CHAPTER 4

Machine learning and its application in microscopic image analysis

F. Xing, L. Yang
University of Florida, Gainesville, FL, United States

CHAPTER OUTLINE

4.1 INTRODUCTION

Microscopic image analysis in pathology is very important for image-based computer-aided diagnosis (CAD), which might provide potential support for early detection of diseases. Due to the advance in microscopic imaging techniques, a huge number of microscopic images are generated every day (Sommer and Gerlich, 2013), which prohibit data manual assessment. Therefore automated microscopic image analysis is in urgent need of digitized specimen evaluation. In addition, computerized approaches can eliminate the inter-observer variations and thus significantly improve the objectivity and reproducibility of the assessment (Foran et al., 2011). This will allow comparative study of diseases, and can potentially support decision making in diagnosis.

Machine learning is a subfield of computer science that aims to develop a set of algorithms to detect patterns within existing data and then uses these uncovered patterns to make predictions on new data (Murphy, 2012; Hastie et al., 2009; Bishop, 2007). Compared with nonlearning-based methods that are usually specifically designed for certain applications and not adaptive to other scenarios (eg, different biology assays), machine learning does not require manual software adaption, and more importantly, it can effectively handle high-dimensional data that are not easily

Machine Learning and Medical Imaging. http://dx.doi.org/10.1016/B978-0-12-804076-8.00004-9

modeled with a few parameters (Sommer and Gerlich, 2013). Machine learning has attracted a great deal of interest in the fields of computer vision, image processing, data mining, medical imaging, computational biology, etc. Specifically, there exist a large number of reports applying various machine learning techniques to automated data analysis in medicine, biomedical images, and digitized pathology specimens (Waljee and Higgins, 2010; Wernick et al., 2010; Tarca et al., 2007; Kourou et al., 2015), which can provide strong support for CAD. In this chapter, we will not attempt to comprehensively review the literature of machine learning-based CAD, but focus on machine learning applications in microscopic image analysis by using recent publications in our own research instead. We will introduce several popular machine learning techniques in digital pathology image analysis and their applications in nucleus/cell detection and segmentation on microscopic images.

4.2 DETECTION

Accurate nucleus/cell detection in microscopic images is a fundamental step for many subsequent computer-aided biomedical image analyses, such as nucleus/cell segmentation, counting, tracking, classification, etc. However, robust object localization, especially in those images exhibiting dense clusters and large variations in object scales, remains a challenging task (Quelhas et al., 2010). In the past few years, a number of methods including spatial filters (Al-Kofahi et al., 2010), kernel-based voting (Parvin et al., 2007; Qi et al., 2012; Xing et al., 2014), and graph partition (Bernardis and Yu, 2010; Zhang et al., 2014) have been proposed to automatically detect nuclei/cells in microscopic images. However, these approaches usually require careful parameter tuning to achieve desired performance on different types of images. This procedure of manual adjustment is tedious and limits the usage of those algorithms. By contrast, machine learning, especially supervised learning, learns processing rules from given image data instead of relying on manual parameter selection (Sommer and Gerlich, 2013). These learning methods have received a large amount of attention in microscopic image analysis (Arteta et al., 2012; Mualla et al., 2013; Cireşan et al., 2013). Currently the nucleus/cell detection is usually formulated as a pixel or superpixel-wise (region-wise) classification problem, and a specific model is learned to map data examples into discrete labels. There exist many classifiers with various feature representations reported in the literature, and here we mainly focus on two recently popular classifiers, support vector machine (SVM) and deep convolutional neural network (CNN), by explaining their applications in fluorescence, bright-field, and phase-contrast microscopy images.

4.2.1 SUPPORT VECTOR MACHINE

The SVM is a nonprobabilistic binary classifier, aiming to find a hyperplane with a maximal margin to separate high-dimensional data points (Cortes and Vapnik, 1995).

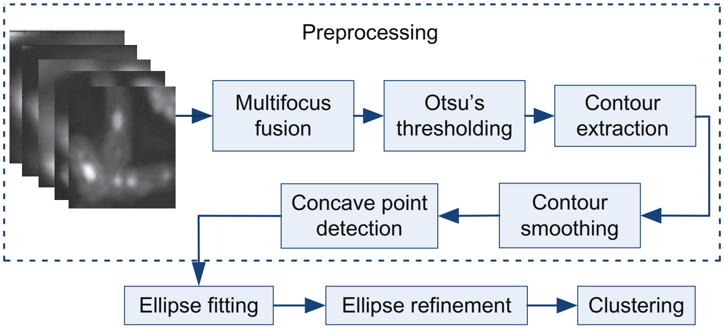

FIG. 4.1

An overview of the proposed myonucleus detection in isolated single muscle fiber fluorescence images.

Given a set of training data $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^N$, where $\boldsymbol{x}_i \in R^p$ and $y_i \in \{1, -1\}$, SVM solves the following problem:

$$\min_{\boldsymbol{w},b,\boldsymbol{\xi}} \frac{1}{2}\boldsymbol{w}^{\mathrm{T}}\boldsymbol{w} + C\sum_{i=1}^{N}\xi_i, \text{ s.t. } y_i(\boldsymbol{w}^{\mathrm{T}}\phi(\boldsymbol{x}_i) + b) \geq 1 - \xi_i, \quad \xi_i \geq 0, \ \forall i, \tag{4.1}$$

where the $\phi(\boldsymbol{x})$ maps the data points into a high-dimensional space. $C > 0$ is a penalty parameter to control the violation, which is represented by the slack variables $\boldsymbol{\xi} = [\xi_1, \cdots, \xi_N]^{\mathrm{T}}$. By using the kernel trick (Scholkopf and Smola, 2001), SVM can produce nonlinear decision boundaries for robust classification.

We have applied a binary SVM classifier to automated myonucleus detection in isolated single muscle fiber fluorescence images (Su et al., 2014, 2013). The framework is shown in Fig. 4.1, which mainly consists of four steps: image preprocessing, robust ellipse fitting, SVM-based ellipse refinement, and mean-shift clustering based on geodesic inner distance. It first fuses a set of z-stack images with a normalized linear combination and extracts foreground edges by applying Otsu's method (Otsu, 1979) to the fused image, then fits a sufficient number of ellipse hypotheses using heteroscedastic errors-in-variable (HEIV) regression (Matei and Meer, 2000), next uses an SVM classifier with a set of sophisticated design features to select the good candidate fittings, and finally applies inner geodesic distance-based clustering to ellipse localization, which determines the locations of myonuclear centers.

4.2.1.1 Image preprocessing

In z-stack imaging, the microscope collects lights from both in-focus and out-of-focus planes. Therefore each image consists of in-focus objects as well as out-of-focus objects. Motivated by Hariharan et al. (2007), we have proposed a modified image fusion algorithm to compose a smooth multifocus image from the z-stack images. Let $i = 1, 2, \ldots, M$ denote the z-stack images. The gradient magnitude of each image is calculated as

$$Z_i(x, y) = \sqrt{I^2_{X(i)}(x, y) + I^2_{Y(i)}(x, y)}, \tag{4.2}$$

where $I^2_{X(i)}(x, y)$ and $I^2_{Y(i)}(x, y)$ are the horizontal and vertical gradients of the ith image, respectively. Different from Hariharan et al. (2007), which picks the pixel $I_i(x, y) = \text{argmax}_i Z_i(x, y)$ as the synthesized image intensity $I_s(x, y)$, we linearly combine all the pixel values in the z-stack to calculate $I_i(x, y)$:

$$g_i(x, y) = \frac{Z_i(x, y)^k}{\sum_{i=1}^{N} Z_i(x, y)^k}, \quad k = 3, \tag{4.3}$$

$$I_s(x, y) = \sum_{i=1}^{M} g_i(x, y) I_i(x, y). \tag{4.4}$$

Our image fusion algorithm can provide the final composed image with smooth boundaries, which greatly increase the robustness of the subsequence automatic detection procedure. We apply Otsu's method (Otsu, 1979) to the fused image for binarization and then contour extraction of myonuclei or myonucleus clumps. Thereafter we use an elliptical Fourier descriptor (Kuhl and Giardina, 1982) to smooth the contours by keeping the first 20 Fourier coefficients, and then exploit a concavity detection algorithm (Yang et al., 2008) to detect concave points, which will be used for subsequent ellipse fitting.

4.2.1.2 Robust ellipse fitting

Based on the detected concave points, we adopt the robust HEIV regression to generate fitting ellipses. Compared with some other traditional methods, HEIV has a weaker dependence on initialization and a faster convergence. The direct least-squares (DLS) method (Fitzgibbon et al., 1999) is biased when the input data points are a short low-curvature segment of a whole ellipse, the geometric distance minimization algorithm (Zhang, 1997) is sensitive to the initialization, and Taubin's method does not provide robust fitting either when only one short curve segment is available (Taubin, 1991). In our approach, an ellipse is modeled using an errors-in-variables model with Gaussian-distributed noise, which can be iteratively solved by reformulating it as a generalized eigenvalue problem.

In order to segment the touching myonuclei, robust ellipse fitting based on the HEIV regression model is performed on the contour pixels. Suppose that there are n concave points for a contour; the original object contour is intersected by these concave points into n segments. We fit candidate ellipses using the different combinations of the contour segments. Thus there are $\binom{n}{k}$ segment combinations if we choose k segments to fit one ellipse. We empirically set $k = \{1, 2, 3\}$ to guarantee that a sufficient number of ellipses using the HEIV regression are generated as input during the subsequent refinement procedure.

4.2.1.3 SVM-based ellipse refinement

Given the candidate ellipses calculated from Section 4.2.1.2, we propose to refine these ellipses and remove the false fittings using a supervised learning technique, an SVM classifier. In order to conduct SVM training and testing, we specifically design several groups of features for ellipse representation. In total we have extracted 42 features for each ellipse hypothesis, and they are summarized as follows.

Group 1

The most intuitive geometric feature to evaluate the ellipse fitting is to measure whether or not the fitting ellipses match with the boundaries of myonuclei. A set of morphological ratios is defined for this type of measurements. Given the fitting ellipse e and the myonucleus contour c, two overlapping ratios are calculated: $r_{aoc} = \frac{A_o}{A_c}$ and $r_{aoe} = \frac{A_o}{A_e}$, where A_c and A_e represent the region areas inside e and c, respectively, and A_o denotes the area of the overlapping region inside both e and c. In addition, the ratio $r_{oace} = \frac{A_o}{A_c+A_e}$ is also calculated.

Considering robustness, the pixel-wise overlapping ratios are measured as well. Let p_o denote the number of overlapping pixels between the fitting ellipse and the contour, p_c represent the number of contour pixels, and $r_{oc} = \frac{p_o}{p_c}$ be the contour pixel-level overlapping ratio. Based on r_{oc}, we design an iterative procedure to assign a match-quality score to each fitting ellipse. In each iteration, we count the contour pixels overlapping with an ellipse, and then sort the ellipses with respect to the number of their overlapping contour pixels. Within each iteration, the contour pixels that overlap with the highest ranked ellipse are removed, and r_{oc} is updated for each remaining ellipse. This iteration terminates when there is a small fraction (decided by a threshold) of contour pixels left. Note that, in each iteration, the ellipse that has the most overlapping pixels with the object contour will be assigned the highest rank. The match-quality of an ellipse is defined as the rank of this fitting ellipse calculated during each iteration. To improve efficiency, we only record the ranks of an ellipse in the first three iterations.

Group 2

The myonuclei are objects with certain biologically meaningful areas. Therefore the ellipse area A_e, the axis ratio (long-to-short) r_{axis}, and the perimeter p_c and the area A_c of the myonucleus contour are also considered as potential geometric features for classification.

