

Overlapping Cell Nuclei Segmentation in Microscopic Images Using Deep Belief Networks

Rahul Duggal
SBILab, Deptt. of ECE
Indraprastha Institute of
Information Technology-Delhi
(IIIT-D), India.
rahulduggal2608@gmail.com

Anubha Gupta^{*}
SBILab, Deptt. of ECE
Indraprastha Institute of
Information Technology-Delhi
(IIIT-D), India.
anubha@iiitd.ac.in

Ritu Gupta[†]
Laboratory Oncology Unit
Dr. BRA.IRCH, AIIMS, Delhi,
India.
drritugupta@gmail.com

Manya Wadhwa
Deptt. of CSE
Indraprastha Institute of
Information Technology-Delhi
(IIIT-D), India.
manya12144@iiitd.ac.in

Chirag Ahuja
SBILab, Deptt. of ECE
Indraprastha Institute of
Information Technology-Delhi
(IIIT-D), India.
chiragahuja.nsit@gmail.com

ABSTRACT

This paper proposes a method for segmentation of nuclei of single/isolated and overlapping/touching immature white blood cells from microscopic images of B-Lineage acute lymphoblastic leukemia (ALL) prepared from peripheral blood and bone marrow aspirate. We propose deep belief network approach for the segmentation of these nuclei. Simulation results and comparison with some of the existing methods demonstrate the efficacy of the proposed method.

CCS Concepts

•Computing methodologies → Neural networks; Supervised learning by classification; Image segmentation; •Applied computing → Life and medical sciences;

Keywords

Overlapping nuclei segmentation; B-ALL; Deep Belief Network, Machine Learning

1. INTRODUCTION

Automated image analysis of cells and tissues is an active research topic in the field of medical informatics. Computer assisted microscopic examination is cost effective and can offer a number of advantages including the ease of deployment. The first step in microscopic image analysis is

cell segmentation that is generally a challenging task [10]. This paper proposes deep learning based fully automated method for segmentation of blasts (immature white blood cells) from the microscopic slides related to B-lineage acute lymphoblastic leukemia (B-ALL) (Fig. 1).

B-Lineage ALL disorder requires study of immature white blood cells (WBCs) or WBC blasts collected from the bone marrow aspirate. Illumination inconsistencies, cell occlusion, and clustering of cells lead to challenges in segmentation. Some of the commonly used nucleus segmentation methods are: intensity thresholding, color based segmentation, contour based methods, K -means segmentation, and watershed method [6, 8, 14, 12, 1]. Recently, machine learning techniques have been employed for cell segmentation. For example, [4] uses inference based Bayes Network for Yeast nucleus segmentation from florescence microscopy images. In [5], deep learning has been used to segment neuronal membranes in electron microscopy images. In [11], hybrid watershed and Support Vector Machine (SVM) classifier based approach has been used for cell nucleus segmentation from pap-smear images. None of these methods relate to segmentation of overlapping WBC blasts. In general, cell segmentation is challenging because different cell types have different shapes, structures, density, etc. (refer to figure-1 of [10]). Hence, segmentation that works on one type of cell type may not work for another type. Owing to this, research papers on cell segmentation paper are generally specific to the type of the cell.

The cytoplasm in the WBC blasts of B-ALL images (Figure-1) is scant and hence, automated B-ALL disease monitoring requires study of nucleus of these blasts. Intensity thresholding and color based segmentation do not work efficiently on these images. Active contour based methods require an initial region of interest as an input. This region of interest may vary from cell to cell. K -means segmentation is a clustering algorithm that partitions the data into clusters. This works considerably better on our images for isolated nucleus segmentation. However, only K -means cannot segment clusters of nuclei and thus, a post processing step is necessary.

^{*}Joint Corresponding Author

[†]Joint Corresponding Author

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

ICVGIP, December 18-22, 2016, Guwahati, India

© 2016 ACM. ISBN 978-1-4503-4753-2/16/12...\$15.00

DOI: <http://dx.doi.org/10.1145/3009977.3010043>

In this paper, we explore segmentation of nuclei of overlapping WBC blasts from nuclei clusters using recent deep learning based methods. We propose to employ machine learning to identify ridges between overlapping nuclei. Since the cytoplasm in the WBC blasts of B-ALL images is scant, ‘cell clusters’ are equivalently referred to as ‘nuclei clusters’ in the rest of the paper.

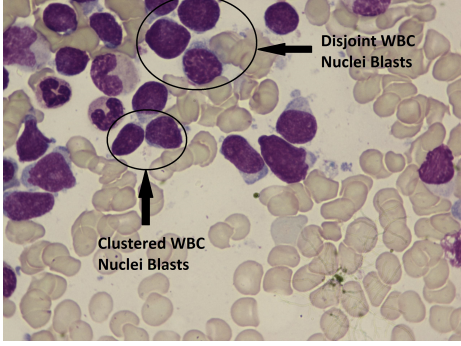


Figure 1: Microscopic Image of B-ALL disorder; WBC blast nuclei are dark purple colored and are regions of interest.

