AIIMDs: An Integrated Framework of Automatic Idiopathic Inflammatory Myopathy Diagnosis for Muscle

Manish Sapkota, *Student Member, IEEE,* Fujun Liu, *Student Member, IEEE,* Yuanpu Xie, *Student Member, IEEE,* Hai Su, *Student Member, IEEE,* Fuyong Xing, *Student Member, IEEE,* and Lin Yang, *Member, IEEE*

***Abstract*—Idiopathic Inflammatory Myopathy (IIM) is a common skeletal muscle disease that relates to weakness and inflammation of muscle. Early diagnosis and prognosis of different types of IIMs will guide the effective treatment. Interpretation of digitized images of the cross section muscle biopsy, which is currently done manually, provides the most reliable diagnostic information. With the increasing volume of images, the management and manual interpretation of the digitized muscle images suffer from low efficiency and high interobserver variabilities. In order to address these problems, we propose the first complete framework of automatic IIM diagnosis system (AIIMDs) for the management and interpretation of digitized skeletal muscle histopathology images. The proposed framework consists of several key components: (1) Automatic cell segmentation, perimysium annotation, and nuclei detection; (2) Histogram based feature extraction and quantification; (3) Content-based image retrieval (CBIR) to search and retrieve similar cases in the database for comparative study; and (4) Majority voting based classification to provide decision support for computer aided clinical diagnosis. Experiments show that the proposed diagnosis system provides efficient and robust interpretation of the digitized muscle image and computer aided diagnosis of IIM.**

***Index Terms*—Computer-aided diagnostic system, muscle, myopathies, microscopic images**

I. Introduction

IDIOPATHIC inflammatory myopathy (**IIM**) is a common muscle disease characterized by weakness and inflammation of the skeletal muscles [1]. IIM affects 5 to 10 people per million, with an average age between 40-50, and the female/male ratio of 2:1 [2]. IIM can be separated into three major categories: (a) polymyositis (**PM**), (b) dermatomyositis (**DM**), and (c) inclusion body myositis (**IBM**) [3]. Differentiation of these different types is clinically important because IIMs can be well treated if diagnosis is made at an early stage.

Studies have shown that muscle biopsies, which are interpreted in a laboratory with special expertise in enzyme histochemistry and myopathology, offer the most definitive diagnostic information for IIMs for the majority of the cases [4]. However, even with the expert knowledge, the misinterpretation of biopsies is still prevalent, causing misdiagnosis of the IIMs [1]. The inherent visual similarity between different cases of IIM makes it extremely challenging to render correct diagnosis [5]. The efficiency of the interpretation is also influenced by the large volume of the cases and the number of images per case that need to be interpreted.

Recent fast developing computer-aided diagnosis (**CAD**) systems could be a promising solution to help the precise diagnosis [6], [7]. For example, Gorelick et al. in [7] showed a promising result in automatic detection and grading of digitized histopathology prostate cancer images using a two-stage classification method. Similarly, Basavanhally et al. in [6] proposed a CAD method to automatically detect and grade the extent of lymphocytic infiltration in digitized breast cancer histopathology images with consistently precise performance.

In this paper, we propose a complete automatic IIM diagnosis system (**AIIMDs**) for muscle. To manage the large volume of image data, we propose to use a content-based image retrieval (**CBIR**) system followed by a majority voting for disease diagnosis. However, the efficient CBIR system for CAD relies on the efficient representation of the images [8], and the prerequisite is to identify the clinically significant image markers. Therefore, we propose an IIM CAD system with an entire pipeline comprised of following three steps: 1) Skeletal muscle image marker (illustrated in Figure 1(a)) identification, which contains muscle fiber (individual contractile units in the muscle) segmentation, perimysium (dense collagen between bundles of muscle fibers [9]) annotation, and nucleus (relatively smaller and elongated or oval structure normally inside muscle fiber) center detection; 2) Muscle image feature extraction, which includes computing the cross section area (**CSA**) of muscle fibers and the physical distribution of nuclei centers; 3) Automatic IIM diagnosis, which involves the establishment of an entire muscle image CBIR module. In the past few decades, although lots of CAD systems have been developed to process and diagnose different radiology and pathology images [10], [11], to the best of our knowledge, this is the first CAD system to process IIM image data. The major contribution of this paper is the first integrated framework of automatic IIM diagnosis shown in Figure 1(b), and it includes:

- An automatic muscle cell segmentation module to handle heavily apposed cells of muscle fibers, perimysium annotation based on learned features, and nuclei detection.
- An effective feature extraction method that is designed specifically for processing skeletal muscle images.
- A CBIR module using high-performance computing for skeletal muscle comparative studies, and an automatic diagnosis module of IIM based on a majority voting of the CBIR results.

M. Sapkota, F. Liu, and F. Xing are with the Department of Electrical and Computer Engineering, and Y. Xie, H. Su and L. Yang are with J. Crayton Pruitt Family Department of Biomedical engineering, University of Florida, Gainesville, FL, 32611 USA. E-mail:msa235@ufl.edu

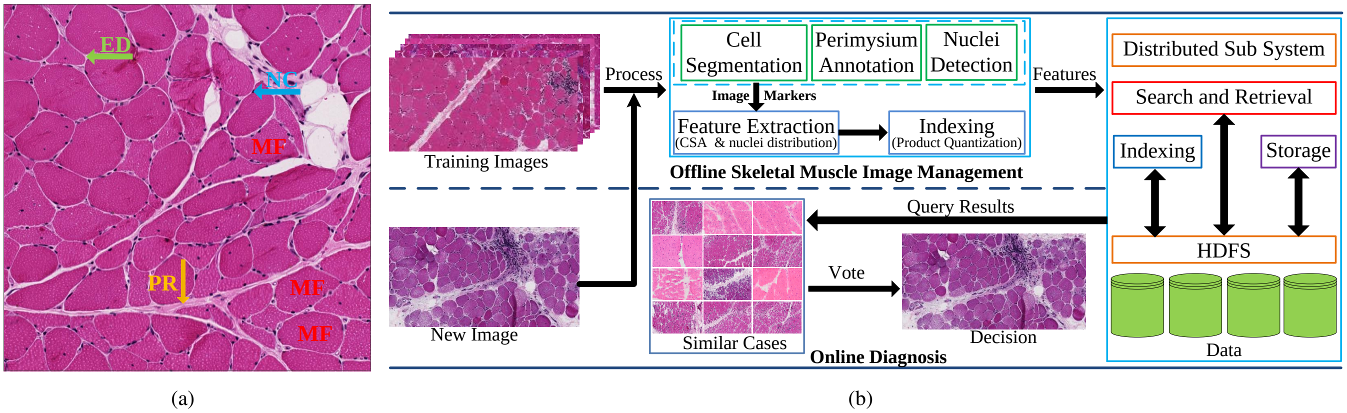


Fig. 1: *(a) An example of the skeletal muscle image illustrating the muscle fibers (MF), perimysium (PR) region pointed by orange arrow, endomysium (ED) region pointed by green arrow and nuclei (NC) pointed by blue arrow. (b) The complete architecture of the proposed CAD system for IIM. The system consists of muscle cell segmentation, perimysium annotation, and nuclei detection for skeletal muscle image marker identification, histogram based feature representation to summarize CSA and nuclei distribution, and a distributed CBIR module for image management. For testing, we process the test image for image marker identification and extract the features, and query the CBIR module to retrieve similar cases. We render the final diagnosis for the test image based on the majority voting on the retrieved images.*

II. Related Work

A. Muscle Cell Segmentation

One of the most reliable information in the diagnosis of IIM is the image morphological features that uniquely distinguish diseased skeletal muscle from healthy specimens. Recent studies suggest that important morphological characteristics of muscle fibers, such as CSA, are critical factors to determine the health condition of the muscle [12]. The prerequisite to quantify CSA and extract other representative morphological features is to accurately segment each individual muscle fiber. Recently, many automatic and semi-automatic methods such as marker-controlled watershed [13], seed detection based active contours [14], and graph-based methods [15] have been proposed, but none of them are specifically designed to process heavily apposed cells like muscle fibers. Liu *et. al* have proposed a two steps cell segmentation method [16] to specifically deal with muscle fibers based on seed detection.

B. Perimysium Annotation and Nuclei Detection

In addition to the CSA, the different types of IIMs exhibit variations in fiber shapes and nuclei distribution patterns in or near the regions of perimysium [17]. For example, diseased skeletal muscle often exhibits dense nuclei distributed around the regions of the perimysium, while the normal skeletal muscle does not. This evidence provides a distinctive image-marker that could be used for effective diagnosis and prognosis of the muscle disease. Therefore, accurate delineation of the perimysium region and robust detection of the muscle fiber nuclei are critical to building the CAD system for muscle.

Many computer algorithms for automatic annotation and detection have been proposed in the literature [18], [19], most of them are based on hand crafted features and heuristic rules. Due to the high variability of the patterns shown in histopathology images, it is difficult to design robust feature descriptors for automatic skeletal muscle image analysis. On the other hand, there is an encouraging evidence that automatically learned representation of biomedical images using deep neural network usually outperforms the handcrafted features in a wide range of applications such as detection, segmentation and diagnosis of different diseases [20]. However, the sliding window-based approaches [20] fail in modeling the global semantic information by exploiting the context information, which could improve performance of perimysium annotation.