Group 3

Concave point depth is a feature designed to measure the distance between concave points and fitting ellipses. This feature design is based on the observation that an accurate ellipse fitting should not have a concave point deeply inside the ellipse. For an ellipse e and a set of concave points q_j $(j = 1, \ldots, M)$, the concave point depth d is defined as the sum of squares of the Euclidean distances from the concave points to ellipse e:

$$d = \sum_{j} dist^2(q_j, e),\ j \in \{j:\ q_j \text{ is in ellipse } e\}, \tag{4.5}$$

where $dist(\cdot)$ denotes the Euclidean distance from q_j to e.

Because the center of an accurate fitting ellipse should not be located near the myonucleus boundary, the distance between the ellipse center and the myonucleus boundary d_{ecc} is also calculated. This feature can help to remove suboptimal fitting ellipses whose centers are close to the object boundaries.

Group 4

The irregularity of the boundaries is defined as $r_{irg} = \frac{n_c}{p_c}$, where p_c denotes the perimeter of a myonucleus contour and n_c represents the number of concave points detected.

Group 5

A set of statistical features is computed to capture the relationships among the ellipses generated from the same contour. Assuming N ellipse fitting candidates are generated from the segments of an object contour, and $f_i, i = (1, 2, \ldots, N)$ represents one specific feature calculated for the ith ellipse e_i, we can generate the following statistics:

$$f_{1i} = f_i - \frac{1}{N}\sum_{i=1}^{N} f_i, \tag{4.6}$$

$$f_{2i} = f_i - \mathbf{median}(f_1, f_2, \ldots, f_N), \tag{4.7}$$

$$f_{3i} = f_i - \mathbf{max}(f_1, f_2, \ldots, f_N), \tag{4.8}$$

$$f_{4i} = f_i - \mathbf{min}(f_1, f_2, \ldots, f_N), \tag{4.9}$$

where the functions $\mathbf{median}(\cdot)$, $\mathbf{max}(\cdot)$, and $\mathbf{min}(\cdot)$ compute the median, maximum, and minimum value of the input feature vector, respectively. These statistical features capture the inter-group variance between one fitting ellipse and the whole group of fitting ellipses for the same object contour. These statistical features are calculated for each feature only in Groups 1 and 3.

Group 6

The following two features are also considered in the classification algorithm: (1) distance from the fitting ellipse centroid to the object centroid and (2) average distance from one ellipse centroid to the centroids of all the other fitting ellipses for the same object contour.

With these specifically designed features, we train a binary SVM classifier to refine the generated candidate ellipses. The training data are manually annotated, with correct fitting ellipses as positive and false fitting as negative. For classification improvement, we can select only the most representative ellipses and the discriminative features (Su et al., 2014). During the testing stage, many false fitting ellipses

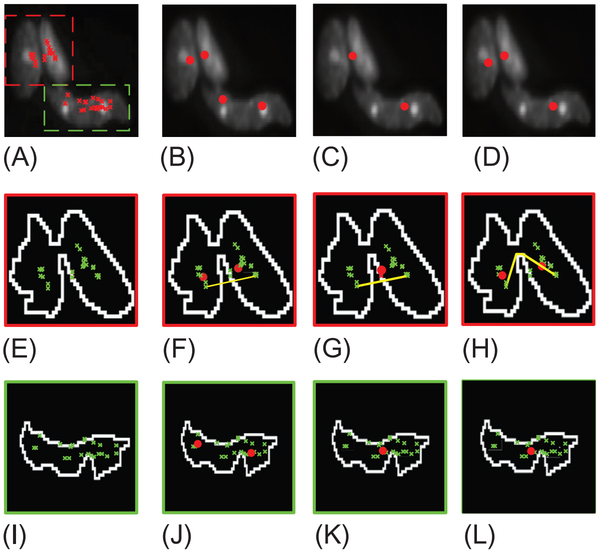

FIG. 4.2

The demonstration of the effectiveness of mean-shift clustering based on inner distance. Gold-standard myonuclei detection should identify two touching nuclei on the top and one single nuclei at the bottom (in total three myonuclei). This is a challenging problem because of the bright dots irregularly distributed inside the myonuclei, which is caused by the DAPI intercalating preferentially into heterochromatin. (A) The initial fitting ellipses' centers generated by the SVM classifier-based refinement. (B) The clustering results (red dots) based on Euclidean distance with bandwidth $bwd_c = 10$. (C) The clustering based on Euclidean distance with bandwidth $bwd_c = 14$. As presented in (B) and (C), it is difficult to find a unified bandwidth that can produce accurate detection results for both cases. However, using inner distance, a unified bandwidth ($bwd_c = 14$) can be used and the accurate detection results for both cases are shown in (D). (E)–(H) Edge images overlaid with SVM refinement results (green crosses) and clustering results (red dots) for the top-left patch in (A). A similar illustration is provided in (I)–(L) for the bottom-right patch in (A). The lines in (F) and (G) denote Euclidean distances between two sample seeds, and the line in (H) denotes an inner distance between two sample seeds.

can be removed, but multiple candidates still remain for a gold-standard myonucleus. Therefore postprocessing is required to achieve final myonucleus detection.

4.2.1.4 Inner geodesic distance-based clustering

In order to remove redundant ellipses, which crowd around a single myonucleus, we apply mean-shift clustering to the candidates after SVM refinement, as shown in Fig. 4.2. Instead of relying on the Euclidean distance, we use the inner geodesic distance for clustering to find the final myonucleus centers. Unlike the Euclidean distance, the inner distance is calculated as the length of the shortest connecting paths that only lay inside the contour. It builds a graph with the centers of candidate ellipses and the concave points as vertices and the links connecting these vertices inside the contour as edges, and then runs a shortest distance algorithm in the graph. Since the inputs of the mean-shift clustering are the coordinates of the points, the inner-distance matrix will be converted into a new coordinate system by harboring the origin at one of the two points and calculating the other relevant distances.

Inner distance is intuitively correct because the real distance between two seeds should be the paths within the object, instead of a direct line that might cross the cell boundaries (Yang et al., 2008). As shown in Fig. 4.2A, the Euclidean-distance based clustering will encounter some significant challenges in selecting proper clustering bandwidth (Fig. 4.2B or C). On the other hand, based on the inner geodesic distance, we can obtain correct clustering results using one unified bandwidth, as shown in Fig. 4.2D. Inner distance is proven to be quite effective in natural shape classification (Ling and Jacobs, 2007). The myonucleus detection using the proposed method on nine sample fused images is shown in Fig. 4.3, which demonstrates that hundreds of myonuclei are correctly located. We compare the proposed framework with three recent state-of-the-art cell detection methods: Laplacian-of-Gaussian filters (LoG)

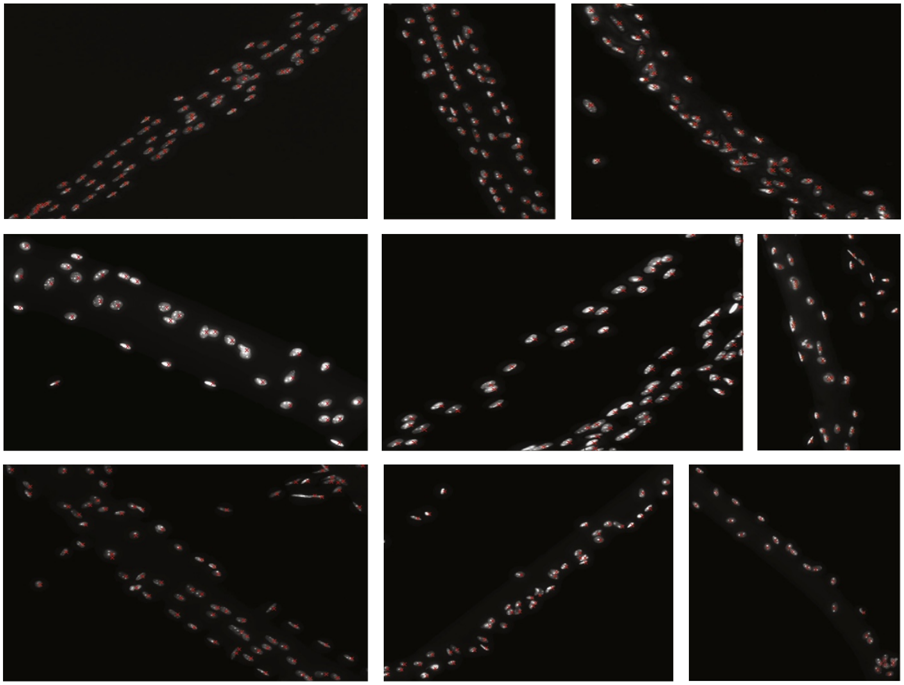

FIG. 4.3

The automated myonucleus detection results on nine randomly selected image patches.

Table 4.1 The Pixel-Wise Seed Detection Accuracy Compared With the Gold Standard

	Mean	Standard Deviation	Min	Max
LoG (Al-Kofahi et al., 2010)	3.72	4.22	0.55	7.99
IRV (Parvin et al., 2007)	4.28	3.86	0.24	**7.93**
SPV (Qi et al., 2012)	2.99	3.54	**0**	7.96
Proposed	**2.6**	**2.90**	0.17	**7.93**

(Al-Kofahi et al., 2010), iterative radial voting (IRV) (Parvin et al., 2007), and single-pass voting (SPV) (Qi et al., 2012). The pixel-wise detection accuracy using over 500 multifocus *z*-stack images (over 1500 myonuclei) is listed in Table 4.1. The best performance in terms of each metric is highlighted in bold. It is clear that the proposed method produces the best accuracy with respect to both mean errors and standard deviations, representing high detection accuracy and reliability.

4.2.2 DEEP CONVOLUTIONAL NEURAL NETWORK

Recently, deep learning-based models, especially CNN, have achieved outstanding performance in both natural (LeCun et al., 2015; Krizhevsky et al., 2012; Farabet et al., 2013) and medical (Liao et al., 2013; Wu et al., 2013; Cireşan et al., 2013) image analyses. In contrast to the conventional supervised learning techniques (eg, SVM) relying on hand-crafted features that need sophisticated design, CNN can automatically learn multilevel hierarchies of features that are invariant to irrelevant variations of samples while preserving relevant information (LeCun et al., 2010). Usually a CNN consists of successive pairs of convolutional and pooling layers, followed by several fully connected layers (LeCun et al., 1998). A convolutional layer learns a set of convolutional filters that will be used to calculate output feature maps, with all units in a feature map sharing the same weights. A max-pooling layer summarizes the activities and picks up the max values over a neighborhood region in each feature map (Hinton et al., 2012), which not only reduces feature dimensionality but also introduces local shift and translation invariance to the neural network. The convolutional-pooling layers are stacked to learn local hierarchical features, based on which the fully connected layers learn more higher level feature representation for classification. The last layer is often a sigmoid layer producing the probability distributions over categories (Krizhevsky et al., 2012; Farabet et al., 2013).

