Skeletal Muscle Cell Segmentation Using Distributed Convolutional Neural Network

Manish Sapkota[1,2], Fuyong Xing[1,2], Fujun Liu[1,2], Lin Yang[1,2]

[1]Dept. of Electrical and Computer Engineering, University of Florida
[2]J. Crayton Pruitt Family Dept. of Biomedical Engineering, University of Florida

Abstract. Morphological characteristics of muscle cells, such as cross sectional areas (CSAs), are critical factors to determine the muscle health. Automatic muscle fiber segmentation is often the first prerequisite. However, it is challenging to achieve effective and efficient skeletal muscle cell segmentation on Hematoxylin and Eosin (H&E) stained muscle images due to the complex nature of histopathology imaging and a large number of cells on a single image. In this paper, we propose to formulate the cell segmentation as a pixel-wise classification problem and train a deep convolutional neural network (CNN) to segment out the cell boundaries. Considering the speed, we apply the CNN model training to multiple graphical processing units (GPUs), and implement the distributed testing on the Spark parallel computing platform. To further improve running time cost, we apply a fast scanning technique to the pixel-wise classification with the learned CNN model. We have presented the segmentation results on a set of 120 H& E stained muscle images using the trained CNN model, and evaluated the proposed framework with accuracy calculation and speed performance.

1 Introduction

Skeletal muscle is one of the major tissues in human body, accounting for about 40% of body mass [12]. Muscle research community has agreed that cross sectional areas (CSAs) play an important role in determining the health and functionality of the muscle, and attempted to accurately compute the CSAs for further analysis. However, manual assessment of the CSAs is labour-intensive and suffers from inter-observer variations. Computer-aided image analysis can significantly improve the objectivity, and automatic muscle cell segmentation usually is the first prerequisite for computer-aided CSA calculation. Due to the complex nature of muscle images, there exist several major challenges for automated muscle cell segmentation (see Figure 1): 1) almost all muscle cells touch with one another and exhibit large scale variations; 2) the intensities of cell boundaries vary significantly; 3) freeze artifacts introduced during sample preparation often create false edges inside the muscle cells [13].

Automatic muscle cell segmentation is achieved by labeling the cell boundaries on digitized skeletal muscle specimens, as shown in right column of Figure 1. It has attracted a great deal of research interests in the medical image

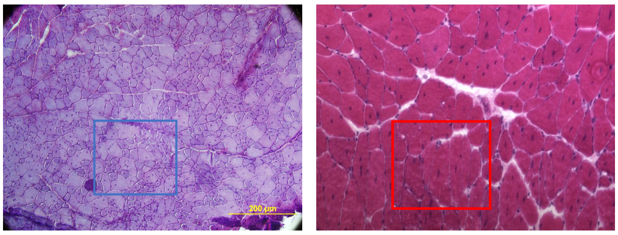

Fig. 1: The challenges for muscle cell segmentation. **Left:** Touching muscle cells with scale variations. **Right:** Muscle cells with weak boundaries.

analysis community. Sertel *et al.* [21] have applied ridge detection to enhance cell boundaries and then employed morphological operations for postprocessing, but it heavily relies on the accuracy of ridge detection. In [7], a concave point based contour splitting algorithm is exploited to decompose muscle cell clumps which exhibit weak boundaries. However, local minimum along the contours often create false concave points, and therefore the subsequent contour decomposition may not be reliable. Recently, dynamic programming is employed to obtain muscle cell segmentation in [13]. It begins with generating a set of segmentation candidates and then selects a subset of region candidates using an Integer Linear Programming scheme. Some other supervised learning techniques have successfully applied to cell segmentation and can be extended to muscle image segmentation. Kong *et al.* [8] have achieved nucleus segmentation by performing pixel classification in histopathology images, and a two-step supervised learning-based classification is reported in [14] for automated cell segmentation.

Recently there is an encouraging evidence that learned representation of biomedical images might perform better than the handcrafted features [24,16,20], and this has boosted the usage of deep learning techniques. Cruz-Roa *et al.* [4] have proposed a deep neural network for automated basal cell carcinoma cancer detection, and a unified deep representation learning model is reported [11] for automatic prostate magnetic resonance image segmentation. A deep convolutional neural network [3] has been successfully applied to mitosis detection in breast cancer histopathology images, and a similar neural network [2] has been applied to membrane segmentation in electron microscopy images. However, none of these methods deal with digitized muscle specimens, which are significantly different from other types of histopathology images. Since these exist weak cell boundaries and freeze artifacts, it is very challenging to achieve automatic accurate muscle cell segmentation.