2. BACKGROUND

Artificial neural network (ANN) are a family of classification techniques inspired by the biological neural networks and are used to estimate or approximate functions that depend on a large number of inputs. The neural interconnections have weights that learn features from input and aid in classification. ANN can be three or more layered with one input layer, one output layer, and one or more hidden layers that aid in capturing data characteristics.

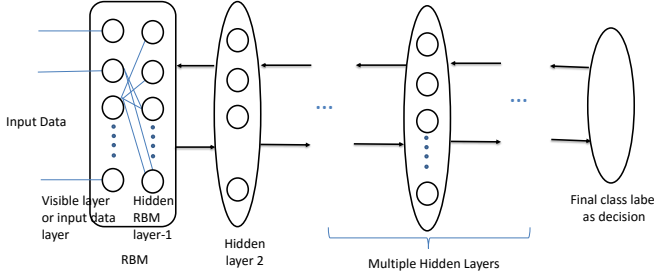


Figure 2: Deep Belief Network Architecture

Another recent development in the field of deep learning is a multilayer network called Deep Belief Network (DBN). It has an architecture similar to ANN wherein neurons are organized into layers with each neuron in a layer being connected to all neurons of the previous layer. There are no intra-layer connections.

A DBN is often conceptualized as a network with stacked RBM layers (refer to Figure-2). It is trained in two steps. The first step is an unsupervised training phase where the stacked RBM layers are trained layer-wise using a greedy algorithm [7]. The second step consists of a supervised learning phase, where the weights learned in step 1 are used as initial weights to start the training of an ANN. DBN’s have been successfully applied to classification tasks [7, 3].

3. PROPOSED WORK

In this paper, we propose a machine learning based approach for segmentation of overlapping WBC blasts from B-lineage ALL microscopic images. The proposed methodology consists of 3 steps:

- Apply K -means clustering in the ‘a’ and ‘b’ channels of Lab color space to extract nuclei of WBC blasts. It should be noted that we are considering overlapping nuclei in our problem.
- Identify nuclei clusters that are required to be broken.
- Apply a machine learning based method to identify pixels lying on the ridge of touching/overlapping nuclei of WBC blasts. Once identified, these ridge pixels are dropped from the image, resulting in segmented nuclei.

The entire approach is summarized in Figure-3.

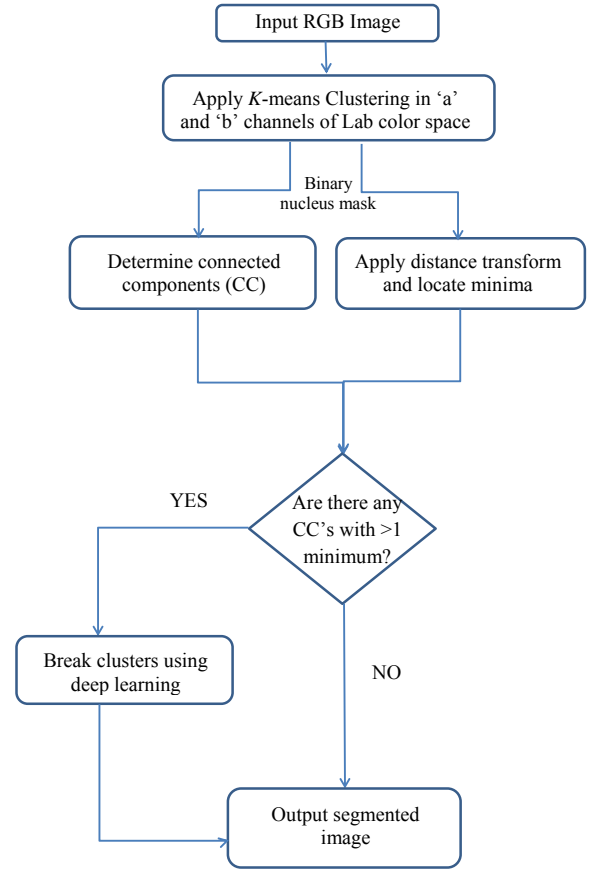


Figure 3: Proposed method outline

3.1 Segmentation of nuclei clusters using K -means clustering

K -means is an unsupervised learning method used for clustering. We would like to segment the entire image into three clusters: background cluster, WBC blast nuclei, and other nucleuss. Since we observed an overlap among the three classes in the RGB color space, we chose to use the Lab color space. On visual inspection, it was found that

the ‘a’ and ‘b’ channels of the Lab color space offer a clean separation of nuclei. Thus, we first convert the image from RGB color space to Lab color space and then apply K -means on only the ‘a’ and ‘b’ channels. Since we captured images at high resolution (2560x1920), we use an optimized GPU implementation of K -means clustering. The output of this step is shown in Figure-4 below.

