



ORIGINAL ARTICLE

Cell segmentation in histopathological images with deep learning algorithms by utilizing spatial relationships

Nuh Hatipoglu^{2,3} · Gokhan Bilgin^{1,2}

Received: 27 July 2016 / Accepted: 13 February 2017 / Published online: 28 February 2017
© International Federation for Medical and Biological Engineering 2017

Abstract In many computerized methods for cell detection, segmentation, and classification in digital histopathology that have recently emerged, the task of cell segmentation remains a chief problem for image processing in designing computer-aided diagnosis (CAD) systems. In research and diagnostic studies on cancer, pathologists can use CAD systems as second readers to analyze high-resolution histopathological images. Since cell detection and segmentation are critical for cancer grade assessments, cellular and extracellular structures should primarily be extracted from histopathological images. In response, we sought to identify a useful cell segmentation approach with histopathological images that uses not only prominent deep learning algorithms (i.e., convolutional neural networks, stacked autoencoders, and deep belief networks), but also spatial relationships, information of which is critical for achieving better cell segmentation results. To that end, we collected cellular and extracellular samples from histopathological images by windowing in small patches with various sizes. In experiments, the segmentation accuracies of the methods used improved as the window sizes increased due to the addition of local spatial and contextual information. Once we compared the effects of training sample size and influence of window size, results revealed that the deep learning algorithms, especially

convolutional neural networks and partly stacked autoencoders, performed better than conventional methods in cell segmentation.

Keywords Histopathological images · Deep learning algorithms · Computer-aided diagnosis systems · Segmentation · Spatial relationships

1 Introduction

Although deep learning has been an active field of research in machine learning, its application with histopathological images is relatively new [27], including with images of cancer. In medical literature, *cancer* is a general term for diseases caused by the uncontrolled proliferation of abnormal cells in a certain part of the body and its carcinomatosis in other parts. By definition, cancer thus does not represent a specific disease, but various disorders manifested by excessive cell proliferation and tissue growth [21].

To better diagnose cancerous diseases in tissues and organs, specimens are collected and examined as part of histopathological investigation, particularly because the early detection of cancer and determination of its developmental stage (i.e., grading) are crucial for initiating an appropriate treatment regimen [55]. Traditionally, specimens taken from patients as serial histological sections are first stained with various specific chemical dyes, including hematoxylin and eosin (H&E), as well as immunohistochemical stains [13], after which pathologists examine cell structures (i.e., morphological features) and cellular distributions (i.e., placement of cells) under a microscope to evaluate the state and stage of cancer. For pathologists, though visualizing histopathological specimens is a current primary diagnostic approach, it is quite a time-consuming

✉ Gokhan Bilgin
gbilgin@yildiz.edu.tr

¹ Department of Computer Engineering, Yildiz Technical University (YTU), 34220 Istanbul, Turkey

² Signal and Image Processing Lab. (SIMPLAB), YTU, Istanbul, Turkey

³ Graduate School of Natural and Applied Science, Yildiz Technical University, 34220 Istanbul, Turkey

process that requires intense concentration. Even given time and focus, the accuracy of any final decision depends heavily on the pathologists' experience. Indeed, in borderline cases, intra- and interobservation variations of pathologists do occur [29].

In response to that problem, the development and use of digital imaging as a second reader system in pathology have enabled rapid, consistent solutions that manual analysis cannot [54]. With high-resolution histopathological images, computer-aided diagnosis (CAD) systems can extract necessary features for classification and grading tasks to be completed by using a variety of techniques [21]. Consequently, CAD systems for histopathological images have become widely popular, especially for analyzing complex cases and massive volumes of medical data [53].

From the perspective of computer science, researchers have discovered several areas of scientific study by using pattern recognition and machine learning algorithms as well as signal, image processing, and statistical analysis for CAD systems [27, 29]. In histopathological image analysis for cancer diagnosis and grading, in which pathologists seek to define the regularity of cell borders, shapes, and distributions with digitized slides [54], cell segmentation early on during analysis has become fundamental to deciding whether tissue regions are cancerous. If the tissues indeed have cancer, then assessing malignancy and risk becomes critical in the design of CAD systems [53]. For that process, cells of interest on a histopathological slide are extracted from the background during the cell segmentation task. However, the many factors affecting the segmentation task—overlapping or touching cells, noise artifacts on cell and nuclei contours, and blurring due to zooming during digitization—all adversely affect the proper extraction of cellular regions by conventional image segmentation [24]. Ideally, after accurate segmentation, cell borders and the distribution of cellular regions and other structures (e.g., nuclei, cytoplasm, stroma, red blood cells, and connective parts) should be in focus for further recognition, grading, and classification of cancer [15].

In recent years, important advances have transformed quantitative histopathological assessments for the

diagnosis, grading, and classification of carcinomatous diseases. As biomedical imaging literature has reported, the first step in any histopathological CAD system is to precisely detect cellular structures. Afterward, various cell identifiers such as cell size, distribution, shape, and cellular morphologies are defined in histopathological images in order to build a robust segmentation model [13, 51, 58]. In general, segmentation can be performed by supervised or unsupervised pattern recognition and machine learning methods [11]. Some cellular information extracted from cellular shapes and distribution as well as from color histograms of cells has been used for histopathological image segmentation [31, 39]. Various mathematical models have also been developed that use machine learning algorithms for cell segmentation [3, 56]. A general framework has even been proposed for cancer diagnosis by watershed-based cell segmentation using high-resolution histopathological images [15], and for segmenting histopathological cervical cell images in particular, an unsupervised machine learning approach has been used [18]. Among other advances, the automated labeling of cancer textures has been studied for colorectal histopathological slides by using quasi-supervised learning [44], and spatial relations in the segmentation of histopathological images have been investigated using Haralick features [2]. Various textural and morphological attributes have been used by way of support vector machines (SVMs) [41, 43], and the classification of histopathological images by using k-nearest neighbors (*k*-NN) has been addressed [30]. As ensemble-based feature selection and classification of leukocytes, random forests (RF) were introduced in [52], and cell segmentation in breast tissue has been studied with deep learning algorithms [48, 57]. The bag-of-features approach for visual pattern analysis in histopathological images has also been proposed [10], and for cell detection and segmentation using breast, liver, bone marrow, intestinal mucosa and kidney tissues, a novel contour-based minimum-model approach has been introduced [55]. Among other studies, the early diagnosis of breast cancer using mammographic and histopathological images has been evaluated [32]. For a useful reference, the current status and future potential of digital pathology have

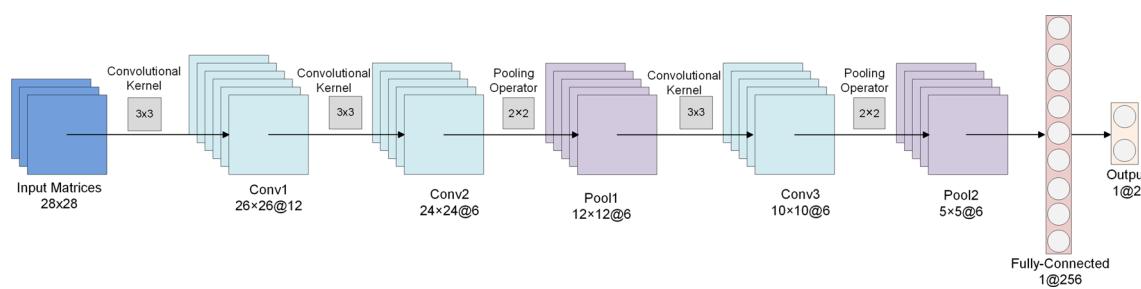


Fig. 1 Illustration of CNN architecture proposed in the study

been reviewed by addressing the issues of nuclei detection, segmentation, and classification in particular [27].

In preliminary versions of our study [22, 23], we presented only convolutional neural network (CNN)-based segmentation with a benchmark breast cancer data set. In this paper, however, we have substantially extended those studies by using two other prominent deep learning methods, namely deep belief networks (DBNs) and stacked autoencoders (SAEs) and adding a new benchmark a renal clear cell carcinoma data set to strengthen the proof of our findings. Overall, we proposed cell segmentation in histopathological images that use deep learning algorithms combined with information about spatial relationships. In related literature, it is difficult to find a benchmark histopathological image data set with cellular ground-truth information for cancer detection and classification tasks. Furthermore, each study has used different image data sets and metrics in evaluating their results. For those reasons, we used the University of California, Santa Barbara (UCSB), biosegmentation benchmark breast cancer data set obtained from David Rimm Laboratory at Yale University [17] and Harvard University's Beck Laboratory benchmark kidney renal clear cell carcinoma data set extracted from the cancer genome atlas portal [26], both of which also provide cellular ground-truth information. For spatial information about cellular and extracellular parts, we collected samples by creating a window structure to contain the neighbors of a pixel from all images in the data set. To evaluate the effects of training sample size, we randomly selected various numbers of cellular and extracellular samples from images in data sets to obtain four differently

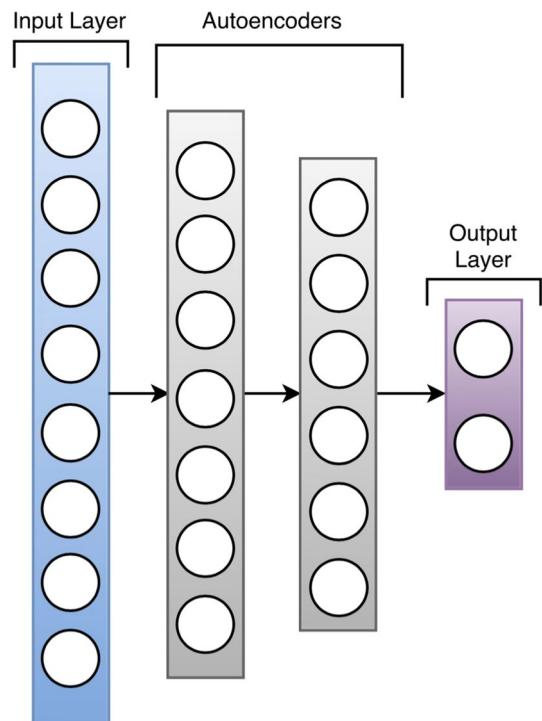


Fig. 3 A general diagram of SAE

sized training sets. Moreover, we evaluated various window sizes to discern the effects of spatial relations for cellular and extracellular pixel samples. Because we employed supervised techniques for segmentation, we compared three different deep learning methods CNN, DBN, and SAE and two prominent conventional machine learning methods: SVM and RF.

