INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN BIOMEDICAL ENGINEERING
Int. J. Numer. Meth. Biomed. Engng. 2010; **00**:1–17
Published online in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/cnm

Phase Contrast Cell Detection Using Multi-level Classification

Ehab Essa[1]*, Xianghua Xie[2]†

[1] *Faculty of Computers and Information Sciences, Mansoura University, Egypt*
[2] *Department of Computer Science, Swansea University, UK*

SUMMARY

In this paper, we propose a fully automated learning based approach for detecting cells in time-lapse phase contrast images. The proposed system combines two machine learning approaches to achieve bottom-up image segmentation. We apply pixel-wise classification using random forests (RF) classifiers to determine the potential location of the cells. Each pixel is classified into four categories (cell, mitotic cell, halo effect, and background noise). Various image features are extracted at different scales to train the RF classifier. The resulting probability map is partitioned using the k-means algorithm to form potential cell regions. These regions are expanded into the neighboring areas to recover some missing or broken cell regions. In order to validate the cell regions, another machine learning method based on the bag-of-features and spatial pyramid encoding is proposed. The result of the second classifier can be a validated cell, a merged cell, or a non-cell. In the case that the cell region is classified as a merged cell, it is split by using the seeded watershed method. The proposed method is demonstrated on several phase contrast image datasets, i.e. U2OS, HeLa, and NIH 3T3. In comparison to state-of-the-art cell detection techniques, the proposed method shows improved performance, particularly in dealing with noise interference and drastic shape variations. Copyright © 2010 John Wiley & Sons, Ltd.

Received . . .

KEY WORDS: cell segmentation; phase contrast imaging; machine learning; random forests; bag-of-features

1. INTRODUCTION

Phase contrast microscopy enables the monitoring and analysis of cell proliferation without the need for staining as it is based on bright field microscopy. It is an optical microscopy that transforms the phase differences in light moving through the specimen into the differences in brightness, which can be observed and converted into an image. A fundamental task in monitoring cell behavior across time is cell localisation and segmentation. Manual cell segmentation or detection is a time-intensive, laborious process and is subjective due to high inter- and intra-operator variability, particularly with the number of cells increasing exponentially over time. Fully automatic cell segmentation is challenging due to the large variability in cell shape, low contrast between the object of interest and the background, and the image artifacts such as halo effect and shade-off.

Cell segmentation methods for phase contrast images can be categorized into region-based approaches [1, 2, 3, 4], active contour methods [5, 6, 7, 8], energy minimization based approaches

† Email: x.xie@swansea.ac.uk. Web: http://csvision.swan.ac.uk

* Correspondence to: Ehab Essa. Computer Science Department, Faculty of Computers and Information Sciences, Mansoura University, Mansoura, 35516, Egypt. Email: ehab_essa@mans.edu.eg

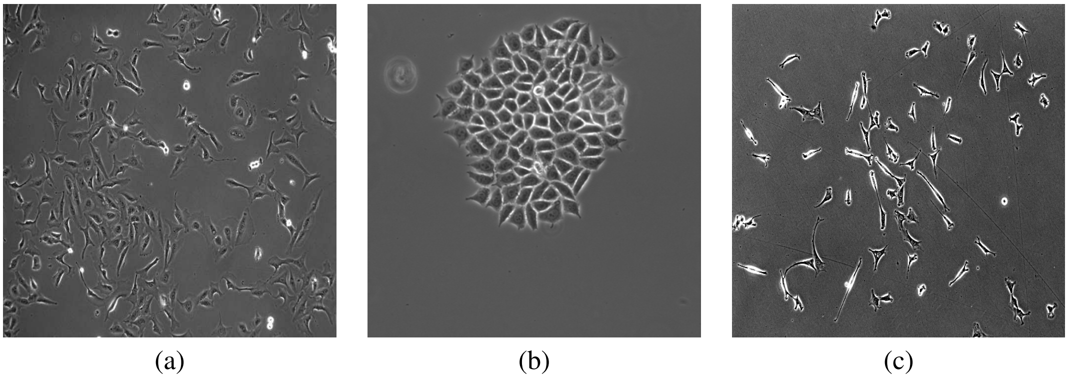


Figure 1. Example of the phase contrast images. (a) U2OS human osteosarcoma cells (672 × 512 pixel). (b) Human cervical cancer HeLa cells (400 × 400 pixel) (c) NIH 3T3 fibroblast cells (696 × 520 pixel).

[9, 10, 11, 12], image restoration based approaches [13, 14, 15], and machine learning-based approaches [16, 17, 18, 15, 19, 20, 21, 22].

Region-based approaches include methods such as thresholding and watershed. Chalfoun *et al.* [3] proposed a morphological watershed-based method to segment cells in colonies, including those that are touching each other. Empirical global thresholding is used to separate the foreground from the background, and then seed points are detected using histogram quantization. Next, each pixel is assigned to the closest seed point using the geodesic distance. Mitotic cells are also detected by applying thresholding and testing the roundness of the obtained region. However, the method has a large number of morphological parameters to be adjusted. Energy minimization based approaches, such as graph cut, have also been introduced to detect and segment cells. Bensch and Ronneberger [12] proposed a cost function for a graph cut based method to encourage dark-to-bright transition by defining asymmetric edge weights in a directed graph.

For image restoration based approaches, the aim is to recover an artifact-free phase contrast image. For example, Yin *et al.* [13] treated the cell segmentation problem as restoring an artifact-free phase contrast image. The image model is defined based on the optical properties of phase contrast imaging, and the restoration problem is formulated using a regularized quadratic cost function. The segmented cell can be found by applying thresholding on the restored images. However, the method can not distinguish between artifacts, e.g. halo effect, and bright cells, e.g. cell in the mitosis cycle. Su *et al.* [14, 15] introduced a dictionary-based restoration method of phase contrast diffraction patterns. The restored image is then grouped into atoms, and the cells are segmented based on a semi-supervised method. However, user initialization is required to propagate the labeling process.

Machine learning-based approaches are used to recognize the cells from the background. The learning methodology can be supervised [17, 16, 18, 22], semi-supervised [15, 19], or unsupervised [21, 20]. For example, the methods proposed in [16, 21] used a learning-based approach in cell detection by selecting an initial set of points corresponding to cell centers and refining the detection by removing the less likely points. However, these methods are not segmenting the cell region. He *et al.* [17] used an AdaBoost classifier to detect cell centers based on wavelet features and an SVM classifier using LBP features to segment cells from the background. The output of both classifiers is fed into a seeded watershed method to obtain the final cell regions. In [20], the cell segmentation is achieved by using random forest classification to learn cell boundaries and then carrying out superpixel clustering to obtain initial cell regions and build an adjacency graph. This graph is then partitioned using correlation clustering segmentation. However, these methods are prune to both under- and over-segment the cells, especially in densely populated regions. Arteta *et al.* [18] proposed to detect a set of candidate cell regions using a maximally stable extremal regions (MSER) detector; each region is scored based on SVM classification. Finally, a subset of these regions that are non-overlapping is selected as cells using dynamic programming. The authors in [23] extended the previous work so that it can split the detected region containing overlapping cells

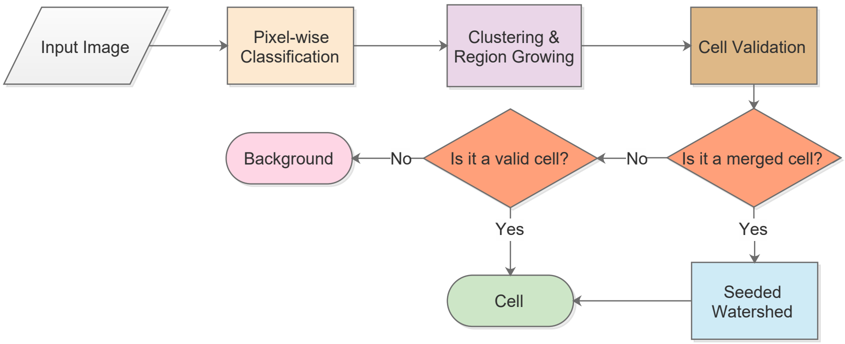


Figure 2. Proposed system overview.

into different individual ones. Akram *et al.* [22] used two convolutional neural networks (CNN) to detect and then segment cells. The first CNN predicts bounding boxes containing the cells, and then the second CNN produces a segmentation mask for each bounding box.