3.2 Identification of nuclei clusters

Nuclei clusters can be robustly identified using the nuclei mask output obtained from the K -means clustering step. We determine the connected components (CC) in the mask. These CCs correspond to individual nuclei as well as nuclei clusters. Further, we mark the center of each nucleus using the distance transform on the nucleus mask.

Distance transform maps each image pixel’s intensity to the value of shortest distance from that pixel to an obstacle. In our case, with the binary nucleus mask (non-zero intensity on nucleus’ pixel), the obstacle implies any zero intensity pixel. In general, various distance metrics, such as Euclidean distance, city-block distance, and chess-board distances are used. We use the city-block distance that is calculated as below in (1):

$$\mathbf{D}(\mathbf{p}, \mathbf{q}) = \|\mathbf{p} - \mathbf{q}\|_1 = \sum_{i=1}^n |p_i - q_i|, \quad (1)$$

where $\mathbf{D}(\mathbf{p}, \mathbf{q})$ is the city-block distance between two points \mathbf{p} and \mathbf{q} in an n -dimensional space.

The local maxima in the transformed image correspond to the centers of individual nucleus. Nuclei clusters are precisely those CCs that have more than one local maxima. This is illustrated in Figure-5. It is to be noted that from this step, only the true positives, i.e., nuclei clusters are required to be identified with high accuracy, whereas any false positives in terms of single nucleus being identified as a cluster is properly dealt by the trained machine learning model presented later.

3.3 Segmentation of clusters using Deep Learning

Every nuclei cluster has ridges that separate the touching/overlapping nuclei. A nucleus is said to be properly segmented if it passes the following two criteria: 1) It is disjoint from other nuclei; and 2) It is not over-segmented, i.e., it is not broken into several parts. In this stage, we propose to learn the ridges of overlapping nuclei. This requires preparing training data containing ridge pixels, marked as region of interest (ROI), and non-ROI pixels from other parts of the nucleus. When converted to grayscale, ridge pixels are characterized by higher intensity values giving them a white appearance as shown in Figure-6.

For non-ROI pixels, we decide to construct training data that consists of inside-the-nuclei pixels and boundary pixels. In other words, non-ROI class can be split further into two classes: non-ROI-boundary and non-ROI-inside pixels. Thus, we consider ridge identification as a 3-class problem, the 3 classes being ROI, non-ROI-boundary, and non-ROI-inside pixels.

ROI pixels are labeled manually for the training data. For labeling the other classes, we consider a 2-D neighborhood of patch size 51x51 on the nuclei cluster mask. ROI pixels labeled earlier are not re-labeled during this process. Any pixel with only non-zero intensity pixels and without any

ROI-labeled pixel (marked earlier) in its 51x51 neighborhood is labeled to *non-ROI-inside* class. Any pixel with no prior labeled class pixel but having atleast one zero intensity pixel in its 51x51 neighborhood is labeled to *non-ROI-boundary* class. The left over pixels are *not added* to the training data, although all image pixels are fed to the classifier during the test phase.

An example training image with all the three labeled classes and the set of un-labeled pixels is shown in Figure-7.

Next, we have trained various models including Artificial neural networks (ANN) with 1 and 2 hidden layers and Deep belief networks (DBN) with 1 and 2 hidden layers to identify ridge (ROI) and non-ridge pixels. The network architecture, training strategy, and feature vectors are described in the following section. All pixels identified as ROI pixels by a trained machine learning model are dropped from the nuclei cluster leading to its segmentation in individual nuclei.

4. RESULTS

In order to segment nuclei clusters, it is important that ROI pixels are accurately marked. The challenge lies in predicting a connected ridge for ROI pixels. If the ridge is broken, dropping the pixels from the image will not segment the nuclei cluster. In this section, we explain training set-up and compare all the models trained.

Data: Data consists of microscopic image slides prepared from bone marrow aspirate of B-Lineage ALL subjects in the Laboratory Oncology at the Hospital. These images are stained with Jenner-Giemsa stain. These microscopic images are first stain normalized. Ten images containing single clusters capturing most variations are manually cropped from the stained slide images. Each of these nuclei cluster images are rotated by 180 degrees in the steps of 10 degrees. This results in a total of 190 images of single nuclei clusters that are used for training. Images are rotated so that the model learns the ridges located at all possible rotations while training. The validation data consists of 6 images, similarly rotated thus yielding 108 images. The test dataset consists of 35 images containing a mix of single and clustered nuclei.

System Configuration: Simulations were carried using MATLAB15b on a Ubuntu 14.04 system with an Intel Xeon(R) CPU E5-2630 v2 @ 2.60GHz 12 processor and a GeForce GTX 980 Graphics card having cache size 25600KB with 48GB RAM.