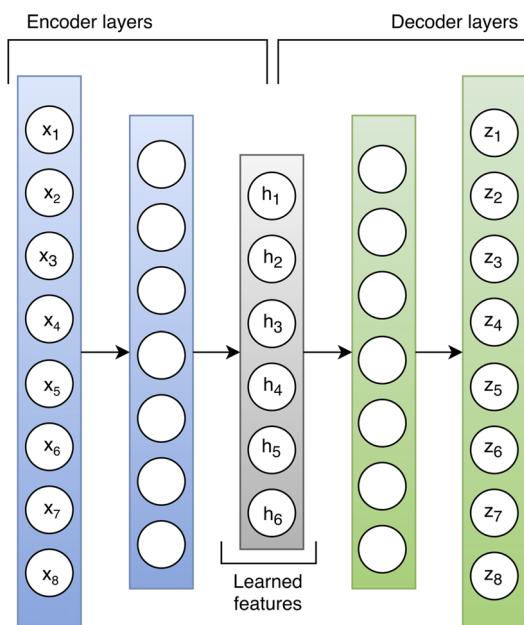


Fig. 2 A general diagram of AE

2 Methods

Deep learning is a highly promising technique in machine learning that chiefly aims to identify the optimal representation of data with nonlinear methods based on neural network architectures and energy-based models [14, 37]. At base, the architectures of those models consist of several neural layers and hidden variables, and all techniques described as deep learning extract the hidden features of the data hierarchically. At the same time, the learned hidden features are invariant to most local changes of the input data. Briefly, deep learning algorithms include CNN, SAE, and DBN and provide a nonlinear transformation of data compared to shallow structures (e.g., Gaussian mixture models, hidden Markov models, and conditional random fields). Consequently, they can model complex relationships in data with multilayered network structures. In terms of usage, deep learning algorithms can be divided

into three classes: probabilistic generative models, discriminative (i.e., separation) models, and hybrid models that combine the first two classes. The generative deep architectures enable an understanding of high-degree correlations of features in observed data for the purposes of pattern analysis and synthesis. Furthermore, they characterize the joint probability distributions with related classes. Deep learning architectures focus on identifying distinctive features for pattern classification tasks, for which a supervised training process is performed naturally by labeled data. Discriminative deep architectures aim to increase the discriminative power of pattern classification properties using the posterior distributions of classes on original input space. In hybrid structures, used to obtain optimal, regular results in applications, the training of layers is performed in an unsupervised manner, with the exception of the training of the last layer, which is conducted in a supervised way using the label information of the data [14]. In our study, we compared the effectiveness of deep learning algorithms CNN, SAE, and DBN for histopathological image segmentation.

2.1 Convolutional neural networks (CNNs)

CNNs are feed-forward, back-propagation neural networks with a special architecture inspired by the human visual system that have achieved success in image, video, audio, and artificial intelligence applications. First proposed by Fukushima in the early 1980s [16] and improved upon by LeCun et al. in the 1990s [12, 35], CNN has been used in various applications, including handwriting classification [8], image classification [33], object identification [45], and the determination of mitotic cells [9].

Unlike other deep learning algorithms that accept vectorial input, the CNN structure uses two-dimensional input data. It consists of convolution and subsampling or pooling layers; in

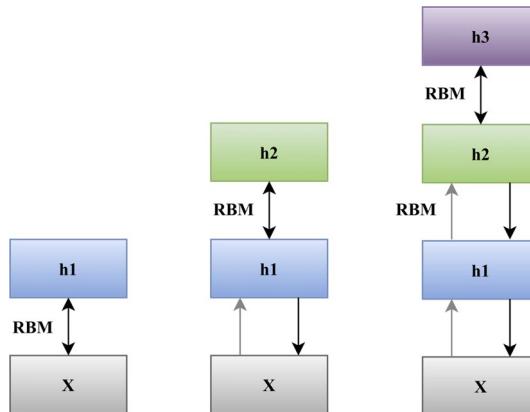


Fig. 4 General diagram of DBN

the former, according to the type of convolution kernel, different feature maps can be extracted. It is possible to use the same kernels in each convolution layer or differently sized kernels. In the latter, the representation of feature maps and down-sampling operations are performed [36]. Figure 1 illustrates the CNN architecture proposed in our study.

In mathematical terms, l is the number of layers in neural network, $f_j(\cdot)$ is the activation function of j th layer, and \mathbf{a}_j^l is the j th feature vector of the l th layer. As such, \mathbf{a}_j^l can be calculated according to Eq. 1:

$$\mathbf{a}_j^l = f_j \left(\sum_{i \in M_j^l} k_{ij}^l \otimes \mathbf{a}_i^{l-1} + b_j^l \right) \quad (1)$$

where \otimes is the two-dimensional convolution operator, b_j^l is the coefficient of the j th feature map, k_{ij}^l is the convolution kernel from i th feature map in the $(l-1)$ th layer to the j th feature map in the l th layer, and M_j^l is the list of features at the $(l-1)$ th layer which is connected to the j th feature map.

In subsampling layer, the down-sampling operation is performed first. The j th feature map is decomposed into 2×2 pixel blocks in the $(l-1)$ th layer. The Z_j^{l-1} matrix is obtained by totaling the four pixels in each block. The calculation of $Z_j^{l-1}(m, n)$ is shown in Eq. 2:

$$Z_j^{l-1}(m, n) = \mathbf{a}_j^{l-1}(2m-1, 2n-1) + \mathbf{a}_j^{l-1}(2m-1, 2n) \\ + \mathbf{a}_j^{l-1}(2m, 2n-1) + \mathbf{a}_j^{l-1}(2m, 2n) \quad (2)$$

The deduction of the j th feature map for subsampling layer l is included in Eq. 3, in which w_i^l represents weights and b_j^l represents bias coefficients of the l th layer.

$$\mathbf{a}_j^l = f_j(Z_j^{l-1} \times w_i^l + b_j^l) \quad (3)$$

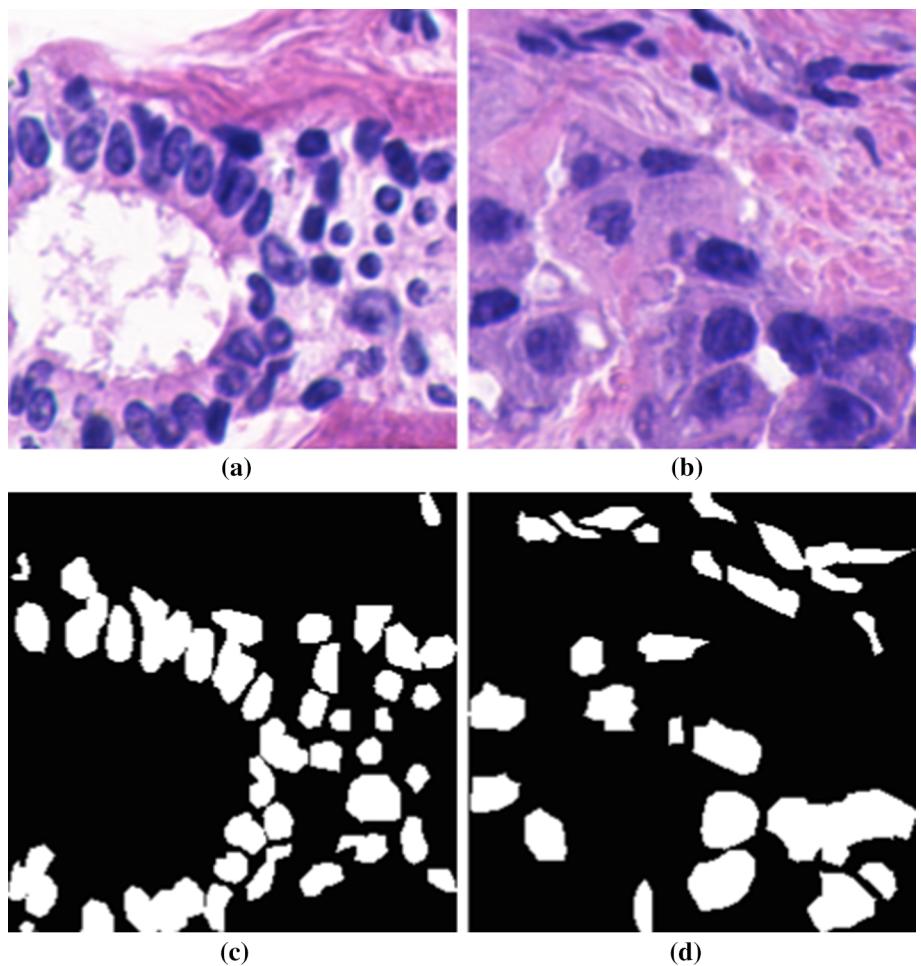
The final stage of CNN is the output layer, which is fully connected to the previous convolution layer. Equation 4 shows the calculation of the output layer of the j th sigmoid neuron, in which L represents the output layer, N^L represents number of sigmoid neurons, $w_{i,j}^L$ shows the weights from feature vector i to the last convolution layer, b_j^L denotes the coefficient of the j th neuron in layer L , and f^L is the sigmoid transfer function.

$$\mathbf{a}_j^L = f^L \left(\sum_{i=1}^{N_{L-1}} w_{i,j}^L \mathbf{a}_i^{L-1} + b_j^L \right) \quad (4)$$

CNN produces a system output in vectorial form ($1 \times N_L$) in the output layer for each input datum [20, 49], which can be expressed as:

$$\mathbf{y} = [y_1^L, y_2^L, \dots, y_{N_L}^L] \quad (5)$$

Fig. 5 Example images of UCSB histopathological data set used in the experiments: **a** benign image sample, **b** malignant image sample, **c** ground truth of the benign image, **d** ground truth of the malignant image



2.2 Stacked autoencoders (SAEs)

Autoencoders (AEs) are discriminative graphical models that seek to reconstruct their input data. AEs (Fig. 2) use unsupervised feed-forward, back-propagation methods to estimate input neurons from output neurons. Stacking the input layer and hidden layers of AEs together layer by layer forms an SAE (Fig. 3) [5]. The most important property of AE architecture is its ability to extract hidden structural features embedded in data by changing the number of units in the hidden layer. A wide hidden layer structure can learn some very interesting features by using a sparsity constraint on a hidden layer [1, 20, 46].

Briefly, AEs perform reconstruction of z with input vector x and encode it to hidden layer h . Afterward, encoded data in hidden layer are decoded and restructured, as expressed mathematically in Eq. 6:

$$\begin{aligned} h(x) &= \text{sigm}(W_1x + b_1) \\ z(x) &= \text{sigm}(W_2h(x) + b_2) \end{aligned} \quad (6)$$

where sigm is the sigmoid activation function, W_1 and W_2 are scalar weight between layers, and b is the bias of those layers.

