chuang, et al., A medical imaging analysis system for trigger finger using an adaptive texture-based active shape model (ATASM) in ultrasound images, p. 21,
[10] N. Martins, S. Sultan, D. Veiga, M. Ferreira, F. Teixeira, M. Coimbra, A new active contours approach for finger extensor tendon segmentation in ultrasound images using prior knowledge and phase symmetry, IEEE J. Biomed. Health Inform. 22 (4) (2018) 1261–1268.
[11] G. Tsechpenakis, Deformable model-based medical image segmentation, in: Multi Modality State-of-the-Art Medical Image Segmentation and Registration Methodologies, Springer US, 2011, pp. 33–67.
[12] S. Bauer, et al., Reliability of a 3 T MRI protocol for objective grading of supraspinatus tendonosis and partial thickness tears, J. Orthop. Surg. Res. 9 (1) (2014) 128.
[13] M. Golman, et al., Rethinking patellar tendinopathy and partial patellar tendon tears: A novel classification system, Am. J. Sports Med. 48 (2) (2020) 359–369.
[14] J. Xu, et al., Three-dimensional spectral-domain optical coherence tomography data analysis for glaucoma detection, PLoS ONE 8 (2) (2013) e55476.
[15] A.D. Belsare, M.M. Mushrif, M.A. Pangarkar, N. Meshram, Breast histopathology image segmentation using spatio-colour-texture based graph partition method: Breast histopathology image segmentation, J. Microsc. (2016).
[16] H. Zhu, et al., A novel lung cancer detection algorithm for CADs based on SSP and level set, THC 25 (2017) 345–355.

[17] A. Signoroni, et al., BS-Net: Learning COVID-19 pneumonia severity on a large chest X-ray dataset, Med. Image Anal. 71 (2021) 102046.
[18] S. Wan, C. Gong, P. Zhong, B. Du, L. Zhang, J. Yang, Multiscale dynamic graph convolutional network for hyperspectral image classification, IEEE Trans. Geosci. Remote Sens. (2020).
[19] Z. Zhou, G. Zhao, R. Kijowski, F. Liu, Deep convolutional neural network for segmentation of knee joint anatomy: Zhou et al., Magn. Reson. Med. 80 (6) (2018) 2759–2770.
[20] H. Su, L. Gao, Y. Lu, H. Jing, J. Hong, L. Huang, Z. Chen, Attention-guided cascaded network with pixel-importance-balance loss for retinal vessel segmentation, Front. Cell Dev. Biol. 11 (2023).
[21] J. Hong, Y. Zhang, W. Chen, Source-free unsupervised domain adaptation for cross-modality abdominal multi-organ segmentation, Knowl.-Based Syst. 250 (2022).
[22] J. Hong, S. Chun-Ho Yu, W. Chen, Unsupervised domain adaptation for cross-modality liver segmentation via joint adversarial learning and self-learning, Appl. Soft Comput. 121 (2022).
[23] S. Li, S. Zhao, Y. Zhang, J. Hong, W. Chen, Source-free unsupervised adaptive segmentation for knee joint MRI, Biomed. Signal Process. Control 92 (2024).
[24] C.-P. Kuok, et al., Segmentation of finger tendon and synovial sheath in ultrasound image using deep convolutional neural network, BioMed. Eng. OnLine 19 (1) (2020) 24.
[25] S. Ourselin, L. Joskowicz, M.R. Sabuncu, G. Unal, W. Wells (Eds.), Medical Image Computing and Computer-Assisted Intervention – MICCAI 2016: 19th International Conference, Athens, Greece, October 17-21, 2016, Proceedings, Part II, Vol. 9901, Springer International Publishing, 2016.
[26] Z. Tian, et al., Graph-convolutional-network-based interactive prostate segmentation in MR images, Med. Phys. 47 (9) (2020) 4164–4176.
[27] T. Gaber, A. Tharwat, A. Ibrahim, V. Snael, A.E. Hassanien, Human thermal face recognition based on random linear oracle (RLO) ensembles, in: 2015 International Conference on Intelligent Networking and Collaborative Systems, 2015, pp. 91–98.