Instead of solving a pixel-wise classification problem, we have proposed a novel CNN-based structured regression model for robust nucleus and cell detection on breast cancer, pancreatic neuroendocrine tumor (NET), and HeLa cervical cancer microscopic images (Xie et al., 2015). We modify the conventional CNN by replacing the last layer (classifier) with a structured regression layer to encode topological information. Meanwhile, instead of working on the label space, regression on the proposed structured proximity space for patches is performed so that the centers of

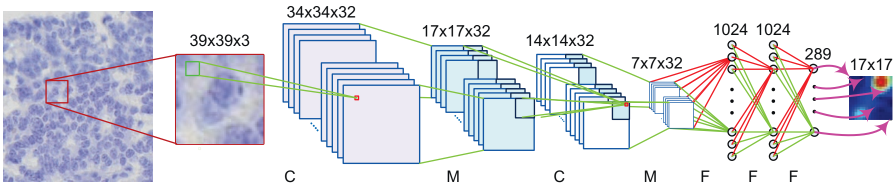

FIG. 4.4

The CNN architecture used in the proposed structured regression model on the NET dataset. C, M, and F represent the convolutional layer, max-pooling layer, and fully connected layer, respectively. The arrows from the last layer illustrate the mapping from the final layer's outputs to the proximity patch.

image patches are explicitly forced to get higher values than their neighbors. These proximity patches are then gathered from all testing image patches and fused to obtain the final proximity map, where the maximum positions indicate the object centroids. The proposed model, whose architecture is listed in Fig. 4.4, is able to handle touching cells, inhomogeneous background noises, and large variations in sizes and shapes, and exhibits superior performance over existing state-of-the-art methods.

4.2.2.1 CNN-based structured regression

Let $\mathcal{X}$ denote the patch space, which consists of a set of $d \times d \times c$ local image patches extracted from c-channel images I. It defines $\mathcal{M}$ as the proximity mask corresponding to image I, and computes the ijth entry in $\mathcal{M}$ as

$$\mathcal{M}_{ij} = \begin{cases} \frac{1}{1+\alpha D(i,j)} & \text{if } D(i,j) \leq r, \\ 0 & \text{otherwise}, \end{cases} \tag{4.10}$$

where $D(i,j)$ represents the Euclidean distance from pixel (i,j) to the manually annotated nucleus or cell center, r is a distance threshold (selected as 5 pixels), and α is the decay ratio (set as 0.8). An image patch $x \in \mathcal{X}$ centered at (u,v) is represented by a quintuple $\{u,v,d,c,I\}$, with the corresponding proximity patch $s \in \mathcal{V}^{d' \times d'}$ that can be represented as $\{u,v,d',\mathcal{M}\}$ (d' is not necessarily equal to d). We are given training data $\{(\boldsymbol{x}^i,\boldsymbol{y}^i) \in (\mathcal{X},\mathcal{Y})\}_{i=1}^{\mathcal{N}}$, where $\mathcal{Y} \subset \mathcal{V}^{p\times 1}$ represents the output space of the structured regression model, and $p = d' \times d'$ denotes the number of units in the last layer. The outputs are $\boldsymbol{y}^i = \Gamma(\boldsymbol{s}^i)$, where $\Gamma : \mathcal{V}^{d' \times d'} \to \mathcal{Y}$ is a mapping function to represent the vectorization operation in column-wise order for a proximity patch. Define $\{\boldsymbol{\theta}_l\}_{l=1}^{L}$ as the parameters corresponding to each of the L layers, the training process of the structured regression model can be formulated as learning a mapping function F represented by $\{\boldsymbol{\theta}_l\}_{l=1}^{L}$, which will map the image space $\mathcal{X}$ to the output space $\mathcal{Y}$. Therefore the optimization problem can be formulated as

$$\arg\min_{\boldsymbol{\theta}_1,\ldots,\boldsymbol{\theta}_L} \frac{1}{\mathcal{N}} \sum_{i=1}^{\mathcal{N}} \mathcal{L}(\psi(\boldsymbol{x}^i;\boldsymbol{\theta}_1,\ldots,\boldsymbol{\theta}_L),\boldsymbol{y}^i), \tag{4.11}$$

where $\mathcal{L}$ is the user-defined loss function. For training sample $(\boldsymbol{x}^i,\boldsymbol{y}^i)$, it is defined as

$$\begin{aligned}\mathcal{L}(\psi(\boldsymbol{x}^i;\boldsymbol{\theta}_1,\ldots,\boldsymbol{\theta}_L),\boldsymbol{y}^i) = \mathcal{L}(\boldsymbol{o}^i,\boldsymbol{y}^i) &= \frac{1}{2}\sum_{j=1}^{p}(y_j^i+\lambda)(y_j^i-o_j^i)^2 \\ &= \frac{1}{2}\left\|(Diag(\boldsymbol{y}^i)+\lambda\mathbf{I})^{1/2}(\boldsymbol{y}^i-\boldsymbol{o}^i)\right\|_2^2,\end{aligned} \tag{4.12}$$

where $\mathbf{I}$ is an identity matrix of size $p \times p$, $\boldsymbol{o}^i = \psi(\boldsymbol{x}^i;\boldsymbol{\theta}_1,\ldots,\boldsymbol{\theta}_L)$ represents the output of the last layer, and $Diag(\boldsymbol{y}^i)$ denotes a diagonal matrix with the jth diagonal element equal to y_j^i.

Eq. (4.11) can be solved using the classical back-propagation algorithm. Since the nonzero region in the proximity patch is relatively small, our model might return a trivial solution. To alleviate this problem, we adopt a weighting strategy (Szegedy et al., 2013) to give more weights to the loss coming from the network's outputs corresponding to the nonzero area in the proximity patch. A small λ indicates strong penalization that is applied to errors coming from the outputs with low proximity values in the training data. Our model is different from Szegedy et al. (2013) which applies a bounding box mask regression approach on the entire image.

We choose the sigmoid activation function in the last layer, that is, $o_j^i = sigm(a_j^i)$ (a_j^i is the jth element of $\boldsymbol{a}^i$ representing the input of the last layer). The partial derivative of (4.12) with respect to the input of the jth unit in the last layer is given by

$$\frac{\partial\mathcal{L}(\boldsymbol{o}^i,\boldsymbol{y}^i)}{\partial a_j^i} = \frac{\partial\mathcal{L}(\boldsymbol{o}^i,\boldsymbol{y}^i)}{\partial o_j^i}\frac{\partial o_j^i}{\partial a_j^i} = (y_j^i+\lambda)(o_j^i-y_j^i)a_j^i(1-a_j^i). \tag{4.13}$$

Based on (4.13), we can evaluate the gradients of (4.11) with respect to the model's parameters in the same way as in LeCun et al. (1998). The optimization is conducted with the mini-batch stochastic gradient descent.

4.2.2.2 CNN architecture

The proposed structured regression model contains several convolutional layers (C), max-pooling layers (M), and fully connected layers (F). Fig. 4.4 illustrates the architectures and mapped proximity patches in the proposed model on the NET dataset. The detailed model configuration is: Input(39 × 39 × 3) − C(34 × 34 × 32) − M(17 × 17 × 32) − C(14 × 14 × 32) − M(7 × 7 × 32) − F(1024) − F(1024) − F(289). The input image size depends on cell scales, and a 39 × 39 patch is large enough to cover a single cell in NET images. Due to the small size of the input image patch, it is sufficient to stack two pairs of C-M layers for feature computation. Meanwhile, multiple F layers are designed to learn more higher level feature representation, which can benefit the final regression. The activation function of the last F (regression) layer is chosen as the sigmoid function, and an ReLu function is used for all the other F and C layers. The sizes of C and M layers are defined as *width* × *height* × *depth*, where *width* × *height* determines the dimensionality of each feature map and *depth* represents the number of feature maps. Since the input image size is relatively small, the filter size is chosen as 6 × 6 for the first convolutional layer and 3 × 3 for the other. The max-pooling layer uses a window of size 2 × 2 with a stride of 2, which has been widely adopted in current object detection algorithms and gives an encouraging performance. Similar CNN architectures are used for breast cancer and HeLa cervical cancer datasets, but with input patch sizes of 49 × 49 × 3 and 31 × 31 × 3, respectively.

4.2.2.3 Structured prediction fusion and cell localization

Given a testing image patch $x = (u, v, d, c, I)$, it is easy to get the corresponding proximity mask as $s = \Gamma^{-1}(\boldsymbol{y})$, where $\boldsymbol{y} \in \mathcal{Y}$ represents the model's output corresponding to $\boldsymbol{x}$. In the fusion process, s will cast a proximity value for every pixel that lies in the $d' \times d'$ neighborhood area of (u, v). For example, pixel $(u+i, v+j)$ in image I will get a prediction s_{ij} from pixel (u, v). In other words, each pixel actually receives $d' \times d'$ predictions from its neighboring pixels. To get the fused proximity map, we average all the predictions from its neighbors for each pixel to calculate its final proximity prediction. After this step, the cell localization can be easily obtained by finding the local maximum positions in the average proximity map. In order to reduce the running time of testing, we present a striding strategy for speed improvement. This is based on the observation that our model generates a $d' \times d'$ proximity patch for each testing patch, and it is feasible to skip a lot of pixels and only test the image patches at a certain stride ss $(1 \leq ss \leq d')$ without significantly sacrificing the accuracy.

4.2.2.4 Experimental results

The proposed model is evaluated on three datasets: 32 breast cancer, 60 pancreatic NET, and 22 phase-contrast HeLa cervical cancer images. All of the data are randomly split into halves for training and testing. For quantitative analysis, we define the gold-standard areas as circular regions within 5 pixels of each annotated cell center. A detected cell centroid is considered to be a true positive (TP) only if it lies within the gold-standard areas; otherwise, it is considered as a false positive (FP). Each TP is matched with the nearest ground-truth annotated cell center. The gold-standard cell centers that are not matched by any detected results are considered to be false negatives (FN). Based on these definitions, we can compute the precision (P), recall (R), and F_1-score as $P = \frac{TP}{TP+FP}$, $R = \frac{TP}{TP+FN}$, and $F_1 = \frac{2PR}{P+R}$, respectively.

We compare the proposed structured regression model (SR-1 means testing with a stride of 1 pixel) with four state-of-the-art methods, including nonoverlapping extremal regions selection (NERS) (Arteta et al., 2012), iterative radial voting (IRV) (Parvin et al., 2007), Laplacian-of-Gaussian filtering (LoG) (Al-Kofahi et al., 2010), and image-based tool for counting nuclei (ITCN) (Byun et al., 2006). In addition to precision, recall, and F_1-score, we also compute the mean and standard deviation of two terms: (1) the absolute difference $\mathbf{E_n}$ between the number of true positive and ground-truth annotations, and (2) the Euclidean distance $\mathbf{E_d}$ between the true positive and the corresponding annotations. The quantitative experimental results are reported in Table 4.2. It is obvious that our method provides better performance than the other methods in all three data sets, especially in terms of F_1-score. Our method also shows strong reliability with the lowest mean and standard deviations in $\mathbf{E_n}$ and $\mathbf{E_d}$ on NET and phase contrast data sets.

Table 4.2 The Comparative Cell Detection Results on Three Data Sets

Data	Methods	P	R	F_1	$\mu_d \pm \sigma_d$	$\mu_n \pm \sigma_n$
Breast cancer	SR-1	**0.919**	0.909	**0.913**	**3.151 ± 2.049**	4.8750 ± 2.553
	NERS (Arteta et al., 2012)	–	–	–	–	–
	IRV (Parvin et al., 2007)	0.488	0.827	0.591	5.817 ± 3.509	9.625 ± 4.47
	LoG (Al-Kofahi et al., 2010)	0.264	**0.95**	0.398	7.288 ± 3.428	**2.75 ± 2.236**
	ITCN (Byun et al., 2006)	0.519	0.528	0.505	7.569 ± 4.277	26.188 ± 8.256
NET	SR-1	0.864	**0.958**	**0.906**	**1.885 ± 1.275**	**8.033 ± 10.956**
	NERS (Arteta et al., 2012)	**0.927**	0.648	0.748	2.689 ± 2.329	32.367 ± 49.697
	IRV (Parvin et al., 2007)	0.872	0.704	0.759	2.108 ± 3.071	15.4 ± 14.483
	LoG (Al-Kofahi et al., 2010)	0.83	0.866	0.842	3.165 ± 2.029	11.533 ± 21.782
	ITCN (Byun et al., 2006)	0.797	0.649	0.701	3.643 ± 2.084	24.433 ± 40.82
Cervical cancer	SR-1	**0.942**	**0.972**	**0.957**	**2.069 ± 1.222**	**3.455 ± 4.547**
	NERS (Arteta et al., 2012)	0.934	0.901	0.916	2.174 ± 1.299	11.273 ± 11.706
	IRV (Parvin et al., 2007)	0.753	0.438	0.541	2.705 ± 1.416	58.818 ± 40.865
	LoG (Al-Kofahi et al., 2010)	0.615	0.689	0.649	3.257 ± 1.436	29.818 ± 16.497
	ITCN (Byun et al., 2006)	0.625	0.277	0.371	2.565 ± 1.428	73.727 ± 41.867

μ_d, σ_d represent the mean and standard deviation of $\mathbf{E_d}$ and μ_n, σ_n represent the mean and standard deviation of $\mathbf{E_n}$.