AEs seek to learn the parameter and identity function of hypothesis ($h_\theta(x) \approx x$). The main idea in the training model of AEs is to minimize the mean difference between input x and reconstruction z . The smallest difference, θ , between x and z can be found by using Eq. 7:

$$\theta = \arg \min_{\theta} = \frac{1}{N} \sum_{i=1}^N \Delta(x^{(i)}, z(x^{(i)})) \quad (7)$$

where N represents the number of samples in the training set, Δ shows the difference between x and z data ($\Delta(x, z) = \|x - z\|^2$). The feed-forward, back-propagation algorithm is used in updating the parameters, and the sparsity algorithm is used for hidden unit activations. For training, an additional penalty component is added to the penalty function to provide low-level activation in hidden layers (i.e., sparsity) [42], as shown in Eq. 8:

$$\phi(x, z) = \|x - z\|^2 + \beta \|h - \rho\| \quad (8)$$

where β indicates the sparsity penalty term and ρ is the sparsity parameter close to zero [19, 50].

2.3 Deep belief networks (DBNs)

DBNs consist of a combination of multiple layers of a restricted Boltzmann machine (RBM). Especially in classification tasks, DBNs exhibit superior performance due to having RBM structures in their architecture. Any RBM is commonly trained in a greedy layer-wise fashion and is used in constructing DBNs. An RBM has two layers in its network structure: visible and hidden layers. Briefly, the layers of DBNs are formed with RBMs connected to each other in a cascaded fashion [46]. The two most important features of DBNs are that:

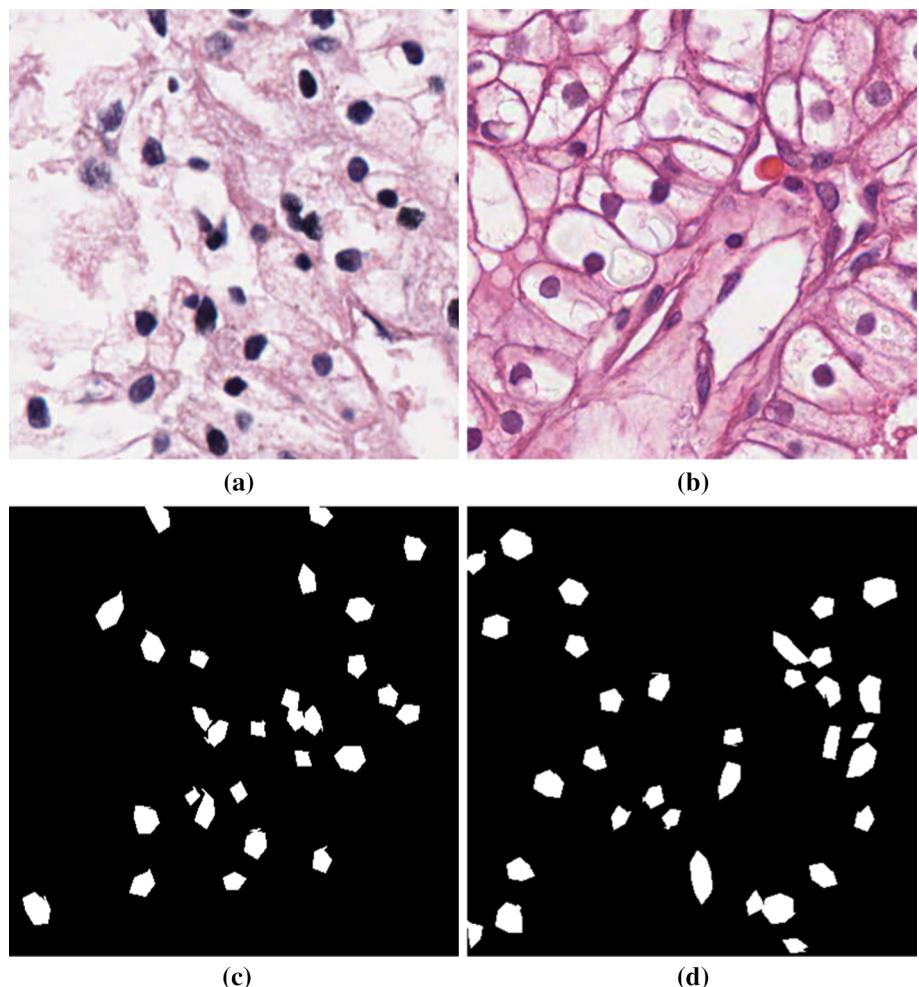
- (i) Learning in the training phase occurs sequentially in successive layers. After the first RBM is trained, the output vector generated in the hidden layer of the RBM by its training data is given to the next RBM as training data, and the same rule applies for subsequent layers.

- (ii) In the reconstruction phase, data flow are provided via the last output vector in the last layer from the last RBM to the first RBM, and the reconstruction vector is observed in the visible layer of the first RBM (Fig. 4).

In the training step of DBN, comparative divergence is applied for each RBM, which then tries to attain thermal equilibrium. Hence, more efficient results can be obtained by using a conventional feed-forward, back-propagation algorithm [25]. The bottom layer represented by x is the visible layer, and the top layer represented by h is the hidden layer. Since units in the visible and hidden layers take binary values, input data are expressed according to a binary system $x_i \in \{0, 1\}^p, h_j \in \{0, 1\}^q$. The scalar weight that allows a connection between visible and hidden units is W . The energy functions are shown in Eqs. 9 and 10, and the Z partition function can be defined as in Eq. 11:

$$E(x, h; \phi) = - \sum_{i=1}^p b_i x_i - \sum_{j=1}^q a_j h_j - \sum_{i=1}^p \sum_{j=1}^q x_i W_{i,j} h_j \quad (9)$$

Fig. 6 Example images of Harvard histopathological data set used in the experiments: **a–b** original image samples, **c–d** ground truths of selected samples, respectively



$$E(\mathbf{x}, \mathbf{h}; \phi) = -\mathbf{b}^T \mathbf{x} - \mathbf{a}^T \mathbf{h} - \mathbf{x}^T \mathbf{W} \mathbf{h} \quad (10)$$

$$Z(\phi) = \sum_{\mathbf{x}, \mathbf{h}} E(\mathbf{x}, \mathbf{h}; \phi) \quad (11)$$

where $\phi = \{b_i, a_j, w_{ij}\}$, in which b_i and a_j are bias terms of visible and hidden layers, and w_{ij} is the weight between visible layer i and hidden layer j .

The probability distribution function belonging to this model is presented in Eq. 12:

$$P(\mathbf{x}, \mathbf{h}; \phi) = \frac{e^{-E(\mathbf{x}, \mathbf{h}; \phi)}}{Z} \quad (12)$$

The conditional probability density function in representing the hidden layer is shown in Eq. 13:

$$P(\mathbf{h}|\mathbf{x}; \phi) = \frac{P(\mathbf{x}|\mathbf{h}; \phi)}{P(\mathbf{h}; \phi)} \quad (13)$$

The probability value of a sample used in a training phase is provided with updated weights and bias values, due to which the energy of some samples is reduced or increased. The contribution to the divisor function in Eq. 14 is especially higher for samples with low energy. The derivative of log likelihood with respect to weights of a training vector is calculated according to Eq. 14:

$$\frac{\partial \log P(\mathbf{x}; \phi)}{\partial W_{ij}} = \langle x_i h_j \rangle_{\text{data}} - \langle x_i h_j \rangle_{\text{reconstruction}} \quad (14)$$

The stochastic steepest descent parameter update method can be used with the obtained results depending on the logarithmic probability of the training set in the RBMs.

$$\begin{aligned} w_{ij} &= \varepsilon (\langle x_i h_j \rangle_{\text{data}} - \langle x_i h_j \rangle_{\text{reconstruction}}) \\ a_i &= \varepsilon (\langle x_i \rangle_{\text{data}} - \langle x_i \rangle_{\text{reconstruction}}) \\ b_j &= \varepsilon (\langle h_j \rangle_{\text{data}} - \langle h_j \rangle_{\text{reconstruction}}) \end{aligned} \quad (15)$$

where ε is the learning coefficient, and the values of x_i and h_j in the RBMs are calculated with conditional probabilities. In case of the random selection of x , the probability of h_j when equals to 1 is calculated by Eq. 16:

$$(h_i = 1 | \mathbf{x}) = \text{sigm}(c_i + W_i \mathbf{x}) \quad (16)$$

where the sigm is the sigmoid function. The probability of any h_i value, which relates to the hidden vector, can be calculated similarly while it equals to 1 [25].

$$P(x_i = 1 | \mathbf{h}) = \text{sigm}(b_i + W_i \mathbf{h}) \quad (17)$$

3 Experimental studies

In the experimental part of our study, we compared three prominent deep learning methods (i.e., CNN, SAE, and

DBN) with the well-known machine learning methods SVM and RF in terms of histopathological cell segmentation. Since the cellular structures are spatially coherent in local regions of images [21, 27], we also took the incorporation of spatial information into account. The proposed approach naturally improved the discrimination of pixels of interest in cellular and extracellular parts of histopathological images. Analogous to an $n \times n$ geometric window that includes the centered pixel and its neighbors, we investigated different window sizes in the field/patch of interest with close spatial relationships and dependencies.

3.1 Histopathological image data

In literature on histopathological image processing, finding a benchmark image data set with cellular ground-truth information ranks among the greatest challenges for researchers. Generally, researchers use their own image data sets, which are not publicly available, and employ different metrics in evaluating their results. In the experimental part of this study, we thus used two benchmark data sets with cellular ground-truth information available for researchers.

The first data set is from David Rimm Laboratory at Yale University and provided by the Center for Bio-Image Informatics at University of California, Santa Barbara (UCSB), and can be downloaded as a part of UCSB's biosegmentation benchmark data set [17]. The UCSB data set includes 58 H&E-stained RGB histopathological breast tissue images: 26 of malignant tissues and 32 of benign tissues from anonymous patients, all sized 896×768 pixels. A part of each image was cropped and labeled by pathologists and formed approximately in the size of 200×200 pixels. Figure 5 shows examples of benign and malignant tissue images from the UCSB data set with accompanying ground truths used in the experiments.

The second data set with cellular ground-truth information is from Beck Laboratory at Harvard University. Original images in data set are obtained from whole slide images (WSIs) of kidney renal clear cell carcinoma in The Cancer Genome Atlas (TCGA) data portal, a large-scale initiative funded by the US National Cancer Institute and US National Human Genome Research Institute. The Harvard data set is obtained from ten WSIs, which represent a range of histological grades of kidney renal clear cell carcinoma for crowdsourcing image annotation for the purpose of nucleus detection and segmentation [26]. Some parts of all WSIs were annotated by a pathologist, and a total of 64 H&E-stained RGB image parts were generated in the size of 400×400 pixels. These images are not classified as benign or malignant. Figure 6 provides example images of the Harvard data set used in the experiments with accompanying ground truths.