Due to the increasing of medical data, many computer-aided image analyses including cell segmentation have been applied to high-performance computing machines. Foran *et al.* have successfully exploited a grid technology called CaGrid to tissue microarray image analysis and achieve significant speed improvement. Qi *et al.* [19] have applied to multiple GPUs to cell detection to reduce running time. In addition, the robustness analysis of medical images [17] has been carried

Table 1: The structure of the CNN used in our algorithm.

Layer No.	Layer Type	Feature Map	Kernel Size
1	Input	51 × 51 ×3	-
2	Convolutional	48 × 48 × 20	4 × 4
3	Max-pooling	24 × 24 × 20	2 × 2
4	Convolutional	22 × 22 × 20	3 × 3
5	Max-pooling	11 × 11 × 20	2 × 2
6	Convolutional	10 × 10 × 20	2 × 2
7	Max-pooling	5 × 5 × 20	2 × 2
8	Fully-connected	500 × 1	-
9	Fully-connected	250 × 1	-
10	Output	2 × 1	-

out using the CometCloud parallel computing platform, and a similar framework is presented in [18] for histopathology image retrieval.

In this paper, we present an automated cell segmentation approach on skeletal muscle images, which is based on a deep convolutional neural network (CNN). The problem is formulated into a pixel-wise classification framework, where a CNN model is trained with raw RGB values of image data and automatically learns a set of hierarchical features for classification. In the testing stage, the learned CNN model will be applied to the images in a sliding window, differentiating pixels in the cell boundaries from other regions (inside cells) to achieve automatic segmentation. We have presented the automatic cell segmentation results based on the CNN model. To improve the running time, we will perform the model training and testing in a distributed framework using multiple machines. This approach will provide efficient and effective muscle cell segmentation results, which can serve as a basis for further image analysis, such as CSA compuation, of skeletal muscle disease.

2 Cell Segmentation Using Deep Convolutional Neural Network

2.1 CNN Architecture

Convolutional neural network (CNN) is a feed-forward network alternatively composed of convolution and max-pooling layers, followed by several fully connected layers [9]. It can provide progressively abstract representation of the input with the increment of the number of layers. The convolutional layer calculates a set of output feature maps by applying multiple kernels (filters) to the input image or feature map, and the activation function is usually chosen as rectified linear units (ReLUs) [15], which can enable fast model training and potentially improves the classification performance.

Max pooling layer is used to preform dimension reduction, and also introduces local shift and translation invariance. It corresponds to a kernel of a certain size with or without overlapping. Usually max-pooling operation is performed separately for each input feature map. An alternative option is the average pooling, which can also reduce feature dimension but is not as robust as the max pooling to local transformation.

Fully-connected layer consists of ReLUs aiming to learn global feature representation. Each unit in the fully-connected layers connects to all the units in the previous layer. The last (Output) layer is a fully-connected layer with a softmax function, which corresponds to two units and is used for final binary classification. The details of the neural network layers are summarized in Table 1.

2.2 CNN Model Training and Testing

In our implementation, the CNN model is trained with a set of small image patches with size $s \times s \times 3$, where $s = 51$ and 3 means the RGB channels. A patch is labeled as positive if its center pixel locates on muscle cell boundaries, otherwise negative. In order to accelerate the training procedure, we train the CNN model by using a parameter server framework based on multiple GPUs [10].

In the testing stage, automatic annotation is achieved by applying the CNN model to new images for pixel-wise classification. The image patches partially outside the image boundaries are ignored. The last layer outputs the probabilities that each pixel is located in the cell boundaries or not, from which we determine the patch labels by choosing the category associated with the higher probability. It is straightforward to apply the model to pixel-wise classification with the sliding window technique [23], but it is computationally expensive on large-size images. Actually there is significant redundancy in the computation of the convolutional layers and the max-pooling layers when the sliding window is used for patch classification. The repeated convolutional computations and max-pooling operations reside in the overlapping region of one sliding window and its successive windows. We employ the fast scanning algorithm [5] to avoid redundancy by directly processing the entire image such that the running time can significantly decrease. The redundancy in one convolutional layer can be removed by applying the convolutional kernel (filter) to the entire image or the output of previous layers, and repeated computation in max-pooling layers is eliminated by a rearrangement strategy [5].