In this work, we propose a machine learning approach to detect and delineate the cell region in microscopy phase contrast images (see example images in Figure 1). Pixel-wise classification is carried out based on a random forest (RF) classifier to achieve low-level image segmentation and produce a probability map of the cell locations. We construct initial cell regions by using k-means clustering and region growing and then validate each region using a bag-of-features (BoF) classifier based on the spatial pyramid feature coding. The main contributions of this paper include the following aspects.

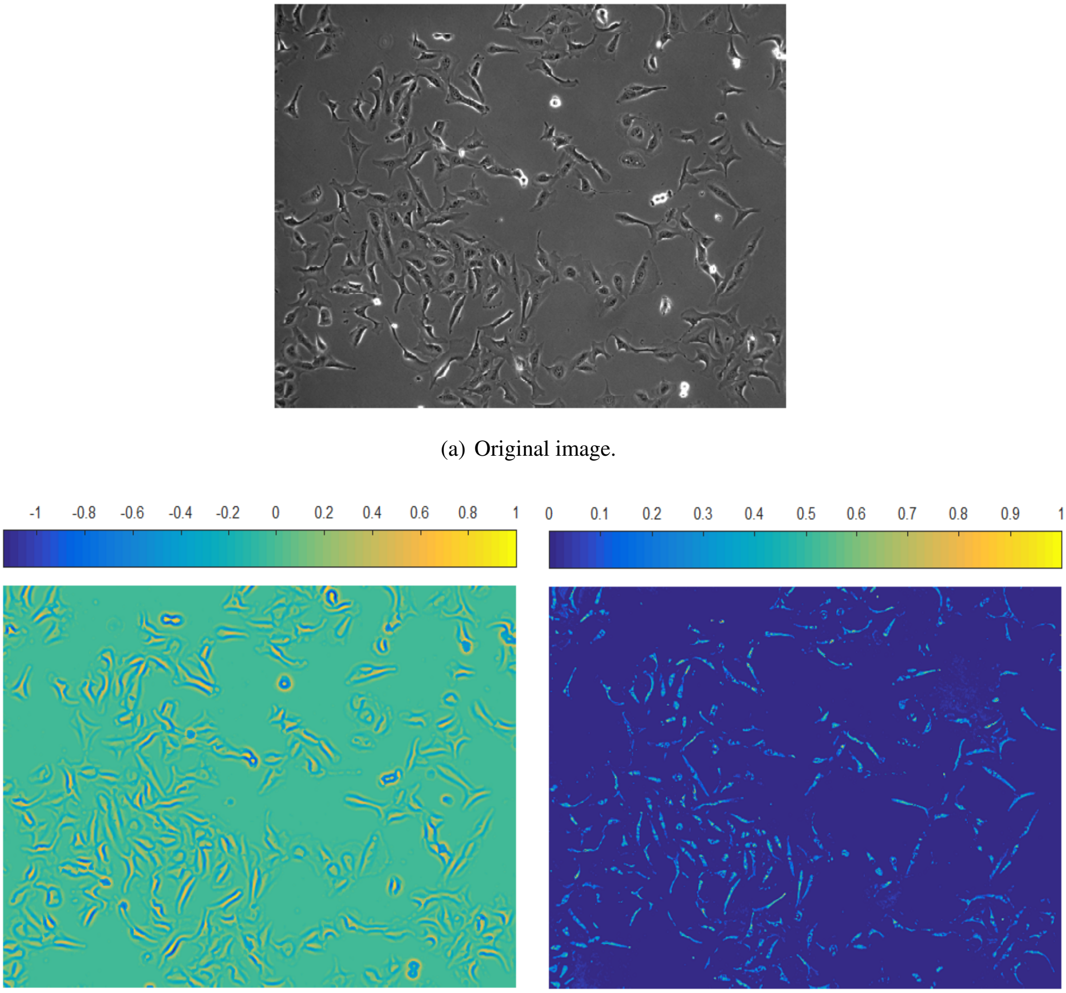
- Combines two machine learning approaches in an automatic cell-detection paradigm. The learning is efficient and can be easily adapted to new data.
- A low-level phase contrast image segmentation using a pixel-wise classification is proposed. The classifier is trained on multi-scale features that enable efficiently highlighting cell regions regardless of its size or cell-division cycle.
- A region-growing method is proposed to retrieve most of the cell region and to examine its relationship with its neighboring regions.
- BoF classification with spatial pyramid feature encoding is proposed to validate the cell regions as a single cell, merged or non-cell. The classifier can handle large variation in cell region size and shape.
- Splitting the merged cells using a seeded watershed method where the seeds are inferred from the pixel-wise classification probability map.

The rest of this paper is organized as follows. Section 2 presents the proposed method in detail. Experimental results and discussion are given in Section 3. Finally, Section 4 concludes this paper.

2. PROPOSED METHOD

Briefly, the proposed method starts with pixel-wise classification using an RF classifier, in order to create a probability map of cell and non-cell regions. The similar pixels are grouped together using the k-means algorithm. These initial regions are expanded slightly to recover some missing structures to form candidate cell regions. The regions are then validated by using the BoF-based classifier that can handle cell region variation. The merged cell is split by using the seeded watershed method. The flow chart in Figure 2 illustrates the key stages of the proposed method.

DOI: 10.1002/cnm



(a) Original image.

(b) Large eigenvalue of the Hessian matrix.

(c) Precondition image restoration.

Figure 3. Example of features extraction from a phase contrast image.

2.1. *Pixel-wise Classification*

There are several machine learning methods that can be used for pixel-wise labeling. Recently, deep learning methods, e.g., CNN have emerged as popular approaches. However, deep learning methods generally require a large quantity of training data in order to obtain good performance due to its significantly larger number of parameters, typically hundreds of millions or even more depending on the complexity of its architecture. RF is much more efficient to train and test and has very few parameters. In this paper, an RF classifier is used to classify image pixels into four classes, i.e. normal migration (dark) cell, mitotic (bright) cell, halo artifact, and background. RF [24] is a set of decision trees that form a strong ensemble classifier to overcome problems such as data outliers and overfitting. The RF uses two randomization techniques to improve its generalization capabilities. The first technique is bagging where each tree is trained on a different subset of the training data sampled randomly with replacement. The second technique is the randomized node optimization, where each node in any tree is trained on a random subset of features to find the best parameter to split the tree.