Fig. 7 General flowchart of the proposed study

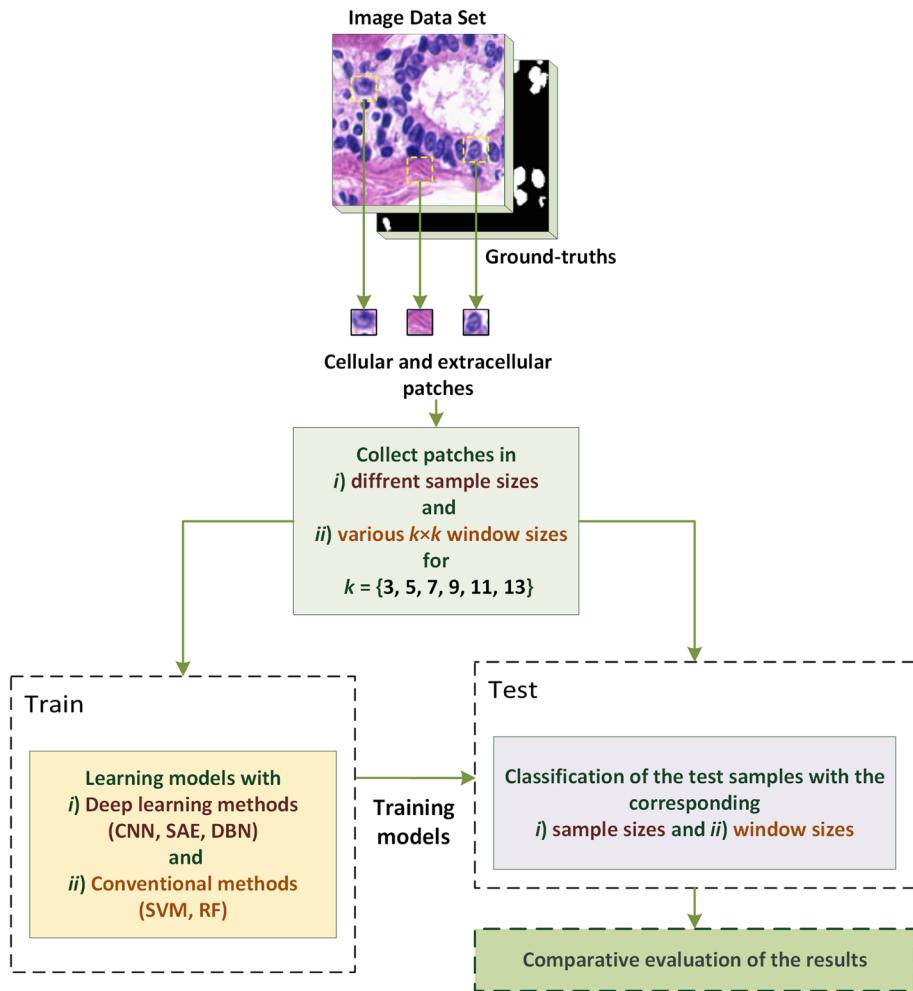


Table 1 The CNN model parameters

Layer	Feature maps	Kernel size
Input	$28 \times 28 \times 3$	3
Conv1	$26 \times 26 \times 12$	3
Conv2	$24 \times 24 \times 6$	3
MaxPool1	$12 \times 12 \times 6$	2
Conv3	$10 \times 10 \times 6$	3
MaxPool2	$5 \times 5 \times 6$	2
Fully connected	256	—
Output	2	—

3.2 Data formation

The general flowchart of the proposed approach is presented in Fig. 7. The processing stages of the flowchart are applied to each data set separately but similarly. As a first step, we collected cellular and extracellular sample patches by creating a window structure to contain the neighbors of a pixel from the each band of the RGB color space in

Table 2 SAE model parameters

Layer	Feature maps
Input	2352
AutoEncoder1	256
AutoEncoder2	128
Output	2

differently sized windows by using supplied ground truths. For training purposes, we collected various numbers of cellular and extracellular samples (i.e., 100, 150, 200, and 250) randomly from each of the 58 histopathological images of the UCSB data set and the 64 histopathological images of the Harvard data set separately. On the other side, we selected test samples three more times from each image (i.e., 300, 450, 600, and 750). Extracellular patches are comprised of intracellular muscular, adipose, red blood cells, and connective tissues. We assumed that the spatial characteristics of cellular and extracellular patches typically have similar statistical distributions according to their

type. We also collected patch samples in variously sized windows in a $k \times k$ format to observe the changing characteristics of the spatial relations with varying window sizes, in which $k = \{3, 5, 7, 9, 11, 13\}$. Ultimately, we compared the effects of training sample size and window size.

In the formation of the training and test sets in the experiments for each data set, we used a 1/4 part of the data set for training, and the remaining 3/4 part for testing. Input training batch data are composed of cellular and extracellular patch samples in a stacked form as *training cubes*. Similar to their formation, test patch samples are also arranged in a stacked form to create *test cubes*. In detail, the proposed training and testing processes are entailed the following:

- (i) The training cube is formed as $k \times k \times n_{tr}$ dimensional data, in which k is the window size equal to $k = \{3, 5, 7, 9, 11, 13\}$ and n_{tr} is the number of training samples. The number of training samples equals the number of cellular and extracellular samples selected from one image multiplied by the number of total images (i.e., 58 for the UCSB data set and 64 for the Harvard data set).
- (ii) An equal number of cellular and extracellular samples are selected randomly from one image (i.e., $n_{cellular} = n_{extracellular} = 100, 150, 200$, and 250 samples), so that $n_{cellular} + n_{extracellular} = 200, 300, 400$, and 500 samples are obtained for four differently sized training sets (that is, for the *UCSB data set*, $n_{tr} = \{n_{cellular} + n_{extracellular}\} \times 58 = 11,600, 17,400, 23,200$, and 29,000 samples, and for the *Harvard data set*, $n_{tr} = \{n_{cellular} + n_{extracellular}\} \times 64 = 12,800, 19,200, 25,600$, and 32,000 samples). In that way, we could comparatively assess the effects of training sample sizes.
- (iii) By changing the $k = \{3, 5, 7, 9, 11, 13\}$ in each differently sized training set, we could evaluate six different training cubes with diverse spatial and contextual information to compare the size effect of the windowing process.
- (iv) Similar to the formation of the training cubes, test cubes are composed of cellular and extracellular patch samples in a stacked form. Since the number of test samples is three times larger than that of training samples, we randomly selected 300, 450, 600, and

750 cellular and extracellular samples from each histopathological image. Thus, test cubes are formed in the size of $k \times k \times n_{tst}$, in which $k = \{3, 5, 7, 9, 11, 13\}$ and n_{tst} is the number of test samples. Consequently, we obtained $n_{cellular} + n_{extracellular} = 600, 900, 1200$, and 1500 samples for four differently sized test sets (that is, for the *UCSB data set*, $n_{tst} = \{n_{cellular} + n_{extracellular}\} \times 58 = 34,800, 52,200, 69,600$ and 87,000 samples, and for the *Harvard data set*, $n_{tst} = \{n_{cellular} + n_{extracellular}\} \times 64 = 38,400, 57,600, 76,800$, and 96,000 samples).

- (v) In that way regard, four test cubes are formed for the corresponding differently sized training sets. Additionally, we could evaluate six different test cubes with diverse spatial and contextual information data (for $k = \{3, 5, 7, 9, 11, 13\}$) separately to observe the size effect of the windowing process for selected training and testing cubes in the classification process.
- (vi) Afterward, each training and testing cube layer (i.e., image patch sample) is resized to 28×28 pixels according to input requirements of all deep learning algorithms. We used the nearest-neighbor interpolation method to extend the size of the image patch samples [38].

3.3 Parameters of methods

In the segmentation of histopathological images, the supervised methods are employed with three deep learning methods (i.e., CNN, DBN and SAE) and two conventional machine learning methods (SVM and RF) for comparison. The deep learning methods are carried out in the Keras environment written in Python [7].

- The CNN architecture is designed with conv1:conv2:pooling1:conv3:pooling2: fully connected layers (as in Fig. 1). The CNN model parameters are presented in Table 1 and are as follows:
 - Input layer: Patch samples with RGB color channels are resized to be 28×28 pixels;
 - Convolutional layer: A two-dimensional convolution of the input feature maps with a 3×3 convolution

		Actual	
		Positive	Negative
Prediction	Positive	True Positive (TP)	False Positive (FP)
	Negative	False Negative (FN)	True Negative (TN)

Fig. 8 Confusion matrix

- kernel and rectified linear units (ReLU) as nonlinear activation function;
- Pooling layer: Subsampling pooling operation over 2×2 non-overlapping windowing on each output feature map;
- Fully connected layer: A total of 256 neurons connected to the output feature maps of pooling layer; and
- Output layer: Two neurons for cellular and extracellular types activated by a softmax classifier model.
- The architecture of SAE is shown in Fig. 3. The SAE model parameters are presented in Table 2 and are as follows:
 - Input layer: Patch size is resized as $28 \times 28 \times 3$, where each patch has three color channels, and input is arranged in a vector form composed of the pixel intensities of a patch with 2352 features;
 - Autoencoder layers: SAE architecture's two autoencoder layers with 256 and 128 neurons, respectively, with the last autoencoder layer connected to the output layer; and

- Output layer: Two neurons for cellular and extracellular types are activated by a sigmoid regression model.
- The architecture of DBN is shown in Fig. 4. The DBN classification model parameters are presented in Table 3:
 - Input layer: Patch size is resized as $28 \times 28 \times 3$, where each patch has three color channels, and input is arranged in a vector form composed of the pixel intensities of a patch with 2352 features;
 - Hidden layers: DBN architecture has two RBM layers with 256 and 128 neurons, and the last RBM layer is connected to the output layer; and
 - Output layer: Two neurons for cellular and extracellular types are activated by a softmax classifier model.

