

Cytopathological image analysis using deep-learning networks in microfluidic microscopy

G. GOPAKUMAR,¹ K. HARI BABU,² DEEPAK MISHRA,² SAI SIVA GORTHY,³ AND GORTHY. R. K. SAI SUBRAHMANYAM^{1,*}

¹Department of Earth and Space Sciences, Indian Institute of Space Science and Technology, Thiruvananthapuram 695547, Kerala, India

²Department of Avionics, Indian Institute of Space Science and Technology, Thiruvananthapuram 695547, Kerala, India

³Department of Instrumentation and Applied Physics, Indian Institute of Science, Bangalore, Karnataka 560012, India

*Corresponding author: gorthysubrahmanyam@iist.ac.in

Received 29 July 2016; revised 27 October 2016; accepted 20 November 2016; posted 21 November 2016 (Doc. ID 272778); published 20 December 2016

Cytopathologic testing is one of the most critical steps in the diagnosis of diseases, including cancer. However, the task is laborious and demands skill. Associated high cost and low throughput drew considerable interest in automating the testing process. Several neural network architectures were designed to provide human expertise to machines. In this paper, we explore and propose the feasibility of using deep-learning networks for cytopathologic analysis by performing the classification of three important unlabeled, unstained leukemia cell lines (K562, MOLT, and HL60). The cell images used in the classification are captured using a low-cost, high-throughput cell imaging technique: microfluidics-based imaging flow cytometry. We demonstrate that without any conventional fine segmentation followed by explicit feature extraction, the proposed deep-learning algorithms effectively classify the coarsely localized cell lines. We show that the designed deep belief network as well as the deeply pretrained convolutional neural network outperform the conventionally used decision systems and are important in the medical domain, where the availability of labeled data is limited for training. We hope that our work enables the development of a clinically significant high-throughput microfluidic microscopy-based tool for disease screening/triaging, especially in resource-limited settings. © 2016 Optical Society of America

OCIS codes: (100.4996) Pattern recognition, neural networks; (100.3008) Image recognition, algorithms and filters; (100.2000) Digital image processing.

<https://doi.org/10.1364/JOSAA.34.000111>

1. INTRODUCTION

Cytopathology [1] is the study and diagnosis of diseases at the cellular level. In general, the process of cytological investigation is quite intricate and time-consuming, due to the manual processing involved at various stages of the test. In the case of manual slide examination, the slide has to be manually moved using a mechanical translation stage to visually assess different regions of the slide to check for the presence of cells with abnormal morphology. This would involve several focus and magnification readjustments, making the overall process quite tedious and time-consuming, and it also reduces the throughput. Efforts in the direction of automating the process of slide imaging-based diagnostic testing have been made with the use of robotics. The extensive use of robotics to automate the process of whole-slide image acquisition raises the cost of systems significantly. Recently microfluidics imaging flow cytometry (mIFC) [2,3] based cell analysis has been gaining interest. The mIFC is a nascent microscopic imaging paradigm employing a flow-cell architecture that ensures spatial separation of cells during image

acquisition, reducing the computational complexity of subsequent image processing for cell localization and characterization. In mIFC, the cells are made in suspension and are allowed to flow through the microfluidics channel, where they are imaged at a higher rate and at good resolution so as to capture morphological features. The entire process of making the cells in suspension, capturing the cell images, and analyzing the cells by the automated system takes altogether only a few minutes, while the slide preparation, staining/labeling, followed by the manual examination demands a few hours. Thus, mIFC gained its importance, particularly in resource-limited settings, offering high throughputs, low cost, and reasonable portability [4,5].

Once the imaging is done, the next step is to build a system to mimic the intelligence of a skilled cytopathologist. Computerized methods have been rapidly evolving to automate the segmentation of cells and its subcellular components such as nuclei [6] and cytoplasm so as to assist the cytopathologist in making a better decision. There are automated systems in the next level for feature extraction and classification. A good review

on these techniques can be found in [7]. Mostly, morphological features, which reflect the size and shape of the cells, have been used to classify cancerous cells [8]. Biologists have identified that the power of the human brain comes from the large number of massively interconnected neurons capable of parallel computation [9]. Several neural network (NN) architectures were studied [10] to provide the expertise to the machines, some of them bypassing the need for hand-engineered features. A review on the use of artificial NNs in cytopathology can be found in [11]. Traditional classification systems were typically modeled to contain steps such as cell segmentation, feature extraction, and classification [support vector machines (SVMs), NNs] [7,12]. For large data sets, SVM takes more time for learning and results in a large number of support vectors, particularly when the decision problem is difficult. On the other hand, diminishing gradient inhibits multiple layers in NNs to fall in local minimums and results in slower learning with back propagation. Another difficulty with NNs and SVMs is that they need labeled data, and the amount of information that the system can learn is restricted by labels. Recently, deep-learning systems are emerging as the reliable and *de facto* models for medical image analysis, and many groups across the globe are quickly entering the field and applying these techniques to a variety of applications. A recent review on deep learning in the medical domain can be found in [13].

The traditional system of analyzing the cells under a microscope offers the power to inspect the morphology of the cells, but suffers from low throughput, particularly because of the time needed for fixation and staining. Flow cytometry [14], on the other hand, offers high throughput by studying the size and complexity of the cells in flow by measuring the amount of light scatter. However, flow cytometry offers only limited ability to analyze the specific morphology of cells. Imaging flow cytometry (IFC) [15–17] is a nascent technology that combines the statistical power of flow cytometry with spatial and quantitative morphology of digital microscopy. These systems employ bulk fluid-handling mechanisms for automating the process of sample image acquisition. Further, these systems employ sophisticated and expensive image acquisition schemes, which use time-delay integration detectors and multiple laser sources. Hence, commercially available IFC systems [18] are bulky and significantly expensive as compared to conventional imaging systems. It is from this perspective that we set our aim to develop a low-cost reliable microfluidic microscopy-based disease screening system for triaging that is more relevant in resource-limited settings. Toward this goal, we are one among the few who have worked on microfluidics-based IFC and have developed a system for cytopathology. The proof of concept of the device can be found in [19], where we have used advanced principal component analysis (PCA) signatures for the classification of leukemia cell lines. A framework for processing cells in mIFC was developed and cell classification based on morphological features of the cells was done in [12]. Though the system offered high throughput at reduced cost, the reported accuracy had to be improved so as to make it clinically usable. In this paper, we explore and propose the possibility of using deep learning for cytopathology for a reliable classification system that is an extension of our earlier work [12,19]. Though we demonstrate the performance of the mIFC-based classification

system on the three important leukemia cell lines K562, MOLT, and HL60, the approach is quite general, as it does not require explicit segmentation and feature extraction. When augmented with other diagnostic modalities, such systems enable early detection of diseases such as cancer.

In real scenarios, more often we would have large amounts of data for training but have a small percentage of them labeled. In order to overcome the difficulty of learning both from labeled as well as unlabeled data, a system that models the structure of the data is needed. In this paper, we introduce one such model, where the idea is to tune the weights to learn a general abstract representation of the structure of the data without considering the labels. This is done by building a deep belief network (DBN) using restricted Boltzmann machines (RBMs) and capturing the distribution of the training vectors using the parameters of the RBM: the weights and biases. The model keeps the efficiency and simplicity of the gradient method for learning. Also, we discuss the use of convolutional neural networks (CNNs) for cytopathology analysis using a network deeply trained on the popular imaging database, ImageNet [20,21]. This also overcomes the need for large labeled data and produces discriminative features for an accurate classification.

