

Chapter 5

A Novel Cell Detection Method Using Deep Convolutional Neural Network and Maximum-Weight Independent Set

Fujun Liu and Lin Yang

Abstract Cell detection is an important topic in biomedical image analysis and it is often the prerequisite for the following segmentation or classification procedures. In this chapter, we propose a novel algorithm for general cell detection problem: First, a set of cell detection candidates is generated using different algorithms with varying parameters. Second, each candidate is assigned a score by a trained deep convolutional neural network (DCNN). Finally, a subset of best detection results are selected from all candidates to compose the final cell detection results. The subset selection task is formalized as a maximum-weight independent set problem, which is designed to find the heaviest subset of mutually nonadjacent nodes in a graph. Experiments show that the proposed general cell detection algorithm provides detection results that are dramatically better than any individual cell detection algorithm.

5.1 Introduction

Cell detection is an important topic in biomedical image analysis because it is often the first step for the following tasks, including cell counting, segmentation, and morphological analysis. Many automatic cell detection algorithms are proposed in recent literatures [1–3]. Parvin et al. proposed an iterative voting algorithm based on oriented kernels to localize cell centers, in which the voting direction and areas were dynamically updated within each iteration. In [2], a simple and reliable cell detector was designed based on a Laplacian of Gaussian filter. A learning-based cell detection algorithm was proposed in [3]. It used an efficient maximally stable extremal regions (MSER) detector [4] to find a set of nested candidate regions that will form a tree

F. Liu · L. Yang (✉)

Department of Electrical & Computer Engineering, Medford, MA, USA

e-mail: Lin.Yang@bme.ufl.edu

L. Yang

J. Crayton Pruitt Family Department of Biomedical Engineering, University of Florida, Gainesville, FL, USA

© Springer International Publishing Switzerland 2017

L. Lu et al. (eds.), *Deep Learning and Convolutional Neural Networks for Medical Image Computing*, Advances in Computer Vision and Pattern Recognition, DOI 10.1007/978-3-319-42999-1_5

graph. Then a nonoverlapping subset of those regions was selected for cell detection via dynamic programming.

All the methods reviewed above give good detection results under certain circumstances. However, in general, they all have some limitations. For example, both [1] and [2] are sensitive to the selection of proper cell diameter parameters. However, finding an appropriate parameter that works under all conditions is extremely difficult when the cells exhibit large size variations. In [3], the algorithm heavily depends on the quality of MSER detector that does not take advantage the prior cell shape information and the performance will deteriorate when the cells overlap with one another.

In this chapter, we propose a novel algorithm for general cell detection that does not require the fine tuning of parameters. First, a set of cell detection candidates is produced from different algorithms with varying parameters. Second, each candidate will be assigned a score using a trained deep convolutional neural network (DCNN) [5, 6]. Third, we will construct a weighted graph that has the detection candidates as nodes and the detection scores (DCNN outputs) as weights (an edge exists between two nodes if their corresponding detection results lie in the same cell). Finally, a subset of mutually nonadjacent graph nodes is chosen to maximize the sum of the weights of the selected nodes. An overview of the algorithm is shown in Fig. 5.1. The selection of the best subset is formulated as a maximum-weight independent set problem (MWIS). MWIS is a combinatorial optimization problem that has been successfully applied in clustering [7], segmentation [8], and tracking [9], etc.

To the best of our knowledge, this is the first work that formulates the general cell detection problem as a MWIS problem, and this is also the first work to introduce DCNN to provide weights to a graph for future combinatorial optimization.

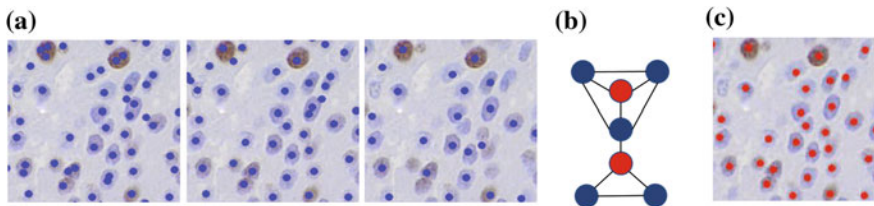


Fig. 5.1 An overview of the proposed general cell detection algorithm. **a** A set of detections candidates generated using multiple detection algorithms. Each candidate is marked with *blue dot*. **b** An undirected weighted graph was constructed from all detection candidates. The *red color* indicates the selected nodes. **c** The final cell detection results using the proposed method