DOI: 10.1002/cnm

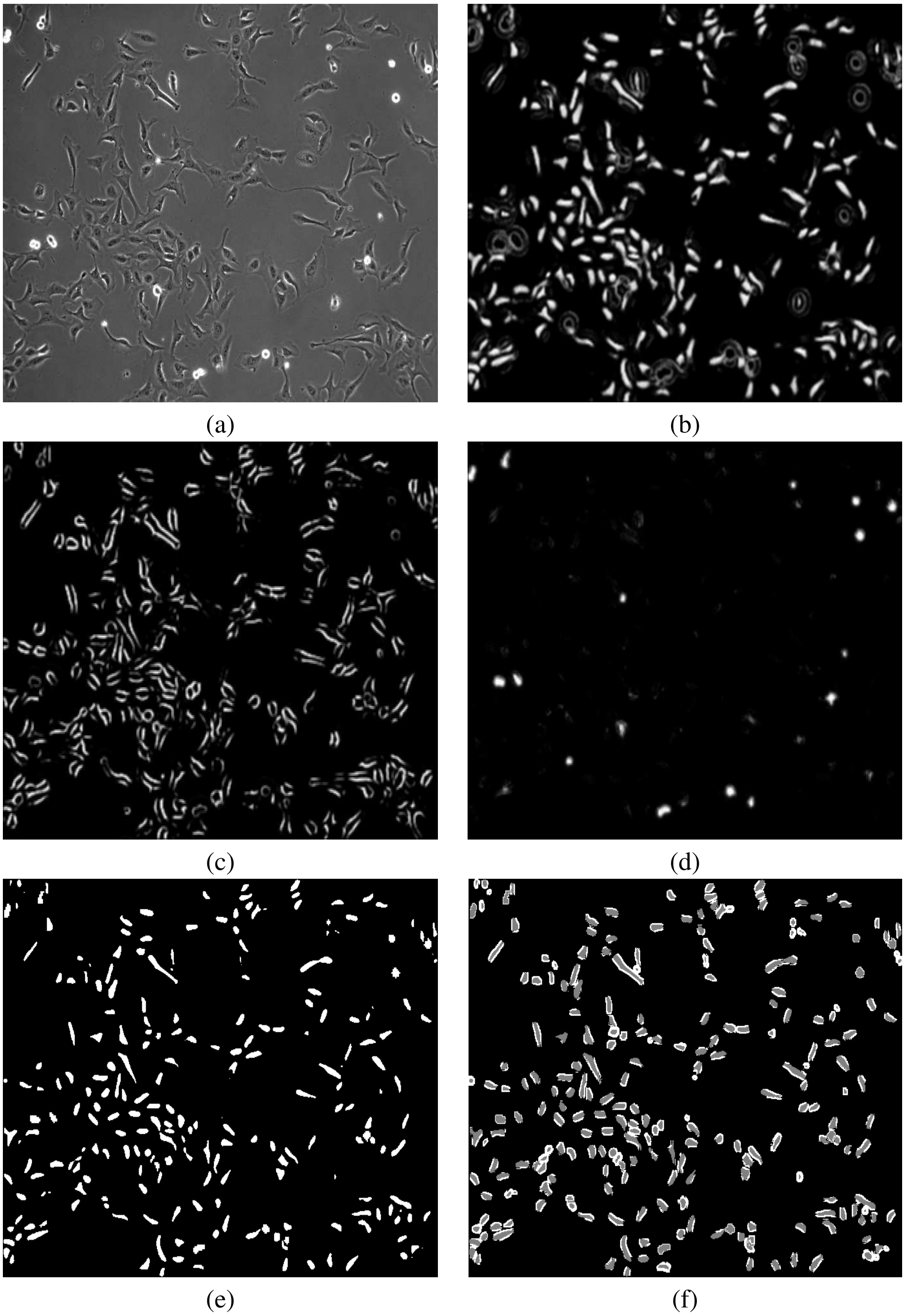


Figure 4. Pixel-wise probability map and clustering results. (a) Original image; (b) Normal cell; (c) Halo artifact; (d) Mitotic cell; (e) Clustering result; (f) Region dilation (white).

In RF training, we extract a set of features from the training images and train a set of trees $t \in \{1, \cdots, T\}$ independently on a randomly sampled subset of the training data. After training, each leaf node stores empirical distribution $p_t(c|v)$ of classes. Testing a new sample v starts from the root node of each tree and follows the splitting functions of each node until it reaches the leaf nodes. The final decision $P(c|v)$ is acquired for each class c by taking the average predication of all trees:

$$P(c|v) = \frac{1}{T}\sum_{t=1}^{T} p_t(c|v). \tag{1}$$

The RF classifier has two important parameters: T the number of trees, and fs the number of randomly sampled features per node. In this paper, the number of trees is set to 500, and $fs = 4\sqrt{l}$ where l is the feature vector length. For each pixel, the features are extracted at two scales of fixed window; 4×4 and 8×8. Here, we compute two features; the histogram of the pre-conditional image [13], and the Hessian matrix feature.

The Hessian matrix is a matrix of the second-order partial derivatives of image I. It has been successfully used to extract ridge features and vessel structures [6, 25]. Hessian matrix H can be written as:

$$H = \begin{pmatrix} I_{xx} & I_{xy} \\ I_{yx} & I_{yy} \end{pmatrix} \tag{2}$$

where $I_{ij} = \frac{\partial^2 I}{\partial i \partial j}$ and is obtained by using a convolution operation between the image I and the second derivative of the Gaussian filter. The eigenanalysis of the Hessian matrix is then performed, and the largest eigenvector/value is extracted as feature. The mitotic cell has a blob-like structure which can be detected by the Hessian matrix feature. Here, for each window scale, the scale parameter of the Gaussian filter is defined as 2 and 4, respectively.

The other extracted feature for the RF classifier is the histogram of the precondition image. The assumption is that we can restore an image s free from artifacts, i.e. halo effect and background noise:

$$u = As \tag{3}$$

where u is an observed image, and A is a matrix composed by the imaging optics. Solving the equation directly by inverting A is highly prone to noise. Instead, the restored image s can be found by minimizing the following quadratic function [13]:

$$O(s) = ||As - g||_2^2 + \omega_s s^{tr}\mathbf{L}s + \omega_r ||\mathbf{\Lambda} s||_1 \tag{4}$$

where $\mathbf{L}$ is defined as a Laplacian matrix to impose spatial smoothness between neighboring pixels, $\mathbf{\Lambda}$ is a diagonal matrix (positive) in the l_1-norm sparseness regularization, and ω_s and ω_r are weighting parameters. To find a close-form solution, the equation (4) is constrained to be nonnegative values and solved iteratively to obtain the restored phase contrast image. For each patch window, the histogram of the restored image is computed where the number of bins is 15 and the histogram is normalized by l_2-norm. Figure 3 shows an example of the large eigenvalue of the Hessian matrix and the restored image. Figures 4 (b), (c), (d) show the probability map from the RF classifier for normal cell, halo artifact, and mitotic cell, respectively.

2.2. Clustering & Region Growing

The RF classifier generates a probability map based on the extracted low level features. Thus, we can determine the probable cell location and discard the background regions. However, it is still difficult to identify the individual cell in densely populated areas. Here, spatial clustering is carried out to analyze the connected component structure and extract the individual cells.

The k-means algorithm is used to locate the central region of the normal and mitotic cells by partitioning the probability map into three classes. The output class is automatically selected by finding which cluster gives the maximum probability to contain cells based on the probability map computed from the first stage. The cluster is working as a binary mask that refers to the center region of candidate cells as shown in Figure 4 (e). Note that the normal (i.e. dark) and mitotic (i.e. bright) cells are clustered separately to reduce the effect of overlapping.

The cell growing process is then applied to extend the cell regions obtained from the k-means clustering to recover some of the missing cell region. Figure 4 (f) shows an example of the cell dilation. The probability maps for normal, mitotic, and halo effect are defined as P_d, P_b, and P_h, respectively. The cell growing process begins by converting these probability maps P_i into a binary

DOI: 10.1002/cnm

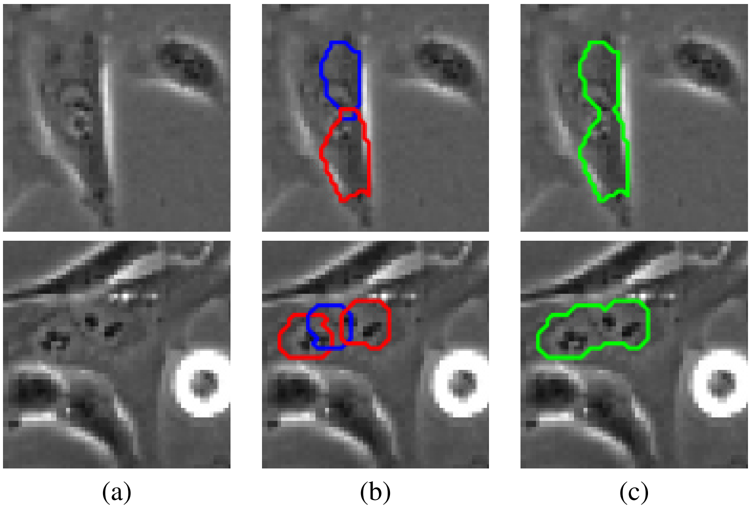


Figure 5. Touching candidate cell regions classified as a single cell in the cell validation stage. (a) An image window of 40×40 pixels. (b) Touching cell regions result after the region growing stage (red, blue). (c) The cell validation classifier tests whether the touching cells are actually a single cell.

mask:

$$M = \begin{cases} 1 & P_d + P_b + P_h > th_a, \\ 0 & otherwise \end{cases} \tag{5}$$

where M represents the searching space that can be picked up in the cell growing process and th_a is defined by applying Otsu's thresholding which splits M by maximizing the between-class variance. Each cell region obtained by the k-means clustering has a binary label R_i (1 for cell and 0 otherwise). The new cell region D_i is defined as:

$$D_i = \begin{cases} 1 & Ed(R_i) * M < th_d, \\ 0 & otherwise \end{cases} \tag{6}$$

where $Ed(.)$ is the Euclidean distance transformation of the cell binary region, and th_d is the searching space parameter that controls the dilation size. Since the aim is to refine the detected cell regions, the dilation parameter is usually small. Here, th_d sets to 5 for normal cell and 3 for the mitotic cell. The cell growing process can easily determine if candidate cell regions are touching each other and this helps in the following cell validation stage to identify if the cell region is broken into small subregions by combining those subregions and testing the united region, as shown in Figure 5.