The major contributions of this paper are (i) the proposal of a highly accurate classification framework that is based on deep learning for unstained IFC data for the first time, and is an improvement over the methods/frameworks proposed in [12,19], thereby moving one step ahead for a reliable triaging tool for cost-effective disease diagnosis, (ii) proposal of a DBN-based cell classifier for better accuracy and faster response, particularly when the availability of labeled medical data is limited, and (iii) the finding that the CNN pretrained on the ImageNet database can generate discriminant features leading to very good classification accuracy of the leukemia cell lines K562, MOLT, and HL60. Just like the DBN-based classifier, a CNN pretrained on ImageNet is very useful in medical fields when there is only limited labeled medical data for training.

This paper is organized such that Section 2 proposes a new and simple framework for cytopathological analysis and classification of IFC data using deep learning. Section 3 discusses the designed DBN for learning the structure of the data, and Section 4 introduces the CNN ImageNet model to extract discriminative features for classification. Results and discussion are provided in Section 5 and conclusions in Section 6.

2. FRAMEWORK FOR ANALYSIS OF CELLS IN MICROFLUIDICS-BASED IFC

The image data set employed for performing the proposed deep learning-based classification has been generated using a relatively new flow imaging modality referred to as “microfluidics-based imaging flow cytometry.” As opposed to conventional IFCs, the method leverages unconventional optics and microfluidics-based sample handling to meet the required imaging throughput and fidelity specifications for cytological investigations. The system enables bright-field microscopic imaging of cells flowing across a custom-designed microfluidic device. The detailed description of the imaging system and the associated microfluidic devices can be found in [19]. For our experiments, raw videos of the flow stream of three leukemia cell lines, K562,

MOLT, and HL60, are captured. These cell lines were obtained from the American Type Culture Collection (ATCC) and are separately cultured. A total of 618 cells (388 HL60, 124 K562, 106 MOLT) were used in the study. The rationale behind demonstrating the usefulness of the presented medical imaging approach with the cultured cell-lines rather than the clinical blood samples is twofold: (i) to have ground truth and thereby be in a position to evaluate the performance of the classification; (ii) classification of red blood cells (RBCs) from white blood cells (WBCs)/cancer cells is fairly easy, and not of much interest in the current work, while the former would contribute to unnecessary bloating of the data set. For the chosen wavelength of illumination (LED having a 405 nm central wavelength), hemoglobin in RBCs absorbs quite a lot of light and appears to be darker (at the front part of cells) and can easily be distinguished from that of WBCs/cancer cells, based on absorption properties and size (RBCs are 6–8 μm , while cancer cells are about 20 μm).

The first major step in automating the cytopathology analysis is segmentation of the cells. Often, segmentation is difficult and computationally intensive. In the proposed approach, rather than going for an accurate segmentation, we look for good classification with the features extracted using deep-learning networks from the roughly localized cells. The rough segmentation is achieved by finding a rectangular bounding box containing the cell. The basic steps in the proposed framework for making cytopathology decision are shown in Fig. 1 and are preprocessing, rough localization, and classification.

In order to enhance features of cells, a simple preprocessing is done by subtracting the background. A background frame for this purpose can be readily captured by keeping only the sheath fluid in the channel. Before subtracting the background, both foreground and background frames are filtered using an average mask of size 5×5 to reduce the effect of noise. The effect of the operation is shown in Fig. 2.

A frame not containing any cell has a very low edge sensitivity threshold (i.e., the gradient energy will be less) compared to the one having cells; this fact is used to discard all such frames [12]. The area of the frame containing the cells will be denser. A morphologically closed and filled canny edge image of the frame can quickly identify the location of the cells. A rectangular bounding box enclosing the cell is then identified.



Fig. 1. Block diagram showing the overview of the system.

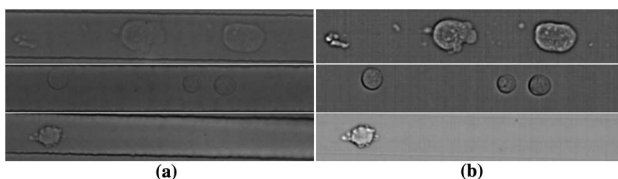


Fig. 2. (a) Frames containing K562 cells (first row), MOLT cells (second row) and HL60 cells (third row) and (b) the corresponding background-subtracted enhanced frames.



Fig. 3. (a) Roughly localized cells from K562 frame in Fig. 2 and (b) bounding box containing leftmost object from 3(a), and leftmost cells from background subtracted frames in Fig. 2.

All objects whose area is less than the minimum size of the cell under study are discarded as cell debris. In order to avoid analyzing a cell multiple times, duplicate elimination is done by comparing local regional properties (area, perimeter, area of the bounding box, and centroid with respect to the bounding box). All nonduplicate cells are then localized by identifying a bounding box [12] enclosing the cell and are used as the data set in this study. The process of rough localization of the cell can be found in [12]. Note that after the background subtraction, cells have better contrast with respect to the background, and since we are looking for a rectangular box bounding the cell, these boxes fit the size of the cell. The roughly localized, leftmost cells from the frames in Fig. 2 are shown in Fig. 3(b).

Recently, deep-learning networks based on RBM and CNN have been found to be effective in learning complex features for higher level visual recognition tasks. In subsequent sections, we introduce these networks, which operate on cell images. The effectiveness in cell classification by these networks is going to be demonstrated with the good result achieved in classifying the leukemia cell lines HL60, K562, and MOLT. Note that the updated framework shown in Fig. 1 does not need fine segmentation and explicit feature extraction, unlike the framework in [12], and uses deep-learning-based classifiers.

3. DBN FOR CLASSIFICATION OF LEUKEMIA CELL LINES

We design a DBN for the classification of leukemia cell lines. A DBN can be thought of as multiple layers of hidden units with connections between the layers but not between the units within each layer. In DBN, every two adjacent layers (except at the fully connected final layer) can be treated as a RBM. Typically, a DBN is trained in a greedy way by training each of the RBMs [22] one at a time starting from the lowest layer. This training is unsupervised, as discussed in the following subsections; this is then followed by discriminative fine-tuning of all the learned weights using back-propagation performed on the available labeled data.

A. RBM

The RBM is a popular generative model. A pictorial representation of a typical RBM architecture is shown in Fig. 4. Note that the connections are undirected, and hence bidirectional. In an RBM, no connections are made between the hidden units in the hidden layer, and no connections are made between the visible units in the visible layer. This helps one to reasonably assume that (i) for a given input at the visible layer, the output at different hidden units are independent and (ii) for a given output observed at the hidden layer, the output that can be induced at different visible units is also independent. Though RBMs can be used as standalone classifiers [23], typically they are used in DBN where a discriminative fine tuning is applied

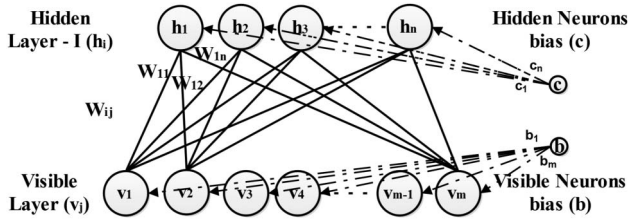


Fig. 4. Architecture of RBM.

on top of the structure of the data learned by the RBM. It has been shown that generative model learning with RBM improved the discriminative classification result in studying functional magnetic resonance imaging (fMRI) images of patients recovering from stroke [24]. In [25], a variant of RBM is used to classify tumor histopathology images. In [26], RBM is used to learn low-dimensional effective features and in [27], RBM is used to learn low-dimensional effective features in manifold learning of brain MRI. The applicability of RBM in medical data is discussed in [28].