5.2 Methodology

5.2.1 Cell Detection Using MWIS

A set of cell detection candidates (points), $P = \{p_1, \dots, p_n\}$, are first generated based on different cell detection algorithms with various parameters. An undirected and weighted graph, $G = (V, E, w)$, is constructed, where the node v_i corresponds to the i -th cell detection candidate p_i , E denotes undirected edges between nodes, and w_i denotes weight for the i -th node v_i . Two nodes v_i and v_j are adjacent, $(v_i, v_j) \in E$, if the Euclidean distance between their respective detection results p_i and p_j is smaller than a threshold λ . A node v_i will be assigned a larger weight value w_i if its corresponding detection result p_i is close to the real cell center, otherwise smaller weight will be assigned. After graph G is constructed, an optimal subset of V will be selected with the constraint that two nodes adjacent to each other will not be selected simultaneously. A subset is represented by an indicator vector $\mathbf{x} = \{x_1, \dots, x_i, \dots, x_n\}$, where $x_i \in \{0, 1\}$. $x_i = 1$ indicates that node v_i is in the subset, and $x_i = 0$ represents that v_i is not in the subset. This best subset selection is then formulated as finding the maximum-weight independent set (MWIS) \mathbf{x}^* .

$$\mathbf{x}^* = \arg \max_{\mathbf{x}} w^T \mathbf{x}, \text{ s. t. } \mathbf{x}^T A \mathbf{x} = 0, x_i \in \{0, 1\}, \quad (5.1)$$

where $A = (a_{ij})_{n \times n}$ is the adjacent matrix, $a_{ij} = 1$ if $(v_i, v_j) \in E$ and $a_{ij} = 0$ otherwise. The diagonal elements of A are zeros. The quadric constraints can be integrated into the object function to reformulate the optimization as

$$\mathbf{x}^* = \arg \max_{\mathbf{x}} \left(w^T \mathbf{x} - \frac{1}{2} \alpha \mathbf{x}^T A \mathbf{x} \right), \text{ s. t. } x_i \in \{0, 1\}, \quad (5.2)$$

where α is a positive regularization parameter to encode the nonadjacent constraints in (5.1).

The MWIS optimization can be solved by some numerical approximation algorithms [8, 10]. In [10], the integer constraints in (5.2) are relaxed, and a graduated assignment algorithm iteratively maximizes a Taylor series expansion of the object function in (5.2) around the previous solution in the continuous domain. The relaxed continuous solution will then be binarized to obtain the discrete solution. This binarization procedure might lead to errors. In order to avoid this type of error, [8] directly seeks a discrete solution in each iteration in maximizing the Taylor series approximation. However, in this case the solution of (5.2) might not satisfy the nonadjacent constraints in (5.1). In our algorithm, unlike all the previous procedures, we propose to find the optimal results iteratively only in the solution space of (5.1).

Denote $f(\mathbf{x})$ as the objective function in (5.2), let $x^{(t)} \in \{0, 1\}^n$ denotes the current solution in the t -th iteration, each iteration consists of the following two steps in our algorithm.

Step 1: For any point $x \in \{0, 1\}^n$ in the neighborhood of $x^{(t)}$, we first find the first-order Taylor series approximation of $f(x)$ as

$$f(\mathbf{x}) \approx T(\mathbf{x}) = f(x^{(t)}) + (x - x^{(t)})^T (w - \alpha Ax^{(t)}) = x^T (w - \alpha Ax^{(t)}) + \text{const}, \quad (5.3)$$

where *const* represents an item that does not depend on x . Define $y^{(t)}$ as the intermediate solution to (5.3), it can be computed by maximizing the approximation $T(x)$ as $y^{(t)} = \mathbb{1}(w - \alpha Ax^{(t)} \geq 0)$, where $\mathbb{1}(\cdot)$ is an indicator function.

Step 2: The solution of (5.3) might not satisfy the nonadjacent constraints listed in (5.1). If this is the case, we need to find a valid solution of (5.1) based on $y^{(t)}$. This is achieved by the following steps: (1) We first sort all the nodes based on their weights with a decreasing order. The nodes with $y^{(t)} = 1$ will be placed in front of the nodes that have $y^{(t)} = 0$. (2) The nodes are then selected from the front of the queue sequentially with a constraint that the picked node will not be adjacent to those that are already chosen.