[28] R. Achanta, et al., SLIC superpixels compared to state-of-the-art superpixel methods, IEEE Trans. Pattern Anal. Mach. Intell. 34 (11) (2012) 2274–2282.
[29] D. Jain, Superpixels and SLIC [Online]. Available: https://darshita1405.medium.com/superpixels-and-slic-6b2d8a6e4f08.
[30] A. Ibrahim, S. Tominaga, T. Horiuchi, A spectral invariant representation of spectral reflectance, Opt. Rev. 18 (2) (2011) 231–236.
[31] A. Vedaldi, S. Soatto, Quick shift and kernel methods for mode seeking, in: Computer Vision, Vol. 5305, ECCV 2008, 2008, pp. 705–718.
[32] F. Meyer, The watershed concept and its use in segmentation: A brief history, 2012, [Online]. Available: http://arxiv.org/abs/1202.0216. (Accessed 4 October 2022).
[33] J. Cheng, J. Liu, Y. Xu, F. Yin, D.W.K. Wong, B.-H. Lee, T.Y. Wong, Superpixel classification for initialization in model-based optic disc segmentation, in: Engineering in Medicine and Biology Society (EMBC), 2012 Annual International Conference of the IEEE.
[34] M.S. Siddiquee, N.S. Pathan, Optic disc segmentation using superpixel based features and random forest classifier, in: 2019 4th Intl. Conference on Electrical Information and Communication Technology, EICT.
[35] S.N. Kumar, et al., Suspicious lesion segmentation on brain, mammograms and breast mr images using new optimized spatial feature based superpixel fuzzy C-means clustering, J. Digit. Imag. 32 (2) (2019) 322–335.
[36] N.B. Prakash, et al., Deep transfer learning for COVID-19 detection and infection localization with superpixel based segmentation, Sustainable Cities Soc. 75 (2021) 103252.
[37] A. Ibrahim, E.-S.M. El-kenawy, Image Segmentation Methods Based on Superpixel Techniques: A Survey, Tech. Rep., 2020, p. 11.
[38] L. Cong, S. Ding, L. Wang, A. Zhang, W. Jia, Image segmentation algorithm based on superpixel clustering, IET Image Process. 12 (11) (2018) 2030–2035.
[39] R.J. Gillies, P.E. Kinahan, H. Hricak, Radiomics: Images are more than pictures, they are data, Radiology 278 (2) (2016) 563–577.
[40] Pyradiomics documentation, 2023, [Online]. Available: https://pyradiomics.readthedocs.io/en/latest/features.html.
[41] Scikit-learn developers, Sklearn.preprocessing.StandardScaler, 2023, [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html. (Accessed 23 July 2024).
[42] T.N. Kipf, M. Welling, Semi-supervised classification with graph convolutional networks, in: ICLR, 2016.
[43] N. Amini, A. Shalbaf, Automatic classification of severity of COVID-19 patients using texture feature and random forest based on computed tomography images, Int. J. Imaging Syst. Technol. (2021).
[44] Unknown, Learning texture features from GLCM for classification of brain tumor MRI images using random forest classifier, WSEAS Trans. Signal Process. (2022).
[45] V.D.P. Jasti, A.S. Zamani, K. Arumugam, M. Naved, H. Pallathadka, F. Sammy, A. Raghuvanshi, K. Kaliyaperumal, Machine learning and image processing for medical image analysis of breast cancer diagnosis, J. Med. Imag. Health Inform. (2021).
[46] T. Alzyadat, S. Praet, G. Chetty, R. Goecke, D. Hughes, D. Kumar, G. Waddington, Automatic segmentation of achilles tendon tissues using deep convolutional neural network, in: Machine Learning in Medical Imaging: 11th Intl. Workshop, MLMI 2020, Held in Conjunction with MICCAI 2020, Springer Intl. Publishing.
[47] I. Scott, D. Connell, D. Moulton, S. Waters, A. Namburete, A. Arnab, P. Malliaras, An automated method for tendon image segmentation on ultrasound using grey-level co-occurrence matrix features and hidden Gaussian Markov random fields.
[48] J. Ma, Y. He, F. Li, L. Han, C. You, B. Wang, Segment anything in medical images, Nature Commun. 15 (2024) Open access.