Given a particular configuration (weights and biases), joint probability of observing visible and hidden vectors (v, h)

$$P(v, h) = \frac{\exp^{-E(v, h)}}{\sum_{v', h'} \exp^{-E(v', h')}}, \quad (1)$$

where

$$E(v, h) = -[h^T W v + b^T v + c^T h]; \quad (2)$$

$$W = \begin{bmatrix} W_{11} & \cdot & \cdot & W_{1m} \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ W_{n1} & \cdot & \cdot & W_{nm} \end{bmatrix}; \quad (3)$$

$$b = \begin{bmatrix} b_1 \\ \cdot \\ \cdot \\ b_m \end{bmatrix}; \quad c = \begin{bmatrix} c_1 \\ \cdot \\ \cdot \\ c_n \end{bmatrix}; \quad v = \begin{bmatrix} v_1 \\ \cdot \\ \cdot \\ v_m \end{bmatrix}; \quad h = \begin{bmatrix} h_1 \\ \cdot \\ \cdot \\ h_n \end{bmatrix}. \quad (4)$$

The probability of observing a particular visible vector v is

$$P(v) = \frac{\sum_h \exp^{-E(v, h)}}{\sum_{v', h'} \exp^{-E(v', h')}}. \quad (5)$$

We can train the RBM so as to maximize $P(v)$ by minimizing the average negative log likelihood using gradient descent. It turns out that [29]

$$\frac{\partial \log P(v)}{\partial \theta} = -\sum_h p(h|v) \frac{\partial E(v, h)}{\partial \theta} + \sum_{v, h} p(v, h) \frac{\partial E(v, h)}{\partial \theta}, \quad (6)$$

i.e.,

$$\frac{\partial \log P(v)}{\partial \theta} = -\mathbb{E}_{h|v} \left[\frac{\partial E(v, h)}{\partial \theta} \right] + \mathbb{E}_{(v, h)} \left[\frac{\partial E(v, h)}{\partial \theta} \right], \quad (7)$$

where \mathbb{E} is the expectation operator, θ is any parameter (weights W_{ij} , and biases b_j and c_i) of RBM. It can be shown that

$$\frac{\partial E(v, h)}{\partial W_{ij}} = -h_i v_j; \quad \frac{\partial E(v, h)}{\partial b_j} = -v_j; \quad \frac{\partial E(v, h)}{\partial c_i} = -h_i, \quad (8)$$

and

$$p(h_i|v) = \frac{\exp(b_i W_{i1} v + c_i h_i)}{1 + \exp(W_{i1} v + c_i)}, \quad (9)$$

$$p(h_i = 1|v) = \frac{\exp(c_i + W_{i1} v)}{1 + \exp(c_i + W_{i1} v)} = \frac{1}{1 + \exp^{-(c_i + W_{i1} v)}}. \quad (10)$$

Similarly,

$$p(v_j = 1|h) = \frac{1}{1 + \exp^{-(b_j + h^T W_{j1})}}, \quad (11)$$

where W_i and W_j denote the i th row and j th column of the weight matrix. Note that based on Eq. (10), the first term of Eq. (7) can be computed analytically. The only term left out in computing the gradient in Eq. (6) is the expectation of $\partial E(v, h)/\partial \theta$ over the model distribution $p(v, h)$. Though samples from the model distribution can be generated using Gibbs sampling [30], computing $p(v, h)$ is intractable due to the partition function at the denominator in Eq. (1). Hinton has shown that the contrastive divergence (CD) [31] algorithm can be used to replace this expectation with a point estimate at $p(h'|v')$. His team has shown that such an estimate captures the direction of the gradient in Eq. (6) and works well for all practical applications. A pictorial depiction of the CD-1 algorithm is shown in Fig. 5. For every training vector v , compute $p(h|v)$ and get h' by sampling. (That is, for each node in the hidden layer, compute $p(h_i = 1|v)$ using Eq. (10) and turn on the node with the computed probability. If the computed probability is greater than a random number selected from a uniformly distributed random variable in the range between 0 and 1, make the output value at the corresponding node 1. Otherwise, set the value as 0.) Now compute the value at visible layer and sample v' using Eq. (11). Finally, recompute $p(h' = 1|v')$ using Eq. (10).

B. Quick Way of Learning an RBM

This subsection summarizes the algorithm used to train an RBM. Figure 5 depicts the procedure of learning the weights W_{ij} . First, take the input at the visible layer and find the output at the hidden layer using the initialized weights (forward pass). Using the output data sampled at the hidden unit, reconstruct the visible data, and from this reconstructed visible data, try to

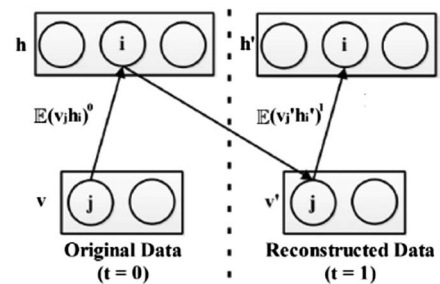


Fig. 5. CD-1 depiction.

produce the data at the hidden unit once again (backward pass). If this reconstruction is error-free, then the difference between the reconstructed data and the data produced at the hidden layer during the forward pass should be zero. The weights need to be adjusted to minimize this error for effective reconstruction

$$\Delta W_{ij} = \varepsilon(\mathbb{E}(v_j h_i)^0 - \mathbb{E}(v'_j h'_i)^1), \quad (12)$$

where \mathbb{E} denotes expectation operation.

The steps involved in learning an RBM are as follows:

1. Initialize W_{ij} from samples randomly selected from normal distribution. Also initialize bias terms (b_j, c_i) to zero.
2. Start with a training vector on the visible units (i.e., the input image of size $m \times n$ is normalized and vectorized). Each component (v_j) is treated as the chance of turning the corresponding visible node ON (set to 1).
3. Update all the hidden units in parallel using Eq. (10).
4. Update all the visible units in parallel using Eq. (11) to get the reconstruction. If the computed probability is greater than a threshold selected from a uniformly distributed random variable in the range [0,1], the corresponding node is turned ON [i.e., the nodes are turned ON (set to 1) with the computed probability].
5. Now, try to reproduce h from the reconstructed input v'_j s using Eq. (10). Let h'_i s denote these reconstructed nodes.
6. Update the weights and biases (W_{ij}, b_j, c_i) as shown below,

$$\begin{aligned} W_{ij}^{t+1} &= W_{ij}^t + \eta \Delta W_{ij}^{t+1} \\ b_j^{t+1} &= b_j^t + \eta \Delta b_j^{t+1} \\ c_i^{t+1} &= c_i^t + \eta \Delta c_i^{t+1} \end{aligned} \quad (13)$$

$$\begin{aligned} \Delta W_{ij}^{t+1} &= \mu \Delta W_{ij}^t + \alpha (\mathbb{E}(v_j h_i)^0 - \mathbb{E}(v'_j h'_i)^1) \\ &= \mu \Delta W_{ij}^t + \alpha [v_j P(h_i|v) - v'_j P(h'_i|v')] \\ \Delta b_j^{t+1} &= \mu \Delta b_j^t + \beta (\Sigma_j v_j - \Sigma_j v'_j) \\ \Delta c_i^{t+1} &= \mu \Delta c_i^t + \gamma (\Sigma_j P(h_i|v) - \Sigma_j P(h'_i|v')). \end{aligned}$$

Here η is the learning parameter, μ is the momentum term, and α, β, γ are the parameters that decide the weights on the corresponding gradient. Note that the gradient [the second term in Eq. (13)] is computed using Eq. (7) and is approximated with a point estimate due to CD-1. In our implementation, the parameters are set by empirical estimation and are $\mu = 0.5, \alpha = \beta = \gamma = 0.1$.