After we find the valid solution, the $x^{(t+1)}$ in the solution space of (5.1) based on $y^{(t)}$ is computed using a local search method by first randomly removing k selected nodes and the probability to remove each node is inversely proportional to its weight, then choosing the maximum weighted node in the queue that are not adjacent to those selected until all nodes are considered. This procedure continues until convergence or maximum iterations reached and the best solution is selected as $x^{(t+1)}$. The reason that we randomly remove k selected nodes is to help the optimization escape from potential local maxima.

5.2.2 Deep Convolutional Neural Network

In this section, we need to calculate the weight w_i for each detection candidate $v_i \in V$ from Sect. 5.2.1. A deep convolutional neural network (DCNN) is trained for this purpose to assign each node a proper score as its weight. In our algorithm, a detection candidate is described by a small rectangle region centering around the detected position. Some training samples are shown in the first row in Fig. 5.2. The patches whose centers are close to the true cell centers are annotated as positive (+1) samples, marked with red rectangles in Fig. 5.2. Patches that have centers far away from true cell centers will be annotated as negative (-1) samples, marked with blue rectangles in Fig. 5.2.

DCNN Architecture: In our algorithm, the input features are the raw intensities of 31×31 image patches around the detected position. Considering the staining variations and the generality of the detection framework, color information is disregarded since they may change dramatically with respect to different staining protocols. The DCNN consists of seven layers: three convolutional (C) layers, two pooling layers, and two fully connected (FC) layers. In our implementation, max pooling (MP) is applied. The MP layers select the maximal activations over nonoverlapping patches

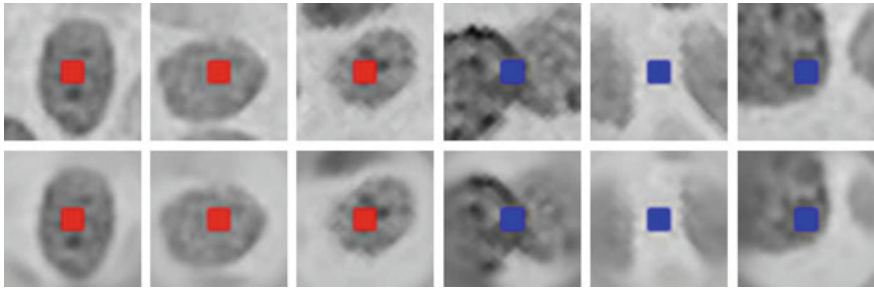


Fig. 5.2 Some training samples for DCNN and their foveation versions. The first row denote original training samples. Positive samples are marked with *red rectangles* and negatives are marked with *blue*. The second row denote the samples after foveation

Table 5.1 The configuration of the proposed DCNN architecture in our algorithm

Layer	Type	Maps (M) and neurons (N)	Filter size	Nonlinearity	Weights
0	I	1M × 31N × 31N	–	–	–
1	C	6M × 28N × 28N	4 × 4	Rectifier	102
2	MP	6M × 14N × 14N	2 × 2	–	–
3	C	12M × 12N × 12N	3 × 3	Rectifier	660
4	MP	12M × 6N × 6N	2 × 2	–	–
5	C	12M × 4N × 4N	3 × 3	Rectifier	1308
6	FC	100N	1 × 1	Rectifier	19300
7	FC	2N	1 × 1	Softmax	202

of the input layers. Except the output layer, where the two-way *softmax* function is used as activation function, the rectifier nonlinear activation functions are used in the convolutional layers and the fully connected layer prior to the output layer. A detailed configuration of the DCNN used in our algorithm is shown in Table 5.1.

Foveation: The task of DCNN is to classify the center pixel of each rectangle patch, so it will be ideal if we can keep the focus on the central region (fovea) and also retain the general structure of the image. Foveation, inspired by the structure of human photoreceptor topography, has been shown to be effective in imposing a spatially variant blur on images [11]. In our algorithm, a Gaussian pyramid is first built for each input image, then all the pyramid layers are resized to the input image scale. In the foveated image, pixels closer to the image center are assigned intensity values in higher resolution layers at the same coordinate, pixels far away

from the centers will be assigned values from lower resolution layers. Some foveation examples are shown in the second row of Fig. 5.2.