Training and Validation sets: Pixels in the training nuclei cluster images are labeled uniquely as described in Section 3.3. The number of pixels in the non-ROI classes greatly outnumber the ROI pixels. Thus, for our training and validation sets, we use all ROI pixels and twice as many non-ROI pixels, sampled randomly from the training images. Further, among the non-ROI pixels, the number of pixels in non-ROI-inside greatly outnumber those in non-ROI-boundary. So we pick twice as many non-ROI-inside as non-ROI-boundary. Table-1 shows the number of pixels considered for each class in the training and validation phase.

Feature Vectors: Our model takes pixel-wise decisions as belonging to one of the three possible classes. This decision is based upon the intensities of the surrounding pixels. Thus, these surrounding pixels become a part of the feature vector. We used gray-level intensity patch of 2D neighborhood of pixel of interest flattened out in the vector form as

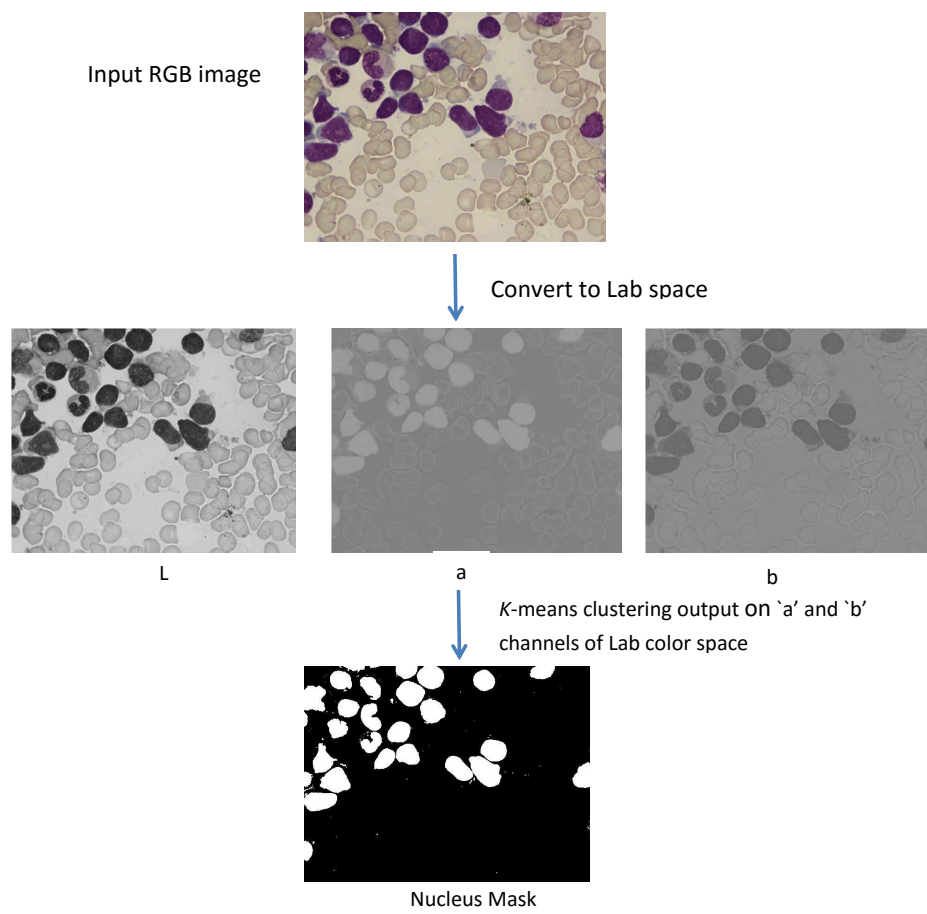


Figure 4: Separation of nuclei clusters using *K*-means Clustering

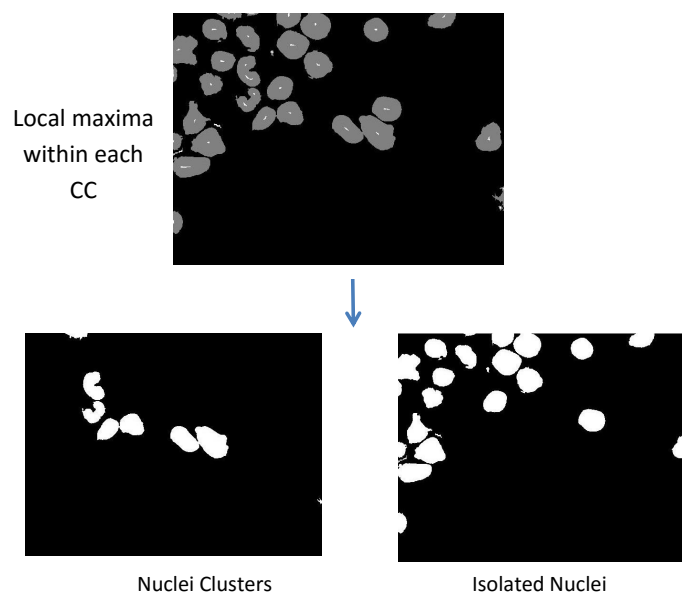


Figure 5: Cluster identification using Distance Transform

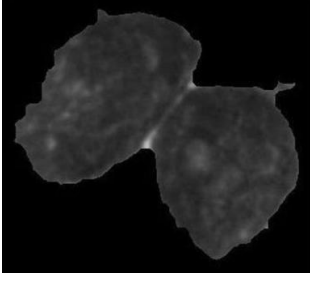


Figure 6: Ridge pixels appear brighter when viewed in grayscale

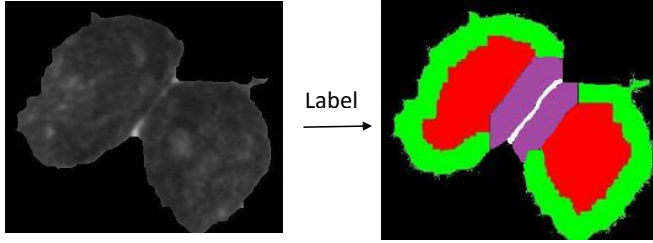


Figure 7: 3 Classes: white corresponds to ROI, red corresponds to non-ROI-inside, and green corresponds to non-ROI-boundary, while purple color pixels are un-labeled and are not considered during the training

Table 1: Number of pixels in each class

Set	ROI	non-ROI-boundary	non-ROI-inside
Training Set	165145	110275	220452
Validation Set	105461	70416	140795

the feature vector. We experimented with 2D patches of varying sizes: 21x21, 31x31, 41x41, 51x51, and 61x61. Best results are obtained with a patch size of 51x51 size.

The following subsections compare the performance of various machine learning models that were trained using this 51x51 patch size that is equal to a feature vector of length 2601.

4.1 Results on 3-Layer ANN

In the 3-layer ANN, the first layer consists of 2601 nodes corresponding to the input feature vector, hidden layer consists of 289 nodes, and the third or the output layer contains 3 nodes. During the training phase, the mean value corresponding to every feature (2601 features in all) with reference to the entire training data is calculated. Hence, we have a mean vector of length 2601. The input training data is pre-processed by subtracting mean vector from the input data so that zero mean data or feature vector is fed to the ANN.