C. Designed and Implemented DBN

The designed DBN for classification of leukemia cell lines is shown in Fig. 6. The system is trained by considering three RBMs and a final fully connected classifier layer. Each of the first three RBMs is trained independently, starting from the first RBM and considering all the data (labeled as well as unlabeled) available for training. The algorithm explained in Section 3.B is used to train the RBMs. Finally, with the labeled data available for training, the parameters are fine-tuned. This is done by a back-propagation algorithm, treating the architecture as a feed-forward (FNN) with five layers. The convergence will be faster, requiring only a few epoch iterations, as the parameters are already trained by RBM learning. The implemented

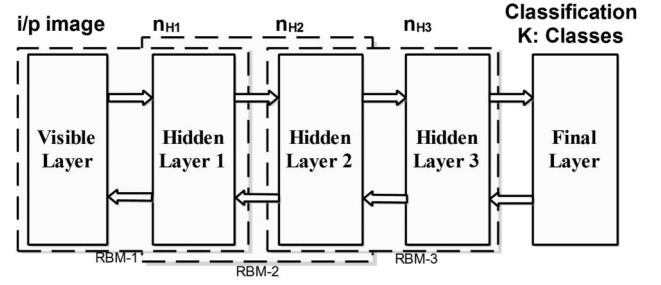


Fig. 6. DBN for classification: depicting the back propagation-based fine-tuning across the layers.

model is tested for classification accuracy, training, and testing time as well as its effectiveness in learning from a mixture of labeled and unlabeled data. The results are discussed in Section 5.

4. FEATURE EXTRACTION USING CNN

A CNN is a feed-forward artificial NN mapping an input vector X into an output vector Y . Internally, it can be thought of a composition of functions, each implementing simple convolutions on an input feature map using learned kernels interleaved with nonlinear and pooling operations followed by locally or fully connected layers [32]. These types of NNs has proved their greater ability to surpass a skilled human in certain classification tasks [33]. The discriminative power of CNN is used in a number of medical cases, such as in classifying breast tissues [34,35] and detecting microcalcifications on mammograms [36], classifying interstitial lung disease [37], detecting pathologic cases in chest X-rays [38], in thyroid cytopathology [39], and in detecting lung nodules in chest radiographs [36]. Cireşan has reported the use of deep max-pooling CNN for detecting mitosis in breast histology images [40]. The transfer learning capability of CNN is studied in [41] and utilized in [42] to annotate the gene expression patterns in a mouse brain. A CNN approach for vision-based classification of cells can be found in [43].

The basic building blocks of a typical CNN are (i) convolution block, (ii) rectified linear unit (ReLU) activation function, and (iii) max pooling. A detailed explanation for each of these building blocks can be found in [44].

A. Convolution Block

Given a multidimensional image X (D image instances), the convolution block finds Y by performing the convolution between the image and a set of learned kernel maps f and by selecting only the valid part of the convolution,

$$X \in \mathbb{R}^{H \times W \times D}, f \in \mathbb{R}^{H' \times W' \times D \times K}, Y \in \mathbb{R}^{H'' \times W'' \times K} \quad (14)$$

$$y_{i''j''k} = b_k + \sum_{i'=1}^{H'} \sum_{j'=1}^{W'} \sum_{d'=1}^D f_{i'j'd'} \times x_{i''+i'-1, j''+j'-1, d'},$$

where $b_k \in \mathbb{R}$ are biases for the nodes in the layer.

B. ReLU Activation Function

Given an input y_{ijd} , the ReLU suppresses the value if it is negative. Thus, the operation performs like an activation function, and it follows the convolution operation:

$$y_{ijd}^1 = \max(0, y_{ijd}). \quad (15)$$

C. Max Pooling

The max pooling operation produces the maximum response of each feature channel in a $H' \times W'$ neighborhood patch and typically follows the ReLU operation:

$$y_{ijd}^2 = \max(y_{i+i'-1, j+j'-1, d}^1) \quad 1 \leq i' \leq H', 1 \leq j' \leq W'. \quad (16)$$

Trained CNNs can outperform humans in certain classification tasks [33]. The only bottleneck in using CNN over back-propagation NNs is the need for a large amount of labeled data for training, which is often limited in the medical domain. In this section, we explore the possibility of doing medical image analysis using a CNN pretrained on a large-scale nonmedical image database, ImageNet (i.e., we wish to investigate the transfer learning capability of CNN for cell analysis). The model readily available [21,44] is heavily trained with several hundreds of non-medical images [20] for each of the 1000 classes (such as birds, cars, tools, etc.). The CNN ImageNet model has 37 layers. The first 36 layers used to generate the cell features for our experiment are shown in Fig. 7. The last three blocks represent the convolution and ReLU operation in fully connected (represented using the subscript FC) layers [44]. Note that the operations in these 36 layers are grouped into eight blocks, where, in each block, the layer is represented by O_{ij} . The O stands for the operation [convolution (C), ReLU (R), max pooling (MP)], i stands for the block ID, and the j stands for the j th instance of the operation O in the block. For each input image, the output of the 36th layer, which is a 4096-element vector, is used as the input to the classifier. As it was heavily trained for the classification of quite a large number of classes, it is reasonable to believe that the features generated have good discriminative power.

D. PreTrained CNNs for the Classification of Leukemia Cells

We use the CNN (Fig. 7) deeply trained on ImageNet [20,21] to classify the leukemia cell lines. The roughly localized cell image is

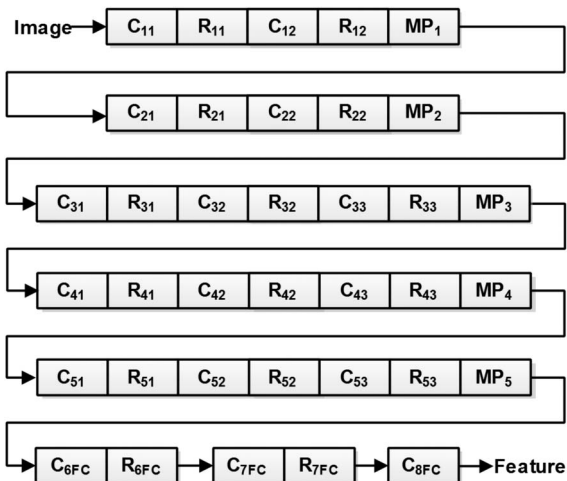


Fig. 7. CNN architecture pretrained ImageNet model used to extract the features for leukemia cell-line classification.

given as input to the network, which will give a 4096-element feature vector at the output. The dimensionality of these vectors is reduced from 4096 to 20 using PCA [45]. The feature vectors of the reduced dimensions are used to train the classifiers. The use of CNN, which was deeply trained on a specific data set, and its use as a general feature extractor in a completely different setting, is an example of transfer learning [38,41,42]. This can be explained as follows. The CNN model used was heavily trained on a few million challenging images of 1000 broad categories. In CNN, the kernel maps are learned by observing local image patches, and hence, the learned detectors (kernels) looks for very local features, such as curves and edges. As most of these features are intrinsic for any image data set, we can transfer the knowledge for other classification settings. Note that this effectively overcomes the difficulty in using deep-learning networks such as CNN due to the lack of sufficient labeled medical images for CNN training.