2.3. *Cell Validation via Spatial Pyramid Coding*

The result of the last stage is a large set of candidate cell regions with different shape and size. In this stage, we propose a classification method based on a spatial pyramid feature coding to validate the cell identity. The aim is to classify the candidate cell regions into three categories: single cell, merged cell, and background. We define single cell (if it is not in mitosis phase) as the region that contains mainly the cell nucleus. The candidate cell regions that are touching after the cell growing phase are merged into a single region and tested in this stage to determine if the touching regions belong to the same cell or multiple cells. Figure 5 shows an example of touching candidate regions classified as a single cell. The proposed cell validation method is built on the bag-of-features (BoF) model. Various BoF representations have been applied successfully not only in object and scene classification [26, 27] but also in medical image classification [28]. The BoF model produces a histogram of feature coding of all locally extracted image descriptors. From a

DOI: 10.1002/cnm

set of training images, feature descriptors are extracted and then clustered into a set of codes called visual words (i.e. codebook). The image features are attached to the nearest code or visual word in the visual vocabulary. Accordingly, the image is represented as a set of codes and generalized as a histogram.

The basic BoF model eliminates the spatial information of the extracted local descriptors. However, including the spatial information has been shown to be useful for the image classification tasks, e.g. [27, 29]. Spatial pyramid coding (SPC) can be used to capture the local information of the extracted features. This method splits the image into sections in various scales, and then it produces a BoF histogram for each section and concatenates all the histograms to form the final representation of the image.

Support vector machines (SVM) classifier is usually adopted along with SPC by using nonlinear Mercer kernels, e.g. the Chi-square kernel. However, training a nonlinear classifier is computational expensive which limit the scalability of the SPC method. In this work, we use locality-constrained linear coding (LLC) [30] which is an SPC based approach that replaces the traditional vector quantization method by locality constraint linear sparse coding. LLC method performs very well with the linear SVM classifier and performs much more efficiently, e.g. as demonstrated in [30, 31, 32].

Cell validation consist of extracting the scale-invariant feature transform (SIFT) features, coding the features using LLC and applying SPC when integrating the final histogram image representations.

2.3.1. Dense SIFT

SIFT is extracting a set of local features that are invariant to scale and rotation. It has been successfully used in many applications, such as object recognition, and image stitching. The ordinary SIFT [33] consists of four steps: detecting scale-space extrema, localizing key points, assigning an overall orientation, and defining a local descriptor. Dense SIFT extracts local features regularly at each possible location, instead of at key point locations, to model a dense representation of the input image. Since extracting the same key points from the input cell images is difficult due to the large deformation of cell shapes, dense SIFT is adopted here.

The local descriptor is defined as a histogram of gradient orientations within a local region centered at the key point. The size of the local region is usually selected as 16×16. The local region is split into 4×4 sub-regions and for each sub-region, a histogram of edge directions with 8 bins is computed. These histograms are then concatenated to form a 128-dimensional feature vector and normalized to unit length to improve contrast and illumination invariancy.

2.3.2. Feature coding

Building visual vocabulary from a set of SIFT feature descriptors $X = \{x_1, x_2, \cdots, x_n\}$ is the second procedure in the proposed cell validation method. K-means algorithm is used to construct the visual vocabulary, $B = \{b_1, b_2, \cdots, b_K\}$. Traditionally, feature coding is achieved by vector quantization, where it assigns each feature vector to one cluster (i.e. visual word) based on a similarity measurement. However, this hard assignment results in loss of information. Yang *et al.* [31] proposed a relaxation of the hard quantization by adding a sparsity regularization term so that the coding problem is equivalent to the standard sparse coding:

$$\arg\min_{C} \sum_{i=1}^{n} \| x_i - Bc_i \|^2 + \lambda \| c_i \|_{\ell^1} \tag{7}$$

where B is a codebook, and $C = [c_1, c_2, \cdots, c_n]$ is a set of codes for feature descriptors X. The second term is the sparsity regularization that is ℓ^1 norm of codes C.

In this paper, LLC is used as the feature coding. In LLC [30], sparsity constraint is replaced with locality constraint to catch the correlations between similar feature descriptors that reduces the

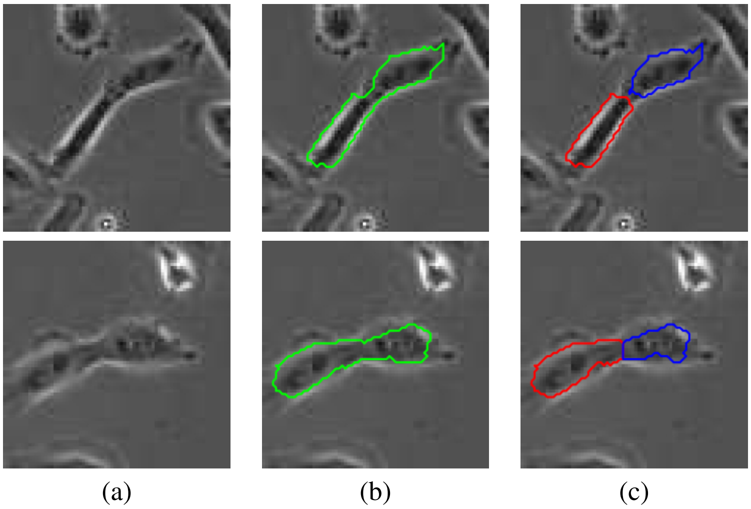


Figure 6. Splitting merged cell using seeded watershed method. (a) An image window of 75×75 pixels; (b) Merged cell (green); (c) After watershed segmentation, two separated cells are created (blue and red).

quantization errors. The optimization problem is defined as:

$$\arg\min_{C} \sum_{i=1}^{n} \parallel x_i - Bc_i \parallel^2 + \lambda \parallel d_i \odot c_i \parallel \tag{8}$$

where $\odot$ denotes to the element-wise multiplication, and d_i is the locality factor which is proportional to the similarity between feature descriptor x_i and codebook B.

2.3.3. Spatial pyramid coding

A pooling of features coding is used as input to the SVM classifier. But, this orderless representation of the features is losing important information about the spatial location of the features. In order to tackle this problem, spatial pyramid coding is used. In SPC, the image is divided into subregions at different resolutions $\{0, \cdots, L\}$ and the local features are then pooled in these subregions and normalized. The total number of subregions at a resolution level l is 2^{2l}. Here, max pooling is applied on the feature coding vectors located inside the same subregion.

SVM classifier [34] finds a hyperplane with the maximal margin that separates the feature space into two classes. The subset of data that locates the position of the hyperplane separator are called support vectors and used to define the decision function. Here, a linear SVM is used, as discussed earlier.

2.4. Splitting Merged Cells

Some cell regions are classified as merged cells by the BoF classifier used in the cell validation phase. In order to split the merged cells, we propose to use the seeded watershed segmentation. In the watershed segmentation algorithm, each local minima of the image gradient is considered as the starting point of water flows. The water is continually spreading to form catchment basins. The neighboring catchment basins will meet at some points forming a dam or watershed. The water will stop rising when reaching watersheds or pre-defined background regions. In the end, the image is segmented into enclosed regions by watersheds at each local minimum. However, the watershed algorithm tends to over-segment the image. In order to overcome this problem, we use the seeded watershed segmentation.