We utilized *LibSVM* library for the SVM classification method with radial basis function (RBF) kernel [4]. For each data set, the best modeling parameters are determined by using a fivefold cross-validation approach. The penalty parameter of SVM (c) is tested between [1–1000] interval with a step size increment of 2 and the (γ) parameter

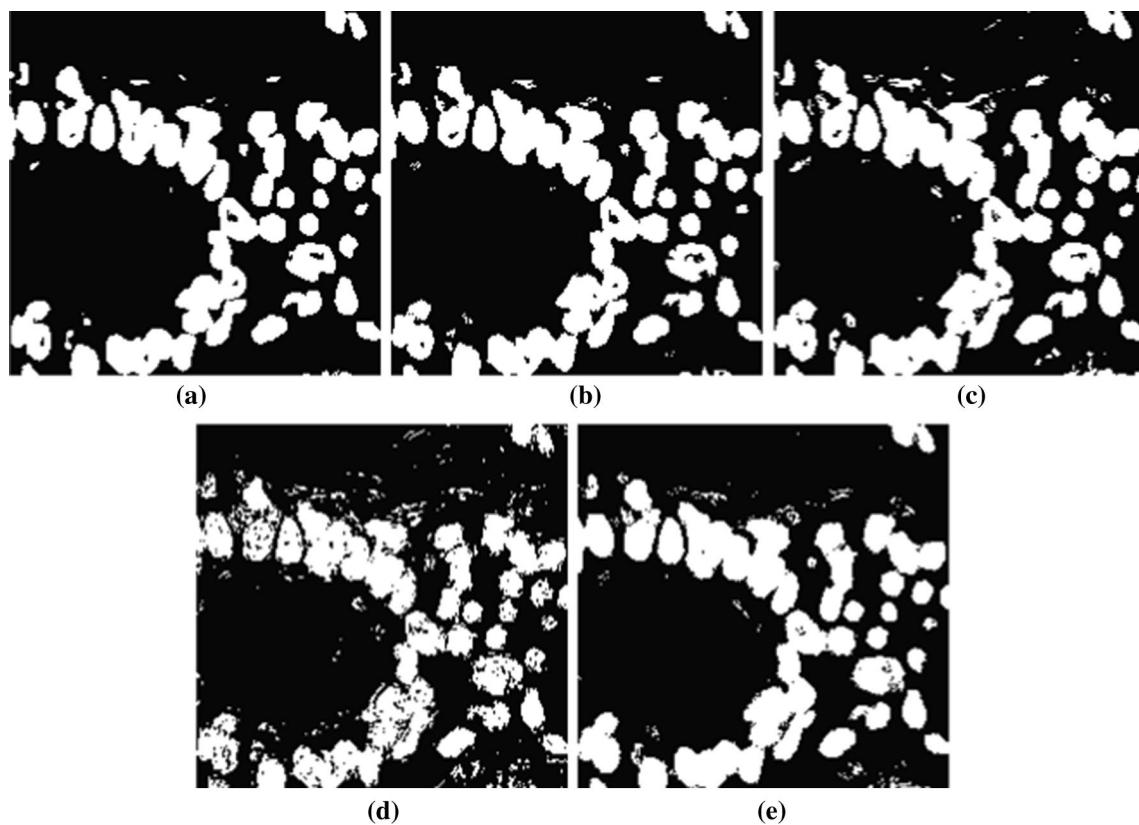


Fig. 9 The segmentation results of the benign cancer patch of Fig. 5a from UCSB data set by **a** CNN, **b** DBN, **c** SAE, **d** SVM, **e** RF classifiers

of RBF kernel function is tested between [0.01100] with a step size increment of 0.1. For the RF method [28], the best forest parameter is evaluated between [25–1000] interval and chosen as 200 in all experiments.

3.4 Evaluation metrics

We evaluated all accuracies by four different metrics—overall accuracy (OA), average precision (AveP), the κ coefficient, and $F1$ -score—as segmentation accuracy metrics using the ground truth of the data set. The evaluation metrics are determined by the confusion matrix of the classifiers, calculated by comparing the location and class information of each ground-truth pixel with the corresponding location and class information in the predicted results after the classification of all test images. In the confusion matrix, shown in Fig. 8, each column represents an actual/ground-truth class, and the values in each row correspond to the predicted labeling of image pixels.

OA is calculated by totaling correctly classified pixels and dividing it by the total number of pixels (Eq. 18). The correctly classified pixels according to ground-truth information are placed along the diagonal of the confusion matrix.

$$OA = \frac{TP + TN}{TP + FP + FN + TN} \quad (18)$$

The κ coefficient/statistic measures the agreement between predicted classification results and ground-truth pixels (Eq. 19). A value of 1 represents the perfect matching, whereas that of 0 represents no agreement.

$$\kappa = \frac{2 \times (TP \times TN - FP \times FN)}{(TP + FP)(FP + TN)(TP + FN)(FN + TN)} \quad (19)$$

$F1$ (or F -measure/ $F1$ -score) can be interpreted as a measure of overlap between the predicted classes and ground-truth information. The $F1$ -score is the harmonic mean of *Precision* (Eq. 20) and the *Recall* (Eq. 21) metrics presented in Eq. 22. The $F1$ -score ranges from 0 to 1; the higher the $F1$ -score, the better the predictive power of the classification process.

$$Precision = \frac{\text{Number of correctly detected pixels}}{\text{Number of manually identified pixels}} = \frac{TP}{TP + FP} \quad (20)$$

$$Recall = \frac{\text{Number of correctly detected pixels}}{\text{Total number of automatically detected pixels}} = \frac{TP}{TP + FP} \quad (21)$$

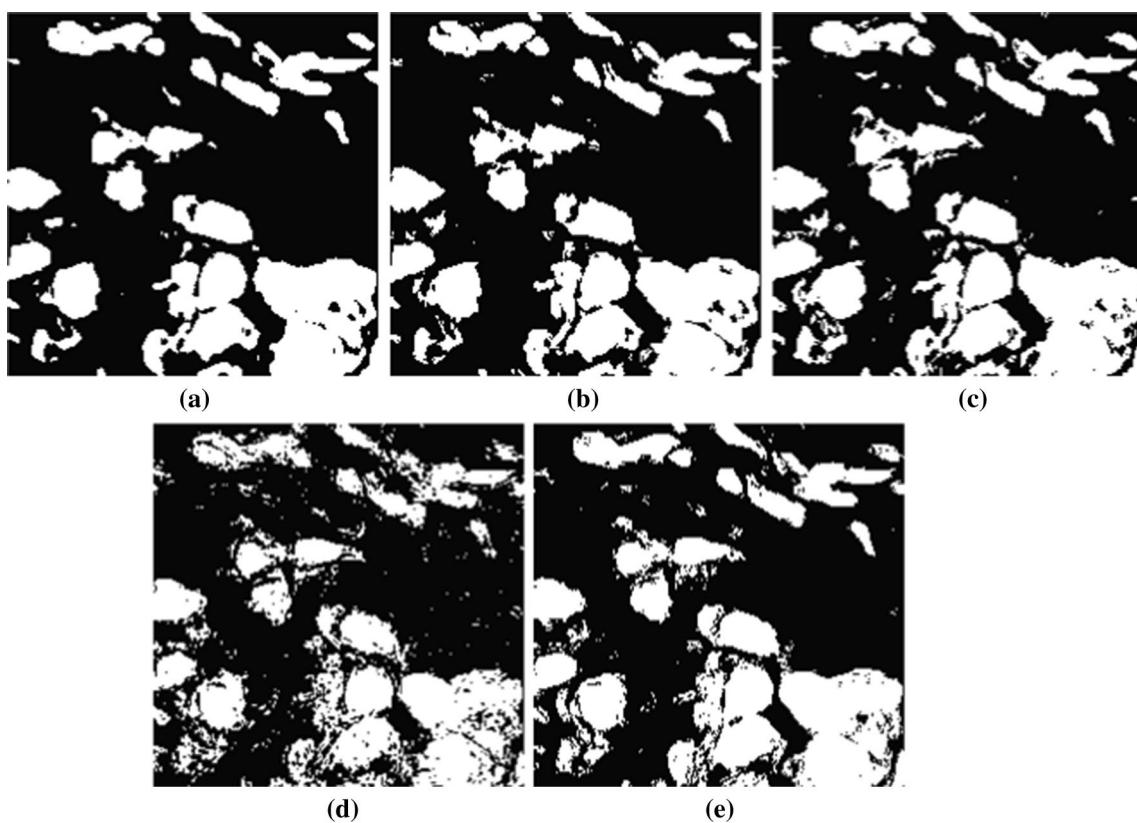


Fig. 10 Segmentation results of the malignant cancer patch of Fig. 5b from UCSB data set by **a** CNN, **b** DBN, **c** SAE, **d** SVM, **e** RF classifiers

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (22)$$

Average precision (AveP) involves the computation of the average value of $p(r)$ over an interval of $r = 0$ and $r = 1$ and the precision $p(r)$ is a function of recall r . AveP can be expressed as in Eq. 23:

$$AveP = \int_0^1 p(r)dr \quad (23)$$

4 Discussion

In the segmentation of histopathological images, three different deep learning methods (i.e., CNN, DBN, and SAE) and two conventional machine learning methods (SVM and RF) are evaluated comparatively. The segmentation accuracies for each test set are presented for UCSB data set in Tables 4, 5, 6, and 7, and for Harvard data set in Tables 8, 9, 10, and 11. In Figs. 9 and 10, segmented images facilitate a visual comparison of all classifiers from the UCSB data set example images shown in Fig. 5a, b, respectively. In the segmentation process of selected UCSB images, n_{tr} is chosen as 11,600 training

samples and window size k is selected as 11. Similarly, segmented images in Figs. 11 and 12 allow visual comparison of all classifiers from the Harvard data set example images in Fig. 6a, b, respectively. In the segmentation process of example Harvard images, n_{tr} is chosen as 25,600 training samples and window size k is selected as 11. We tested entire patches for every pixel in images to obtain segmented images (Figs. 9, 10, 11, 12).

As Tables 4, 5, 6, and 7 representing the results of UCSB data set, CNN performs better than other deep learning algorithms and conventional methods in segmentation accuracy according to OA, AveP, κ , and $F1$ -score metrics. An examination of Tables 8 and 9 representing the Harvard data set reveals that CNN also outperforms the other methods, though SAE ranks first, as Tables 10 and 11 show, in which CNN comes in second with a small difference according to OA, AveP, κ , and $F1$ -score metrics. In all tables representing the two data sets, SVM performs worse than expected among other methods in all experiments. However, RF can achieve results as good as those of the other deep learning methods.