5. RESULTS AND DISCUSSION

In this section, we discuss the classification of unlabeled unstained leukemia cell lines K562, MOLT, and HL60 imaged using the low-cost, high-throughput microscopic imaging paradigm: the microfluidics-based imaging flow cytometer. Altogether, 618 cells (124 K562, 106 MOLT, and 388 HL60) were used in the experiment. The classification of these cell lines by deep-learning networks (DBN and CNN) is compared for training time, testing time, and accuracy. The ability of the system for semisupervised learning is also considered. The classification accuracy achieved is compared with the SVM-based system in [12], where the features reflecting size, shape, texture, and complexity of the finely segmented cells are used for classification. All the simulations were performed on a Windows PC with Intel(R) Core (TM) i5 CPU at 2.54 GHz, 16 GB RAM, with MATLAB version 8.2.0.701 (R2013b).

The DBN model used to train and test the cell image data set is shown in Fig. 6. It has a visible layer, three hidden layers, and an output layer. The number of neurons in the visible layer is made the same as the number of pixels in the image. Assuming that the input image is of size $m \times n$, the number of neurons is made as mn . In our implementation, all roughly segmented cells are made to standard size $m \times n$. The parameters $m = 52$ and $n = 48$ are selected. (These are the mean number of rows and columns observed for the roughly segmented cell images.) Being a three-class classification problem, there are three neurons in the output layer. The numbers of neurons in the hidden layer are $n_{H1} = 100$, $n_{H2} = 300$, and $n_{H3} = 1000$. The number of RBMs in DBN is fixed as three for our cell-line data set after experimenting with one, two, three, and four layers of RBMs in the stack. The number of hidden layers is also fixed in a similar way. This configuration is then trained by choosing different percentages of input data.

A. DBN Classifier on Roughly Segmented Cell Images

In our experiment, we have treated the normalized pixel intensity values of the cell images as the corresponding probability with which the visible nodes are turned ON in RBM. In order to assess the quality of training achieved, cross-validation experi-

ments are conducted at different folds ($K = 2, 3, 4, 5, 10$). In one cross-validation experiment, training data from each class is randomly divided into K parts, each containing almost the same number of samples. Now, a cross-validation test set is constructed by selecting one part from each class. The system is trained by all other samples and tested with this test set. The cross-validation testing is repeated K times so that all samples are used for testing in some step. The entire experiment is repeated 100 times, each time with different combinations of samples for training as well as testing. The mean classification accuracy (mean) as well as standard deviation (std) is computed. The same set of experiments is conducted using SVMs with linear kernel and FNNs with back propagation as the classifiers. Being a three-class classification problem, the majority voting-based classification strategy is adopted to decide the class label in SVM. The results are shown in Table 1. It can be seen that the highest accuracy (with minimum std) in classifying cell images is achieved by RBM and is shown in Fig. 8. For FNN, we have used only one hidden layer. When the number of hidden neurons is varied ($H_n = 5, 10, 25, 50, 100, 250$), the average cross-validation accuracy did not improve significantly for $H_n > 10$. So, the number of hidden neurons in FNN is fixed as 10 in our implementation.

B. DBN Classifier: The Ability to Learn from Unlabeled Data

In order to check the effectiveness of RBM-based classifiers in learning the structure of the data rather than learning the labels, the following testing strategy has been done. All the data are divided into three sets. **Case 1** holds 50%, 30%, and 20%

Table 1. Cross-Validation Accuracy—FFN, SVM, and RBM on Cell Images

Kfold	10	5	4	3	2
FFN	95.56 (2.57)	95.31 (1.92)	95.18 (1.73)	95.14 (1.49)	94.50 (1.38)
SVM	94.75 (2.71)	94.65 (1.73)	94.48 (1.64)	94.44 (1.47)	93.96 (1.16)
RBM	97.60 (0.48)	97.11 (0.62)	96.50 (0.60)	95.52 (0.20)	94.51 (0.17)

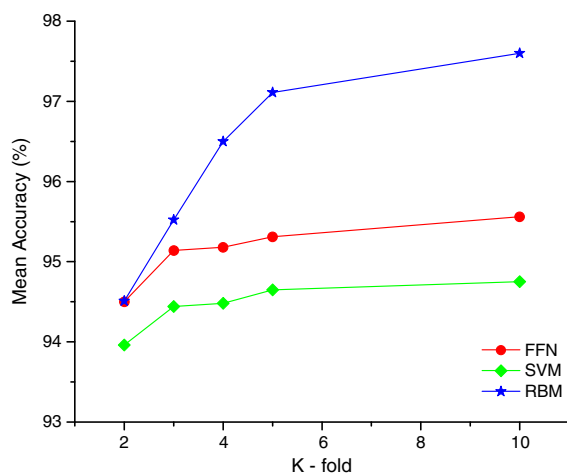


Fig. 8. Effectiveness of RBM classification on raw cell images.

samples, respectively, and **Case 2** holds 67%, 13% and 20% samples, respectively. The first set, along with class labels, and the second set, without class labels, are used for training, and the system is tested with samples from the second and third sets. Note that we have used samples from the second set for testing, since we did not use their labels during the training process. For both the cases (**Case 1 and Case 2**), the experiment is repeated 100 times (each time reinitializing the weights) and the average classification accuracy (mean) as well as std are computed. The results are shown in Table 2. Note that RBM makes use of samples from set 1 and set 2 for training but uses labeled data (set 1) only during back propagation to fine-tune the system. The training error for every pass through the entire training data (an epoch) is computed, and the average error for each epoch over 100 iterations is plotted in Fig. 9 for **Case 1**. The training error profile is similar to that of the normal FNN, and in our case the training converges close to 80 epochs. The result of classification is shown in Table 2. The accuracies are slightly better than the results shown in Table 1 [two folds and three folds (i.e., 50% and 67% training data)]. Note that the unlabeled data cannot be used for learning FNN and SVM.

C. Classification of Features Generated Using PreTrained CNNs

The architecture of the CNN used to generate discriminative features from the cell images is shown in Fig. 7. As noted earlier in Section 4, the output of the 36th layer, which is a 4096-element feature vector, is used for CNN-based classification. Before using the 4096 features by classifier, the dimension of the feature is reduced from 4096 to 20 using PCA [45]. We have selected the number of principal components starting from 13

Table 2. Classification Accuracy—Learn Structure from Data

Cases	Training Accuracy		Testing Accuracy	
	mean	std	mean	std
1	99.98	0.08	95.22	0.71
2	99.99	0.05	95.93	0.86

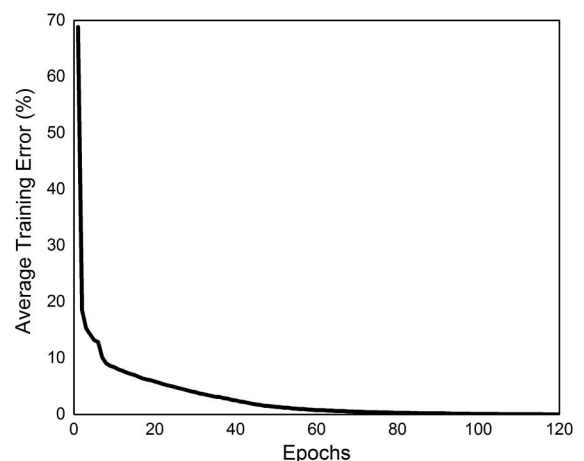


Fig. 9. Training error of the RBM model during discriminative fine-tuning.