The seeded watershed algorithm uses the seed positions, which can be defined by the user interaction or some morphological operations, as the starting regions of water flooding. Seeds usually defined for both object (e.g. cell nucleus) and background. Here, object seeds are defined

DOI: 10.1002/cnm

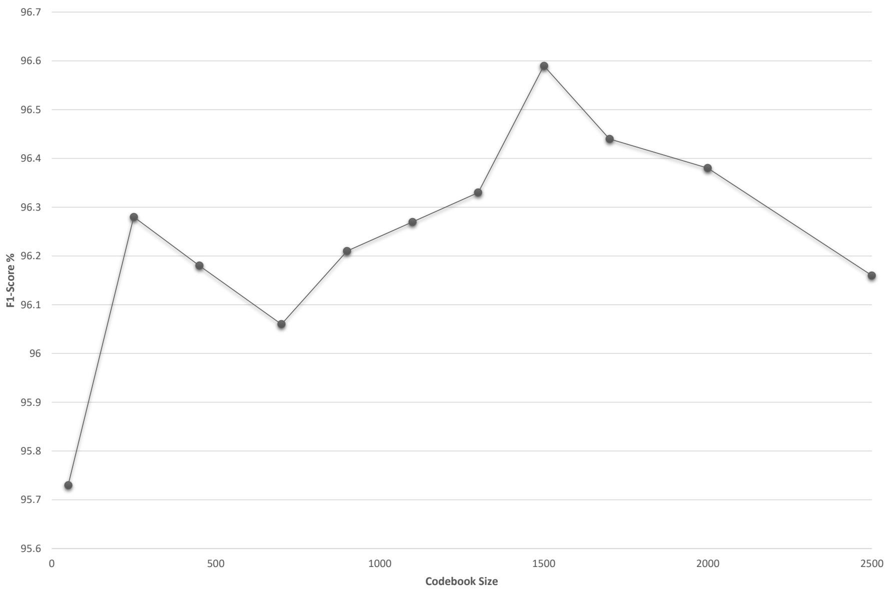


Figure 7. Codebook size vs. F1-Score.

using the h-maxima transform of the probability map of the merged cells generated by the pixel-wise classifier as discussed in Section 2.1. The h-maxima suppress all the local maxima in the probability map that has a height less than h level. A high h-value means the merged cells will remain unchanged with one seed. Since the probability map value is in the range between 0 to 1, h is defined at 0.20 to keep the regions with high probability as seeds. The background seeds defined at the regions that have the lowest probability (i.e. equal to zero). The object and background seeds information are combined and used as input for watershed segmentation. Figure 6 shows some examples of merged cells detected during the validation phase and after applying the seeded watershed algorithm.

3. EXPERIMENTAL RESULTS

3.1. Dataset and Evaluation

The proposed method is tested on three datasets of phase contrast images: U2OS human osteosarcoma cells, human cervical cancer HeLa cells, and NIH 3T3 fibroblast live cells. These datasets include many challenging problems, such as high density of cells, deformable cell shapes, a variability of cell sizes, and missing boundaries.

The U2OS dataset contains 97 time-lapse images recorded at a resolution of 672×512 pixels. Only 10 images are assigned for training and the rest for testing. The cells are growing from 90+ to 400+ cells. The groundtruth carried out every 10th frame, in total 2431 cells are annotated to perform the quantitative analysis. The HeLa dataset is provided with Arteta *et al.* method[†]. It contains 22 phase contrast images separated equally into training and testing. The total number of the annotated cells for evaluation is 1073. The groundtruth manually defines the cell nuclei. The NIH 3T3 dataset[‡] contains 238 images acquired every quarter hour for more than 62 hours using a 10X (0.3 NA) objective on Zeiss Axiovert 200M microscope. The evaluation is carried out every 10 frames (i.e.

[†] http://www.robots.ox.ac.uk/~vgg/software/cell_detection/
[‡] https://isg.nist.gov/deepzoomweb/data/dissemination

23 images) with a total number of 790 annotated cells. The last 6 images in the sequence are used for training. Figure 1 shows an example of the phase contrast images from the three datasets.

An automated segmented cell is a truly detected cell (TDC), if the cell center is very close to a point in the groundtruth (i.e. radius distance $\leq p$). If not, the cell is a false detected cell (FDC). The remaining groundtruth centers without a matching is considered as a false negative cell (FNC). Here, five evaluation metrics are used to measure cell detection performance: precision (%), recall (%), F_1-score (%), Euclidean distance D_e between detected cell centers and the matched ground truth, and also, the absolute difference D_n of the ground truth and detected cell numbers per frame. For each dataset, p is defined as the average radius of the single objects. Precision, recall and F_1-score are defined as:

$$Precision = \frac{TDC}{TDC + FDC}, \tag{9}$$

$$Recall = \frac{TDC}{TDC + FNC}, \tag{10}$$

$$F_1 = 2 * \frac{Precision * Recall}{Precision + Recall}. \tag{11}$$

Table I. U2OS cell detection quantitative comparison. Using Precision (%), recall (%), F_1-score (%), and mean ± standard deviation of D_e and D_n.

	Precision	Recall	F_1-score	$\mu_{D_e} \pm \sigma_{D_e}$	$\mu_{D_n} \pm \sigma_{D_n}$
Yin *et al.* Method [13]	83.26	87.58	85.36	3.50 ± 2.68	13.6 ± 8.78
Arteta *et al.* “singletons” Method [23]	94.63	84.12	89.07	3.74 ± 2.89	27 ± 9.54
FogBank Method [3]	81.99	92.51	86.93	4.48 ± 3.05	31.20 ± 16.23
Bensch *et al.* [12] (Asym. cost)	87.62	86.80	87.21	4.15 ± 3.14	5.30 ± 4.29
Bensch *et al.* [12] (Sym. cost)	93.65	83.18	88.10	4.14 ± 2.88	27.2 ± 11.89
Proposed Method	95.98	97.20	96.59	2.94 ± 2.07	4.3 ± 2.83

3.2. *U2OS cells*

Figure 7 shows the effect of different codebook size on the BoF classifier in terms of F_1 measure for the detected cells. The codebook size of 1500 gives the best result where F_1-score is 96.59%. The performance beyond 1500 codebook seems show no improvements.

The proposed method is compared to four different methods as shown in Table I: Yin *et al.* method [13], Arteta *et al.* “singletons” method [23], FogBank method[§] [3] and Bensch *et al.* [12] method[¶]. In Yin *et al.* [13], an image restoration technique is introduced. The final cell region is obtained by applying adaptive thresholding. In Arteta *et al.* [23], the authors used an SVM classifier to score the candidate cell regions obtained by the MSER method. Then the optimal cell regions are selected by using dynamic programming. We refer this method as “singletons”. FogBank method [3] applied a similar principle of the watershed morphology by detecting some seed points and cluster each pixel based on its distance to the seed points. This method involves a large number parameter. However, the parameter settings are not provided in [3]. For fair comparison, we empirically test their method with different parameters on the training set and use the ones achieved the best result for comparison. In Bensch *et al.* [12], we tested two different configurations for the cost function: asymmetric and symmetric cost functions. Asymmetric boundary cost means the cost of the transition between two neighboring nodes in one direction is not the same as in the reverse direction. Asymmetric

[§] https://isg.nist.gov/deepzoomweb/resources/csmet/pages/fogbank_segmentation/fogbank_segmentation.html

[¶] http://lmb.informatik.uni-freiburg.de/Publications/2015/BR15/

DOI: 10.1002/cnm

cost endorses dark-bright intensity transitions. The parameters are adjusted using a grid search as suggested by the authors.