Our recent work [21] has used recurrent neural networks (**RNNs**) to model the semantic information by considering the context information as chain structured data and successfully applied it to annotate perimysium in 2D muscle images. Specifically, this work presents a 2D spatial clockwork RNN (**SCW-RNN**), which is an extension to the chain structured clockwork RNN (**CW-RNN**) [22]. It directly encodes the 2D contextual information of the entire image into the representation of local patches. Meanwhile, it leverages the structured regression [23] to compute a complete prediction mask for each local patch, and thus avoids inefficient patch-wise classification.

C. Content-based Image Retrieval

Given a new skeletal muscle image, it is common to search for relevant cases in the database that exhibit similar image content, and this can be achieved by CBIR. A typical CBIR framework often contains two stages: offline and online [8]. For a CBIR system based on machine learning techniques, in the offline stage, image signatures (features) are extracted from the image data, and prediction models are learned based on these image signatures. In the online stage, the features of the query images are calculated and applied to the database to retrieve the most similar content images in the database.

Due to the gradually increasing amount of patient data, scalable or real-time (sub)image retrieval techniques have been proposed, such as, offline registration and online searching combined to retrieve similar X-ray images [24] and, hashing-based method to retrieve breast cancer images [25]. In [26], proposed a product quantization (**PQ**) method to provide efficient indexing and matching in the carefully designed feature space. Similarly a parallel retrieval system based on

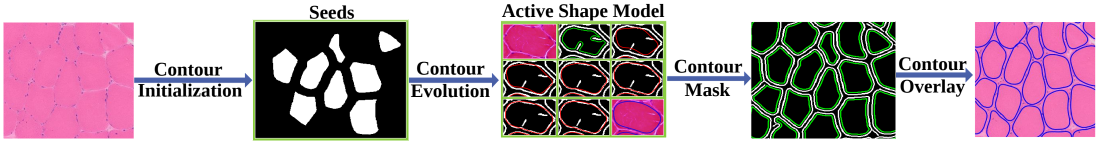


Fig. 2: *The flowchart of the proposed muscle cell segmentation method.*

a demand-driven master-worker parallelization platform for prostate cancer images [26] has also been proposed.

III. Muscle Cell Segmentation

In this section, we introduce a novel muscle cell segmentation method, which consists of two steps [16] for segmentation: (1) initial muscle cell geometric center (seed) detection, and (2) cell boundary evolution with the active shape model [27]. The seed detection step estimates the number and the locations of cells, and guides the subsequent model for final segmentation. This two-step segmentation approach can improve the representation of the diseased muscle images, avoiding incorrect measurement of CSA due to missing muscle fiber detection. The flowchart is shown in Figure 2.

A. Learning Based Seed Detection

The first step of automatic muscle image segmentation is to find the geometric center of each individual muscle fiber. We treat these centers as seeds to initiate the subsequent contour evolution step. We propose to use a pixel-wise classification method for seed detection, where each pixel is labeled as muscle fiber or non-muscle fiber by a learned classification model. In order to improve the robustness, especially for cases where the endomysium region, i.e. the region between two cells [9], is not clear, we combine the pixel-wise classification with the Ultrametric Contour Maps (**UCM**) [28].

For pixel-wise segmentation, we utilize the image texture information as local image intensity distribution feature to construct a model. Textons [29], which are defined as repetitive local features that humans perceive as being discriminative between textures, is adapted to generate a compact representation of textures. We use the multiple scale Schmid filter bank [30] composed of 13 rotation invariant filters to obtain filter responses for all local image patches:

$$F(r,\sigma,\tau) = F_0(\sigma,\tau) + \cos\left(\frac{\pi r \tau}{\sigma}\right) e^{-\frac{r^2}{2\sigma^2}}, \tag{1}$$

where $r = \sqrt{x^2+y^2}$ for each pixel location (x, y) in the image, τ is the number of cycles of the harmonic function for Gaussian filters, σ is the scaling factor for the Gaussian distribution, and $F_0(\sigma,\tau)$ is the mean frequency response at zero frequency. The responses computed using Equation (1) are clustered with K-means. Cluster centers are treated as elementary atoms and are put together to construct a large codebook. In our implementation, we set the number of cluster centers to 32, which is learned offline and kept constant throughout the experiments.

To train our classification model, we extract the same amount of positive and negative patches from the muscle fiber and nonmuscle fiber regions in training images based on the manually labeled ground-truth masks. The patches with their center pixels located within a muscle fiber region are considered positive samples, and negative otherwise. The training patches are represented with a texton histogram based on the codebook, where each pixel of the training patch is assigned to its closest texton.

In order to achieve scale invariance, the texton histogram is extracted from 5 different window sizes (4, 8, 16, 32, and 64 pixels respectively) and concatenated into one large feature vector. This concatenated texton histogram is used as feature to train a logistic boosting classifier [31] to distinguish muscle fiber and non muscle fiber regions. In the testing stage, the boosting model is applied to test images for pixel-wise classification to obtain binary region information.

Sometimes, the aforementioned pixel-wise classification can combine multiple seeds into one seed, especially for regions exhibiting narrow or unclear endomysium regions. In order to improve the robustness of the seed detection, we apply UCM, which is a soft boundary image representation of the hierarchical segmentation operator (HSO) [28], to get the edge information from the skeletal muscle image. Next, the output of pixel-wise classification (BW_{px}) and the edge information (BW_e) obtained from UCM are combined, which removes the false positives and false negatives, as:

$$BW_s = BW_{px} \cap BW_e, \tag{2}$$

where $\cap$ is the intersection operation. Finally, we perform connected component analysis to detect seeds. After seed detection, only those seeds with area larger than a predefined threshold are considered, and these seeds are used to initialize a deformable model for segmentation.

B. Cell Segmentation by Balloon Snake

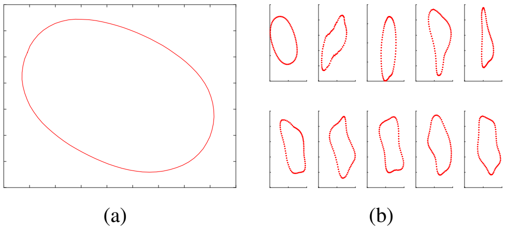


Fig. 3: *(a) The mean shape of the training contours; (b) The shapes that represent the first 10 largest eigenvectors.*

The next step in the automatic muscle image segmentation is to find the boundaries of the muscle fibers initialized from the seeds. In this work, we propose to use the parameterized active contour model [32] with active shape constraints [27] to generate smooth boundaries. Although the non-parameterized contour evolution method such as level set [33] could also be used with shape constraints for segmentation, we choose parameterized method because it preserves the topological structures of the muscle cell fibers in the image. Additionally, parameterized methods are convenient to use in conjunction with shape constraints in the active shape modeling.

A snake is an active contour evolution model that is attracted towards salient features (e.g. intensity, edges, etc) in the image $I(x, y)$. The position of the snake is represented parametrically by $v(s) = (x(s), y(s))$, $s \in [0, 1]$, whose components denote the image coordinates [34], [35]. It is defined to minimize the energy function

$$E = \int_0^1 \frac{1}{2}[\alpha|v'(s)|^2 + \beta|v''(s)|^2] + E_{ext}(v(s))ds, \quad (3)$$

where $v'(s)$ and $v''(s)$ are the first and second derivatives representing the internal energy that control the regularity of the curve, and α and β controls the elasticity and rigidity of the curve, respectively. $E_{ext}(v(s))$ in (3) is the external energy, which is a function of the salient features (e.g. the image gradient) in the image data. By incorporating an extra pressure force, the balloon snake [32] calculates its internal and external forces as follows:

$$F_{int}(v) = \alpha v''(s) - \beta v''''(s), \quad (4)$$

$$F^B_{ext}(v) = \gamma \mathbf{n}(s) - \lambda \frac{\bigtriangledown E_{ext}(v(s))}{||E_{ext}(v(s))||}, \quad (5)$$

where $\mathbf{n}(s)$ represents the normal vector to the snake at the specific point on $v(s)$, γ is the corresponding parameter, and λ represents the weight for the normalized gradient. The first term in (5) serves as the pressure to enforce the snake to inflate or deflate (the sign of γ determines the inflation or deflation), while the second acts as the smoothing term that facilitates convergence to the object boundaries. The values for parameters γ and λ are usually chosen of the same magnitude in the order of pixel size, and λ is set slightly larger than γ to stop the inflation at the edge point. In our implementation, we follow the suggestions in [32] and set the values of γ, λ, α, and β to 0.6, 1, 0.05, and 0 respectively. They work generally well for the images in our database. An example of automatic segmentation with blue overlaid boundaries of the segmented fibers is illustrated in Figure 2.