Spatial information is crucial for achieving better cell segmentation accuracies. In the findings of the present study, segmentation accuracy metrics of all methods

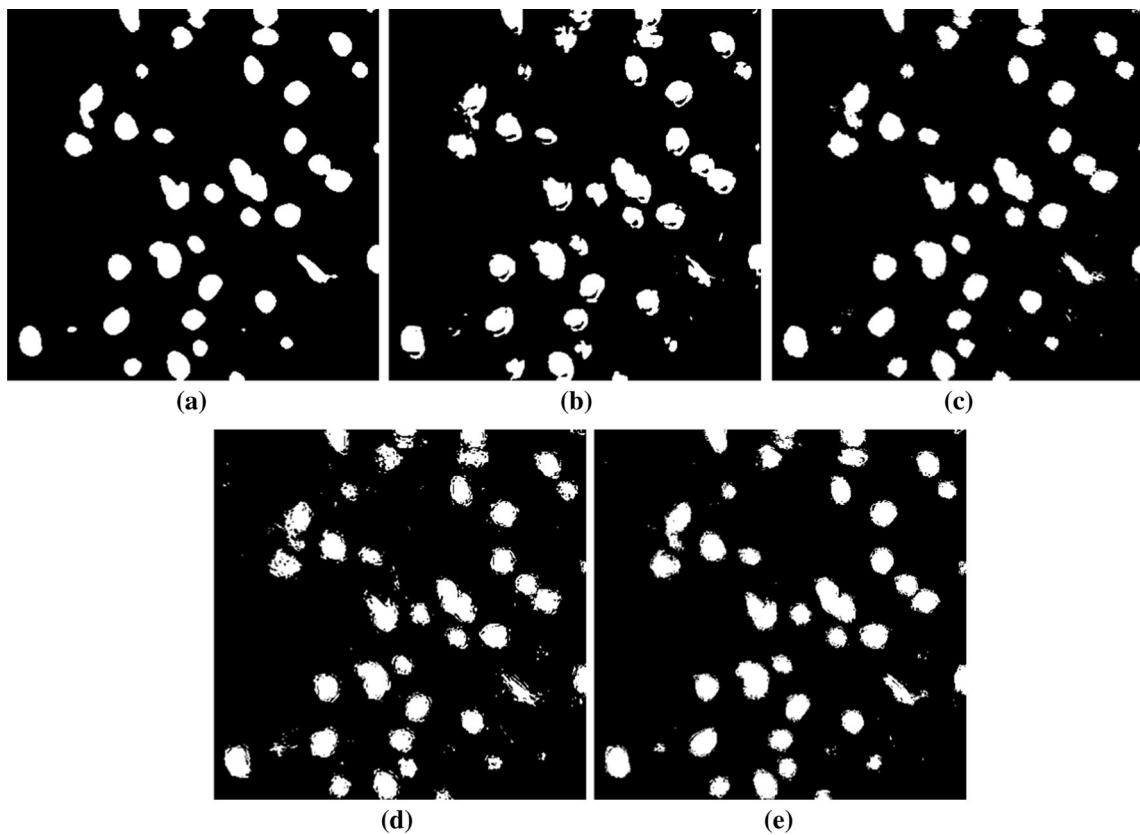


Fig. 11 Segmentation results of the image patch of Fig. 6a from Harvard data set by **a** CNN, **b** DBN, **c** SAE, **d** SVM, **e** RF classifiers

increased as the window size increased. Greater segmentation accuracy metric values emerged especially for $k = 9$ and $k = 13$ in all tables, largely because the spatial information hidden in larger patches revealed cellular and extracellular patterns with increasing window size better. From the pattern recognition and machine learning perspective, extracted features of cellular and extracellular patches become more discriminative by adding spatial and contextual information with larger local windowing.

Lastly, as the results of previous studies (Tables 12, 13) per available metrics from reference studies for the UCSB and Harvard benchmark data sets show, the results of deep learning algorithms are better than those of previous studies and conventional methods. In a reference study [47], OA values seemed greater than those of the proposed methods as 92.45%. However, those authors choose 30 images from the entire 58 images as the training set to optimize the segmentation algorithm and remaining part of the data set as the testing set in the reference study. Compared with our study, the authors used far more training samples in their experiments, though the number of test samples used is incomparable.

5 Conclusions

Pathologists can use CAD systems as a second reader to analyze high-resolution histopathological images for the purposes of detecting and grading cancer. Cell segmentation is particularly important in designing computer-aided cancer detection systems. In this paper, we have proposed cell segmentation in digital images using the spatial and contextual relationships of cellular and extracellular pixels by creating a window structure to contain the neighbors of a pixel in various window sizes. We comparatively evaluated prominent deep learning methods with conventional machine learning methods in the segmentation phase, and our results showed that CNN and, at times, SAE outperformed other methods by way of segmentation accuracy. As expected, segmentation accuracies of all methods increased in all metrics as the window size increased. Altogether, deep learning algorithms are promising CAD methods for evaluating histopathological images, the results of which spatial information can improve, since the extracted features of pixels in cellular and extracellular areas become more discriminative with the addition of spatial and contextual information via larger local windowing.

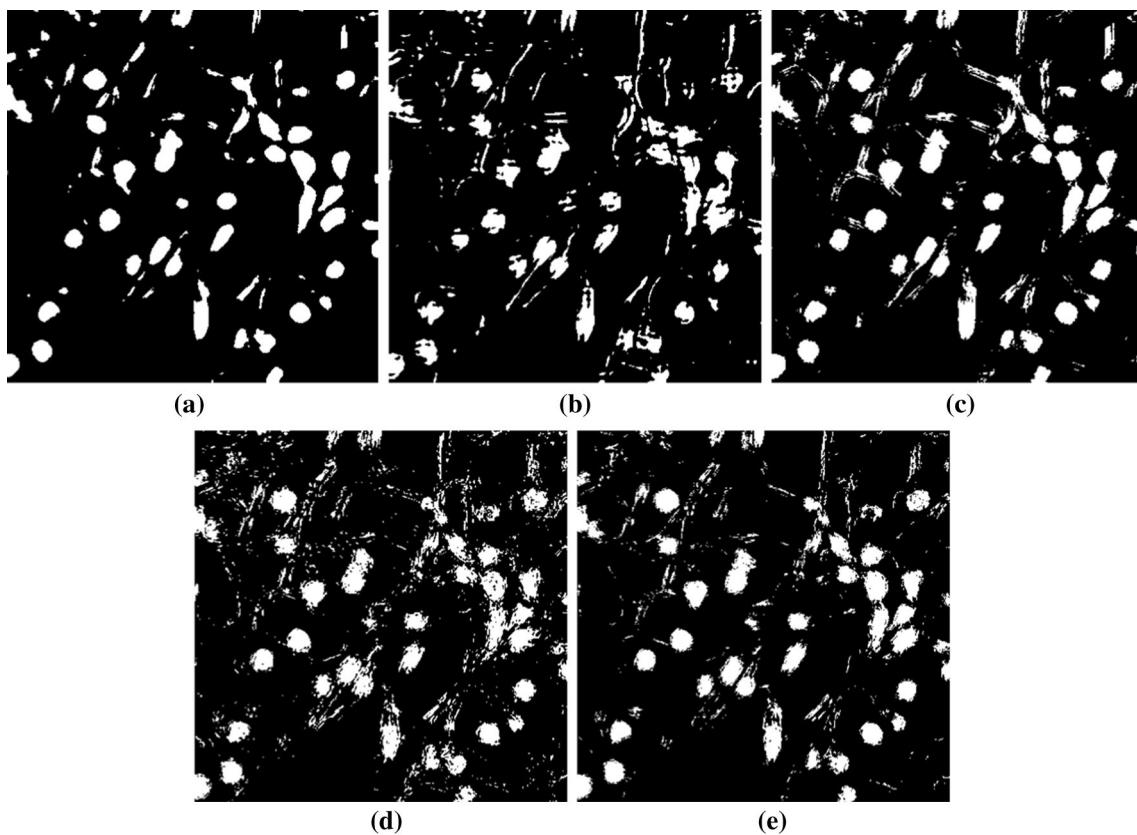


Fig. 12 Segmentation results of the image patch of Fig. 6b from Harvard data set by **a** CNN, **b** DBN, **c** SAE, **d** SVM, **e** RF classifiers

Table 4 UCSB data set segmentation accuracies of all classifiers for $n_{\text{st}} = 34,800$ test samples in different windows sizes using OA, AveP, κ , and $F1$ -score metrics

Window size	CNN	SAE			DBN			SVM			RF			
		OA	AveP	κ	OA	AveP	κ	OA	AveP	κ	OA	AveP	κ	$F1$
3 × 3	87.457	92.079	0.749	87.905	87.112	92.355	0.742	87.715	87.155	0.736	87.289	85.770	89.184	0.736
5 × 5	87.848	92.094	0.757	88.210	87.287	91.910	0.746	87.728	86.724	0.735	89.853	86.960	84.017	0.735
7 × 7	88.069	92.605	0.761	88.518	87.425	92.023	0.749	87.865	86.919	0.738	87.093	83.652	87.705	0.738
9 × 9	88.704	93.649	0.774	89.273	87.687	90.913	0.754	87.735	87.307	0.745	87.491	85.175	88.858	0.747
11 × 11	88.491	93.485	0.770	89.067	87.638	91.521	0.753	87.885	86.956	0.743	87.152	86.663	89.791	0.743
13 × 13	89.345	93.625	0.787	89.772	87.747	90.783	0.755	87.738	86.939	0.744	87.054	86.971	89.982	0.744

Table 5 UCSB data set segmentation accuracies of all classifiers for $n_{\text{st}} = 52,200$ test samples in different windows sizes using OA, AveP, κ , and $F1$ -score metrics

Window size	CNN	SAE			DBN			SVM			RF			
		OA	AveP	κ	OA	AveP	κ	OA	AveP	κ	OA	AveP	κ	$F1$
3 × 3	87.441	92.304	0.749	87.953	87.044	91.482	0.741	87.421	86.783	0.733	86.979	85.795	89.114	0.726
5 × 5	87.471	92.665	0.749	88.072	87.289	90.547	0.746	87.316	86.931	0.736	87.184	84.431	88.230	0.699
7 × 7	87.592	93.916	0.752	88.469	87.492	91.870	0.750	87.874	87.168	0.740	87.493	84.193	88.130	0.686
9 × 9	88.946	93.125	0.779	89.336	88.023	92.583	0.760	88.477	87.314	0.746	87.613	85.149	88.859	0.696
11 × 11	88.521	94.170	0.770	89.249	87.805	91.038	0.756	87.864	87.241	0.746	87.532	86.567	89.747	0.716
13 × 13	89.303	93.756	0.786	89.770	87.994	92.339	0.760	88.390	87.122	0.746	87.356	87.411	90.353	0.721

Table 6 UCSB data set segmentation accuracies of all classifiers for $n_{\text{st}} = 69,600$ test samples in different windows sizes using OA, AveP, κ , and $F1$ -score metrics

Window size	CNN	SAE						DBN						SVM						RF					
		OA	AveP	κ	$F1$	OA	AveP	κ	$F1$																
3 × 3	87.243	92.077	0.745	87.740	87.286	91.285	0.746	87.547	87.022	90.047	0.735	87.287	86.176	89.380	0.729	86.540	86.477	89.368	0.727	86.746					
5 × 5	88.157	92.645	0.763	88.597	87.519	91.970	0.750	87.922	87.050	90.061	0.737	87.322	84.185	88.021	0.697	84.407	86.978	89.754	0.736	87.213					
7 × 7	88.733	92.556	0.775	89.022	87.638	91.280	0.753	87.813	87.188	90.161	0.740	87.458	84.449	88.250	0.688	84.601	87.504	90.144	0.745	87.723					
9 × 9	88.701	93.984	0.774	89.348	87.904	91.315	0.758	88.026	87.438	90.319	0.747	87.735	85.410	88.989	0.698	85.511	87.576	90.233	0.746	87.756					
11 × 11	89.213	94.090	0.784	89.775	88.040	91.975	0.761	88.325	87.390	90.315	0.747	87.648	86.768	89.933	0.718	86.940	87.544	90.195	0.741	87.740					
13 × 13	90.151	95.365	0.793	90.446	88.632	92.247	0.773	88.859	87.524	90.405	0.748	87.791	87.567	90.535	0.723	87.640	87.651	90.268	0.742	87.863					