As presented in Table I, the overall evaluation metrics show better performance of the proposed method over the state of art. The precision and recall of the proposed method are 95.98%, 97.20%, compared to 83.26%, 87.58% for Yin *et al.* method, 94.63%, 84.12% for Arteta *et al.* method, 81.99%, 92.51% for FogBank method, and 87.62%, 86.80% for Bensch *et al.* method using asymmetric cost. We found that asymmetric boundary cost of Bensch *et al.* method is no better than the symmetric cost. In fact, symmetric boundary cost provided better cell detection results. Table I also shows the mean and standard deviation, D_e and D_n, for each method. The proposed method has a smaller distance 2.94 ± 2.07 between detected cell centers and the groundtruth than all other methods. In terms of the absolute difference between the number of the detected cells and the groundtruth, the proposed method has much smaller difference of 4.3 ± 2.83, compared to 13.6 ± 8.78 for Yin *et al.* method, and 27 ± 9.54 for Arteta *et al.* method.

Figure 8 shows some typical results of the proposed method, Yin *et al.*, and Arteta *et al.* methods. The segmented cell is highlighted using green contour. The groundtruth points and the detected cell center are shown in yellow and red, while the false detected cell in magenta and the false negative in blue. In contrast to Yin *et al.* and Arteta *et al.* methods, the proposed method is effective in detecting both normal migration cell and mitotic (bright) cell. Arteta *et al.* method tends to over-segment the cell. The identification and segmentation of the cells provide the basis for tracking the cell growth over time.

3.3. HeLa cells

The proposed method is compared to four cell detection methods: CellStat method [35], Arteta *et al.* "singletons" method [23], Akram *et al.* method [22], and SemiPlanarCC [20]. The F_1-score is used to evaluate the cell detection as commonly used in the literature.

As shown in Table II, the proposed method achieves a good performance compared to the other detection methods. CellStat applied adaptive thresholding to segment the image into clusters. A circular hough transform is used to detect cell centers, then the cell contour is recovered by using a dynamic programming method. However, it has a poor performance with F_1-score of 35.0% since determining cell centers from a large cluster is problematic. Also, the method cannot segment low contrast cells. Arteta *et al.* method performed better with F_1-score of 87.0%, however, it failed to highlight the bright cells. Akram *et al.* used a convolutional neural network for cell detection and non-maxima suppression to remove duplicate cell regions. The F_1-score is 93.20%. SemiPlanarCC [20] computes superpixels of the cell boundary probability map to construct a weighted region adjacency graph that converts the segmentation problem into graph partitioning and solves it by a correlation clustering method. It also adds a constraint on the valid cell size. However, it tends to produce many false-positive cell regions. The proposed method achieves F_1-score 96.29% compared to SemiPlanarCC F_1-score 95.0%.

Table II. The quantitative comparison of HeLa cell detection.

	F_1-score (%)
CellStat Method [35]	35.0
Arteta *et al.* "singletons" Method [23]	87.0
Akram *et al.* Method[22]	93.20
SemiPlanarCC [20]	95.0
Proposed Method	96.29

3.4. NIH 3T3 cells

We also test the proposed method on a publicity available dataset, NIH 3T3 fibroblast cells. The images have less populated cells but have strong halo effect. The proposed method was compared to two different methods: FogBank method [3], and Bensch *et al.* [12] method, as shown in table III. The proposed method performed better in F_1-score and recall: 90.92% and 94.43% respectively, while Bensch *et al.* [12] with asymmetric cost gave better precision at 94.21%. Figure 10 shows a comparison between the proposed method, FogBank method [3], and Bensch *et al.* [12] method with symmetric boundary cost. FogBank method tends to over segment the cells especially when cells are touching each other. Bensch *et al.* [12] method suffered from higher false negative cells since it can not separate the touching cells. Overall, the proposed method performed better despite the strong halo effect.

Table III. NIH 3T3 cell detection quantitative comparison.

	Precision	Recall	F_1-score	$\mu_{D_e} \pm \sigma_{D_e}$	$\mu_{D_n} \pm \sigma_{D_n}$
FogBank Method [3]	81.96	87.67	84.72	4.14 ± 4.87	2.83 ± 2.20
Bensch *et al.* [12] (Asym. cost)	94.21	79.08	85.99	4.12 ± 4.23	5.45 ± 5.46
Bensch *et al.* [12] (Sym. cost)	89.62	86.05	87.80	3.63 ± 3.81	2.91 ± 3.43
Proposed Method	87.66	94.43	90.92	3.62 ± 3.31	2.65 ± 2.18

4. CONCLUSION

We presented an automated segmentation method that combines two machine learning methods based on RF and BoF classifiers in order to detect cells in microscopy phase contrast images. We adopted RF classifier to create a probability map of cell locations. K-means algorithm and region-growing are proposed to form candidate cell regions. The BoF based on spatial pyramid coding features is used to validate the cell region and detect the merged cell which can be split later by using a seeded watershed algorithm. The proposed method showed better results compared to the state-of-the-art cell detection methods.

ACKNOWLEDGMENT

We are grateful to Rachel Errington and Nick White (Cardiff University, UK) for providing U2OS dataset. We thank Carlos Arteta (Oxford University, UK) for interesting discussion regarding the related work.

REFERENCES

1. Dewan M, Ahmad M, Swamy M. Tracking biological cells in time-lapse microscopy: An adaptive technique combining motion and topological features. *IEEE Transactions on Biomedical Engineering* 2011; **58**(6):1637–1647.
2. Debeir O, Adanja I, Warzee N, Van Ham P, Decaestecker C. Phase contrast image segmentation by weak watershed transform assembly. *IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, 2008; 724–727.
3. Chalfoun J, Majurski M, Dima A, Stuelten C, Peskin A, Brady M. Fogbank: a single cell segmentation across multiple cell lines and image modalities. *BMC Bioinformatics* 2014; **15**(1):431.
4. CHALFOUN J, MAJURSKI M, PESKIN A, BREEN C, BAJCSY P, BRADY M. Empirical gradient threshold technique for automated segmentation across image modalities and cell lines. *Journal of Microscopy* 2015; **260**(1):86–99.
5. Xie X, Mirmehdi M. Level-set based geometric colour snake with region support. *ICIP*, 2003; 153–156.
6. Ersoy I, Bunyak F, Mackey M, Palaniappan K. Cell segmentation using hessian-based detection and contour evolution with directional derivatives. *ICIP*, 2008; 1804–1807.