2.3 Spark Implementation

In order to improve the running time for large-scale images, we have implemented the testing based on the Spark framework [22]. Given a large-scale image, we partition the image into several smaller overlapping regions with the parallel scheme in Spark, and use the **map** and **collect** functions for the distributed operations on multiple machines. In each machine, fast scanning is applied to

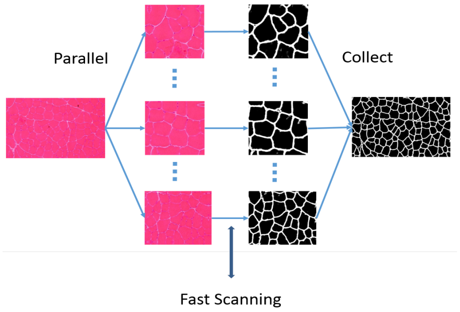

Fig. 2: Illustration of distributed testing for muscle cell segmentation.

the testing. After the computation, we collect the results from multiple nodes and merge the partitions into a single image with the same size as the input image (see Figure 2).

3 Experiments

3.1 Data Collection

We collect the skeletal muscle images from our collaborators at University of Kentucky. The dataset consists of images with both normal and disease muscle cells, and three subtypes of muscle diseases are dermatomyositis (DM), inclusion-body myositis (IBM), and polymyositis (PM). 120 images are uniformly cropped from over 10 whole-slide scan muscle specimens, which are captured at 20× magnification. Each type of muscle images has approximately the same data size. Since the CNN model is trained using small patches sampling from those images, in total we will have millions of image patches for training and testing.

3.2 Evaluation of Time Cost

We evaluate the running time for both training and testing. We train several CNN models based on multiple core CPUs, one GPU, and two GPUs, and the training time with approximately 2 million samples for 5 epochs is shown in the left panel of Figure 3. As we can see, compared with CPU, GPU can provide much faster training. Using two GPUs is slightly slower than the single GPU (TK40C) is due to the communication between two GPUs, which accounts for high time cost.

We perform the distributed testing on AWS. The running time in testing highly depends on the the number of image partitions, as shown in Figure 3. For a testing image, the smaller image patches are, the more segments we will obtain. Therefore, if the image segments are not partitioned into multiple groups, we do not make full use of the distributed machine sources such that the running time with a small partition number is relatively high (middle panel of Figure

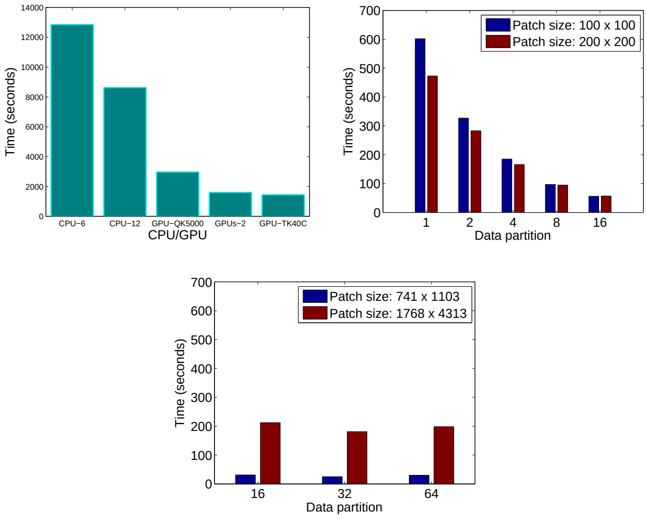

Fig. 3: **Left:** The running time of training. CPU-6 and CPU-12 corresponds to 6 and 12 CPU cores, respectively. GPU-2 represents the case that 2 GPUs are used for training. **Middle** and **Right:** The running time on AWS with respective to the number of partitions on medium-size and large-size images, respectively.

3, here we use 8 nodes). The more partitions we have, the low time cost is. In the right panel of Figure 3, we show the comparative running time between two large images. As we can see, large images needs more time to process regardless of the number of the partitions.