Because of the bottom-up approach of active contour, the inhomogeneity inside the object may affect the contour evolutions to create bizarre segmentation results. In our algorithm, we propose to create the shape models to provide top-down constraints to the contour evolutions. At the learning stage, a set of training shapes is annotated by the human. Each training contour is aligned to a reference shape and the four points lying on the major and minor axes, and the uniformly sampled points between them are used as the control points. Procrustes analysis is utilized to find the optimal translation and rotation for the alignment. With all the training shapes aligned, each contour is represented by the coordinates of the uniformly sampled points. Let $\mathbf{x}_i \in \mathbb{R}^{2n \times 1}$, where n denotes the number of points that are uniformly sampled from the contour such that the distance between two consecutive points in a contour is not larger than an empirically set threshold. During the experiment, we found out that the segmentation results are not sensitive to this threshold and set $n = 77$. The covariance matrix is then computed as:

$$\mathbf{CV} = \frac{1}{N}\sum_{i=1}^{N}(\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^T, \quad (6)$$

where N is the number of training samples, $\bar{\mathbf{x}}$ is the mean of the training data (Figure 3(a)), and $(\cdot)^T$ is the vector transpose operation.

Principal component analysis (PCA) is applied to the training dataset to find the dimensions with the largest variation. In our implementation, we keep the eigenvectors corresponding to the first 10 largest eigenvalues that contain the 99% of the energy (Figure 3(b)). In testing, a sample $\mathbf{x}$ can be represented by the mean and a linear combination of the eigenvectors as:

$$\mathbf{x} = \bar{\mathbf{x}} + \mathbf{V}\mathbf{b}, \quad (7)$$

where $\mathbf{V}$ denotes the matrix formed by the reserved eigenvectors and $\mathbf{b}$ is the linear coefficients that can be solved by the least square optimization.

IV. Perimysium Annotation and Nuclei Detection

A. Spatial Clockwork Recurrent Neural Network

An RNN is a neural network with recurrent connections that allow the network to learn new patterns based upon the past input patterns. For a simple RNN (**SRNN**), at each time step, the output at the current hidden state $\mathbf{h}^t$ and the output state $\mathbf{o}^t$ is given by:

$$\mathbf{h}^t = f(\mathbf{W}\mathbf{x}^t + \mathbf{U}\mathbf{h}^{(t-1)} + \mathbf{b}_h), \quad (8)$$

$$\mathbf{o}^t = g(\mathbf{V}\mathbf{h}^t + \mathbf{b}_o), \quad (9)$$

where $\mathbf{x}^t$ is the current input, $f(.)$ and $g(.)$ are the nonlinear activation functions, $\mathbf{W}$ is the weighted connections between the input and hidden units, $\mathbf{U}$ is the recurrent weighted connections of hidden units, $\mathbf{V}$ is the weighted connections between the hidden and output units, and $\mathbf{b}_h$ and $\mathbf{b}_o$ are the bias terms for the hidden and output units, respectively.

One drawback of the SRNN is that it suffers from exploding or vanishing gradient problem when the time step becomes increasingly large, which prevents the SRNN to learn long-term temporal dependencies. CW-RNN [22] has addressed this problem by partitioning the hidden units into M different modules ($\mathbf{h}^m|m = 1, ..., M$), each of size k, and associating each module with its clock (or temporal) period ($T_m \in T_1, ..., T_M$). At anytime step t, only the neurons in module m, where t mod $T_m = 0$, will be updated. This allows for the units with slower update rates to preserve the long-term information. Additionally, this approach simplifies the network to have a less number of active weights by allowing the connections in the direction of hidden units of slower modules to faster modules, and not necessary vice versa. In this work, we present an SCW-RNN by extending the idea of CW-RNN to image domain for region annotation. Since the static images do not exhibit sequential information, we modify the CW-RNN such that at any time state it can receive information from its predecessors in both the dimensions of the image (see [21] for details).

B. Perimysium Annotation

Perimysium annotation focuses on labeling the region between the bundles of muscle fibers, as illustrated by green overlaid region in the annotation result of Figure 4 and yellow overlaid regions of Figure 5 (b). Formally, given a set of

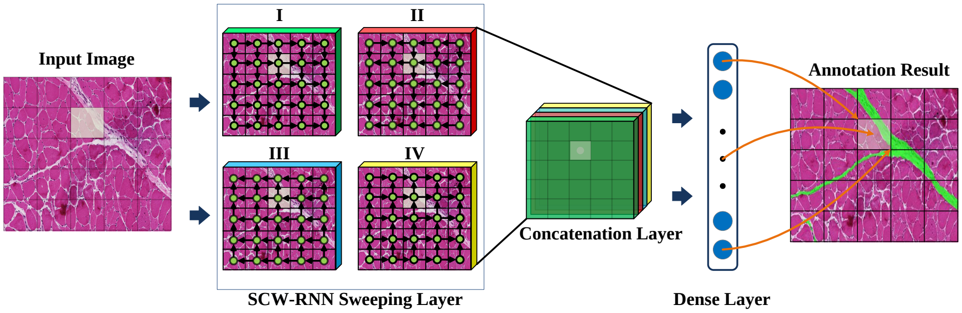


Fig. 4: *An example architecture of the proposed SCW-RNN. The SCW-RNN sweeping layer and the dense layer represent the proposed modified RNN and fully connected layer, respectively. I, II, III, and IV show the different sweepings directions (indicated by the directed arrows) in the SCW-RNN sweeping layer to encode the context information from a 4-connected neighborhood for each local patch. The concatenation layer merges the activations from four sweepings together to represent the global context information for each local patch. The brown arrows illustrate the mapping between the output of the dense layer and the predicted perimysium mask (green overlay on the original image) for each local patch.*

training RGB image patches $I_i \in R^{r \times c \times 3}, i = 1, ..., N$ with dimensionality $r \times c$ for each of the 3 channels, we aim to learn an RNN-based mapping function to assign one of the binary class labels to each pixel (positive if pixel belongs to perimysium regions and negative otherwise). We train a SCW-RNN to model and predict the perimysium regions. In order to ensure that SCW-RNN processes the image, we divide the image into smaller local patches and arrange them in an acyclic sequence. Furthermore, the input image (and/or feature map) is swept from four different corners, and the activations from four directional sweepings are concatenated and fed to the successive layers to produce the final prediction as illustrated in Figure 4. This allows for each local patch to receive information from 4-connected neighborhood (and it could be easily extended to 8-connected neighborhood). We use a structured regression loss [23] and train the entire network end-to-end using standard BPTT [36] algorithm to optimize for annotation. During test, the trained network is applied to the input image to produce the annotation mask with the same size as the input image.

C. Nuclei Detection

Given a skeletal muscle image, nuclei detection is defined as seeking a set $C_{nu} = \{cnu_1, cnu_2, ..., cnu_n\}$ of centroid coordinates of the nuclei. Unlike perimysium regions, nuclei in the Haematoxylin and Eosin (**H&E**) stained images (*dark red dots in Figure 5 (a) and marked with green crosses in Figure 5 (b)*) exhibit consistent patterns. Hence, in our algorithm, we apply a simple adaptive thresholding based approach [37] to detect nuclei centroid in the skeletal muscle image.

V. Muscle Image Features Extraction

In clinical diagnosis, IIM are categorized into three major subsets of diseases according to ICD-10: PM, DM, and IBM. The accurate and objective diagnosis of IIM is important because it affects the treatment decision (using corticosteroids, intravenous immunoglobulin, or immunosuppressive medications, etc.) and patients' prognoses. However, subtype recognition based on H&E stained slides could be hampered by the pathologists' experience or even personal preferences. As a definitive diagnostic test to differentiate IIM subtypes and exclude other disorders, pathological evaluation is necessary but it is also the most common cause of misdiagnosis due to incorrect interpretations [1], [38]. When pathologists' decisions are assisted by CAD, it can significantly improve objectivity, reproducibility, and effectiveness [6], [11].

IIMs are characterized by diverse profusion of cell population, which has skeletal muscle cells with irregular shapes and different sizes, and mononuclear cell infiltration in the extra cellular matrix. Especially, DM shows greater degree of mononuclear cell infiltration and small regenerating muscle cells in/near perimysium region; this is different from PM and IBM, which exhibit mononuclear cell infiltration more towards endomysium regions with degenerating elongated muscle cells. Experts in the research community of IIMs identify and evaluate the qualitative features such as sizes and shapes of muscle fibers, and nuclei distribution in ECM to diagnose the types of disease [39].

In our system, we propose to quantitatively extract the histopathology features such as CSA and distance between nuclei from perimysium using accurately segmented muscle cells, annotated perimysium, and detected nuclei to evaluate the types of IIMs according to the the aforementioned suggestions. We compute the features for each cell and nucleus in the image, and compose a histogram based feature representation [40] for further analysis or classification. All the computed features are scaled and normalized to the range of $[0, 1]$ for retrieval and subsequent classification.