DOI: 10.1002/cnm

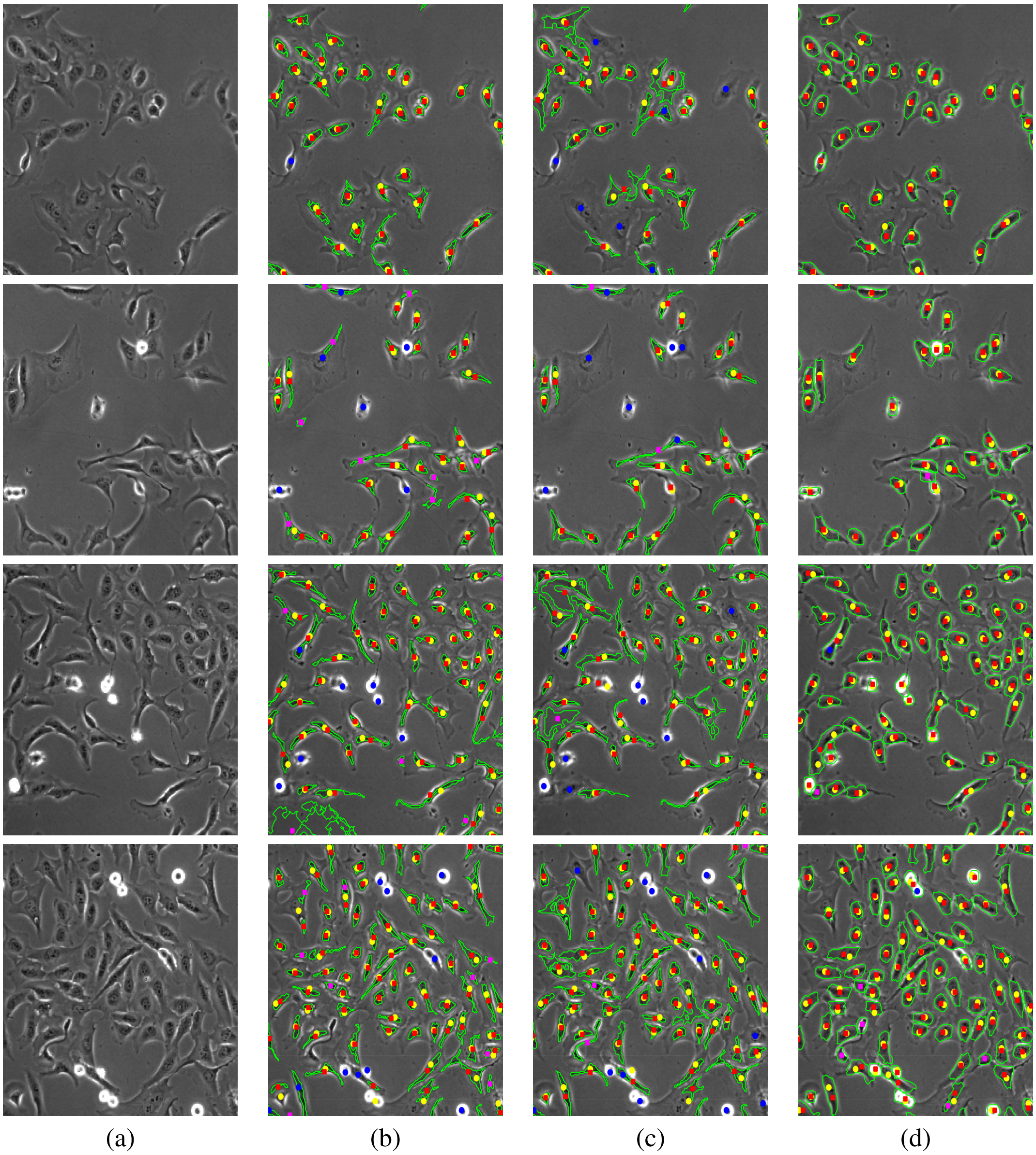


Figure 8. Comparison between the proposed method and two different cell detection methods. The segmented cell is highlighted by green color, the groundtruth and the detected cell centers shown in yellow and red, the false detected cell in magenta color and the false negative cell in blue color. (a) An image window of 197 × 197 pixels. (b) Yin *et al.* method; (c) Arteta *et al.* "singletons" method; (d) Proposed method.

7. Li F, Zhou X, Zhao H, Wong STC. Cell segmentation using front vector flow guided active contours. *MICCAI*, 2009; 609–616.
8. Seroussi I, Veikherman D, Ofer N, Yehudai-Resheff S, Keren K. Segmentation and tracking of live cells in phase-contrast images using directional gradient vector flow for snakes. *Journal of Microscopy* 2012; **247**(2):137–146.
9. Russell C, Metaxas D, Restif C, Torr P. Using the pn potts model with learning methods to segment live cell images. *ICCV*, 2007; 1–8.
10. Pan J, Kanade T, Chen M. Heterogeneous conditional random field: Realizing joint detection and segmentation of cell regions in microscopic images. *CVPR*, 2010; 2940–2947.
11. Massoudi A, Sowmya A, Mele K, Semenovich D. Employing temporal information for cell segmentation using max-flow/min-cut in phase-contrast video microscopy. *EMBC*, 2011; 5985–5988.

DOI: 10.1002/cnm

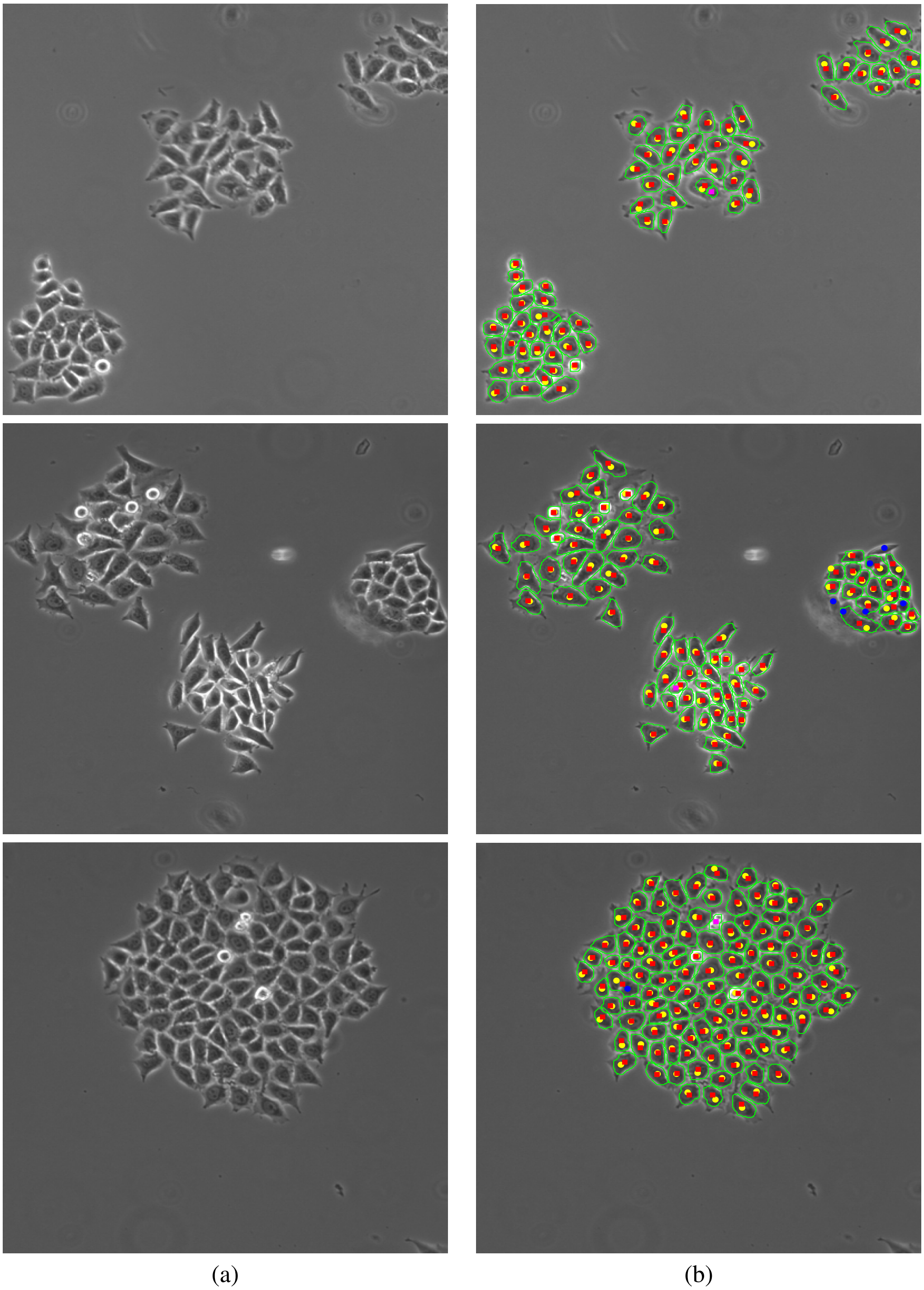


Figure 9. Examples of HeLa cell detection. Ground-truth (yellow), true positive cell (red), false positive cell (magenta), false negative cell (blue) and segmentation results (green). (a) Original image. (b) Proposed method.

12. Bensch R, Ronneberger O. Cell segmentation and tracking in phase contrast images using graph cut with asymmetric boundary costs. *IEEE 12th International Symposium on Biomedical Imaging (ISBI)*, 2015; 1220–1223.
13. Yin Z, Kanade T, Chen M. Understanding the phase contrast optics to restore artifact-free microscopy images for segmentation. *Medical Image Analysis* 2012; **16**(5):1047 – 1062.