4.3 SEGMENTATION

Unlike nucleus or cell detection that only needs to locate the object centroids, segmentation aims to separate individual nuclei or cells by delineating their boundaries. It is a prerequisite for many quantitative image analyses including cellular characteristic description (eg, size, shape, texture, and other imagenomics), and thus might provide diagnosis and prognosis support for improved characterization and personalized treatment. However, it is not easy to achieve robust automated nucleus or cell segmentation. First, there usually exist a lot of noise in microscopic images, especially histopathological images; second, nuclei or cells exhibit significant scale and intracellular intensity variations; finally, many nuclei or cells are clustered or even partially overlap with each other such that no clear boundaries exist.

Many state-of-the-art approaches have been applied to nucleus or cell segmentation on specific medical images. The watershed transform was perhaps the most popular in the early days, but it is prone to oversegmentation and requires region merging (Lin et al., 2003; Wählby et al., 2002) or marker controlling (Zhou et al., 2005; Adiga et al., 2006). The graph cut framework (Chang et al., 2012; Lou et al., 2012) or Voronoi diagrams (Jones et al., 2005) are also used for cell segmentation, but they usually need high time cost for large-scale images or might fail to handle weak cell boundaries. Another type of widely used methods are deformable models (Xing and Yang, 2013; Qi et al., 2012), but the level set-based implementation (Ali and Madabhushi, 2012; Yan et al., 2008) is usually computationally expensive and probably produces undesired object topology changes. In addition, it often needs experts to carefully tune the parameters. On the contrary, the learning techniques (Kong et al., 2011; Janssens et al., 2013) do not require sophisticated parameter selection or predefined processing. In this section, we will introduce two machine learning-based techniques, random forests and sparsity-based dictionary learning, for nucleus or cell segmentation in microscopic images.

4.3.1 RANDOM FORESTS

A random forest (Breiman, 2001) $\mathcal{F} = \{T_t\}$ is an ensemble of decision trees T_t, which are trained independently on randomly selected samples $S = \{s_i = (\boldsymbol{x}_i \in \mathcal{X}, y_i \in \mathcal{Y})\}$, where $\mathcal{X}$ and $\mathcal{Y}$ denote the input features and output labels, respectively. A decision tree $T_t(\boldsymbol{x})$ produces the prediction results by recursively branching a feature sample $\boldsymbol{x} \in \mathcal{X}$ left or right down the tree until a leaf node is reached. For a decision forest, the predictions $T_t(\boldsymbol{x})$ from individual trees are combined together using an ensemble model. Majority voting and averaging are typical ensemble choices for classification and regression problems, respectively.

During the training of a decision tree, at each node n, a split function $h(\boldsymbol{x}, \boldsymbol{\theta}_n)$ is chosen to split the samples S_n into left S_n^L or right S_n^R. The split function $h(\boldsymbol{x}, \boldsymbol{\theta}_n)$ is optimized by maximizing the information gain:

$$\mathcal{I}(S_n) = H(S_n) - \left(\frac{|S_n^L|}{|S_n|} H(S_n^L) + \frac{|S_n^R|}{|S_n|} H(S_n^R) \right), \tag{4.14}$$

where $H(\cdot)$ is the class entropy function. The split function $h(\boldsymbol{x}, \boldsymbol{\theta}_n)$ can be an arbitrary classifier. A common choice is a stump function that is found to be computationally efficient and effective in practice (Dollar and Zitnick, 2014). The training procedure continues to split the samples until either a maximum depth is reached, or too few samples are left, or information gain falls below a certain threshold.

We have proposed a structured edge detection algorithm based on random forests for muscle cell segmentation on hematoxylin and eosin (H&E)-stained microscopic images (Liu et al., 2015). The structured edge detection, which can better capture inherent muscle image edge structures, is achieved by extending a random decision forest framework. It is noted in Kontschieder et al. (2011) and Dollar and Zitnick (2014) that, by storing structure information instead of class probabilities at the leaf nodes of the random decision trees, random decision forests can be conveniently used for structured learning. Therefore edge masks rather than edge probability values will be stored at the leaf nodes in our proposed structured edge detection algorithm. In order to accurately segment each muscle cell exhibiting both strong and weak boundaries, a hierarchical segmentation method is proposed, which takes a set of partitions produced by using a segmentation algorithm with varying parameters as inputs, and selects a best subset of nonoverlapping partition regions as the final results.

An overview of our proposed muscle image segmentation algorithm is shown in Fig. 4.5. Given an image patch, (1) an edge map is generated by the proposed structured edge detection algorithm; (2) an Ultrametric Contour Map (UCM) (Arbelaez et al., 2011) is constructed and a set of segmentation candidates is generated by adjusting the thresholds of UCM; and (3) an efficient dynamic programming-based subset selection algorithm is then used to choose the best regions for muscle image segmentation based on a constructed tree graph.

4.3.1.1 Structured edge detection

Since a decision tree classifier generates the actual prediction at the leaf nodes, more information (instead of only class likelihoods) can be stored at the leaf nodes. For example, in Kontschieder et al. (2011), structured class label information is stored at leaf nodes for semantic image segmentation. Similar to Dollar and Zitnick (2014) and Chen et al. (2015), we have stored edge structure information at the leaf nodes for structured muscle image edge detection. Different from traditional edge detection algorithms (Arbelaez et al., 2011), which take an image patch x as an input and compute the probability of the edge existence at the center pixel p, the output of our proposed structured edge detection algorithm is an edge mask around the central pixel p instead of the likelihood value. After the decision tree is learned, the median or mean of the edge masks sent to the leaf node will be stored as the leaf node output, as shown in Fig. 4.5.

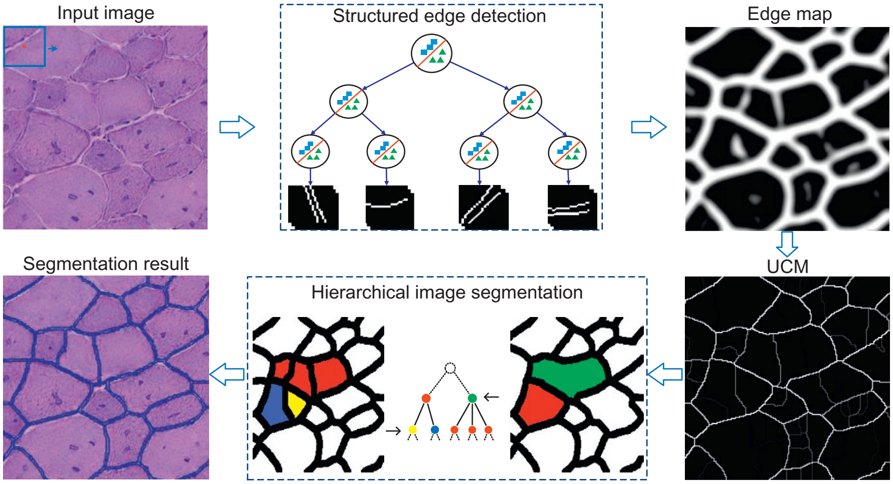

FIG. 4.5

An overview of the proposed muscle image segmentation algorithm.

The information gain criterion in Eq. (4.14) is effective in practice for decision tree training. In order to follow this criterion, the edge masks must be explicitly assigned proper class labels at each internal node of the tree during the training stage. One straightforward idea is to group the edge masks at a node into several clusters by an unsupervised clustering algorithm such as k-means or mean-shift (Comaniciu and Meer, 2002), and then treat each cluster *id* as the class label for the sample belonging to that cluster. However, the edge masks $\mathcal{Y}$ do not reside in the Euclidean space so that direct grouping may not generate desired results. In addition, clustering in a high-dimension space ($y \in \mathbb{R}^{256\times 1}$ for a 16 × 16 edge mask) is computationally expensive. To address this problem, we propose to reduce the high-dimension edge masks $\mathcal{Y} \in \mathbb{R}^n$ to a lower dimensional subspace $\mathcal{Z} \in \mathbb{R}^m$ ($m << n$) using an autoencoder (Hinton and Salakhutdinov, 2006) before clustering the edge masks. For notation convenience, we use the matrix form and vector form of edge mask space $\mathcal{Y}$ interchangeably in this section.

Although the transformed data $\boldsymbol{z} \in \mathcal{Z}$ is used to choose a split function $h(\boldsymbol{x}, \boldsymbol{\theta}_n)$ during the training of the decision tree, only the original edge masks are stored at leaf nodes for the prediction. Several sample edge masks learned and stored at the leaf nodes are shown in Fig. 4.6. As one can tell, many edge structures are unique for muscle cell boundaries, which demonstrates the effectiveness of the structured edge detection procedure. The proposed structured edge detection algorithm takes a 32 × 32 image patch as input and generates a 16 × 16 edge mask around the input's center pixel. The image patch is represented with the same high-dimensional feature used in Dollar and Zitnick (2014) and Arbelaez et al. (2014), and it is effective and computationally efficient. In total, two million samples are randomly generated to

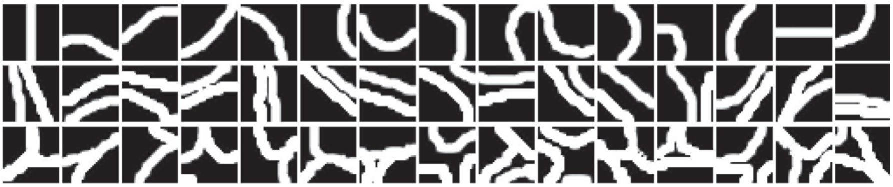

FIG. 4.6

Several sample edge masks learned and stored at the leaf nodes of the random decision trees.

train the structured decision random forest, which consists of eight decision trees. The autoencoder model used in our work consists of an encoder with layers of sizes $(16 \times 16) - 512 - 256 - 30$ and a symmetric decoder. The autoencoder model is trained once offline and applied to all decision trees, and the data compression is only performed at the root node.

4.3.1.2 Hierarchical image segmentation

Recently, the hierarchical strategy has been successfully applied to image segmentation (Farabet et al., 2013; Uzunbaş et al., 2014). In general, the hierarchical image segmentation consists of two steps: candidate region generation and selection. Specifically, a collection of segmentation candidates is first generated by running some existing segmentation algorithms with different parameters. Usually, an undirected graph is constructed from these partition candidates, in which an edge exists between two touching or overlapping regions. Next, based on some domain-specific criteria, a subset of nonoverlapping regions is selected as the final segmentation results. For example, Felzenszwalb's method (Felzenszwalb and Huttenlocher, 2004) with multiple levels is used to generate the segmentation candidate pool, and an optimal purity cover algorithm (Farabet et al., 2013) can be adopted to select the most representative regions. In Uzunbaş et al. (2014), the watershed segmentation method with different thresholds gives a collection of partitions, and then a conditional random field (CRF)-based learning algorithm is utilized to find the best ensembles as final segmentation.