Figure 5 displays two structure-level features that can differentiate between DM, PM, and normal muscles: The spatial distribution of 1) nuclei and 2) muscle fiber CSA with respect to the position of the perimysium. The spatial distribution of muscle fiber CSA is represented as a histogram. The distance of the nuclei with respect to the perimysium regions is also presented as a histogram. For a given image with automatic segmented perimysium region and detected nuclei, we compute the minimum distance between the nucleus center and the perimysium boundary, which is defined as:

$$d(x, P_m) = \begin{cases} -min\{d(x, y) : y \in B_p\} \text{ if } x \in P_m, \\ min\{d(x, y) : y \in B_p\} \text{ otherwise,} \end{cases} \tag{10}$$

where x is the center of the nucleus, B_p is the set of

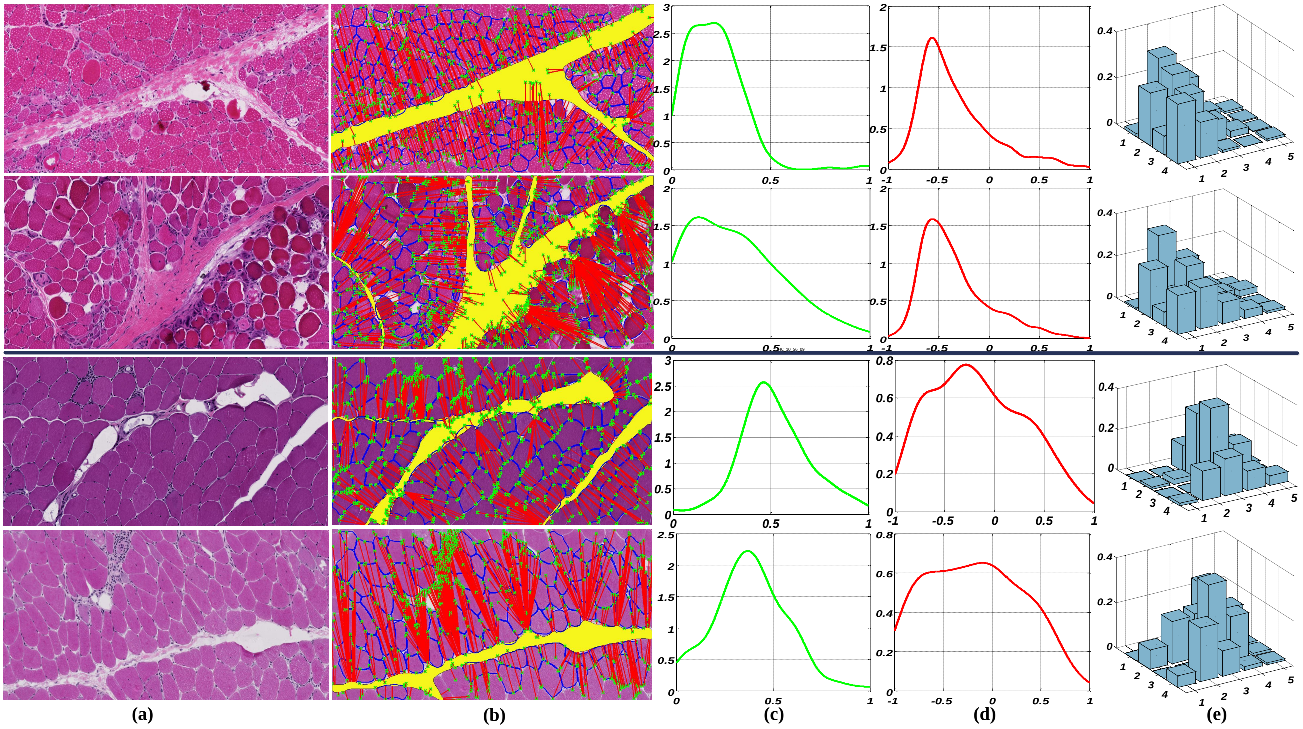


Fig. 5: *An example to show the variations of CSA and nuclei distribution of skeletal muscle images representing different types of IIMs (images in the* ***top two-rows*** *are cases exhibiting DM and the* ***bottom two-rows*** *are cases exhibiting PM). (a) The original images; (b) The annotated perimysium (yellow), detected nuclei (green crosses), and segmented muscle fibers (blue contours) overlaid on the original image; (c), (d), and (e) CSA distribution, nuclei distribution, and histogram plot of the combined features (bin size of 10 for each features has been used for better illustration). We can observe that the same type of IIMs exhibit similar distributions.*

points in the perimysium boundary, and P_m is the set of points in the perimysium. The computed distances are then summarized with a standard vector quantization procedure and then represented as the nuclei distribution spatial histogram.

As one can tell, PM tends to have both perimysial and endomysial lymphocytic inflammations and a scattered pattern of myofiber atrophy. PM also sometimes shows lymphocyte invasions into otherwise non-necrotic fibers. In DM, the inflammations and atrophic myofibers are predominantly perifascicular.

VI. Automatic Diagnosis

Currently, given a new image, muscle researchers are unable to identify previous cases that exhibit similar morphometric measures. One-by-one manual eyeballing based comparisons are impractical even within a small dataset. We, therefore, propose to develop a CBIR unit to rapidly identify and visualize content-based similar muscle images. This function would be exceptionally useful for investigating morphology-biology co-correlations, as well as complex patient cases in which the diagnosis is unclear and evidence-based computer-aided recommendations are needed.

Although CBIR based majority voting can help doctors to make a better diagnosis, a lot of muscle images hinder the success of efficient CBIR. To address these problems, we have implemented a fast CBIR system using PQ [41]. In addition, in order to accommodate the requirement for high-throughput analysis, we have implemented the indexing step during the retrieval within a distributed system, where the MapReduce technique [42] is used to cope with the large scale dataset.

A. PQ-based Search

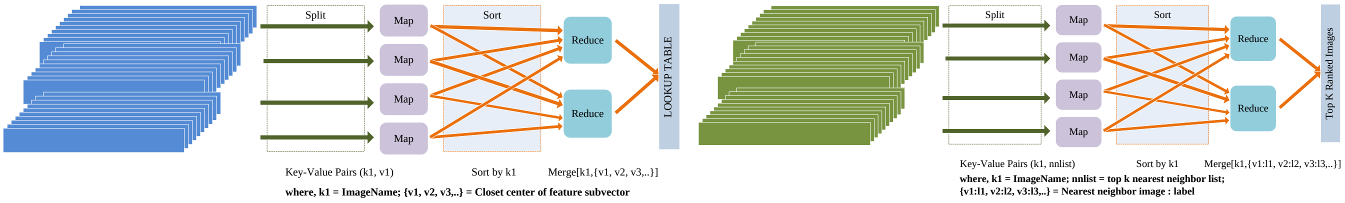
PQ is a type of vector quantization technique that maps feature vectors into short codes for fast similarity search and retrieval [41]. PQ splits the vectors into several distinct subvectors, and then quantizes each subvector separately. Given a vector $x \in R^{d\times 1}$ in a finite set X of data points (image features), PQ quantizer finds a function q that maps x into a codeword $c \in C$ with the constraint of Cartesian product. The objective function of PQ [41] is formulated as:

$$\min_{q(x)} \sum_{x\in X} ||x-q(x)||_2^2,\ s.t.\ q(x) \in C = C_1 \times C_2 \times\ \dots\ \times C_m. \tag{11}$$

Due to the Cartesian product constraints, the nearest neighbor of x is generated by concatenating its nearest sub-codewords. We can solve (11) by applying K-means to each sub-problem and concatenating the sub-codewords to form a single quantized vector. This PQ-encoding can help to reduce the computational complexity and the memory requirement during similarity searching. In our method, PQ is used to index and manage the existing skeletal muscle images during the training phase, and quickly search for similar cases in the testing phase.

B. MapReduce-based Search and Retrieval

MapReduce is a programming framework to support parallel computing for large-scale datasets using computer clusters [42]


Split
Map
Map
Map
Map
Sort
Reduce
Reduce
LOOKUP TABLE
Key-Value Pairs (k1, v1)
Sort by k1
Merge[k1,{v1, v2, v3,..}]
where, k1 = ImageName; {v1, v2, v3,..} = Closet center of feature subvector
Split
Map
Map
Map
Map
Sort
Reduce
Reduce
Top K Ranked Images
Key-Value Pairs (k1, nnlist)
Sort by k1
Merge[k1,{v1:l1, v2:l2, v3:l3,..}]
where, k1 = ImageName; nnlist = top k nearest neighbor list;
{v1:l1, v2:l2, v3:l3,..} = Nearest neighbor image : label

(a) (b)

Fig. 6: *(a) Offline phase: The lookup table will be constructed with partial PQ indexing. The operation is performed in each of the Map methods obtained from the training signatures distributed in all the machines in the cluster. (b) Online phase: Multiple/single query will be performed on the indexing structure in different machines to get the nearest matches.*

Algorithm 1: Encode database vectors with MapReduce
Input: Image signatures, PQ centers, number of subvectors: nS, and centers: nC.
Output: Lookup Table

```
Function Map (key, value)
    subvectors ← split(value,nS,nC)
    code ← []
    for $i^{th}$ subvector
        $j$ ← index of closest center
        code ← code + j + ” ”
    end for
    emit(key, code)
end

Function Reduce(key,values)
    emit(key, center)
end
```

Algorithm 2: The online query using MapReduce
Input: Image signatures, lookup table, database labels, PQ centers, number of subvectors: nS, and centers: nC, k.
Output: Retrieved images with labels

```
Function Map (key, value)
    feats ← split(value)
    nnlist ← PQTopKSearch(feats,k)
    emit(key, nnlist)
end

Function Reduce(key,values)
    items ← split(values)
    for each item in items
        ret ← ret + item + ”(” + get_label(item) + ”) ”
    end
    emit(key, ret)
end
```

Using two major functions Map and Reduce, this programming model takes a set of input key/value pairs, and produces a set of output key/value pairs. The Map function processes the input key/value pairs to generate a set of intermediate key/value pairs, and the Reduce function merges all the intermediate values associated with the same intermediate key. The programming in this functional style is automatically distributed and executed on a large cluster of machines.