3.3 Evaluation of Model

We perform both qualitative and quantitative analysis on the proposed model. Figure 4 shows the cell segmentation results on eight sample muscle images. As one can tell, the CNN-based algorithm can produce very impressive performance, and it can provide promising results on those weak cell boundaries. To quantitatively analyze the pixel-wise segmentation accuracy, we calculate the precision P, recall R, and F_1-score as follows

$$P = \frac{|S \cap G|}{|S|},\ R = \frac{|S \cap G|}{|G|},\ F_1 = \frac{2 * P * R}{P + R}, \qquad (1)$$

where S denotes the segmentation result and G is the ground truth. We compare the proposed CNN model with two state of the arts: Isoperimetric graph partition (ISO) [6], which produces high quality segmentations as a spectral method with improved speed and stability, and global probability of boundary detector (gPb) [1], which is widely adopted to segmenting natural images. Table 2 lists the

Table 2: Comparison with the state of the arts

Method	Precision	Recall	F_1-score
ISO [6]	0.90 ± 0.06	0.74 ± 0.14	0.81 ± 0.10
gPb [1]	0.91 ± 0.05	0.70 ± 0.10	0.79 ± 0.08
DCNN	0.95 ± 0.04	0.77 ± 0.14	0.84 ± 0.11

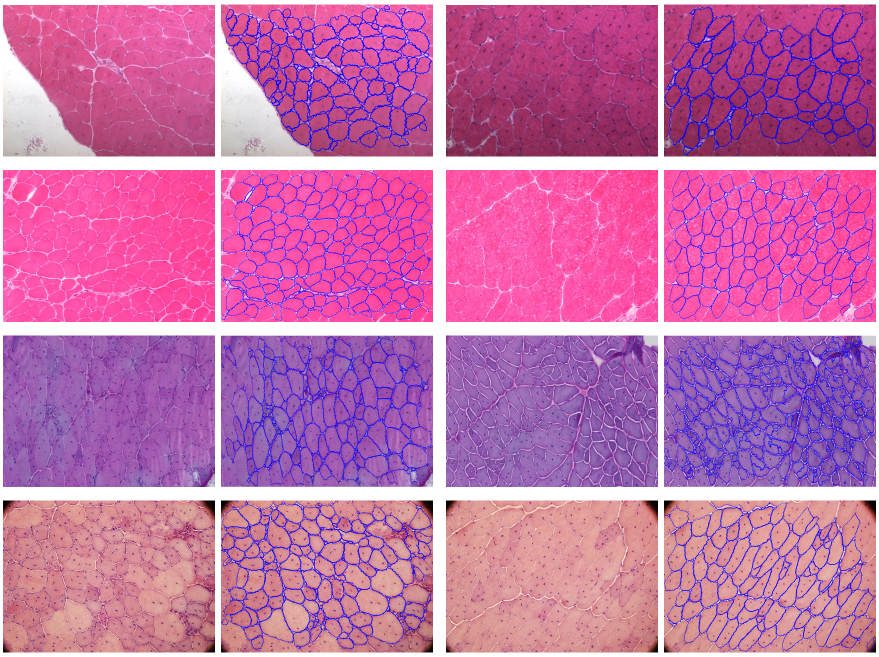

Fig. 4: Cell segmentation results on 8 muscle images using the proposed CNN model. Columns 1 and 3 denote the original images, and columns 2 and 4 represent the corresponding desired cell segmentation results. Note that cell touching image boundaries are ignored.

comparative performance of the three methods, which demonstrates that the proposed DCNN model produce the best segmentation results.

4 Conclusion

In this report, we have presented a deep convolutional neural network (CNN) model for automatic cell segmentation on skeletal muscle images. We achieve cell segmentation by detecting the cell edges, which is formulated into a pixel-wise classification problem. In order to improve the training time, we apply multiple GPUs to the CNN model training. In addition, we have implemented the distributed testing on the Spark framework for high performance computing. The experiments demonstrate the effectiveness and efficiency of the proposed framework in terms of the segmentation accuracy and running time.