The learning rate was set to 0.1, L2-regularization strength was set to 0.01 and the batch size was fixed at 256. The model was trained over 150 epochs. The confusion matrix

using the trained model on the validation set is shown in Table-2. This is to note that the validation set was pre-processed to have zero mean input by subtracting the mean vector computed using the training data.

Table 2: Confusion matrix for 3-layer ANN

	Predicted ROI	Predicted non-ROI -Boundary	Predicted non-ROI -I nside
Actual ROI	0.9053	0.0252	0.0695
Actual non-ROI -Boundary	0.0025	0.9686	0.029
Actual non-ROI -I nside	0.0026	0.0127	0.9847

4.2 Results on 4-Layer ANN

In the 4-layer ANN, the first layer consists of 2601 nodes corresponding to the input feature vector, two hidden layers consist of 289 and 121 nodes, and the fourth layer contains 3 nodes. The same pre-processing and training parameters are employed as those used for the training of 3-layer ANN above. The confusion matrix using the trained model on the validation set is shown in Table-3.

Table 3: Confusion matrix for 4-layer ANN

	Predicted ROI	Predicted non-ROI -Boundary	Predicted non-ROI -I nside
Actual ROI	0.9085	0.0274	0.0641
Actual non-ROI -Boundary	0.0022	0.9693	0.0284
Actual non-ROI -I nside	0.0039	0.0132	0.9829

4.3 Results on 3-Layer DBN

In the 3-layer DBN, the first layer consists of 2601 nodes corresponding to the input feature vector, hidden layer consists of 289 nodes, and the third layer contains 3 nodes. For DBN, the input pre-processing step is not necessary since it only requires the input data to lie between [0,1].

For DBN, the training is performed in two stages. The unsupervised RBM layers were trained first. For this, the learning rate was set at 0.1, L2-regularization strength was set at 0.01, and a batch size of 256 was used. The unsupervised training was performed over 50 epochs. Weights of RBM layers were then used to initialize ANN connection weights. All training parameters were kept same. During the training of ANN, number of epochs was set to 150. The confusion matrix using the trained 3-layer DBN model on the validation set is shown in Table-4.