(since the number of features in [12] was 13) and set it as 20, since there was no considerable increase in the mean accuracy beyond 20. Once the feature descriptors are generated, the classifiers SVM, FFN, naive Bayes (NBS), and K [nearest neighbor (KNN, $K = 5$)] are applied on top of it. The cross-validation experiments are conducted as explained earlier, and the result is shown in Table 3. These result shows that the accuracy is better and consistent across different cross-validation experiments by different classifiers. Though the CNN was not trained on any medical images, it could extract highly discriminant features. This can be explained as follows. As noted in Section 4.D, the pretrained CNN model used was heavily trained on a few million challenging images of 1000 broad categories. In CNN, the kernel maps are learned by observing local image patches, and hence the learned maps look for very local features such as curves and edges. As most of these features are relevant in classifying any image data set, we can transfer the knowledge learned by a well-trained CNN, such as the ImageNet CNN model used in this study. This is confirmed by the better and consistent accuracy (as shown in Table 3) by different classifiers.

D. Classification on Morphometric Features Extracted from Finely Segmented Cells

Table 4 shows the result of cross-validation experiments performed using cellular features [12] by different classifiers. The roughly segmented cells used as input to the RBM are processed further to segment out the cells, and cellular features as explained in [12] are extracted and performed the classification. The results in Tables 1, 3, 4 show that the deep-learning methods are superior to this method.

Table 5 shows the time taken for different processing steps involved in the classification in [12], the ImageNet CNN model [21], and the RBM classifier model. It can be seen that the DBN model not only gives better classification accuracy, but also a faster response when compared to the method described in [12].

Table 3. Cross-Validation Accuracy on CNN Features

Kfold	10	5	4	3	2
SVM	97.80 (1.82)	97.69 (1.38)	97.62 (1.20)	97.47 (1.02)	97.19 (0.87)
FFN	98.26 (1.57)	98.18 (1.16)	98.16 (1.02)	98.05 (0.93)	97.96 (0.72)
NBS	98.40 (1.51)	98.37 (1.08)	98.36 (0.90)	98.40 (0.78)	98.38 (0.53)
KNN	98.42 (1.49)	98.40 (1.04)	98.37 (0.94)	98.35 (0.81)	98.36 (0.59)

Table 4. Cross-Validation Accuracy on Features in [12]

Kfold	10	5	4	3	2
SVM	95.22 (2.83)	94.47 (1.85)	94.63 (1.52)	94.20 (1.40)	93.57 (1.16)
FFN	92.92 (3.48)	92.74 (2.63)	92.57 (2.35)	92.47 (2.23)	91.76 (3.00)
NBS	90.89 (3.60)	90.77 (2.34)	90.70 (2.18)	90.67 (1.74)	90.35 (1.44)
KNN	80.40 (4.81)	79.93 (3.13)	79.48 (2.90)	78.94 (2.63)	77.75 (0.59)

Table 5. Run-Time Analysis

Method	Rough Locali- zation (18 ms)	Fine Localiza- tion [12] (42 ms)	Simple Features [12] (14 ms)	Complex Features [21] (324 ms)	Classi- fier	Total Time (ms)
Fine segmentation and classification [12]	✓	✓	✓		SVM (1 ms)	75
CNN-ImageNet [21]	✓			✓	SVM (1 ms)	342
DBN classifier model	✓				RBM (0.05 ms)	18.05

E. Performance with Small Percentage of Training Data

In order to check the effectiveness of deep-learning networks in classification, experiments are carried out, even with less amount of training data; the results are shown in Table 6. The entire data set is divided into (S1, S2, S3) such that S1 and S2 together hold the training data (50% of the data set), while S3 holds the testing data (the remaining 50% of the data set). We assume that the class labels of data contained in S2 are unavailable, and cannot be used in a supervised learning system. Still, the data from S2 can be used to train the individual RBMs for initializing the weights. The results of classification, i.e., the classification by DBN, SVM (linear) classifier on the CNN-ImageNet model generated features, and SVM on morphological features [12] are shown for different training cases: 5%, 10%, 20%, 30%, and 40% labeled training data in S1. The results shown are the mean and std for 100 runs; each time random samples are selected from the set for training. The mean accuracy achieved for the said 100 runs, when used 5, 10, 20, 30, 40, 50, and 75% of the available data for training, is provided in the graph in Fig. 10. Note that, unlike the data provided for RBM in Table 6, only the labeled data are used for training and the remaining data are used for testing. It

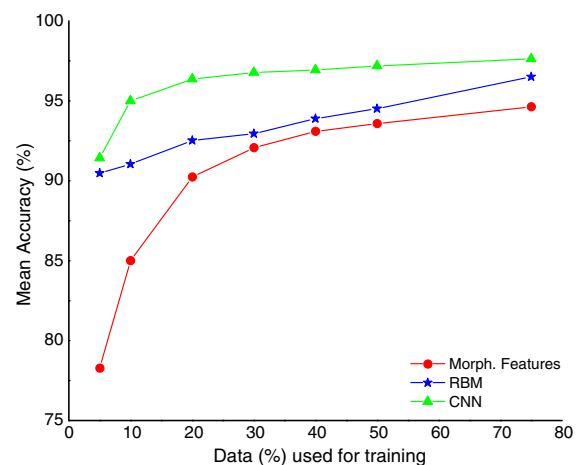


Fig. 10. Classification accuracy of RBM and SVM on features generated by CNN-ImageNet as well as on the morphometric features [12].

Table 6. Classification Accuracy: Training with Less Than 50% Samples

Labeled Data (%)	Morph. Features [12]	RBM	CNN Features [44]
5	78.29 (3.72)	90.88 (1.18)	91.43 (3.33)
10	85.00 (2.74)	91.40 (1.06)	95.00 (1.40)
20	90.23 (1.49)	93.15 (0.85)	96.36 (0.97)
30	92.06 (1.26)	93.55 (0.61)	96.77 (0.76)
40	93.09 (1.13)	94.01 (0.49)	96.93 (0.90)

can be seen that the classification by the deep-learning method is effective, even for the small number of training samples in the case of RBM. Also note that, though the CNN-ImageNet model was heavily pretrained on nonmedical images, the generated features are very good from a classification point of view. These are reflected by the high average accuracy as well as low std for RBM and CNN (see Table 6).