In our implementation, an UCM (Arbelaez et al., 2011; Arbelaez, 2006), which defines a duality between closed, nonself-intersecting weighted contours and a hierarchy of regions, is used to generate a pool of segmentation candidates. Because of the nice property of UCM where the segmentation results using different thresholds are nested into one another, we can construct a tree graph for this pool of segmentation candidates. The final step is to solve this tree graph-based problem using dynamic programming.

Given a set of segmentation candidates generated with different thresholds using UCM, an undirected and weighted tree graph, $G = (V, E, \mathbf{w})$, is constructed, where $V = \{v_i, i = 1, 2, \ldots, n\}$ represents the nodes with each v_i corresponding to a segmented region S_i. E denotes the edges of the graph. The $w(v_i)$ is learned via a general random decision forest classifier to represent the likelihood of S_i as a real muscle cell. An adjacent matrix $A = \{a_{ij}|i, j = 1, \ldots, n\}$ is then built with $a_{ij} = 1$ if $S_i \subset S_j$ or $S_j \subset S_i$, and otherwise 0. Denote $\mathbf{x} \in \{0, 1\}^n$ the indicator vector, where its element is equal to 1 if the corresponding node is selected, otherwise 0. Finally, the constrained subset selection problem is formulated as

$$\mathbf{x}^* = \arg\max_{\mathbf{x} \in \mathcal{X}} \mathbf{w}^{\mathrm{T}}\mathbf{x}, \tag{4.15}$$

where $\mathcal{X}$ denotes all possible valid configurations of $\mathbf{x}$. Considering the special tree graph structure, we can efficiently solve (4.15) via the dynamic programming approach with a bottom-up strategy.

In order to ensure that solving (4.15) selects the desired regions, each candidate region (node) must be assigned an appropriate muscle cell likelihood score w. In our algorithm, each candidate region is discriminatively represented with a feature vector that consists of a descriptor to model the convexity of the shape, and two histograms to describe the gradient magnitudes of the pixels on the cell boundary and inside the cell region. These morphological features are proposed based on the following observations: (1) the shape of a muscle cell is nearly convex; (2) the cell boundaries often exhibit higher gradient magnitudes; (3) the intensities within the cell regions should be relatively homogeneous.

4.3.1.3 Experimental results

We have tested the proposed approach on 120 H&E-stained muscle cell images captured at 10× magnification. Each image contains around 200 muscle cells. The images are randomly split into two sets of equal size, one for training and the other for testing. The gold standard of each individual muscle cell is manually annotated. To quantitatively analyze the pixel-wise segmentation accuracy, we calculate the precision $P = \frac{|S \cap G|}{|S|}$, recall $R = \frac{|S \cap G|}{|R|}$, and F_1-score $= \frac{|2PR|}{|P+R|}$, where S denotes the segmentation result and G is the gold standard.

We compare the proposed method with three state-of-the-art methods: isoperimetric graph partition (ISO) (Grady and Schwartz, 2006), global probability of boundary detector (gPb) (Arbelaez et al., 2011), and a six-layer deep CNN. The comparative boxplots of F_1-scores are shown in Fig. 4.7. We can see that our structured edge detection-based segmentation algorithm outperforms the others on digitized muscle specimens. The quantitative comparative results are shown in Table 4.3, where we report the average and standard variance of F_1-score, precision, and recall. In the table, we provide the proposed structured detection-based segmentation algorithm with and without using the presented hierarchical segmentation (denoted as Prop. w. H. and Prop. w.o. H., respectively). It is worth noting that the proposed structured edge detection algorithm indeed performs better than DCNN for the muscle image dataset. One potential reason is that we do not have sufficient training data to learn a DCNN model that is sufficient to capture all the edge variations. A larger DCNN model with more training data might be able to achieve better performance. With

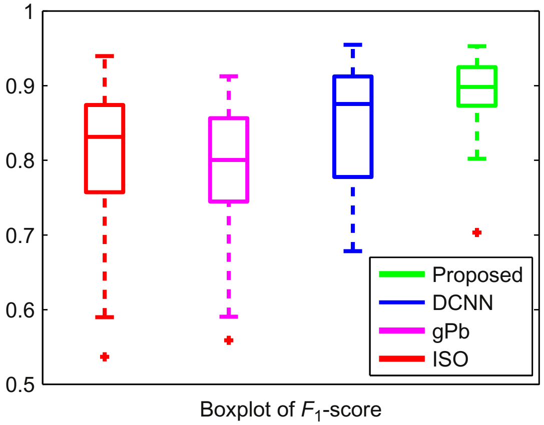

FIG. 4.7

The comparative segmentation results of the proposed muscle image segmentation algorithm with three state-of-the-art methods.

Table 4.3 The Pixel-Wise Segmentation Accuracy

	F_1-Score		**Prec.**		**Rec.**	
Method	**Mean**	**Std**	**Mean**	**Std**	**Mean**	**Std**
ISO (Grady and Schwartz, 2006)	0.8050	0.0993	0.8988	0.0589	0.7429	0.1369
gPb (Arbelaez et al., 2011)	0.7904	0.0780	0.9123	0.0515	0.7011	0.0962
DCNN	0.8388	0.1073	**0.9464**	**0.0411**	0.7666	0.1441
Prop. w.o. H.	0.8815	0.0523	0.8861	0.0551	0.8783	0.0587
Prop. w. H.	**0.8974**	**0.0422**	0.9078	0.0433	**0.8888**	**0.0543**

respect to speed, the current DCNN model takes 28 seconds to generate an edge map for segmentation on an image of size 1024 × 768, which is much slower than our structured edge detection, which takes less than 1 s.

4.3.2 SPARSITY-BASED DICTIONARY LEARNING

Sparse representation has received a great deal of attention in the fields of machine learning and computer vision due to its state-of-the-art performance in various applications including object segmentation. The K-SVD algorithm (Aharon et al., 2006), which takes its name from K computations of singular value decomposition (SVD) for dictionary basis update (Rubinstein et al., 2010), and its variants are widely used for dictionary learning. K-SVD learns a dictionary from a set of data points and emphasizes its generative power, and the purpose is to minimize the overall reconstruction errors. On the other hand, supervised dictionary learning techniques (Mairal et al., 2008; Zhang and Li, 2010; Jiang et al., 2011) have also been proposed to enforce its discriminative power, which can benefit object classification. Recently a selection-based dictionary learning algorithm was reported in Liu et al. (2013), which selects most representative data points as dictionary bases.

Defining $X = \{\boldsymbol{x}_i | i = 1 \cdots N\}$ as a set of N input data points with p-dimensions, the dictionary learning can be formulated as finding a repository $\boldsymbol{\Phi} \in R^{p \times K}$ for a sparse representation of X:

$$\min_{\boldsymbol{\Phi}, \{\boldsymbol{\alpha}_i\}_{i=1,2,\ldots,N}} \sum_{i=1}^{N} ||\boldsymbol{x}_i - \boldsymbol{\Phi}\boldsymbol{\alpha}_i||_2^2 + \lambda||\boldsymbol{\alpha}_i||_m, \tag{4.16}$$

where $\boldsymbol{\alpha}_i$ represents the ith sparse coefficient, $m = 0$ or 1 indicates using ℓ_0 or ℓ_1 penalty, and λ is the regularization parameter controlling the sparsity of α_i. For sparse coefficient computation, orthogonal matching pursuit (OMP) (Tropp, 2004) can be applied to solve the sparse representation problem for ℓ_0 penalty. When using ℓ_1 penalty, it is a Lasso problem (Tibshirani, 1994), which can be efficiently solved by the LARS algorithm (Efron et al., 2004).

Segmentation models with shape priors can handle weak or misleading object boundaries so that they can significantly improve the accuracy (Cootes et al., 1995; Zhang et al., 2012). Recently the sparse shape model has been shown to be more effective than the principal component analysis-based shape prior due to its insensitiveness to object occlusion (Zhang et al., 2011, 2012). However, using all training shapes is inefficient during sparse reconstruction on a large dataset at run-time, and thus data summarization or dictionary learning is usually required for efficient runtime optimization. We have presented a robust sparsity-based dictionary learning algorithm for nucleus shape modeling, which is incorporated into an integrated framework for automated nucleus segmentation in microscopic images (Xing and Yang, 2013; Xing et al., 2016). In contrast to K-SVD (Aharon et al., 2006), this method directly selects the most representative nucleus shapes from the training dataset as dictionary bases. The robustness of the dictionary learning method is achieved by minimizing an integrated square error with a sparse constraint. In order to simultaneously and efficiently segment multiple nuclei, we combine the top-down shape prior model and a bottom-up deformable model with locality and repulsion constraints. Given a microscopic image, it begins with a deep CNN model to generate a probability map, on which an iterative region merging approach is performed for shape initializations. Next, it alternately performs shape deformation using an efficient local repulsive deformable model, and shape inference using the shape prior derived from the sparse shape model. The flowchart of nucleus segmentation is shown in Fig. 4.8.

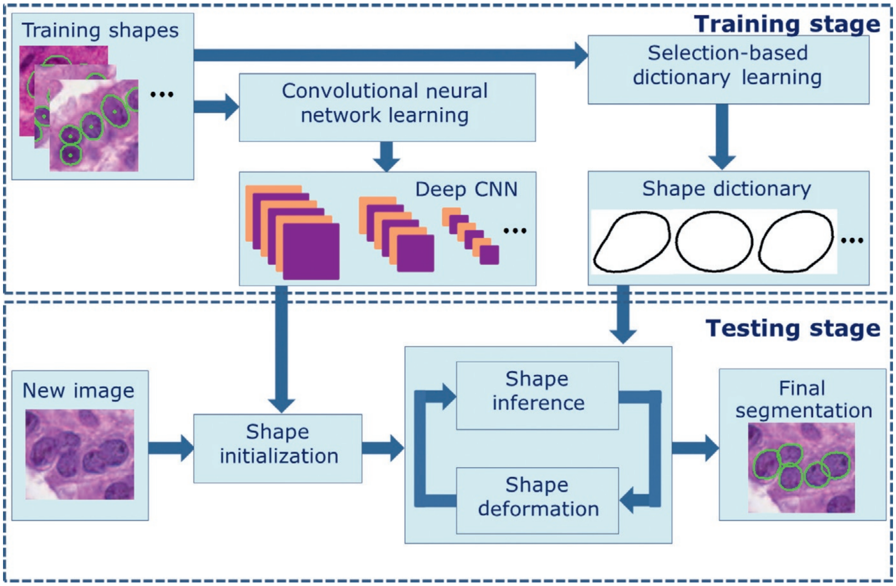

FIG. 4.8

The flowchart of the proposed framework for automated nucleus segmentation in microscopic images.

4.3.2.1 CNN-based shape initialization

In order to facilitate the subsequent contour evolution step, we need to obtain robust initializations. For this purpose, we first learn a deep CNN to generate a probability map which presents nucleus regions, and then apply an iterative region merging algorithm to the map to create the initial shape markers. One of the CNN structures (for breast cancer microscopic images) used in our algorithm is listed in Table 4.4. The last layer is a softmax layer producing probability distributions over classes (Farabet et al., 2013). Our CNN is trained using raw pixel values (the YUV color space) of small image patches with a certain size, centered on the pixel itself. In the testing stage, we apply the learned CNN model to small patches cropped from a new testing image using a pixel-wise sliding window of the same size as training patches. For each testing image, the CNN model with a sliding-window technique creates a probability map where each pixel is assigned a probability of being close to nucleus centers, and those lying in the background would have lower probabilities. Furthermore, an iterative region merging algorithm (Xing et al., 2016) is applied to the probability map to generate initial shapes for the subsequent shape deformation.