Based on the MapReduce architecture, similar muscle image content retrieval is achieved by implementing the PQ indexing in a distributed environment. For our application, we propose to build a distributed sub-system of medical image retrieval using MapReduce that processes the single/multiple query image(s) from the user and retrieves the similar cases based on the measurements of the image features. This sub-system is of particular interest in the biomedical community, because it serves as a core module for a complete CAD system. The MapReduce architecture of the distributed sub-systems is illustrated in Figure 6. It consists of two phases: 1) Offline phase (Figure 6(a)), in which the MapReduce directs the training image features into different machines to build the indexing structure; 2) Online phase (Figure 6(b)), in which the MapReduce distributes the query image(s) features to all the machines, and performs searching, counting, and sorting.

Algorithm 1 demonstrates the MapReduce based distributed implementation of the training PQ, which consists of encoding database vectors to generate a lookup table by finding the nearest cluster center for each subvector. The cluster centers are generated using cartesian K-means, which is also implemented in the MapReduce framework. For online query, we first compute the distance matrix between the query image and cluster centers, and then search and rank images in the database using the lookup table and the distance matrix as illustrated in Algorithm 2.

VII. Experiments

A large dataset of digitized histopathological skeletal muscle images, which consists of different types of IIM, are used in the experiments. The dataset has 718 skeletal muscle images (including challenging cases that exhibit weak cell boundaries, freeze artifacts, and dense mononuclear cell infiltration in the regions of perimysium and endomysium) randomly cropped from whole slide scanned muscle biopsies corresponding to 41 individual subjects. The images contain two major categories of IIM: **PM** and **DM**. The whole slide scanned images are prepared by the Medical College of Wisconsin Neuromuscular Laboratory (MCWNL) on an Hamamatsu NanoZoomer Microscope. The images are captured at a 40× objective with pixel resolution of 0.25 micron. All of the ground-truths are provided by MCWNL. All of the images are analyzed by 3 independent pathologists and a final label is assigned to the images based on the common consensus among them.

A. Muscle Cell Segmentation

First, we evaluate the performance of the proposed muscle cell segmentation method using randomly selected 260 muscle images. In addition, we select the equal number of images from each class to train the classification model for seed detection and the shape modeling for active contours. The results are compared to those of a state-of-the-art mean-shift (MS) [43], isoperimetric (ISO) [44], marker controlled watershed (MWS), and repulsive level set (RLS) [33]. Quantitative assessment is conducted to compare results of the different methods with the ground-truth annotations. All the state-of-the-arts methods are implemented using the source codes provided by the authors, and default parameters settings are used to conduct the experiments. For our proposed method, we discussed the

TABLE I: *The summary of the quantitative comparison of the muscle fiber segmentation results using the proposed method and other methods with ground-truth annotations (STD = Standard deviation).*

Methods	DC (%)			SN (%)			HD (pixels)			MAD (pixels)		
	Mean	Median	$STD^{\dagger}$	Mean	Median	$STD^{\dagger}$	Mean	Median	$STD^{\dagger}$	Mean	Median	$STD^{\dagger}$
MS	0.73	0.77	0.15	0.88	0.94	0.18	15.26	13.35	8.36	8.31	7.32	4.61
ISO	0.77	0.80	0.12	0.85	0.89	0.14	16.12	11.75	19.54	10.65	8.07	10.54
MWS	0.87	0.88	0.04	0.83	0.83	0.04	6.19	5.30	3.14	4.98	4.40	2.14
RLS	0.87	0.88	0.05	0.87	0.88	0.06	5.64	4.78	3.14	4.25	3.69	2.00
our	**0.93**	**0.93**	**0.04**	**0.96**	**0.97**	**0.05**	**3.93**	**3.18**	**3.64**	**2.88**	**2.37**	**2.35**

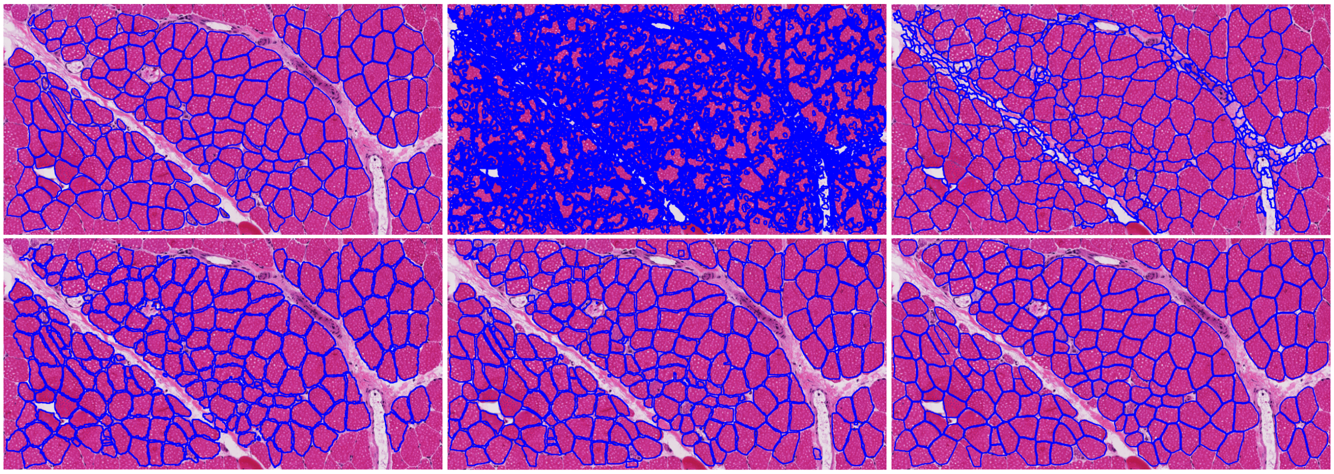


Fig. 7: *The automatic segmentation results (overlaid blue contours) on a skeletal muscle image using different methods.* ***Row 1, from left to right:*** *The ground-truth segmentation provided by human experts, segmentation results using MS [43] and ISO [44];* ***Row 2, from left to right:*** *The segmentation results using MWS, RLS [33] and our proposed method, respectively.*

parameter selection in Section III. Four quantitative metrics are used for evaluation, including dice coefficient (DC), sensitivity (SN), Hausdroff distance (HD), and mean absolute distance (MAD):

$$DC = \frac{2|I_a \cap I_g|}{(|I_a| + |I_g|)}, \tag{12}$$

$$SN = \frac{|I_a \cap I_g|}{|I_g|}, \tag{13}$$

$$HD = \max\left\{\frac{1}{N}\sum_{i=1}^{N} d(sa_i, S_g), \frac{1}{M}\sum_{j=1}^{M} d(sg_j, S_a)\right\}, \tag{14}$$

$$MAD = \frac{1}{2}\left\{\frac{1}{N}\sum_{i=1}^{N} d(sa_i, S_g) + \frac{1}{M}\sum_{j=1}^{M} d(sg_j, S_a)\right\}, \tag{15}$$

where I_a and I_g denote the automatic segmentation result and manual annotation, respectively. $S_a = \{sa_1, sa_2, \dots, sa_N\}$ and $S_g = \{sg_1, sg_2, \dots, sg_M\}$ are the pixels in the edges of the automatic and the ground-truth segmentation results respectively, and $d(p_i, Q) = \min_j \|p_i - q_j\|$.

Table I summarizes the segmentation results evaluated with different metrics. Experimental results show that the proposed segmentation method in general has higher performance than all the other methods according to all the comparison metrics. Higher values of DC and SN in the results demonstrate that there is better overlapping of cell areas between the ground-truth and the proposed method. In addition, lower values of HD and MAD indicate that there is higher degree of agreement between the cell edges obtained using our proposed method and the ground-truth. It is to be noted that all of the segmentation results using the proposed method are obtained in less than one minute even with a MATLAB implementation.

To further explain the superior performance of our method compared to other methods, we take a look at the qualitative results. Figure 7 shows the segmentation results (blue) on one of the test image. The images in the first row are the ground-truth annotation, and results using MS and ISO. Similarly, the images in the second row are the results using MWS, RLS and our proposed method, respectively. Non-seed based methods (MS and ISO) suffer from over segmentation and boundary overlapping. On the other hand, seed-based methods (RLS, MWS and ours) naturally provide better performance because of the relatively high intensity contrast between the muscle fibers and cell boundaries, and the proposed method produces the best segmentation results. This is because the shape constraints introduced in our method encourage the formation of smoother contours and ensure the evolving contours are consistent to the training shapes.