Table 7 UCSB data set segmentation accuracies of all classifiers for $n_{\text{st}} = 87,000$ test samples in different windows sizes using OA, AveP, κ , and $F1$ -score metrics

Window size	CNN	SAE						DBN						SVM						RF					
		OA	AveP	κ	$F1$	OA	AveP	κ	$F1$																
3 × 3	87.702	92.308	0.754	88.156	87.070	91.708	0.741	87.505	87.039	90.083	0.738	87.271	86.543	89.649	0.734	86.893	86.651	89.500	0.729	86.911					
5 × 5	88.293	92.977	0.766	88.788	87.686	91.073	0.754	87.785	87.226	90.193	0.740	87.490	84.812	88.510	0.713	84.975	87.142	89.886	0.737	87.360					
7 × 7	88.749	93.701	0.775	89.320	87.843	91.076	0.757	87.905	87.316	90.203	0.743	87.652	84.660	88.476	0.697	84.691	87.536	90.145	0.747	87.778					
9 × 9	89.048	93.856	0.781	89.592	87.786	93.138	0.756	88.434	87.454	90.363	0.747	87.709	85.494	89.055	0.702	85.587	87.496	90.201	0.747	87.652					
11 × 11	89.076	94.015	0.782	89.650	87.909	91.314	0.758	88.030	87.126	90.124	0.744	87.387	87.044	90.184	0.725	87.155	87.419	90.103	0.748	87.617					
13 × 13	89.306	93.347	0.786	89.674	88.552	92.055	0.771	88.743	87.628	90.508	0.747	87.858	88.089	90.892	0.725	88.260	87.819	90.425	0.751	87.984					

Table 8 Harvard data set segmentation accuracies of all classifiers for $n_{\text{test}} = 38,400$ test samples in different windows sizes using OA, AveP, κ , and F1-score metrics

Window size	CNN	SAE			DBN			SVM			RF									
		OA	AveP	κ	F1															
3 × 3	88.078	90.950	0.762	88.043	87.870	90.489	0.757	87.730	87.106	90.231	0.741	87.216	86.312	90.113	0.735	85.916	86.376	89.353	0.726	86.573
5 × 5	89.482	92.215	0.790	89.512	89.547	91.897	0.791	89.467	86.744	89.884	0.748	86.958	86.078	89.762	0.714	85.843	86.545	89.462	0.742	86.759
7 × 7	90.253	93.587	0.805	90.486	89.966	92.452	0.799	89.960	86.882	89.990	0.757	87.087	87.859	90.787	0.738	87.965	87.344	90.050	0.757	87.542
9 × 9	90.487	94.215	0.810	90.820	89.737	92.368	0.795	89.756	87.295	90.290	0.764	87.499	88.747	91.263	0.753	89.021	87.505	90.231	0.763	87.638
11 × 11	90.870	94.597	0.817	91.212	90.555	94.379	0.811	90.910	86.956	90.052	0.764	87.152	89.367	91.655	0.780	89.682	87.103	89.838	0.760	87.343
13 × 13	91.195	94.450	0.824	91.445	90.898	94.116	0.818	91.130	87.037	90.104	0.764	87.240	89.804	91.921	0.782	90.154	87.617	90.271	0.764	87.790

Table 9 Harvard data set segmentation accuracies of all classifiers for $n_{\text{test}} = 57,600$ test samples in different windows sizes using OA, AveP, κ , and F1-score metrics

Window size	CNN	SAE			DBN			SVM			RF									
		OA	AveP	κ	F1															
3 × 3	88.191	90.476	0.764	87.964	88.304	90.932	0.766	88.207	86.848	89.988	0.739	87.025	86.638	90.270	0.737	86.340	86.268	89.353	0.725	86.573
5 × 5	89.368	92.920	0.787	89.616	89.590	92.323	0.792	89.628	86.917	89.968	0.748	87.188	86.447	89.993	0.725	86.273	86.846	89.682	0.744	87.055
7 × 7	90.385	93.847	0.808	90.654	90.205	93.394	0.804	90.400	87.122	90.044	0.763	87.492	87.562	90.537	0.733	87.703	87.285	89.902	0.759	87.599
9 × 9	90.806	94.329	0.816	91.102	90.561	93.566	0.811	90.728	87.350	90.269	0.768	87.630	88.557	91.183	0.749	88.774	87.691	90.247	0.765	87.941
11 × 11	90.866	94.895	0.817	91.271	90.073	92.389	0.801	90.025	87.241	90.184	0.771	87.532	89.604	91.836	0.775	89.902	87.473	90.130	0.770	87.682
13 × 13	91.238	95.036	0.825	91.602	90.719	94.078	0.814	90.976	87.168	90.160	0.771	87.421	89.804	91.982	0.776	90.203	87.685	90.292	0.771	87.885

Table 10 Harvard data set segmentation accuracies of all classifiers for $n_{\text{test}} = 76,800$ test samples in different windows sizes using OA, AveP, κ , and $F1$ -score metrics

Window size	CNN	SAE						DBN						SVM						RF					
		OA	AveP	κ	$F1$	OA	AveP	κ	$F1$																
3 × 3	88.586	91.208	0.772	88.512	88.441	90.957	0.769	88.319	87.008	90.050	0.742	87.255	86.539	90.290	0.740	86.149	86.477	89.368	0.729	86.746					
5 × 5	89.740	92.570	0.795	89.815	89.616	92.567	0.792	89.717	87.117	90.101	0.756	87.401	86.201	89.860	0.724	85.965	86.978	89.754	0.748	87.213					
7 × 7	90.009	92.973	0.800	90.137	90.393	93.200	0.808	90.502	87.162	90.144	0.761	87.430	87.645	90.698	0.736	87.681	87.504	90.144	0.761	87.723					
9 × 9	90.595	94.006	0.812	90.860	90.667	93.270	0.813	90.738	87.448	90.316	0.768	87.758	88.901	91.412	0.756	89.133	87.576	90.233	0.767	87.756					
11 × 11	91.100	94.272	0.822	91.329	90.820	93.959	0.816	91.031	87.390	90.315	0.771	87.648	89.574	91.812	0.773	89.876	87.544	90.195	0.768	87.740					
13 × 13	91.210	93.988	0.824	91.353	90.885	94.420	0.818	91.187	90.690	92.954	0.794	90.728	90.162	92.218	0.776	90.463	87.651	90.268	0.769	87.863					

Table 11 Harvard data set segmentation accuracies of all classifiers for $n_{\text{test}} = 96,000$ test samples in different windows sizes using OA, AveP, κ , and $F1$ -score metrics

Window size	CNN	SAE						DBN						SVM						RF					
		OA	AveP	κ	$F1$	OA	AveP	κ	$F1$																
3 × 3	88.241	90.904	0.765	88.150	88.408	91.374	0.768	88.430	87.063	90.099	0.774	87.297	85.213	89.897	0.725	85.923	86.651	89.500	0.726	86.911					
5 × 5	89.414	93.100	0.788	89.698	89.517	92.816	0.790	89.706	87.208	90.171	0.757	87.483	86.572	90.186	0.729	86.303	87.142	89.886	0.746	87.360					
7 × 7	90.307	93.945	0.806	90.614	90.261	93.624	0.805	90.502	87.433	90.295	0.767	87.757	87.372	90.206	0.735	87.123	87.536	90.145	0.764	87.778					
9 × 9	90.959	94.327	0.819	91.226	90.472	92.653	0.809	90.415	87.393	90.328	0.768	87.636	88.786	91.350	0.755	88.999	87.496	90.201	0.766	87.652					
11 × 11	91.235	94.123	0.825	91.405	90.914	93.765	0.818	91.060	90.496	92.734	0.799	90.584	89.571	91.844	0.776	89.846	87.419	90.103	0.769	87.617					
13 × 13	91.456	94.036	0.829	91.565	91.032	94.098	0.821	91.234	87.722	90.589	0.781	87.936	90.196	92.292	0.780	90.455	87.819	90.425	0.772	87.984					

Table 12 Results of the proposed methods and the previous studies on the UCSB data set for comparison

Methods	OA	AveP	κ	F1
Superpixel based [6]	69.64	—	—	—
Color decomposition + SVM [40]	80.20	—	—	—
Multiple image model [34]	89.55	—	—	77.33
Sparse reconstruction + DCN [47]	92.45	—	—	83.93
SVM	88.08	90.89	0.72	88.26
RF	87.81	90.42	0.75	87.98
CNN	90.15	95.36	0.79	90.44
SAE	88.63	92.24	0.76	88.39
DBN	87.52	90.40	0.74	87.79

Table 13 Results of the proposed methods and the previous studies on the Harvard data set for comparison

Methods	OA	AveP	κ	F1
Crowdsourcing [26]	—	—	—	66.41
SVM	90.19	92.29	0.78	90.45
RF	87.81	90.42	0.77	87.98
CNN	91.45	94.03	0.82	91.56
SAE	91.03	94.09	0.82	91.23
DBN	90.69	92.95	0.79	90.72

Acknowledgements This research has been supported by Yildiz Technical University, Scientific Research Projects Coordination Department, Project Number: 2014-04-01-KAP01.

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.