Table 4.4 One of the CNN Structure in Our Algorithm

Layer No.	Layer Type	Output Size	Filter Size
1	Input	55 × 55 × 3	—
2	Convolutional	50 × 50 × 48	6 × 6
3	Max-pooling	25 × 25 × 48	2 × 2
4	Convolutional	22 × 22 × 48	4 × 4
5	Max-pooling	11 × 11 × 48	2 × 2
6	Fully connected	1024 × 1	—
7	Fully connected	1024 × 1	—
8	Output	2 × 1	—

4.3.2.2 Sparsity-based shape modeling

We model the nucleus shape priors with sparse representation. For a large nucleus shape dataset, it is intuitive to select a subset of the data as a shape repository that can sufficiently represent the whole dataset. This summarization can help remove outliers that are not true representatives of the dataset and might reduce the computational time for runtime optimization due to the decreased object-space dimension (Elhamifar et al., 2012). Based on these considerations, we propose a novel selection-based dictionary learning method for sparse representation by minimizing a locality-constrained integrated squared error (ISE). Scott (2001) has shown that minimizing the ISE is analogous to minimizing the objective function: $\int g(\boldsymbol{x}|\boldsymbol{\theta})^2 d\boldsymbol{x} - \frac{2}{N}\sum_{i=1}^{N} g(\boldsymbol{x}_i|\boldsymbol{\theta})$, where $g(\boldsymbol{x}|\boldsymbol{\theta})$ is a parametric model with parameter $\boldsymbol{\theta}$ and N is the number of data points $\{\boldsymbol{x}_i\}_{i=1}^{N}$. In a sparse shape model with N nucleus shapes $\{v_i\}_{i=1}^{N}$ aligned by Procrustes analysis, we have $v_i = B\boldsymbol{\alpha}_i + \boldsymbol{\epsilon}_i$, where $B = [\boldsymbol{b}_1\ \boldsymbol{b}_2, \ldots, \boldsymbol{b}_K]$ ($\{\boldsymbol{b}_k \in R^{2m}\}_{k=1}^{K}$ are bases) is the shape dictionary, $\boldsymbol{\alpha}_i$ denotes the sparse coefficient, and $\boldsymbol{\epsilon}_i$ is the residual for the ith shape. Therefore we can model the residual density with function $g(\boldsymbol{\epsilon}|\boldsymbol{\theta})$ and minimize the objective function as follows:

$$\min_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) = \min_{\boldsymbol{\theta}} \left[\left(\int g(\boldsymbol{\epsilon}|\boldsymbol{\theta})^2\, d\boldsymbol{\epsilon} - \frac{2}{N}\sum_{i=1}^{N} g(\boldsymbol{\epsilon}_i|\boldsymbol{\theta}) \right) \right.$$
$$\left. + \lambda \sum_{i=1}^{N}\sum_{k=1}^{K} |\boldsymbol{\alpha}_{ik}| ||v_i - \boldsymbol{b}_k||^2 \right], \quad \text{s.t. } 1^{\mathrm{T}}\boldsymbol{\alpha}_i = 1, \quad \forall i, \tag{4.17}$$

where $\boldsymbol{\epsilon}_i = v_i - B\boldsymbol{\alpha}_i$ and $\boldsymbol{\alpha}_i = [\alpha_{i1}\ \alpha_{i2}\ \ldots\ \alpha_{iK}]^{\mathrm{T}}$. The first two terms form the L_2E criteria, which is robust to outliers (Scott, 2001). The last term constrains local representation of bases with weighted sparse codes, and is used to encourage each nucleus to be sufficiently represented by its neighboring dictionary bases to preserve similarity, which is essential in the sparse reconstruction. The constraint $1^{\mathrm{T}}\boldsymbol{\alpha}_i = 1, \forall i$, ensures the shift invariance. The residual is modeled with multivariate normal distribution: $\boldsymbol{\epsilon}_i \sim N(0, \sigma^2 I_{2m})$. In this way $g(\boldsymbol{\epsilon}_i|\boldsymbol{\theta}) = \xi\phi(\boldsymbol{\epsilon}_i|0, \sigma^2 I_{2m})$, where ξ denotes the percentage of the inlier shapes that need to be estimated and ϕ is the probability density function of a multivariate normal distribution. Based on (4.17), the dictionary B and sparse coefficients $\{\boldsymbol{\alpha}_i\}_{i=1}^{N}$ can be calculated by estimating $\boldsymbol{\theta} = \{\xi, B, \boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2, \ldots, \boldsymbol{\alpha}_N, \sigma^2\}$.

Eq. (4.17) can be solved by performing dictionary basis selection and coefficient computation alternately. As $J(\boldsymbol{\theta})$ in Eq. (4.17) is differentiable with respect to $\{\boldsymbol{b}_k\}_{k=1}^{K}$, projection-based gradient descent is utilized for minimization to update the bases, which are directly selected shapes within each iteration. For coefficient calculation, we keep the dictionary fixed. Based on the sparse reconstruction criterion, the sparse coding objective function can be rewritten as

$$\min_{\{\boldsymbol{\alpha}_i\}_{i\in A}} \left[\sum_{i\in A} ||v_i - B\boldsymbol{\alpha}_i||^2 + \lambda \sum_{k=1}^{K} |\alpha_{ik}|||v_i - \boldsymbol{b}_k||^2 \right], \; s.\,t.\; 1^{\mathrm{T}}\boldsymbol{\alpha}_i = 1, \quad i \in A, \tag{4.18}$$

where A is the set of indices corresponding to estimated inlier shapes. Locality-constrained linear coding (LLC) (Wang et al., 2010) is applied to Eq. (4.18) for coefficient computation, where the neighboring bases are defined in terms of the Euclidean distances between the shape and dictionary bases. Due to large shape variations of nuclei, it would be more effective to build multiple subpopulation shape prior models based on clustered shapes (Xing et al., 2016).

4.3.2.3 Local repulsive deformable model

Combined with shape prior modeling, we propose an efficient shape deformation method based on the Chan-Vese model (Chan and Vese, 2001) for nucleus segmentation. In order to enhance the robustness, we add an edge detector into the original Chan-Vese model combined with the region-based data fitting term to better move contours towards nucleus boundaries. In addition, we introduce a repulsive term (Zimmer and Olivo-Marin, 2005) to handle touching or overlapping nuclei. More importantly, we observe that each nucleus is often surrounded by a limited number of adjacent nuclei, and only its neighboring nuclei make dominant repulsive contributions to its shape deformation during contour evolution. Therefore for each nucleus we do not need to calculate the repulsion from all the other nuclei on the image, but only from its nearest neighbors in a local coordinate system. In this way, the computational cost will be significantly reduced. Consider an image I containing N nuclei, denoted by $v_i (i = 1, \ldots, N)$ with v_i representing the ith contour. The energy function for nucleus segmentation combining the driving and repulsive mechanisms thus can be expressed as follows:

$$\begin{aligned} &\lambda_1 \sum_{i=1}^{N} \int_{\Omega_i} (I(\mathbf{x}) - u_i)^2 \, d\mathbf{x} + \lambda_2 \int_{\Omega_0} (I(\mathbf{x}) - u_0)^2 \, d\mathbf{x} \\ &+ \lambda_3 \sum_{i=1}^{N} \int_0^1 \mathrm{e}(v_i(s)) \, ds + \omega \sum_{i=1}^{N} \sum_{j\in V_i} \int_{\Omega_i \cap \Omega_j} 1 \, d\mathbf{x} + \sum_{i=1}^{N} \gamma |v_i|, \end{aligned} \tag{4.19}$$

where Ω_i and Ω_0 represent the region inside v_i and outside all the contours (background), and u_i and u_0 represent the average intensity of Ω_i and Ω_0, respectively. The third term with $e(v_i(s))$ is the edge detector and is chosen as $-||\nabla I(v_i(s))||^2$ ($s \in [0, 1]$ is the parameter for contour representation), and the fourth term denotes the repulsion preventing contours from crossing with each other. The last term represents the length of v_i.

Using the Euler-Lagrange equations associated with the minimization of (4.19), we can get the following evolution equation:

$$\begin{aligned}\frac{\partial v_i}{\partial t} =& |\frac{\partial v_i}{\partial s}|\mathbf{n}_i(-\lambda_1(I-u_i)^2 + \lambda_2(I-u_0)^2 - \lambda_3 \nabla e(v_i) \\ & - \omega \sum_{j \in V_i} z_j(v_i) + \gamma \rho(v_i)),\end{aligned} \tag{4.20}$$

where $\mathbf{n}_i$ is the normal unit vector of v_i, and $z_j(\mathbf{x})$ represents the indicator function: $z_j(\mathbf{x}) = 1$ if $\mathbf{x} \in \Omega_j$, otherwise 0. $\rho(\cdot)$ denotes the curvature. Given the initial shapes, we can iteratively evolve the contours toward desired nucleus boundaries.

Given initial contours, the proposed segmentation framework alternately performs shape deformation with the repulsive active contour model and shape inference with the sparse shape prior model. The shapes always expand from inside nuclei, one per nucleus, and evolve towards nucleus boundaries. In the active contour model, contours move based on image appearance information until Eq. (4.20) reaches a stable state, where the associated energy function achieves a minimum value; in the shape inference stage, contours evolve based on the high-level shape prior to constrain the shapes. This alternative operation scheme of combining bottom-up and top-down information has been successfully applied to biomedical image segmentation (Zhang et al., 2011, 2012).

4.3.2.4 Experimental results

The proposed framework has been extensively tested on three types of pathology specimens: brain tumor, pancreatic neuroendocrine tumor (NET), and breast cancer. The segmentation results using our method on three sample slide digitized images are shown in Fig. 4.9. As we can see, thousands of nuclei are correctly segmented. For quantitative analysis, we compare the proposed method with six state-of-the-art methods: mean shift (MS) (Comaniciu and Meer, 2002), isoperimetric graph partition (ISO) (Grady and Schwartz, 2006), superpixel (SUP) (Mori, 2005), marker-based watershed (MWS), graph-cut and coloring (GCC) (Al-Kofahi et al., 2010), and repulsive level set (RLS) (Qi et al., 2012), on 30 brain tumor, 51 NET, and 35 breast cancer images. In order to quantitatively analyze the pixel-wise segmentation accuracy, we apply multiple metrics, including Dice similarity coefficient (*DSC*), Hausdorff distance (*HD*), and mean absolute distance (*MAD*), to the evaluation of the algorithms. Letting Ω_{sr} and Ω_{gs} represent the regions inside the automatic segmentation contour v_{sr} and the gold standard contour v_{gs}, respectively, the metrics are defined as (Zhou et al., 2013).