B. Perimysium Annotation

Next, we evaluate the performance of the automatic perimysium annotations. The ground truth of perimysium regions is manually annotated by several experts. For evaluation of the perimysium annotation, we sample approximately 420 digitized muscle images, out of which 190 images are randomly selected for testing and the rest are used for training. For quantitative analysis, we calculate precision (P), recall (R),

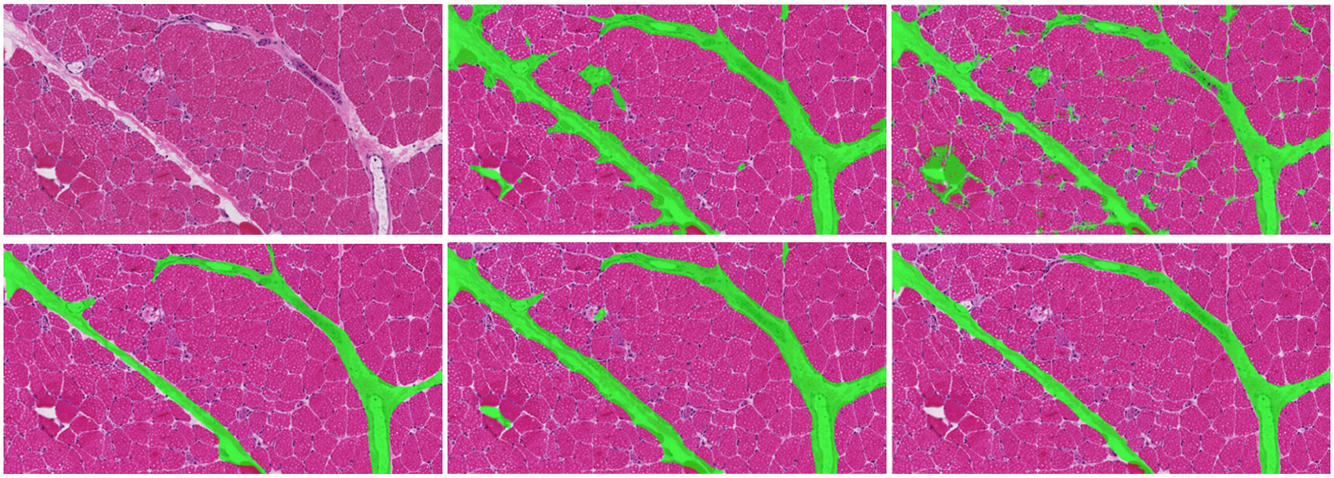


Fig. 8: *The automatic perimysium annotation results (green regions) on an example skeletal muscle image using different methods.* ***Row 1, from left to right:*** *The original image, and results obtained using MLP-48 and CNNP [20];* ***Row 2, from left to right:*** *The ground-truth annotation provided by human experts, and results obtained using CNN-NMS [45] and our proposed SCW-RNN(8), respectively.*

TABLE II: *The network architecture. Dense represents the fully connected layer applied individually to every time step. SCR represents sweeping layer of SCW-RNN, where 4 arrows indicate the 4 sweeping directions. The $Inputs$ row specifies the layer ID of each layer's inputs. Layer 7 takes the concatenation of the output from layer 3,4,5 and 6 as input.*

Layer ID	1	2	3-6	7	8
Layer	**Input**	**Dense**	**SCR**(↘↙↗↖)	**Dense**	**Dense**
Size	300	100	384	100	100
Inputs	-	1	2	[3,4,5,6]	7

TABLE III: *The quantitative automatic perimysium annotation results (Precision (P), Recall (R) and F_1-score) compared with the ground-truth using SCW-RNN and other methods. Table also shows the average annotation time (Avg. Time) in seconds for each of the methods on the test images.*

	P(%)	R(%)	F_1(%)	Avg. Time (Sec)
MLP-10	0.75	0.79	0.76	7.14
MLP-48	0.79	0.80	0.79	17.73
CNNP [20]	0.80	0.60	0.66	-
CNN-NMS [45]	0.81	**0.85**	0.83	203.46
SCW-RNN(4)	**0.84**	0.83	0.83	1.92
SCW-RNN(8)	0.83	**0.85**	0.83	2.63

and F_1-score defined as:

$$P = \frac{N_{TP}}{N_{TP} + N_{FP}}, \ R = \frac{N_{TP}}{N_{TP} + N_{FN}}, \ F_1 = \frac{2PR}{P + R}, \tag{16}$$

where N_{TP}, N_{FP}, and N_{FN} denote the numbers of true positive, false positive, and false negative pixels respectively.

The detailed architecture of the proposed SCW-RNN used for evaluation is summarized in TableII. It consists of a fully connected layer, followed by 4 spatial sweeping layers that sweeps the input feature map in different directions, and two fully connected layers. The model is trained using RMSprop algorithm with the size of the none-overlapping local patches, the learning rate, the number of different modules M, and the size of each modules k set to $10 \times 10 \times 3$, 0.003, 4, and 48, respectively. The time period is set to exponential series: $T_i = 2^i$. The SCW-RNN architecture and all of the parameters are determined empirically.

We compare the performance of the proposed SCW-RNN with several other state-of-the-arts deep learning based methods, including: CNN-NMS [45], which is a famous 10-layer architecture used to segment neuronal membranes and uses a large input window (95×95) to capture the context information; and our previous work, CNNP [20], which is an end-to-end CNN architecture. To further demonstrate the capability of our proposed method to handle the spatial context information, we train a plain MLP network (MLP-10) that shares a similar architecture for comparison. We also try a larger window size (48×48) for MLP, denoted as MLP-48.

The Experimental results are summarized in Table III. In Table III, SCW-RNN(4) and SCW-RNN(8) denote the two different variants of the proposed method with 4 and 8 connected neighborhood, respectively. As we can see, our proposed method outperforms any other comparative methods. It suggests that it is necessary to use context information in perimysium annotation, as evidenced by the improved performance of SCW-RNN(8) over SCW-RNN(4). The CNN-NMS [45], which uses a large window size (95×95), provides a slightly worse P value and an equivalent R value compared to our methods, but it's testing time is almost 100 times slower than our methods. Similar conclusion could be drawn from the qualitative perimysium annotation result on a digitized skeletal muscle image, as shown in Figure 8. It can be observed that other methods suffer from over-segmentation of the perimysium, which is significantly reduced in the proposed method. The consistently better results is attributed to the ability of the proposed SCW-RNN to learn the global context information, which is the key to differentiate perimysium from other anatomically similar structures in ECM.

C. Automatic Diagnosis

Finally, we evaluate the performance of the automatic diagnosis of the proposed system. We analyse the results of skeletal muscle image search and retrieval, and classification both

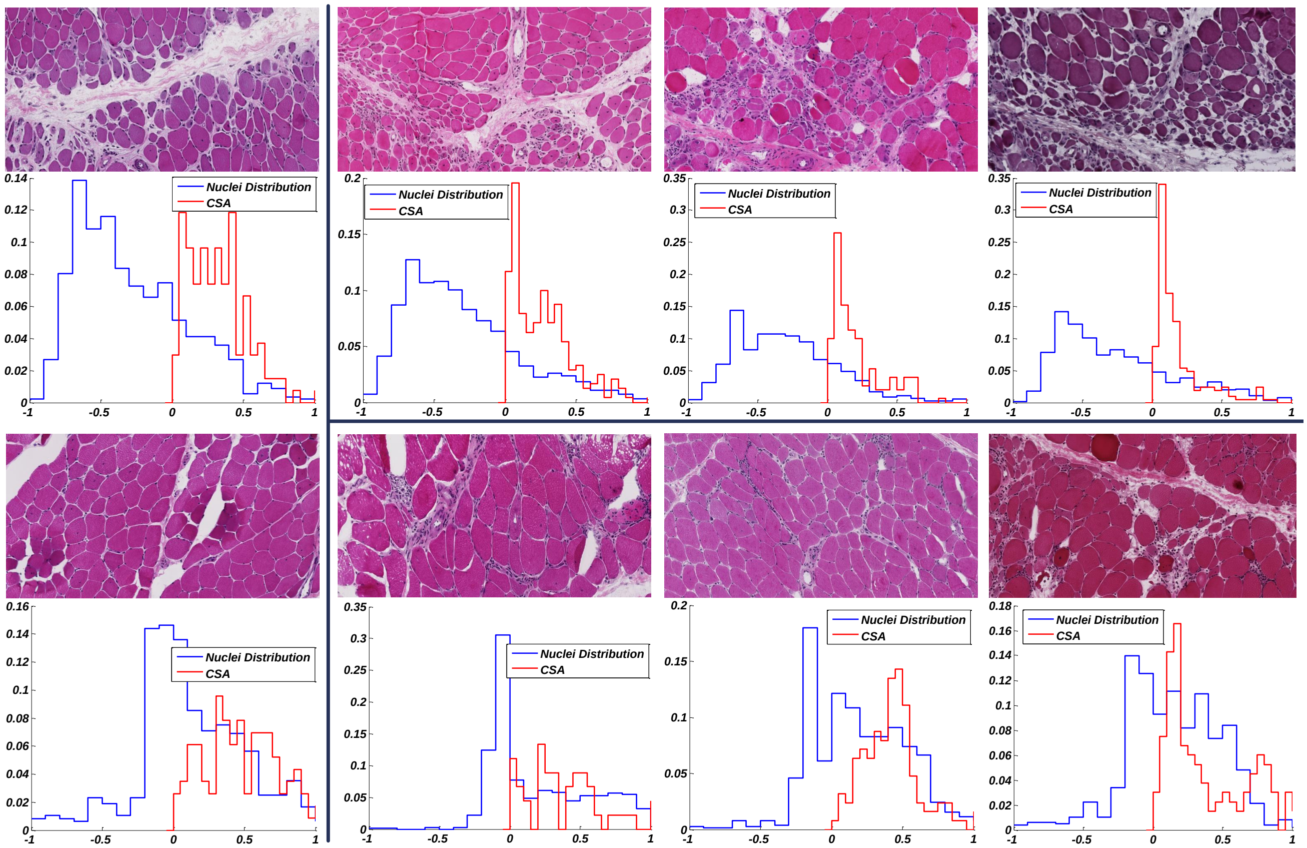
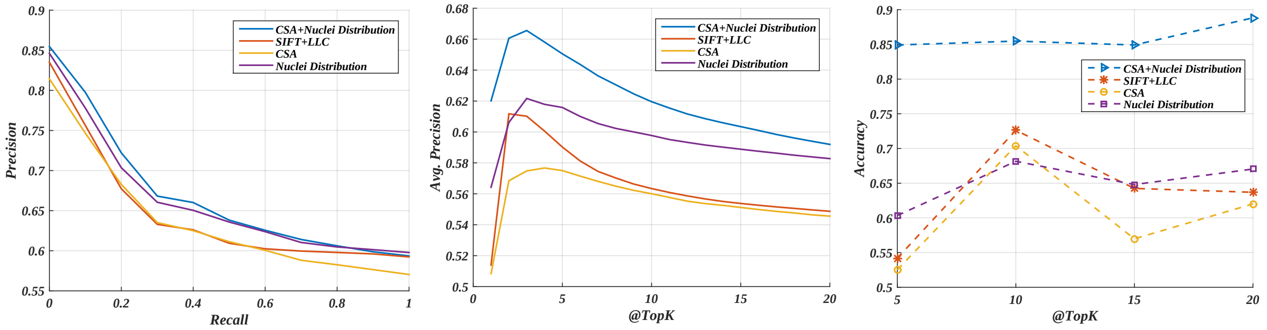