Table 4: Confusion matrix for 3-layer DBN

	Predicted ROI	Predicted non-ROI -Boundary	Predicted non-ROI -I nside
Actual ROI	0.9122	0.0305	0.0573
Actual non-ROI -Boundary	0.0037	0.9726	0.0237
Actual non-ROI -I nside	0.0047	0.0185	0.9768

4.4 Results on 4-Layer DBN

In the 4-layer DBN, the first layer consists of 2601 nodes corresponding to the input feature vector, the two hidden layers consist of 289 and 121 nodes, and the fourth layer contains 3 nodes. We set the same training parameters that we used for the training of 3-layer DBN and a similar training strategy is followed. The confusion matrix using the trained 4-layer DBN model on the validation set is shown in Table-5.

Table 5: Confusion matrix for 4-layer DBN

	Predicted ROI	Predicted non-ROI -Boundary	Predicted non-ROI -Inside
Actual ROI	0.9231	0.0241	0.0528
Actual non-ROI -Boundary	0.0051	0.9713	0.0236
Actual non-ROI -Inside	0.0044	0.0206	0.975

Pixels identified as ROI or ridge pixels are finally dropped (or made to go to zero intensity). Again connected components are extracted from the image that leads to segmented cells.

From the above confusion matrix results, we note that the 4-layer DBN outperforms all other models. Although we note only about 2% increase in accuracy between 4-layer ANN and 4-layer DBN, this improvement is significant because incorrectly classified ridge pixels will not be dropped and hence, may lead to joint connection between the two nuclei.

We obtained inferior performance with the further increase of hidden layers. This is to note that deeper neural networks have been shown to suffer with the vanishing gradient problem [9]. Hence, in general, 2 hidden-layer network is preferred as used here, i.e., 4-layer DBN with two hidden layers.

In this work, convolutional neural networks (CNN) have not been used, although their performance is observed to be good in many applications. CNNs have translation invariance property that can pose a problem in the current application because a pixel having a ridge anywhere in its corresponding bounding box may be mislabeled as a ridge pixel owing to the property of translation invariance. This will misclassify a lot of pixels that are not ridge pixels. Since the classification accuracy of ANNs has been shown to be initialization dependent [13], we have used DBN wherein once weights are initialized using RBM model, the fine tuning is done as a conventional ANN. Hence, comparison has been shown of DBN with randomly initialized ANN.

4.5 Comparative performance with different methods

We present segmentation results over a set of 35 test images containing a wide mix of individual and nucleus clusters with varying textures and having ridges at varied angles. Recently, Arslan et al. (2014) in [2] have proposed the cell segmentation method that implements a hybrid approach of color and shape-based watershed segmentation and active contours. We used the source code provided at <http://www.doc.ic.ac.uk/~sa1013/codes.html> by the authors for comparison with our results. In addition, we implemented intensity thresholding and *K*-means clustering

method for segmentation.

Table-6 presents comparative results using these methods. A total of 151 individual nuclei and 31 nuclei clusters were counted in these 35 test images. Table-6 shows the count on number of individual nuclei and the number of nuclei clusters that were correctly segmented. From this table, we note that the method of [2] comes closer in performance to our proposed method. Thus, in Table-7, we present classifier performance with the proposed methodology and with method of [2] on test images in terms of True Positive Rate (TPR) (2), False Discovery Rate (FDR) (3) and F-score (4) defined as below:

$$TPR = \frac{TP}{TP + FN} \quad (2)$$

$$FDR = \frac{FP}{TP + FP} \quad (3)$$

$$TPR = \frac{2TP}{2TP + FP + FN} \quad (4)$$

where TP denotes true positives (no. of WBC blast nuclei correctly segmented), FN denotes false negatives (no. of WBC blast nuclei not segmented correctly), and FP denotes false positives (no. of other cells shown in output image after segmentation as cells/nuclei of interest).