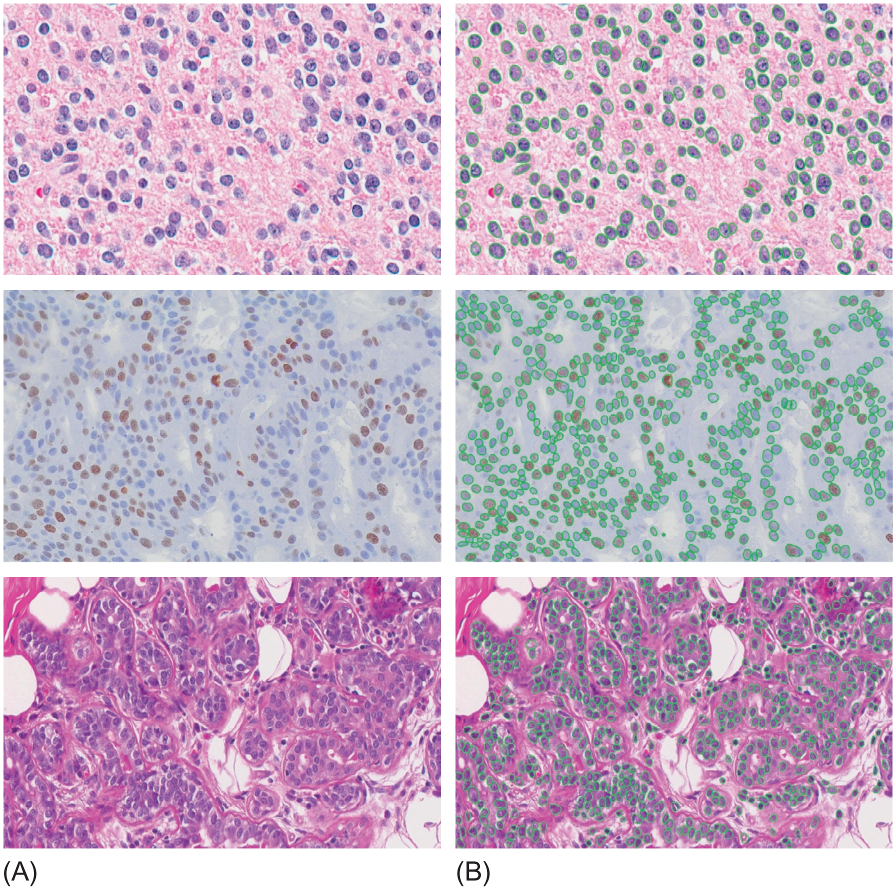

FIG. 4.9

Segmentation results using the proposed method on three sample images of the brain tumor (top), NET (middle), and breast cancer (bottom) datasets. Note that all the nuclei in the connected regions touching image boundaries are ignored. (A) Original images. (B) Segmentation results.

$$
\begin{aligned}
DSC &= \frac{2|\Omega_{sr} \cap \Omega_{gs}|}{|\Omega_{sr}| + |\Omega_{gs}|}, \\
HD &= \max\{\sup_{s} d(v_{sr}(s), v_{gs}), \sup_{s} d(v_{gs}(s), v_{sr})\}, \\
MAD &= \frac{\int_0^1 d(v_{sr}(s), v_{gs})|v'_{sr}(s)|ds}{2|v_{sr}|} + \frac{\int_0^1 d(v_{gs}(s), v_{sr})|v'_{gs}(s)|ds}{2|v_{gs}|},
\end{aligned}
\tag{4.21}
$$

Table 4.5 Comparative Pixel-Wise Segmentation Accuracy on Brain Tumor, NET, and Breast Cancer Datasets

	Brain Tumor			**NET**			**Breast Cancer**		
Brain	***DSC***	***HD***	***MAD***	***DSC***	***HD***	***MAD***	***DSC***	***HD***	***MAD***
MS	0.74	9.98	5.23	0.66	7.01	4.35	0.49	20.83	13.17
ISO	0.70	10.15	6.63	0.48	10.02	8.14	0.56	17.19	11.59
SUP	0.75	10.89	5.32	0.75	6.62	3.77	0.68	17.14	9.32
GCC	0.81	7.07	3.85	0.61	6.37	5.00	0.59	16.84	10.75
MWS	0.81	7.21	3.57	0.82	4.12	2.33	0.73	11.12	6.66
RLS	0.80	8.51	4.59	0.84	2.71	2.26	0.77	10.50	6.30
Proposed	0.85	5.06	3.26	0.92	2.41	1.58	0.80	8.60	6.24

where $d(v_{sr}(s), v_{gs})$ denotes the minimum distance from point s to the contour v_{gs}, sup means the supremum, and $|v_{sr}|$ represents the length of v_{sr}. A large DSC, or a small HD/MAD, indicates high segmentation accuracy. Table 4.5 displays the DSC, HD, and MAD values using MS (Comaniciu and Meer, 2002), ISO (Grady and Schwartz, 2006), SUP (Mori, 2005), MWS, GCC (Al-Kofahi et al., 2010), RLS (Qi et al., 2012), and the proposed method. It is clear that the proposed method provides the best performance, especially in terms of HD that calculates the largest error for each segmentation.

4.4 SUMMARY

This chapter presents several popular machine learning techniques and their applications in nucleus/cell detection and segmentation in microscopic images. Specifically, it discusses a SVM-based myonucleus detection in z-stack fluorescence images, and a deep CNN-based regression model for nucleus detection in bright-field and phase-contrast microscopy images. Due to the encouraging performance of the deep learning on object detection, recently deep neural network has attracted much interest. In addition, it introduces a random forest-based structured edge detection algorithm for muscle cell segmentation in H&E-stained images, and a sparsity-based dictionary learning approach for object shape prior modeling, which is used for nucleus segmentation in both H&E- and Ki-67-stained histopathology images. There exist many other machine learning-based software tools for microscopic image analysis, and more details can be found in Sommer and Gerlich (2013) and Eliceiri et al. (2012).

Currently one challenging problem is to design an efficient and effective algorithm for nucleus/cell detection and segmentation in large-scale microscopic images, such as whole slide imaging (WSI) specimens. WSI provides richer information that might better support CAD, and thus it is necessary to conduct image analysis on WSI specimens. However, these WSI images usually have billions of pixels, which might provide an obstacle for many algorithms. Therefore designing a scalable algorithm that can adapt to large-scale images could attract increasing attention in microscopic image analysis.

REFERENCES

Adiga, U., Malladi, R., Fernandez-Gonzalez, R., de Solorzano, C.O., 2006. High-throughput analysis of multispectral images of breast cancer tissue. IEEE Trans. Image Process. 15 (8), 2259–2268.

Aharon, M., Elad, M., Bruckstein, A., 2006. K-SVD: an algorithm for designing overcomplete dictionaries for sparse representation. IEEE Trans. Signal Process. 54 (11), 4311–4322.

Al-Kofahi, Y., Lassoued, W., Lee, W., Roysam, B., 2010. Improved automatic detection and segmentation of cell nuclei in histopathology images. IEEE Trans. Biomed. Eng. 57 (4), 841–852.

Ali, S., Madabhushi, A., 2012. An integrated region-, boundary-, shape-based active contour for multiple object overlap resolution in histological imagery. IEEE Trans. Med. Imaging 31 (7), 1448–1460.

Arbelaez, P., 2006. Boundary extraction in natural images using ultrametric contour maps. In: Conference on Computer Vision and Pattern Recognition Workshop—CVPRW'06, pp. 182–182.

Arbelaez, P., Maire, M., Fowlkes, C., Malik, J., 2011. Contour detection and hierarchical image segmentation. IEEE Trans. Pattern Anal. Mach. Intell. 33 (5), 898–916.

Arbelaez, P., Pont-Tuset, J., Barron, J., Marques, F., Malik, J., 2014. Multiscale combinatorial grouping. In: 2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 328–335.

Arteta, C., Lempitsky, V., Noble, J.A., Zisserman, A., 2012. Learning to detect cells using non-overlapping extremal regions. In: Medical Image Computing and Computer-Assisted Intervention—MICCAI 2012, vol. 7510, pp. 348–356.

Bernardis, E., Yu, S., 2010. Finding dots: segmentation as popping out regions from boundaries. In: 2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 199–206.

Bishop, C., 2007. Pattern Recognition and Machine Learning. Springer, New York.

Breiman, L., 2001. Random forests. Mach. Learn. 45 (1), 5–32.

Byun, J.Y., Verardo, M.R., Sumengen, B., Lewis, G.P., Manjunath, B.S., Fisher, S.K., 2006. Automated tool for the detection of cell nuclei in digital microscopic images: application to retinal images. Mol. Vis. 12, 949–960.

Chan, T.F., Vese, L.A., 2001. Active contours without edges. IEEE Trans. Image Process. 10 (2), 266–277.

Chang, H., Han, J., Spellman, P.T., Parvin, B., 2012. Multireference level set for the characterization of nuclear morphology in glioblastoma multiforme. IEEE Trans. Biomed. Eng. 59 (12), 3460–3467.

Chen, Y., Yang, J., Yang, M., 2015. Extracting image regions by structured edge prediction. In: 2015 IEEE Winter Conference on Applications of Computer Vision (WACV), pp. 1060–1067.

Cireşan, D.C., Giusti, A., Gambardella, L.M., Schmidhuber, J., 2013. Mitosis detection in breast cancer histology images with deep neural networks. In: Medical Image Computing and Computer-Assisted Intervention—MICCAI 2013, vol. 8150, pp. 411–418.

Comaniciu, D., Meer, P., 2002. Mean shift: a robust approach toward feature space analysis. IEEE Trans. Pattern Anal. Mach. Intell. 24 (5), 603–619.

Cootes, T.F., Taylor, C.J., Cooper, D.H., Graham, J., 1995. Active shape models—their training and application. Comput. Vis. Image Und. 61 (1), 38–59.

Cortes, C., Vapnik, V., 1995. Support-vector networks. Mach. Learn. 20 (3), 273–297.

Dollar, P., Zitnick, C.L., 2014. Fast edge detection using structured forests, pp. 1–12. arXiv 1406.5549 [cs.CV].

Efron, B., Hastie, T., Johnstone, I., Tibshirani, R., 2004. Least angle regression. Ann. Stat. 32 (2), 407–499.

Elhamifar, E., Sapiro, G., Vidal, R., 2012. See all by looking at a few: sparse modeling for finding representative objects. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1600–1607.

Eliceiri, K.W., Berthold, M.R., Goldberg, I.G., Ibanez, L., Manjunath, B.S., Martone, M.E., Murphy, R.F., Peng, H., Plant, A.L., Roysam, B., Stuurman, N., Swedlow, J.R., Tomancak, P., Carpenter, A.E., 2012. Biological imaging software tools. Nat. Methods 9 (7), 697–710.

Farabet, C., Couprie, C., Najman, L., LeCun, Y., 2013. Learning hierarchical features for scene labeling. IEEE Trans. Pattern Anal. Mach. Intell. 35 (8), 1915–1929.

Felzenszwalb, P.F., Huttenlocher, D.P., 2004. Efficient graph-based image segmentation. Int. J. Comput. Vis. 59 (2), 167–181.

Fitzgibbon, A.W., Pilu, M., Fisher, R.B., 1999. Direct least-squares fitting of ellipses. IEEE Trans. Pattern Anal. Mach. Intell. 21 (5), 476–480.

Foran, D.J., Yang, L., Chen, W., Hu, J., Goodell, L.A., Reiss, M., Wang, F., Kurc, T., Pan, T., Sharma, A., Saltz, J.H., 2011. Imageminer: a software system for comparative analysis of tissue microarrays using content-based image retrieval, high-performance computing, and grid technology. J. Am. Med. Inform. Assoc. 18 (4), 403–415.

Grady, L., Schwartz, E., 2006. Isoperimetric graph partitioning for image segmentation. IEEE Trans. Pattern Anal. Mach. Intell. 28 (3), 469–475.

Hariharan, H., Koschan, A., Abidi, M., 2007. Multifocus image fusion by establishing focal connectivity. In: IEEE International Conference on Image Processing—ICIP 2007, vol. 3, pp. 321–324.

Hastie, T., Tibshirani, R., Friedman, J., 2009. The Elements of Statistical Learning: Data Mining, Inference, and Prediction, second ed. Springer, New York.

Hinton, G.E., Salakhutdinov, R.R., 2006. Reducing the dimensionality of data with neural networks. Science 313 (5786), 504–507.

Hinton, G.E., Srivastava, N., Krizhevsky, A., Sutskever, I., Salakhutdinov, R., 2012. Improving neural networks by preventing coadaptation of feature detectors, pp. 1–18. arXiv 1207.0580 [cs.NE].