DOI: 10.1002/cnm

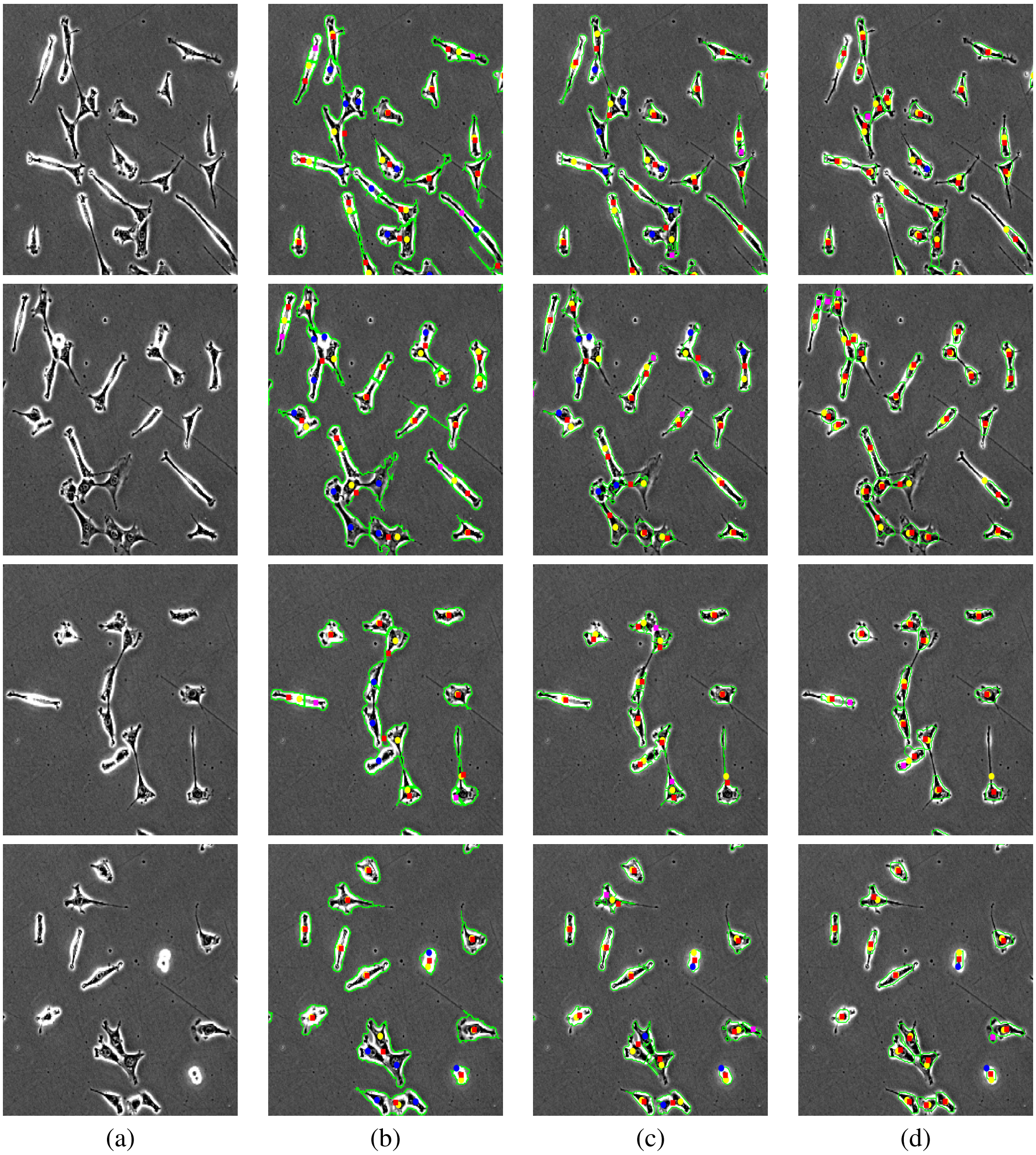


Figure 10. Comparison between the proposed method and FogBank method [3], and Bensch *et al.* [12] method with symmetric boundary cost. The segmented cell is highlighted by green color, the groundtruth and the detected cell centers shown in yellow and red color, the false detected cell in magenta color and the false negative cell in blue color. (a) An image window of 197 × 197 pixels. (b) FogBank method; (c) Bensch *et al.* method with symmetric boundary cost; (d) Proposed method.

14. Su H, Yin Z, Kanade T, Huh S. *MICCAI*, chap. Phase Contrast Image Restoration via Dictionary Representation of Diffraction Patterns. Springer Berlin Heidelberg, 2012; 615–622.
15. Su H, Yin Z, Huh S, Kanade T. Cell segmentation in phase contrast microscopy images via semi-supervised classification over optics-related features. *Medical Image Analysis* 2013; **17**(7):746 – 765.
16. Pan J, Kanade T, Chen M. Learning to detect different types of cells under phase contrast microscopy. *Microscopic Image Analysis with Applications in Biology Workshop*, 2009.
17. He W, Wang X, Metaxas D, Mathew R, White E. Cell segmentation for division rate estimation in computerized video time-lapse microscopy. *SPIE*, vol. 6431, 2007; 1–8.
18. Arteta C, Lempitsky V, Noble J, Zisserman A. Learning to detect cells using non-overlapping extremal regions. *MICCAI*, *Lecture Notes in Computer Science*, vol. 7510. 2012; 348–356.

19. Nketia T, Rittsher J, Noble JA. Utilizing phase retardation features for segmenting cells in phase contrast microscopy images. *MIUA*, 2014; 191–196.
20. Zhang C, Yarkony J, Hamprecht F. Cell detection and segmentation using correlation clustering. *MICCAI*, *Lecture Notes in Computer Science*, vol. 8673. 2014; 9–16.
21. Mualla F, Schöll S, Sommerfeldt B, Maier A, Steidl S, Buchholz R, Hornegger J. Unsupervised unstained cell detection by sift keypoint clustering and self-labeling algorithm. *MICCAI*, *Lecture Notes in Computer Science*, vol. 8675. 2014; 377–384.
22. Akram SU, Kannala J, Eklund L, Heikkilä J. Cell segmentation proposal network for microscopy image analysis. *Workshop on Deep Learning in Medical Image Analysis (DLMIA)*, 2016; 21–29.
23. Arteta C, Lempitsky V, Noble JA, Zisserman A. Detecting overlapping instances in microscopy images using extremal region trees. *Medical Image Analysis* 2016; **27**:3 – 16.
24. Breiman L. Random forests. *Machine Learning* 2001; **45**(1):5–32.
25. Frangi AF, Niessen WJ, Vincken KL, Viergever MA. *MICCAI*, chap. Multiscale vessel enhancement filtering. Springer Berlin Heidelberg: Berlin, Heidelberg, 1998; 130–137.
26. Csurka G, Dance CR, Fan L, Willamowski J, Bray C. Visual categorization with bags of keypoints. *In Workshop on Statistical Learning in Computer Vision, ECCV*, 2004; 1–22.
27. Lazebnik S, Schmid C, Ponce J. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. *CVPR*, 2006; 2169–2178.
28. Barata C, Emre Celebi M, Marques J. Improving dermoscopy image classification using color constancy. *IEEE Journal of Biomedical and Health Informatics* 2015; **19**(3):1146–1152.
29. Huang Y, Wu Z, Wang L, Tan T. Feature coding in image classification: A comprehensive study. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 2014; **36**(3):493–506.
30. Wang J, Yang J, Yu K, Lv F, Huang T, Gong Y. Locality-constrained linear coding for image classification. *CVPR*, 2010; 3360–3367.
31. Yang J, Yu K, Gong Y, Huang T. Linear spatial pyramid matching using sparse coding for image classification. *CVPR*, 2009; 1794–1801.
32. Chatfield K, Lempitsky V, Vedaldi A, Zisserman A. The devil is in the details: an evaluation of recent feature encoding methods. *Proceedings of the British Machine Vision Conference*, 2011; 76.1–76.12.
33. Lowe DG. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision* 2004; **60**(2):91–110.
34. Cortes C, Vapnik V. Support-vector networks. *Machine Learning* 1995; **20**(3):273–297.
35. Kvarnström M, Logg K, Diez A, Bodvard K, Käll M. Image analysis algorithms for cell contour recognition in budding yeast. *Opt. Express* 2008; **16**(17):12 943–12 957.