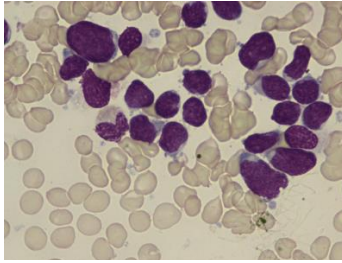
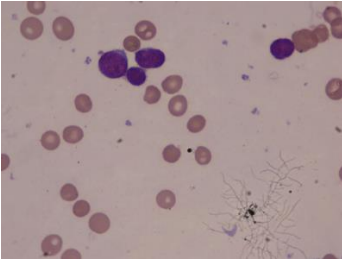
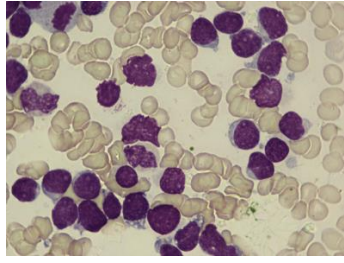
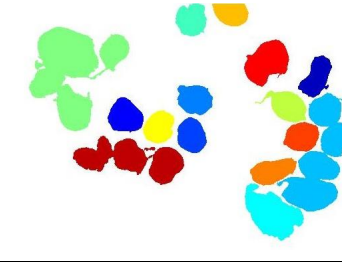
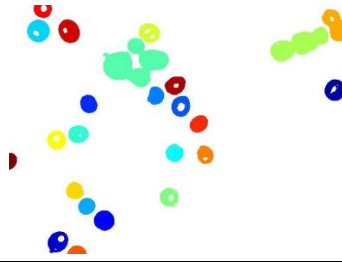
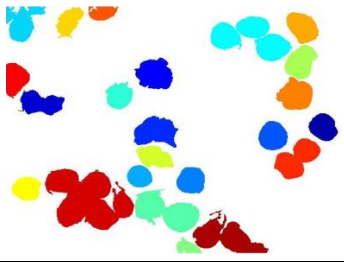
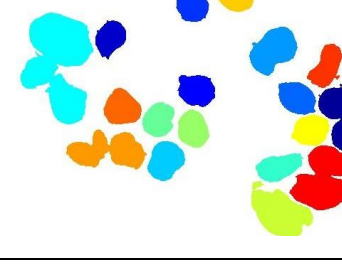
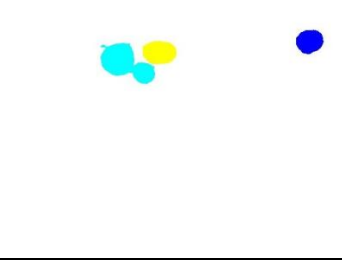
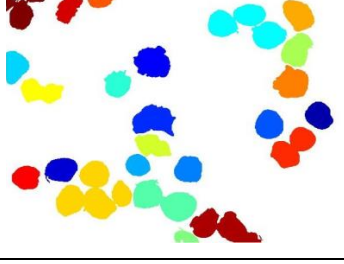
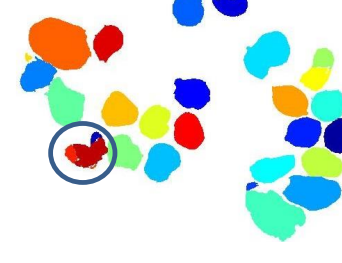
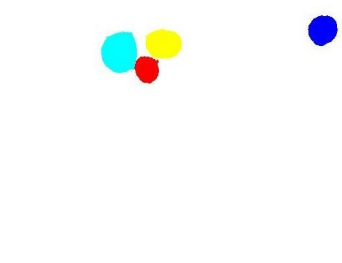
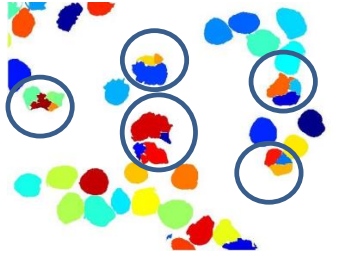
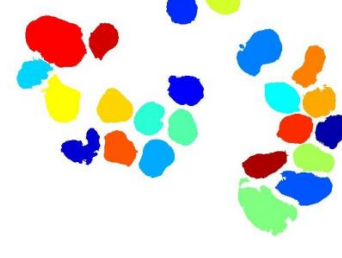
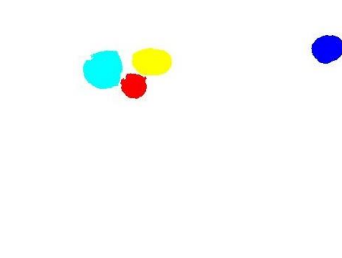
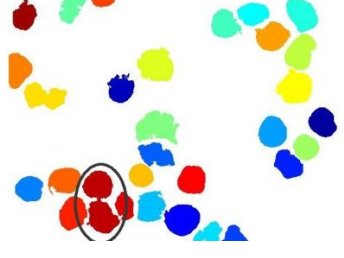
In medical domain, both FN and FP are critical and should be as low as possible. Or, in other words, we should have high values for TPR and F-score, while FDR should be low. The performance of the proposed methodology is consistently better compared to the method of [2] in terms of TPR, FDR, and F-score. Also, it is much faster compared to the method of [2]. Visual results on three test images are shown in Figure-8.

From Figure-8, we note that the method of [2] has a higher no. of false detection and also leads to over-segmentation of nuclei in some cases. Over-segmentation is owing to the use of watershed in the pipeline. On the other hand, the proposed algorithm is robust. Even if an individual nucleus is initially mis-identified as a cluster, it is not segmented because it does not have a ridge in between. This advantage of the proposed method is visible in the higher individual nucleus count in Table-6 and higher TPR, lower FDR, and higher F-score in Table-7.

Intensity thresholding and *K*-means clustering, although faster, are not able to extract all correct cells and cannot segment nuclei from clusters either.