Fig. 9: *The step plot of the proposed histogram features (red plot is the CSA distribution histogram and blue plot is nuclei spatial distribution histogram) for a sample query image (**left**) and the top rank retrieved images in the database (**right**). Images in the **first row** exhibit the case of DM and the ones in the **second row** exhibit the case of PM. We can observe that the same disease does exhibit similar histogram features.*

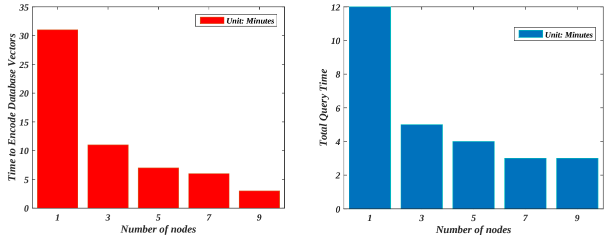


(a) (b) (c)

Fig. 10: *(a) The precision recall curve for PQ-based image search and retrieval. (b) Average precision@K for the first 20 top ranked results. (c) The classification accuracy computed with the majority voting on the rank retrieved images.*

qualitatively and quantitatively. For evaluation, we follow the common CBIR experimental setup [25], [41], and randomly sample 25% of the images (from all 718 images) for query and use the rest for database generation. All of the images are processed for cell segmentation, perimysium annotation, and nuclei detection to obtain the final histogram-based image representation with dimension $d = 400$, which is finally used for image rank retrieval and classification.

First, we perform qualitative evaluation of the designed image features using quantization-based similarity search. In the proposed PQ-based search and retrieval method, there are two important parameters: the number of subvectors and the number of cluster centers. In our implementation, we set them as 8 and 256, respectively, as suggested in [41], and we found that they work well for our case. Figure 9 shows the top 3 ranked and retrieved images given multiple query images. As one can see, the designed image features in our system produce satisfactory results evaluated based on visual similarity. We further examine the labels of the query images and the returned images to identify the disease category, and all of the returned images are in the same disease categories as the respective query images. To validate the discriminative property of the designed image features, we can take a closer look at the histogram plot of features as shown in Figure 9.



(a) (b)

Fig. 11: *The running time for encoding database vectors (a) and online query (b) using histogram based features in MapReduce. This time plot (b) represents 12000 queries in a database that contains* 600000 *images. We can see that each query takes approximately 60 milli-second using one node, and 15 mill-second using 9 nodes.*

We observe that the query image and retrieved images with the same labels show similar feature histogram, which is quite different compared to images with different labels.

Figure 10(a) shows the quantitative precision-recall (PR) curves using the proposed PQ search. We compare our histogram based feature with state-of-the-art dense scale-invariant feature transform (SIFT) with locality-constrained linear coding (LLC) (SIFT+LLC) [46]. For SIFT+LLC, we create a codebook of 512 representative features and encode all the images to approximately 21000-dimensional feature vectors using the default LLC parameter settings in [46]. We also plot the PR curves of individual features (i.e. CSA and nuclei distribution) for comparison. The PR curve is obtained by averaging the precision at standard recall levels over all the test images. Formally, the precision is defined as:

$$P = \frac{1}{N_q} \sum_{i=1}^{N_q} \frac{1}{K} \sum_{k=1}^{K} p(r_{ik}), \tag{17}$$

where P is the interpolated precision at standard recall levels r: $p(r) = \max_{\hat{r} \geq r} p(\hat{r})$, and the N_q and K are the number of query images and returned images, respectively. It can be observed that, in general, the combination of the CSA and nuclei distribution provides the best performance. Furthermore, we demonstrate the average precision at different rank threshold K (avg. precision@K), as shown in Figure 10(b). Avg. precision@K is computed as an average of the precision: $P = \frac{N_{TP}}{(N_{TP}+N_{FP})}$, where the number of true positives (N_{TP}) are the retrieved images with same labels as query images and the number of false positives (N_{FP}) are the retrieved images with labels different than the query images, for each rank position of relevant images in the top K results. Figure 10(b) shows similar PR curves, and illustrates that the combination of the CSA and nuclei distribution can be used as reliable features in image retrieval of similar cases of IIMs.

We then evaluate the classification performance of the proposed framework by applying a majority voting to the top retrieved images. Figure 10(c) displays the classification accuracy with respect to the top K retrieved images. We can observe that the results are consistent with the image retrieval, where the best classification accuracy is achieved with the combination of CSA and nuclei distribution. The accurate results of the retrieval and subsequent classification illustrate the effectiveness of the proposed feature representation, which captures clinically significant information in the image, and the searching algorithm, which can be used for decision support.

Finally, we evaluate the scalability of our distributed CBIR system in terms of running time. For this purpose, we create a much larger dataset consisting of approximately 600000 training image patches and 12000 testing image patches. All of the distributed experiments are conducted in Amazon AWS using the general purpose large size nodes. First, we analyze the running time to build the database by varying the number of computational nodes, as illustrated in Figure 11. As one can observe, the execution time decreases significantly as we increase the number of clusters from 1 to 9. In addition, we record the online query time using MapReduce with different number of nodes. We calculate the average of the time cost for all the testing query images. As we can see, the execution time for the online query decreases significantly as the cluster size increases from 1 to 7. However, there is not a large variation in running time when the node size is scaled up from 7. This might be explained mainly because of the communication overhead between the master node and the slaves.

VIII. Conclusion and Future Work

We have presented a complete framework of IIM diagnosis system with automated cell segmentation, perimysium annotation, and nuclei detection, the novel feature representations that aligns well with the clinically relevant image markers for IIM diagnosis, and the relevant image retrieval system to manage and study the muscle biopsy images that are collected over time period. Experimental results of the classification based on a majority voting provides a promising evidence that the proposed framework can be used as a second opinion in the diagnosis of different types of myopathies.

Currently, the performance of the proposed framework highly depends on the designed features and histogram based representation. In the future, we will focus on better feature representation. For instance, we plan to incorporate learning based features using deep architectures that could provide significant performance boost. Additionally, the proposed CBIR system is based on real-valued features, and thus it demands high storage spaces that will grow with the number of images in the database. This might limit the scalability of the system. In our future work, we will work on the hashing based binary representation, which provides much faster retrieval performance with lower storage requirement. Furthermore, we will work to quantitatively evaluate the interrater or annotator reliability using standard tests, such as kappa measures. We will also compare to the kappa statistics of human pathologists.

Acknowledgment

This research is funded, in part, by NIH R01 AR065479-02.

References

[1] M. C. Dalakas, "Muscle biopsy findings in inflammatory myopathies," *Rheumatic Disease Clinics of North America*, vol. 28, no. 4, pp. 779–798, 2002.

[2] K. N. Mustafa and S. S. Dahbour, "Clinical characteristics and outcomes of patients with idiopathic inflammatory myopathies from jordan 1996–2009," *Clinical rheumatology*, vol. 29, no. 12, pp. 1381–1385, 2010.

[3] S. Rayavarapu, W. Coley, T. B. Kinder, and K. Nagaraju, "Idiopathic inflammatory myopathies: pathogenic mechanisms of muscle weakness," *Skelet Muscle*, vol. 3, no. 1, p. 13, 2013.

[4] F. L. Mastaglia, M. J. Garlepp, B. A. Phillips, and P. J. Zilko, "Inflammatory myopathies: clinical, diagnostic and therapeutic aspects," *Muscle & nerve*, vol. 27, no. 4, pp. 407–425, 2003.

[5] A. A. Amato and R. C. Griggs, "Unicorns, dragons, polymyositis, and other mythological beasts," *Neurology*, vol. 61, no. 3, pp. 288–289, 2003.