F. Comparison of Class-Specific Accuracy

In order to show that the classification is not biased to any one class, in Table 7 we report the confusion matrix for each of the systems developed. The results shown here are for the 10-fold ($K = 10$) cross-validation experiment. As explained in Section 5.A, 90% of the samples are used for training and the remaining 10% of data is used for testing. Such an experiment is repeated 10 times, such that every sample is used for testing exactly once. Note that Table 7(A) reports the results from SVM run on morphological features [12], Table 7(B) reports the results from the classification by RBM, and Table 7(C) reports the results from SVM run on CNN features. For example, the first row of 7(A) shows that out of the total 388 (373 + 8 + 7) HL60 cells, 373 are correctly classified as HL60, eight are wrongly classified as K562, and seven are wrongly classified as MOLT. We have also presented in Table 8 the associated precision and recall. These measures are based on the number of true positives (TPs), number of false positives (FPs), and number of false negatives (FNs). The precision is defined as $TP/(TP + FP)$ and recall is defined as $TP/(TP + FN)$. As these measures are primarily meant for binary classification, we have considered a one-versus-all strategy. Thus, we have three cases. In Case 1, HL60 constitutes a positive class and K562&MOLT together form a negative class. Similarly, Case 2 is K562 versus HL60&MOLT, and Case 3 is MOLT versus HL60&K562. These measures tell us how much the system has learned to pick a particular class. The result in Table 8 shows that there is significant improvement in both measures with deep networks when compared to the method in [12].

From the results shown so far, it is understood that RBM and CNN bypass the step of specific important feature extraction that differentiate these cell lines and performs a better job

Table 8. Comparison of Precision and Recall (One Against All)

	Case 1		Case 2		Case 3	
	Precision	Recall	Precision	Recall	Precision	Recall
Method [12]	98.42	96.13	89.92	93.55	90.00	93.40
RBM	98.96	98.45	96.77	96.77	93.52	95.28
CNN Features	99.48	99.23	95.20	95.97	94.34	94.34

than using a few important features, as in [12]. The RBM-based classifier not only classifies the data, but also extracts the structure of the data, while CNN extracts discriminative features of the data.

Note that the proposed system is fully automatic, with less setup as well as lower running cost compared to cost of the whole-slide scanners. The employed system uses microfluidic conveyor belts (channels) to enable automated sample handling. This has significantly lowered the setup cost compared to commercially available IFCs, which employ an expensive image acquisition scheme and bulk fluid handling mechanisms. In the context of developing diagnostic system for low resource settings, the most important parameter for feasible deployment is the cost per test. While regular cytological testing requires the usage of large sample and dye volumes, the presented system uses microfluidic devices for sample handling, and thereby requires significantly fewer sample and dye volumes, which brings down the overall cost per test. Performing the test a single time would require simply running the sample through the microfluidic device and analyzing using the acquired images. Thus, the cost per test would include the costs of microliter volumes of reagents and a replaceable plastic chip. This makes the proposed system significantly cost-effective in terms of both setup cost as well as cost-per-test. Further, the proposed deep-learning-based methods are quite general so that they can be used for cell classification studies in a microfluidic microscopic setup without the need for precise segmentation as well as explicit feature extraction, while still producing an improved level of accuracy. The system could better capture the inter-class variability of the cancerous cell lines compared to the morphometric and textural features explored in [12] and has significantly improved the precision as well as recall of the system when compared to the PCA signature-based methods [19] and the SVM trained-on-morphometric features [12]. The central idea of the proposed method is to employ inexpensive optofluidic instrumentation for automated image acquisition and subsequent deep learning-based cell recognition. While the optofluidic architecture enables cost-effective automation of image acquisition, the proposed deep learning-based system enables implementation of effective decision-making, even in resource-limited settings, where personnel trained in the art

Table 7. Comparison of Class-Specific Accuracy

	A. Morphologic Features [12]			B. Classification by RBM			C. CNN Features		
	HL60	K562	MOLT	HL60	K562	MOLT	HL60	K562	MOLT
HL60	373	8	7	382	2	4	385	1	2
K562	4	116	4	1	120	3	1	119	4
MOLT	2	5	99	3	2	101	1	5	100

of diagnostic decision-making are scarce/unavailable. As demonstrated, the image extraction and identification perform with a fair level of accuracy, thereby moving a promising step toward implementation of good quality health care, even in resource-limited settings.

6. CONCLUSIONS

In this paper, we have proposed an approach for automatic cytopathologic analysis using deep-learning methods. The DBN is found effective compared to the classifier system proposed in [12]. The RBM model not only improved the classification accuracy, but also avoided the more demanding accurate segmentation of cells. We have also noted the capability of RBM-based systems for learning the structure of the data rather than learning labels, which will be very helpful in the medical image domain, where often a large data set is available for training but only a small fraction is labeled. We have also studied the applicability of CNN for cell analysis and found that a readily available CNN extensively trained on the nonmedical image database ImageNet produces good discriminative features for classifying the leukemia cell lines K562, MOLT, and HL60. In our investigation, we have found that deep-learning methods outperformed the conventional systems in the classification of these cell lines. To the best of our knowledge, such reporting on cytopathology images is the first of its kind, and we believe that it holds great promise in terms of enabling cost-effective cancer screening in resource-limited settings.

Acknowledgment. Dr. Sai Siva Gorthi gratefully acknowledges the support extended, during his postdoctoral research work, by Dr. Ethan Schonbrun and Dr. Dian Schaak, of Rowland Institute at Harvard. It helped him in acquiring the data sets of different cell lines used in this work. The author Gopakumar would like to acknowledge Mr. Verendra Kalyan Jagannadh, Research Scholar at Optics & Microfluidics Lab, Department of Instrumentation & Applied Physics, IISc for constructive discussions that helped him in preparing the paper.