References

- Bengio Y (2009) Learning deep architectures for AI. Found Trend Mach Learn 2(1):1–127
- Bilgin G (2013) Evaluation of spatial relations in the segmentation of histopathological images. In: IEEE 21st signal processing and communications applications conference, pp 1–4
- Bunyak F, Hafiane A, Palaniappan K (2011) Histopathology tissue segmentation by combining fuzzy clustering with multiphase vector level sets. In: Arabnia HR, Tran Q-N (eds) Software tools and algorithms for biological systems. Springer, Berlin, pp 413–424
- Chang CC, Lin CJ (2011) Libsvm: a library for support vector machines. ACM Trans Intell Syst Technol 2(3):27
- Chen Y, Lin Z, Zhao X, Wang G, Gu Y (2014) Deep learning-based classification of hyperspectral data. IEEE J Sel Top Appl 7(6):2094–2107
- Cheng L, Ye N, Yu W, Cheah A (2012) A bag-of-words model for cellular image segmentation. In: Loméni N, Racoceanu D, Gouaillard A (eds) Advances in bio-imaging: from physics to signal understanding issues. Springer, Berlin, pp 209–222
- Chollet F (2015) Keras. <http://github.com/fchollet/keras>
- Ciresan DC, Meier U, Gambardella LM, Schmidhuber J (2011) Convolutional neural network committees for handwritten character classification. In: IEEE international conference on document analysis and recognition, ICDAR’11, pp 1135–1139
- Ciresan DC, Giusti A, Gambardella LM, Schmidhuber J (2013) Mitosis detection in breast cancer histology images with deep neural networks. In: 16th International conference of medical image computing and computer-assisted intervention (MICCAI’13). Springer, Berlin, pp 411–418
- Cruz-Roa A, Caicedo JC, Gonzalez FA (2011) Visual pattern mining in histology image collections using bag of features. Artif Intell Med 52(2):91–106
- Cruz-Roa A, Xu J, Madabhushi A (2015) A note on the stability and discriminability of graph-based features for classification problems in digital pathology. In: Proceedings of the SPIE, vol 9287, pp 928703–928710
- Cun L, Boser B, Denker JS, Henderson D, Howard RE, Hubbard W, Jackel LD (1990) Handwritten digit recognition with a back-propagation network. In: Touretzky DS (ed) Advances in neural information processing systems. Morgan Kaufmann, Los Altos, pp 396–404
- Demir C, Yener B (2005) Automated cancer diagnosis based on histopathological images: a systematic survey. Rensselaer Polytechnic Institute, Technical Report
- Deng L, Yu D (2014) Deep learning: methods and applications. Found Trend Signal Process 7(3–4):197–387
- Dundar MM, Badve S, Bilgin G, Raykar V, Jain R, Sertel O, Gurcan MN (2011) Computerized classification of intraductal breast lesions using histopathological images. IEEE Trans Biomed Eng 58(7):1977–1984
- Fukushima K (1980) Neocognitron: a self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. Biol Cybern 36(4):193–202
- Gelasca ED, Obara B, Fedorov D, Kvilekval K, Manjunath B (2009) A biosegmentation benchmark for evaluation of bioimage analysis methods. BMC Bioinform 10(1):1
- Gençtaş A, Aksoy S, Önder S (2012) Unsupervised segmentation and classification of cervical cell images. Pattern Recognit 45(12):4151–4168
- Goodfellow I, Lee H, Le QV, Saxe A, Ng AY (2009) Measuring invariances in deep networks. In: Advances in neural information processing systems, NIPS’09, pp 646–654
- Goodfellow I, Bengio Y, Courville A (2016) Deep learning. MIT Press, Cambridge ([in preparation](#)). <http://www.deeplearning-book.org>
- Gurcan MN, Boucheron LE, Can A, Madabhushi A, Rajpoot NM, Yener B (2009) Histopathological image analysis: a review. IEEE Rev Biomed Eng 2:147–171
- Hatipoglu N, Bilgin G (2014) Classification of histopathological images using convolutional neural network. In: IEEE 4th international conference on image processing theory, tools and applications, IPTA’14, pp 1–6
- Hatipoglu N, Bilgin G (2015) Segmentation of histopathological images with convolutional neural networks using fourier features. In: IEEE 23th signal processing and communications applications conference, SIU’2015, pp 455–458
- He L, Long LR, Antani S, Thoma GR (2012) Histology image analysis for carcinoma detection and grading. Comput Meth Prog Biol 107(3):538–556
- Hinton GE, Salakhutdinov RR (2006) Reducing the dimensionality of data with neural networks. Science 313(5786):504–507
- Irshad H, Montaser-Kouhsari L, Waltz G, Bucur O, Nowak J, Dong F, Knoblauch NW, Beck AH (2014) Crowdsourcing image annotation for nucleus detection and segmentation in computational pathology: evaluating experts, automated

- methods, and the crowd. In: Pacific symposium on biocomputing, PSB'15. NIH Public Access, pp 294–305
27. Irshad H, Veillard A, Roux L, Racoceanu D (2014) Methods for nuclei detection, segmentation, and classification in digital histopathology: A review—current status and future potential. *IEEE Rev Biomed Eng* 7:97–114
 28. Jaiantilal A (2009) Classification and regression by randomforest-matlab. <http://code.google.com/p/randomforest-matlab>
 29. Jothi JAA, Rajam VMA (2016) A survey on automated cancer diagnosis from histopathology images. *Artif Intell Rev.* doi:[10.1007/s10462-016-9494-6](https://doi.org/10.1007/s10462-016-9494-6)
 30. Karakis R, Tez M, Guler I (2011) Classification the axillary lymph node status of breast cancer patients with the analysis of pattern recognition. In: IEEE 19th conference on signal processing and communications applications, SIU'2011, pp 988–991
 31. Ko B, Seo M, Nam JY (2009) Microscopic cell nuclei segmentation based on adaptive attention window. *J Digit Imaging* 22(3):259–274
 32. Korkmaz SA, Korkmaz MF, Poyraz M (2016) Diagnosis of breast cancer in light microscopic and mammographic images textures using relative entropy via kernel estimation. *Med Biol Eng Comput* 54(4):561–573
 33. Krizhevsky A, Sutskever I, Hinton GE (2012) Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems, NIPS'12, pp 1097–1105
 34. Law YN, Lee HK, Ng MK, Yip AM (2012) A semisupervised segmentation model for collections of images. *IEEE Trans Image Process* 21(6):2955–2968
 35. LeCun Y, Bottou L, Bengio Y, Haffner P (1998) Gradient-based learning applied to document recognition. *Proc IEEE* 86(11):2278–2324
 36. LeCun Y, Kavukcuoglu K, Farabet C, et al (2010) Convolutional networks and applications in vision. In: IEEE international symposium on circuits and systems, ISCAS'10, pp 253–256
 37. LeCun Y, Bengio Y, Hinton G (2015) Deep learning. *Nature* 521(7553):436–444
 38. Lehmann TM, Gonner C, Spitzer K (1999) Survey: interpolation methods in medical image processing. *IEEE Trans Med Imaging* 18(11):1049–1075
 39. Li G, Liu T, Nie J, Guo L, Chen J, Zhu J, Xia W, Mara A, Holley S, Wong S (2008) Segmentation of touching cell nuclei using gradient flow tracking. *J Microsc* 231(1):47–58
 40. Li X, Plataniotis KN (2015) Color model comparative analysis for breast cancer diagnosis using H&E stained images. In: SPIE medical imaging, international society for optics and photonics, p 94200L
 41. Naik S, Doyle S, Agner S, Madabhushi A, Feldman M, Tomaszewski J (2008) Automated gland and nuclei segmentation for grading of prostate and breast cancer histopathology. In: IEEE 5th international symposium on biomedical imaging: from nano to macro, ISBI'2008, pp 284–287
 42. Ng A, Ngiam J, Foo CY, Mai Y, Suen C (2012) Unsupervised feature learning and deep learning tutorial (online). Accessed 2016-01-07
 43. Ojansivu V, Linder N, Rahtu E, Pietikäinen M, Lundin M, Joensuu H, Lundin J (2013) Automated classification of breast cancer morphology in histopathological images. *Diagn Pathol* 8(1):1–4
 44. Onder D, Sarigol S, Karacali B (2013) Automated labelling of cancer textures in colorectal histopathology slides using quasi-supervised learning. *Micron* 47:33–42
 45. Quab M, Bottou L, Laptev I, Sivic J (2015) Is object localization for free?-weakly-supervised learning with convolutional neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition, CVPR'15, pp 685–694
 46. Palm RB (2012) Prediction as a candidate for learning deep hierarchical models of data. Master's thesis, Technical University of Denmark, Palm
 47. Pan X, Li L, Yang H, Liu Z, Yang J, Zhao L, Fan Y (2017) Accurate segmentation of nuclei in pathological images via sparse reconstruction and deep convolutional networks. *Neurocomputing* 229:88–99. doi:[10.1016/j.neucom.2016.08.103](https://doi.org/10.1016/j.neucom.2016.08.103)
 48. Petersen K, Nielsen M, Diao P, Karssemeijer N, Lillholm M (2014) Breast tissue segmentation and mammographic risk scoring using deep learning. In: Breast imaging: proceedings of 12th international workshop on digital mammography, IWDM'14. Springer, Berlin, pp 88–94
 49. Phung SL, Bouzerdoum A (2009) Matlab library for convolutional neural networks. Tech. rep., ICT Research Institute, Visual and Audio Signal Processing Laboratory, University of Wollongong
 50. Poultney C, Chopra S, Cun YL, et al (2006) Efficient learning of sparse representations with an energy-based model. In: Advances in neural information processing systems, NIPS'06, pp 1137–1144
 51. Ross NE, Pritchard CJ, Rubin DM, Dusé AG (2006) Automated image processing method for the diagnosis and classification of malaria on thin blood smears. *Med Biol Eng Comput* 44(5):427–436
 52. Saraswat M, Arya KV (2014) Feature selection and classification of leukocytes using random forest. *Med Biol Eng Comput* 52(12):1041–1052
 53. Veta M, van Diest PJ, Kornegoor R, Huisman A, Viergever MA, Pluim JP (2013) Automatic nuclei segmentation in H&E stained breast cancer histopathology images. *PloS ONE* 8(7):e70,221
 54. Veta M, Pluim JP, van Diest PJ, Viergever MA (2014) Breast cancer histopathology image analysis: a review. *IEEE Trans Biomed Eng* 61(5):1400–1411
 55. Wienert S, Heim D, Saeger K, Stenzinger A, Beil M, Hufnagl P, Dietel M, Denkert C, Klauschen F (2012) Detection and segmentation of cell nuclei in virtual microscopy images: a minimum-model approach. *Sci Rep* 2:503
 56. Wittenberg T, Grobe M, Munzenmayer C, Kuziela H, Spinnler K (2004) A semantic approach to segmentation of overlapping objects. *Method Inform Med* 43(4):343–353
 57. Xu J, Luo X, Wang G, Gilmore H, Madabhushi A (2016) A deep convolutional neural network for segmenting and classifying epithelial and stromal regions in histopathological images. *Neurocomputing* 191:214–223
 58. Xu Y, Zhu JY, Eric I, Chang C, Lai M, Tu Z (2014) Weakly supervised histopathology cancer image segmentation and classification. *Med Image Anal* 18(3):591–604



Nuh Hatipoglu received the B.S. degree in Chemistry, Sakarya University, Turkey, in 2006 and M.S. degree in Computer Engineering, Trakya University, Turkey, in 2008. Currently, he is a Ph.D. student with the Computer Engineering, Yildiz Technical University. His research interests include biomedical image processing, hyperspectral image processing, machine learning, and deep learning. He is the member of Signal and Image Processing Laboratory (SIMPLAB) in Department of Computer Engineering at Yildiz Technical University.



Gokhan Bilgin received B.Sc., M.Sc., and Ph.D. degrees in Electronics and Telecommunication engineering from Yildiz Technical University (YTU), Istanbul, Turkey, in 1999, 2003, and 2009, respectively. He worked as postdoctorate researcher at Computer Aided Diagnosis and Knowledge Discovery Laboratory (CADKD) of the Department of Computer and Information Science in Indiana University–Purdue University Indianapolis (IUPUI). He is currently working as Assistant

Professor in Department of Computer Engineering at Yildiz Technical University. His research interests are in the areas of image and signal processing, machine learning, and pattern recognition with applications to biomedical engineering and remote sensing. He is the founder of Signal and Image Processing Laboratory (SIMPLAB) in Department of Computer Engineering at Yildiz Technical University.