Janssens, T., Antanas, L., Derde, S., Vanhorebeek, I., den Berghe, G.V., Grandas, F.G., 2013. Charisma: an integrated approach to automatic H&E-stained skeletal muscle cell segmentation using supervised learning and novel robust clump splitting. Med. Image Anal. 17 (8), 1206–1219.

Jiang, Z., Lin, Z., Davis, L.S., 2011. Learning a discriminative dictionary for sparse coding via label consistent k-SVD. In: 2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1697–1704.

Jones, T.R., Carpenter, A.E., Golland, P., 2005. Voronoi-based segmentation of cells on image manifolds. In: Computer Vision for Biomedical Image Applications (CVBIA), vol. 3765, pp. 535–543.

Kong, H., Gurcan, M., Belkacem-Boussaid, K., 2011. Partitioning histopathological images: an integrated framework for supervised color-texture segmentation and cell splitting. IEEE Trans. Med. Imaging 30 (9), 1661–1677.

Kontschieder, P., Rota Bulo, S., Bischof, H., Pelillo, M., 2011. Structured class-labels in random forests for semantic image labelling. In: 2011 IEEE International Conference on Computer Vision (ICCV), pp. 2190–2197.

Kourou, K., Exarchos, T.P., Exarchos, K.P., Karamouzis, M.V., Fotiadis, D.I., 2015. Machine learning applications in cancer prognosis and prediction. Comput. Struct. Biotech. J. 13, 8–17.

Krizhevsky, A., Sutskever, I., Hinton, G.E., 2012. Imagenet classification with deep convolutional neural networks. Adv. Neural Inform. Process. Syst. 25, 1097–1105.

Kuhl, F.P., Giardina, C.R., 1982. Elliptic Fourier features of a closed contour. Comput. Graph. Image Process. 18 (3), 236–258.

LeCun, Y., Bottou, L., Bengio, Y., Haffner, P., 1998. Gradient-based learning applied to document recognition. Proc. IEEE 86 (11), 2278–2324.

LeCun, Y., Kavukcuoglu, K., Farabet, C., 2010. Convolutional networks and applications in vision. In: Proceedings of 2010 IEEE International Symposium on Circuits and Systems (ISCAS), pp. 253–256.

LeCun, Y., Bengio, Y., Hinton, G., 2015. Deep learning. Nature 521 (28), 436–444.

Liao, S., Gao, Y., Oto, A., Shen, D., 2013. Representation learning: a unified deep learning framework for automatic prostate MR segmentation. In: Medical Image Computing and Computer-Assisted Intervention—MICCAI 2013, vol. 8150, pp. 254–261.

Lin, G., Adiga, U., Olson, K., Guzowski, J.F., Barnes, C.A., Roysam, B., 2003. A hybrid 3d watershed algorithm incorporating gradient cues and object models for automatic segmentation of nuclei in confocal image stacks. Cytometry A 56A (1), 23–36.

Ling, H., Jacobs, D.W., 2007. Shape classification using the inner-distance. IEEE Trans. Pattern Anal. Mach. Intell. 29 (2), 286–299.

Liu, B., Huang, J., Kulikowski, C., Yang, L., 2013. Robust visual tracking using local sparse appearance model and k-selection. IEEE Trans. Pattern Anal. Mach. Intell. 35 (12), 2968–2981.

Liu, F., Xing, F., Zhang, Z., Mcgough, M., Yang, L., 2015. Robust muscle cell quantification using structured edge detection and hierarchical segmentation. In: Medical Image Computing and Computer-Assisted Intervention—MICCAI 2015, vol. 9351, pp. 324–331.

Lou, X., Koethe, U., Wittbrodt, J., Hamprecht, F., 2012. Learning to segment dense cell nuclei with shape prior. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1012–1018.

Mairal, J., Bach, F., Ponce, J., Sapiro, G., Zisserman, A., 2008. Discriminative learned dictionaries for local image analysis. In: 2008 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1–8.

Matei, B., Meer, P., 2000. A general method for errors-in-variables problems in computer vision. In: 2000 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), vol. 2, pp. 18–25.

Mori, G., 2005. Guiding model search using segmentation. In: 2005 IEEE International Conference on Computer Vision (ICCV), vol. 2, pp. 1417–1423.

Mualla, F., Scholl, S., Sommerfeldt, B., Maier, A., Hornegger, J., 2013. Automatic cell detection in bright-field microscope images using sift, random forests, and hierarchical clustering. IEEE Trans. Med. Imaging 32 (12), 2274–2286.

Murphy, K.P., 2012. Machine Learning: A Probabilistic Perspective, first ed. MIT Press, Cambridge, MA.

Otsu, N., 1979. A threshold selection method from gray-level histograms. IEEE Trans. Syst. Man Cybern. 9 (1), 62–66.

Parvin, B., Yang, Q., Han, J., Chang, H., Rydberg, B., Barcellos-Hoff, M.H., 2007. Iterative voting for inference of structural saliency and characterization of subcellular events. IEEE Trans. Image Process. 16 (3), 615–623.

Qi, X., Xing, F., Foran, D.J., Yang, L., 2012. Robust segmentation of overlapping cells in histopathology specimens using parallel seed detection and repulsive level set. IEEE Trans. Biomed. Eng. 59 (3), 754–765.

Quelhas, P., Marcuzzo, M., MendoncÌğa, A., Campilho, A., 2010. Cell nuclei and cytoplasm joint segmentation using the sliding band filter. IEEE Trans. Med. Imaging 29 (8), 1463–1473.

Rubinstein, R., Bruckstein, A.M., Elad, M., 2010. Dictionaries for sparse representation modeling. Proc. IEEE 98 (6), 1045–1057.

Scholkopf, B., Smola, A.J., 2001. Learning With Kernels: Support Vector Machines, Regularization, Optimization, and Beyond. MIT Press, Cambridge, MA.

Scott, D.W., 2001. Parametric statistical modeling by minimum integrated squared error. Technometrics 43, 274–285.

Sommer, C., Gerlich, D.W., 2013. Machine learning in cell biology—teaching computers to recognize phenotypes. J. Cell Sci. 126 (24), 5529–5539.

Su, H., Xing, F., Lee, J.D., Peterson, C.A., Yang, L., 2013. Learning based automatic detection of myonuclei in isolated single skeletal muscle fibers using multi-focus image fusion. In: 2013 IEEE 10th International Symposium on Biomedical Imaging (ISBI), pp. 432–435.

Su, H., Xing, F., Lee, J.D., Peterson, C.A., Yang, L., 2014. Automatic myonuclear detection in isolated single muscle fibers using robust ellipse fitting and sparse representation. IEEE/ACM Trans. Comput. Biol. Bioinform. 11 (4), 714–726.

Szegedy, C., Toshev, A., Erhan, D., 2013. Deep neural networks for object detection. Adv. Neural Inform. Process. Syst. 26, 2553–2561.

Tarca, A.L., Carey, V.J., Chen, X.W., R., R., Drăghici, S., 2007. Machine learning and its applications to biology. PLoS Comput. Biol. 3 (6), e116.

Taubin, G., 1991. Estimation of planar curves, surfaces, and nonplanar space curves defined by implicit equations with applications to edge and range image segmentation. IEEE Trans. Pattern Anal. Mach. Intell. 13 (11), 1115–1138.

Tibshirani, R., 1994. Regression shrinkage and selection via the lasso. J. R. Stat. Soc. B 58, 267–288.

Tropp, J.A., 2004. Greed is good: algorithmic results for sparse approximation. IEEE Trans. Inform. Theory 50 (10), 2231–2242.

Uzunbaş, M., Chen, C., Metaxsas, D., 2014. Optree: a learning-based adaptive watershed algorithm for neuron segmentation. In: Medical Image Computing and Computer-Assisted Intervention—MICCAI 2014, vol. 8673, pp. 97–105.

Wählby, C., Lindblad, J., Vondrus, M., Bengtsson, E., Björkesten, L., 2002. Algorithms for cytoplasm segmentation of fluorescence labelled cells. Anal. Cell. Pathol. 24 (2), 101–111.

Waljee, A.K., Higgins, P.D., 2010. Machine learning in medicine: a primer for physicians. Am. J. Gastroenterol. 105 (6), 1224–1226.

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

Wernick, M.N., Yang, Y., Brankov, J.G., Yourganov, G., Strother, S.C., 2010. Machine learning in medical imaging. IEEE Signal Process. Mag. 27 (4), 25–38.

Wu, G., Kim, M., Wang, Q., Gao, Y., Liao, S., Shen, D., 2013. Unsupervised deep feature learning for deformable registration of MR brain images. In: Medical Image Computing and Computer-Assisted Intervention—MICCAI 2013, vol. 8150, pp. 649–656.

Xie, Y., Xing, F., Kong, X., Su, H., Yang, L., 2015. Beyond classification: structured regression for robust cell detection using convolutional neural network. In: Medical Image Computing and Computer-Assisted Intervention—MICCAI 2015, vol. 9351, pp. 358–365.

Xing, F., Yang, L., 2013. Robust selection-based sparse shape model for lung cancer image segmentation. In: Medical Image Computing and Computer-Assisted Intervention—MICCAI 2013, vol. 8151, pp. 404–412.

Xing, F., Su, H., Neltner, J., Yang, L., 2014. Automatic ki-67 counting using robust cell detection and online dictionary learning. IEEE Trans. Biomed. Eng. 61 (3), 859–870.

Xing, F., Xie, Y., Yang, L., 2016. An automatic learning-based framework for robust nucleus segmentation. IEEE Trans. Med. Imaging 35 (2), 550–566.

Yan, P., Zhou, X., Shah, M., Wong, S.T.C., 2008. Automatic segmentation of high-throughput RNAi fluorescent cellular images. IEEE Trans. Inform. Tech. Biomed. 12 (1), 109–117.

Yang, L., Tuzel, O., Meer, P., Foran, D.J., 2008. Automatic image analysis of histopathology specimens using concave vertex graph. In: Medical Image Computing and Computer-Assisted Intervention—MICCAI 2008, vol. 5241, pp. 833–841.

Zhang, Z., 1997. Parameter estimation techniques: a tutorial with application to conic fitting. Image Vis. Comput. 15 (1), 59–76.

Zhang, Q., Li, B., 2010. Discriminative k-SVD for dictionary learning in face recognition. In: 2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2691–2698.

Zhang, S., Zhan, Y., Dewan, M., Huang, J., Metaxas, D.N., Zhou, X.S., 2011. Deformable segmentation via sparse shape representation. In: Medical Image Computing and Computer-Assisted Intervention—MICCAI 2011, pp. 451–458.

Zhang, S., Zhan, Y., Metaxas, D.N., 2012. Deformable segmentation via sparse shape representation and dictionary learning. Med. Image Anal. 16 (7), 1385–1396.

Zhang, C., Yarkony, J., Hamprecht, F.A., 2014. Cell detection and segmentation using correlation clustering. In: Medical Image Computing and Computer-Assisted Intervention—MICCAI 2014, vol. 8673, pp. 9–16.

Zhou, X., Liu, K.Y., Bradley, P., Perrimon, N., Wong, S.T.C., 2005. Towards automated cellular image segmentation for RNAI genome-wide screening. In: Medical Image Computing and Computer-Assisted Intervention—MICCAI 2005, vol. 3749, pp. 885–892.

Zhou, X., Huang, X., Duncan, J.S., Yu, W., 2013. Active contour with group similarity. In: 2013 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2969–2976.

Zimmer, C., Olivo-Marin, J.C., 2005. Coupled parametric active contours. IEEE Trans. Pattern Anal. Mach. Intell. 27 (11), 1838–1842.