Table 6: Comparative performance of different methods

S. No.	Method	Average Time per Image (seconds)	Segmented	
			Individual nuclei	Nuclei clusters
	Total number	-	151	31
1	Intensity Thresholding	0.217	78	0
2	<i>K</i> -means Clustering	6.4982	122	0
3	Arslan et al. (2014) [2]	327.54	126	28
4	Proposed 4-layer DBN	40.98	151	28

Method	Original Image 1	Original Image 2	Original Image 3
			
Intensity Thresholding			
K-means Clustering			
Arslan et al. (2014) [2]			
Proposed methodology with 4-layer DBN			

** Circles show over-segmentation results/not segmented

Figure 8: Results on 3 test images with different methods

5. CONCLUSION AND FUTURE WORK

In this paper, we have proposed a 3-stage pipeline for

segmentation of overlapping white blood cell nuclei from B-lineage ALL microscopic images. We experimented with 3- and 4-layer Artificial Neural Networks (ANNs) and Deep

Table 7: Comparison on number of nuclei segmented

S. No.	Method	Average Time per image (seconds)	True Positive Rate (TPR)	False Discovery Rate (FDR)	F-Score
1	Arslan et al. (2014) [2]	327.55	0.93	0.27	0.81
2	Proposed 4-layer DBN	40.98	0.97	0.16	0.89

Belief Networks (DBNs). The performance of 4-layer Deep Belief Network was observed to be the best. The proposed method outperformed some existing and a most recent method on a test set of 35 test images. Although some nuclei clusters having a very thin or light ridge pose a challenge and our method could not completely segment such clusters. In the future, we will try to work around this problem.

6. ACKNOWLEDGMENTS

Authors gratefully acknowledge the research funding support (Grant Number: 1(7)/2014-ME&HI) from the Ministry of Communication and IT, Govt. of India for this research work.

7. REFERENCES

- [1] D. Anoraganingrum, S. Kröner, and B. Gottfried. Cell segmentation with adaptive region growing. *ICIAP Venedig, Italy*, pages 27–29, 1999.
- [2] S. Arslan, E. Ozyurek, and C. Gunduz-Demir. A color and shape based algorithm for segmentation of white blood cells in peripheral blood and bone marrow images. *Cytometry Part A*, 85(6):480–490, 2014.
- [3] Y. Bengio, P. Lamblin, D. Popovici, H. Larochelle, et al. Greedy layer-wise training of deep networks. *Advances in neural information processing systems*, 19:153, 2007.
- [4] S. Chen, T. Zhao, G. J. Gordon, and R. F. Murphy. A novel graphical model approach to segmenting cell images. In *Computational Intelligence and Bioinformatics and Computational Biology, 2006, IEEE Symposium on*, pages 1–8. IEEE, 2006.
- [5] D. Ciresan, A. Giusti, L. M. Gambardella, and J. Schmidhuber. Deep neural networks segment neuronal membranes in electron microscopy images. In *Advances in Neural Information Processing Systems*, pages 2843–2851, 2012.
- [6] W. Gao, Y. Tang, and X. Li. Segmentation of microscopic images for counting leukocytes. In *Bioinformatics and Biomedical Engineering, 2008. 2nd International Conference on*, pages 2609–2612. IEEE, 2008.
- [7] G. E. Hinton, S. Osindero, and Y.-W. Teh. A fast learning algorithm for deep belief nets. *Neural computation*, 18(7):1527–1554, 2006.
- [8] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *International Journal of Computer Vision*, 1(4):321–331, 1988.
- [9] H. Larochelle, Y. Bengio, J. Louradour, and P. Lamblin. Exploring strategies for training deep neural networks. *Journal of Machine Learning Research*, 10(Jan):1–40, 2009.
- [10] E. Meijering. Cell segmentation: 50 years down the road [life sciences]. *Signal Processing Magazine, IEEE*, 29(5):140–145, 2012.
- [11] M. Orozco-Monteagudo, C. Mihai, H. Sahli, and A. Taboada-Crispi. Combined hierarchical watershed segmentation and svm classification for pap smear cell nucleus extraction. *Computación y Sistemas*, 16(2):133–145, 2012.
- [12] J. M. Sharif, M. Miswan, M. Ngadi, M. Salam, and M. Jamil. Red blood cell segmentation using masking and watershed algorithm: A preliminary study. In *Biomedical Engineering (ICoBE), 2012 International Conference on*, pages 258–262. IEEE, 2012.
- [13] I. Sutskever, J. Martens, G. E. Dahl, and G. E. Hinton. On the importance of initialization and momentum in deep learning. *ICML (3)*, 28:1139–1147, 2013.
- [14] M. Yan, J. Cai, J. Gao, and L. Luo. K-means cluster algorithm based on color image enhancement for cell segmentation. In *Biomedical Engineering and Informatics (BMEI), 2012 5th International Conference on*, pages 295–299. IEEE, 2012.