[6] A. N. Basavanhally, S. Ganesan, S. Agner, J. P. Monaco, M. D. Feldman, J. E. Tomaszewski, G. Bhanot, and A. Madabhushi, "Computerized image-based detection and grading of lymphocytic infiltration in her2+ breast cancer histopathology," *Biomedical Engineering, IEEE Transactions on*, vol. 57, no. 3, pp. 642–653, 2010.

[7] L. Gorelick, O. Veksler, M. Gaed, J. A. Gómez, M. Moussa, G. Bauman, A. Fenster, and A. D. Ward, "Prostate histopathology: Learning tissue component histograms for cancer detection and classification," *Medical Imaging, IEEE Transactions on*, vol. 32, no. 10, pp. 1804–1818, 2013.

[8] H. Müller, N. Michoux, D. Bandon, and A. Geissbuhler, "A review of content-based image retrieval systems in medical applicationsclinical benefits and future directions," *International journal of medical informatics*, vol. 73, no. 1, pp. 1–23, 2004.

[9] R. K. Sayed, E. C. de Leonardis, J. A. Guerrero-Martínez, I. Rahim, D. M. Mokhtar, A. M. Saleh, K. E. Abdalla, M. J. Pozo, G. Escames, L. C. López *et al.*, "Identification of morphological markers of sarcopenia at early stage of aging in skeletal muscle of mice," *Experimental gerontology*, vol. 83, pp. 22–30, 2016.

[10] L. Yang, O. Tuzel, W. Chen, P. Meer, G. Salaru, L. A. Goodell, and D. J. Foran, "Pathminer: A web-based tool for computer-assisted diagnostics in pathology," *Information Technology in Biomedicine, IEEE Transactions on*, vol. 13, no. 3, pp. 291–299, 2009.

[11] J.-s. Guan, L.-Y. Lin, G.-l. Ji, C.-M. Lin, T.-L. Le, and I. J. Rudas, "Breast tumor computer-aided diagnosis using self-validating cerebellar model neural networks," *Acta Polytechnica Hungarica*, vol. 13, no. 4, 2016.

[12] J. Mula, J. D. Lee, F. Liu, L. Yang, and C. A. Peterson, "Automated image analysis of skeletal muscle fiber cross-sectional area," *Journal of Applied Physiology*, vol. 114, no. 1, pp. 148–155, 2013.

[13] X. Zhou, K.-Y. Liu, P. Bradley, N. Perrimon, and S. T. Wong, "Towards automated cellular image segmentation for rnai genome-wide screening," in *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2005*. Springer, 2005, pp. 885–892.

[14] F. Xing, H. Su, J. N., and L. Yang, "Automatic ki-67 counting using robust cell detection and online dictionary learning," *Biomedical Engineering, IEEE Transactions on*, vol. 61, no. 3, pp. 859–870, 2014.

[15] G. M. Faustino, M. Gattass, S. Rehen, and C. de Lucena, "Automatic embryonic stem cells detection and counting method in fluorescence microscopy images," in *Biomedical Imaging: From Nano to Macro, 2009. ISBI'09. IEEE International Symposium on*. IEEE, 2009, pp. 799–802.

[16] F. Liu, A. Mackey, R. Srikuea, K. Esser, and L. Yang, "Automated image segmentation of haematoxylin and eosin stained skeletal muscle cross-sections," *Journal of microscopy*, vol. 252, no. 3, pp. 275–285, 2013.

[17] C. Malm and J. Yu, "Exercise-induced muscle damage and inflammation: re-evaluation by proteomics," *Histochemistry and cell biology*, vol. 138, no. 1, pp. 89–99, 2012.

[18] Y. Xu, J. Zhang, I. Eric, C. Chang, M. Lai, and Z. Tu, "Context-constrained multiple instance learning for histopathology image segmentation," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2012, pp. 623–630.

[19] H. Xu, C. Lu, and M. Mandal, "An efficient technique for nuclei segmentation based on ellipse descriptor analysis and improved seed detection algorithm," *Biomedical and Health Informatics, IEEE Journal of*, vol. 18, no. 5, pp. 1729–1741, Sept 2014.

[20] M. Sapkota, F. Xing, H. Su, and L. Yang, "Automatic muscle perimysium annotation using deep convolutional neural network," in *2015 IEEE 12th International Symposium on Biomedical Imaging (ISBI)*. IEEE, 2015, pp. 205–208.

[21] Y. Xie, Z. Zhang, M. Sapkota, and L. Yang, "Spatial clockwork recurrent neural network for muscle perimysium segmentation," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2016, pp. 185–193.

[22] J. Koutnik, K. Greff, F. Gomez, and J. Schmidhuber, "A clockwork rnn," *arXiv preprint arXiv:1402.3511*, 2014.

[23] Y. Xie, F. Xing, X. Kong, H. Su, and L. Yang, "Beyond classification: structured regression for robust cell detection using convolutional neural network," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2015, pp. 358–365.

[24] K. Simonyan, A. Zisserman, and A. Criminisi, "Immediate structured visual search for medical images," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2011, pp. 288–296.

[25] X. Zhang, W. Liu, and S. Zhang, "Mining histopathological images via hashing-based scalable image retrieval," in *2014 IEEE 11th international symposium on biomedical imaging (ISBI)*. IEEE, 2014, pp. 1111–1114.

[26] L. Yang, X. Qi, F. Xing, T. Kurc, J. Saltz, and D. J. Foran, "Parallel content-based sub-image retrieval using hierarchical searching," *Bioinformatics*, vol. 30, no. 7, pp. 996–1002, 2013.

[27] B. Van Ginneken, A. F. Frangi, J. J. Staal, B. M. ter Haar Romeny, and M. A. Viergever, "Active shape model segmentation with optimal features," *IEEE transactions on medical imaging*, vol. 21, no. 8, pp. 924–933, 2002.

[28] P. Arbelaez, "Boundary extraction in natural images using ultrametric contour maps," in *Computer Vision and Pattern Recognition Workshop, 2006. CVPRW'06. Conference on*. IEEE, 2006, pp. 182–182.

[29] B. Julesz, "Textons, the elements of texture perception, and their interactions." *Nature*, vol. 290, no. 5802, pp. 91–97, 1981.

[30] C. Schmid, "Constructing models for content-based image retrieval," in *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, vol. 2. IEEE, 2001, pp. II–39.

[31] R. E. Schapire, "The boosting approach to machine learning: An overview," in *Nonlinear estimation and classification*. Springer, 2003, pp. 149–171.

[32] L. D. Cohen, "On active contour models and balloons," *CVGIP: Image understanding*, vol. 53, no. 2, pp. 211–218, 1991.

[33] X. Qi, F. Xing, D. J. Foran, and L. Yang, "Robust segmentation of overlapping cells in histopathology specimens using parallel seed detection and repulsive level set," *Biomedical Engineering, IEEE Transactions on*, vol. 59, no. 3, pp. 754–765, 2012.

[34] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *International journal of computer vision*, vol. 1, no. 4, pp. 321–331, 1988.

[35] F. Xing and L. Yang, "Robust nucleus/cell detection and segmentation in digital pathology and microscopy images: A comprehensive review," *IEEE Reviews in Biomedical Engineering*, vol. 9, pp. 234–263, 2016.

[36] P. J. Werbos, "Backpropagation through time: what it does and how to do it," *Proceedings of the IEEE*, vol. 78, no. 10, pp. 1550–1560, 1990.

[37] N. Otsu, "A threshold selection method from gray-level histograms," *Automatica*, vol. 11, no. 285-296, pp. 23–27, 1975.

[38] N. Chahin and A. G. Engel, "Correlation of muscle biopsy, clinical course, and outcome in pm and sporadic ibm," *Neurology*, vol. 70, no. 6, pp. 418–424, 2008.

[39] R. Mantegazza and P. Bernasconi, "Inflammatory myopathies: dermatomyositis, polymyositis and inclusion body myositis," 2000.

[40] G. Litjens, B. E. Bejnordi, N. Timofeeva, G. Swadi, I. Kovacs, C. Hulsbergen-van de Kaa, and J. van der Laak, "Automated detection of prostate cancer in digitized whole-slide images of h and e-stained biopsy specimens," in *SPIE Medical Imaging*. International Society for Optics and Photonics, 2015, pp. 94 200B–94 200B.

[41] H. Jegou, M. Douze, and C. Schmid, "Product quantization for nearest neighbor search," *IEEE transactions on pattern analysis and machine intelligence*, vol. 33, no. 1, pp. 117–128, 2011.

[42] J. Dean and S. Ghemawat, "Mapreduce: simplified data processing on large clusters," *the Communications of the ACM*, vol. 51, no. 1, pp. 107–113, 2008.

[43] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 24, no. 5, pp. 603–619, 2002.

[44] G. Mori, "Guiding model search using segmentation," in *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, vol. 2. IEEE, 2005, pp. 1417–1423.

[45] D. Ciresan, A. Giusti, L. M. Gambardella, and J. Schmidhuber, "Deep neural networks segment neuronal membranes in electron microscopy images," in *Advances in neural information processing systems*, 2012, pp. 2843–2851.

[46] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong, "Locality-constrained linear coding for image classification," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE, 2010, pp. 3360–3367.