REFERENCES

1. R. Nayar, *Cytopathology in Oncology* (Springer, 2014).
2. D. A. Basiji, W. E. Ortyl, L. Liang, V. Venkatachalam, and P. Morrissey, "Cellular image analysis and imaging by flow cytometry," *Clin. Lab. Med.* **27**, 653–670 (2007).
3. E. Schonbrun, S. S. Gorthi, and D. Schaak, "Microfabricated multiple field of view imaging flow cytometry," *Lab. Chip* **12**, 268–273 (2012).
4. V. K. Jagannadh, B. P. Bhat, L. A. Nirupa, and S. S. Gorthi, "High-throughput miniaturized microfluidic microscopy with radially parallelized channel geometry," *Anal. Bioanal. Chem.* **408**, 1909–1916 (2016).
5. V. K. Jagannadh, R. S. Murthy, R. Srinivasan, and S. S. Gorthi, "Automated quantitative cytological analysis using portable microfluidic microscopy," *J. Biophoton.* **9**, 586–595 (2016).
6. S. Di Cataldo, E. Ficarra, A. Acquaviva, and E. Macii, "Segmentation of nuclei in cancer tissue images: contrasting active contours with morphology-based approach," in *8th IEEE International Conference on Bioinformatics and BioEngineering (BIBE)* (2008), pp. 1–6.
7. H. Irshad, A. Veillard, L. Roux, and D. Racocanu, "Methods for nuclei detection, segmentation, and classification in digital histopathology: a review—2014; current status and future potential," *IEEE Rev. Biomed. Eng.* **7**, 97–114 (2014).
8. J.-P. Thiran and B. Macq, "Morphological feature extraction for the classification of digital images of cancerous tissues," *IEEE Trans. Biomed. Eng.* **43**, 1011–1020 (1996).
9. G. Deco, V. K. Jirsa, P. A. Robinson, M. Breakspear, and K. J. Friston, "The dynamic brain: from spiking neurons to neural masses and cortical fields," *PLoS Comput. Biol.* **4**, e1000092 (2008).
10. Z. Shi and L. He, "Current status and future potential of neural networks used for medical image processing," *J. Multimedia* **6**, 244–251 (2011).
11. A. Pouliakis, E. Karakitsou, N. Margari, P. Bountris, M. Haritou, J. Panayiotides, D. Koutsouris, and P. Karakitsos, "Artificial neural networks as decision support tools in cytopathology: past, present, and future," *Biomed. Eng. Comput. Biol.* **7**, 1–18 (2016).
12. G. Gopakumar, V. K. Jagannadh, S. S. Gorthi, and G. R. K. S. Subrahmanyam, "Framework for morphometric classification of cells in imaging flow cytometry," *J. Microsc.* **261**, 307–319 (2016).
13. H. Greenspan, B. van Ginneken, and R. M. Summers, "Guest editorial: deep learning in medical imaging: overview and future promise of an exciting new technique," *IEEE Trans. Med. Imaging* **35**, 1153–1159 (2016).
14. M. Ormerod and D. Novo, *Flow Cytometry: A Basic Introduction* (M. G. Ormerod, 2008).
15. N. Barteneva, E. Fasler-Kan, and I. Vorobjev, "Imaging flow cytometry: coping with heterogeneity in biological systems," *J. Histochem. Cytochem.* **60**, 723–733 (2012).
16. L. M. Niswander, K. E. McGrath, J. C. Kennedy, and J. Palis, "Improved quantitative analysis of primary bone marrow megakaryocytes utilising imaging flow cytometry," *Cytometry Part A* **85**, 302–312 (2014).
17. E. J. Beers, L. Samsel, L. Mendelsohn, R. Saiyed, K. Y. Fertrin, C. A. Brantner, M. P. Daniels, J. Nichols, J. P. McCoy, and G. J. Kato, "Imaging flow cytometry for automated detection of hypoxia-induced erythrocyte shape change in sickle cell disease," *Am. J. Hematol.* **89**, 598–603 (2014).
18. Amnis, "The ImageStreamX Mark II Imaging Flow Cytometer," https://www.amnis.com/documents/brochures/ISX-MKII%20Brochure_Final_Web.pdf.
19. V. K. Jagannadh, G. Gopakumar, G. R. K. S. Subrahmanyam, and S. S. Gorthi, "Microfluidic microscopy-assisted label-free approach for cancer screening: automated microfluidic cytology for cancer screening," *Med. Biol. Eng. Comput.*, 1–8 (2016).
20. J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: a large-scale hierarchical image database," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2009).
21. K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman, "Return of the devil in the details: delving deep into convolutional nets," in *British Machine Vision Conference* (2014).
22. G. E. Hinton, "A practical guide to training restricted Boltzmann machines," in *Neural Networks: Tricks of the Trade*, G. Montavon, G. B. Orr, and K.-R. Müller, eds. (Springer, 2012), pp. 599–619.
23. H. Larochelle, M. Mandel, R. Pascanu, and Y. Bengio, "Learning algorithms for the classification restricted Boltzmann machine," *J. Mach. Learn. Res.* **13**, 643–669 (2012).
24. T. Schmah, G. E. Hinton, S. L. Small, S. Strother, and R. S. Zemel, "Generative versus discriminative training of rbms for classification of fmri images," in *Advances in Neural Information Processing Systems 21*, D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, eds. (Curran Associates, 2009), pp. 1409–1416.
25. N. Nayak, H. Chang, A. Borowsky, P. Spellman, and B. Parvin, "Classification of tumor histopathology via sparse feature learning," in *IEEE 10th International Symposium on Biomedical Imaging (ISBI)* (2013), pp. 1348–1351.
26. G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science* **313**, 504–507 (2006).
27. T. Brosch and R. Tam, "Manifold learning of brain MRIs by deep learning," in *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, Vol. **8150** of Lecture Notes in Computer Science (Springer, 2013), pp. 633–640.
28. M. Aalto, "Classification of Medical Data Using Restricted Boltzmann Machines," Master's thesis (University of Tampere, 2014).
29. Y. Bengio, "Learning deep architectures for AI," *Found. Trends Mach. Learn.* **2**, 1–127 (2009).

30. S. Geman and D. Geman, "Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images," *IEEE Trans. Pattern Anal. Mach. Intell.* **PAMI-6**, 721–741 (1984).
31. G. E. Hinton, "Training products of experts by minimizing contrastive divergence," *Neural Comput.* **14**, 1771–1800 (2002).
32. Y. LeCun and Y. Bengio, "Convolutional networks for images, speech, and time series," in *The Handbook of Brain Theory and Neural Networks* (MIT Press, 1998), pp. 255–258.
33. K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: surpassing human-level performance on ImageNet classification," *arXiv:1502.01852 [cs.CV]* (2015).
34. B. Sahiner, H.-P. Chan, N. Petrick, D. Wei, M. A. Helvie, D. D. Adler, and M. M. Goodsitt, "Classification of mass and normal breast tissue: a convolution neural network classifier with spatial domain and texture images," *IEEE Trans. Med. Imag.* **15**, 598–610 (1996).
35. Y. Xu, T. Mo, Q. Feng, P. Zhong, M. Lai, and E. I. C. Chang, "Deep learning of feature representation with multiple instance learning for medical image analysis," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2014), pp. 1626–1630.
36. S.-C. B. Lo, H.-P. Chan, J.-S. Lin, H. Li, M. T. Freedman, and S. K. Mun, "Artificial convolution neural network for medical image pattern recognition," *Neural Netw.* **8**, 1201–1214 (1995).
37. Q. Li, W. Cai, X. Wang, Y. Zhou, D. Feng, and M. Chen, "Medical image classification with convolutional neural network," in *13th International Conference on Control Automation Robotics Vision (ICARCV)* (2014), pp. 844–848.
38. L. W. Yaniv Bar, I. Diamant, and H. Greenspan, "Deep learning with non-medical training used for chest pathology identification," *Proc. SPIE* **9414**, 94140V (2015).
39. E. Kim, M. Corte-Real, and Z. Baloch, "A deep semantic mobile application for thyroid cytopathology," *Proc. SPIE* **9789**, 97890A (2016).
40. D. C. Cireşan, A. Giusti, L. M. Gambardella, and J. Schmidhuber, *Mitosis Detection in Breast Cancer Histology Images with Deep Neural Networks* (Springer, 2013), pp. 411–418.
41. W. Zhang, R. Li, T. Zeng, Q. Sun, S. Kumar, J. Ye, and S. Ji, "Deep model based transfer and multi-task learning for biological image analysis," in *IEEE Transactions on Big Data* (IEEE, 2016), p. 1.
42. T. Zeng, R. Li, R. Mukkamala, J. Ye, and S. Ji, "Deep convolutional neural networks for annotating gene expression patterns in the mouse brain," *BMC Bioinf.* **16**, 1–10 (2015).
43. P. Buyssens, A. Elmoataz, and O. Lézoray, *Multiscale Convolutional Neural Networks for Vision-Based Classification of Cells* (Springer, 2013), pp. 342–352.
44. A. Vedaldi and K. Lenc, "MatConvNet—convolutional neural networks for MATLAB," *CoRR* abs/1412.4564 (2014).
45. I. Jolliffe, *Principal Component Analysis*, Springer Series in Statistics (Springer, 2002).