References

1. Arbelaez, P., Maire, M., Fowlkes, C., Malik, J.: Contour detection and hierarchical image segmentation. TPAMI 33(5), 898–916 (2011)
2. Ciresan, D.C., Gambardella, L.M., Giusti, A., Schmidhuber, J.: Deep neural networks segment neuronal membranes in electron microscopy images. In: NIPS. pp. 2852–2860 (2012)
3. Cireşan, D.C., Giusti, A., Gambardella, L.M., Schmidhuber, J.: Mitosis detection in breast cancer histology images with deep neural networks. In: MICCAI. pp. 411–418 (2013)
4. Cruz-Roa, A.A., Ovalle, J.E.A., Madabhushi, A., Osorio, F.A.G.: A deep learning architecture for image representation, visual interpretability and automated basal-cell carcinoma cancer detection. In: MICCAI. pp. 403–410 (2013)
5. Giusti, A., Cireşan, D.C., Masci, J., Gambardella, L.M., Schmidhuber, J.: Fast image scanning with deep max-pooling convolutional neural networks. arXiv preprint arXiv:1302.1700 (2013)
6. Grady, L., Schwartz, E.: Isoperimetric graph partitioning for image segmentation. TPAMI 28(3), 469–475 (2006)
7. Janssens, T., Antanas, L., Derde, S., Vanhorebeek, I., Van den Berghe, G., Güiza Grandas, F.: Charisma: An integrated approach to automatic h&e-stained skeletal muscle cell segmentation using supervised learning and novel robust clump splitting. Medical image analysis 17(8), 1206–1219 (2013)
8. Kong, H., Gurcan, M., Belkacem-Boussaid, K.: Partitioning histopathological images: an integrated framework for supervised color-texture segmentation and cell splitting. IEEE Trans. Med. Imaging (TMI) 30(9), 1661–1677 (2011)
9. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: NIPS. pp. 1097–1105 (2012)
10. Li, M., Andersen, D.G., Park, J.W., Smola, A.J., Ahmed, A.: Scaling distributed machine learning with the parameter server. In: OSDI. pp. 583–598 (2014)
11. Liao, S., Gao, Y., Oto, A., Shen, D.: Representation learning: A unified deep learning framework for automatic prostate mr segmentation. In: MICCAI. vol. 8150, pp. 254–261 (2013)
12. Liu, F., Mackey, A.L., Srikuea, R., Esser, K.A., Yang, L.: Automated image segmentation of haematoxylin and eosin stained skeletal muscle cross-sections. Journal of Microscopy 252(3), 275–285 (2013)
13. Liu, F., Xing, F., Yang, L.: Robust muscle cell segmentation using region selection with dynamic programming. In: Biomedical Imaging (ISBI), 2014 IEEE 11th International Symposium on. pp. 521–524 (2014)
14. Mao, K.Z., Zhao, P., Tan, P.H.: Supervised learning-based cell image segmentation for p53 immunohistochemistry. IEEE Trans. Biomed. Eng. (TBME) 53(6), 1153–1163 (2006)
15. Nair, V., Hinton, G.: Rectied linear units improve restricted boltzmann machines. In: ICML. pp. 807–814 (2010)

16. Prasoon, A., Petersen, K., Igel, C., Lauze, F., Dam, E., Nielsen, M.: Deep feature learning for knee cartilage segmentation using a triplanar convolutional neural network. In: MICCAI. vol. 8150, pp. 246–253 (2013)
17. Qi, X., Kim, H., Xing, F., Foran, D.J., Yang, L.: The analysis of image feature robustness using cometcloud. Journal of Pathology Informatics 3(33), 1–13 (2012)
18. Qi, X., Wang, D., Rodero, I., Diaz-Montes, J., Gensure, R., Xing, F., Zhong, H., Goodell, L., Parashar, M., Foran, D., Yang, L.: Content-based histopathology image retrieval using cometcloud. BMC Bioinformatics 15(1), 287 (2014)
19. Qi, X., Xing, F., Foran, D.J., Yang, L.: Gpu enabled parallel touching cell segmentation using mean shift based seed detection and repulsive level set. In: High Performance Computing (HP) workshop associated with MICCAI (2010)
20. Roth, H.R., Lu, L., Seff, A., Cherry, K.M., Hoffman, J., Wang, S., Liu, J., Turkbey, E., Summers, R.M.: A new 2.5 d representation for lymph node detection using random sets of deep convolutional neural network observations. In: MICCAI. pp. 520–527 (2014)
21. Sertel, O., Dogdas, B., Chiu, C.S., Gurcan, M.N.: Microscopic image analysis for quantitative characterization of muscle fiber type composition. Computerized Medical Imaging and Graphics 35(7), 616–628 (2011)
22. Spark, A.: `https://spark.apache.org/` (2015), [Online]
23. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In: CVPR. vol. 1, pp. I–511–I–518 vol.1 (2001)
24. Wu, G., Kim, M., Wang, Q., Gao, Y., Liao, S., Shen, D.: Unsupervised deep feature learning for deformable registration of mr brain images. In: MICCAI. vol. 8150, pp. 649–656 (2013)