DCNN Training: Several cell detection algorithms [1, 2] with varying parameters are chosen to generate training samples for the DCNN. All the true cell centers are manually annotated in training images. The detected results within a certain distance τ_1 to the annotated cell centers are marked as positive training samples, others that locate far away from the centers (measured by τ_2) are marked as negative training samples, where $\tau_2 \geq \tau_1$. Each training sample is further rotated by seven angles. In our implementation, a mini-batch of size 10, which is a compromise between the standard and stochastic gradient descent forms, is used to train the DCNN. The learning rate is initiated as 0.01, and decreases as the number of epoches increases.

5.3 Experiments

The proposed algorithm is tested with two datasets: (1) 24 neuroendocrine (NET) tissue microarray (TMA) images, and (2) 16 lung cancer images. Each image contains roughly 150 cells. For each dataset, twofold cross-validation is used to evaluate the accuracy. All the true cell centers are manually labeled by doctors. An automatic detection is considered as true positive (*TP*) if the detected result is within a circle centered at the ground-truth annotation with a radius r . The detected results that do not fall into the circle will be labeled as false positive (*FP*). All missed true cell centers are counted as false negative (*FN*). The results are reported in terms of precision ($P = \frac{TP}{TP+FP}$) and recall ($R = \frac{TP}{TP+FN}$). Both the maximum-weight independent set (MWIS) and the deep convolutional neural network (DCNN) are evaluated in the following sections.

First, in order to justify the proposed two-step iterative algorithm to solve the MWIS problem stated in Eq. (5.1), we have compared it with a commonly used greedy non-maximum suppression (NMS) method [6], which keeps selecting an available node with the highest score and then removing the node and its neighbors until all the nodes are checked. As defined before, two nodes are considered as neighbors if their Euclidean distance is smaller than a threshold parameter λ . Taking detection results obtained from [1–3] as inputs, we generate a set of detection results for both the proposed algorithm and NMS by changing the parameter λ . The comparison of the converged object function values of Eq. (5.1) achieved by the proposed algorithm (Ours) and NMS are shown in Fig. 5.3a, d. The comparative results of detection accuracy (F_1 score) are shown in Fig. 5.3b, e for Net and Lung dataset, respectively. We can observe that: (1) Both the proposed algorithm and NMS method are insensitive to parameter λ , and (2) the proposed algorithm consistently produces solutions of better qualities in terms of maximizing the object function in Eq. (5.1) and outperforms the NMS method in most cases in terms of detection accuracy, F_1 score. For both methods, the detect candidates with scores below than 0 (1 denotes positive and -1 denotes negative while training) will not be considered.

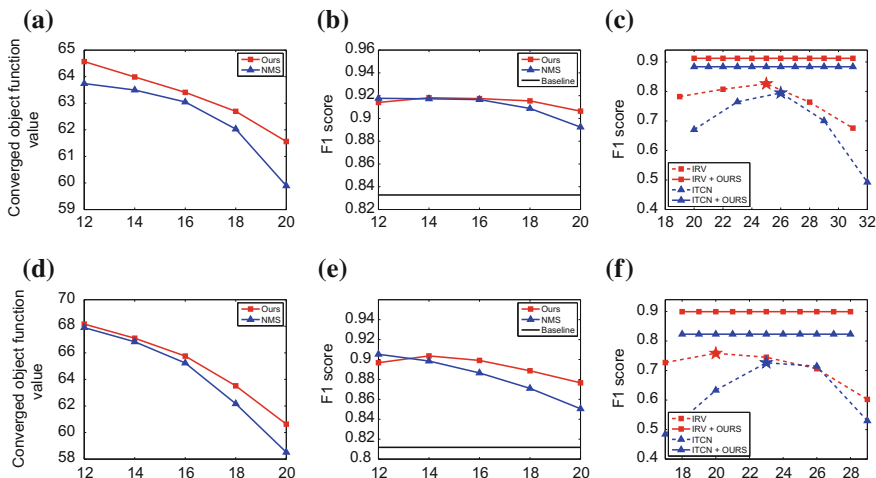


Fig. 5.3 The evaluation of the proposed general cell detection algorithm using two different datasets. The first and second row denotes results of the NET and Lung datasets, respectively. **a** and **d** The comparison of the object values of Eq. (5.1) achieved by the proposed algorithm (Ours) and NMS with different parameter λ . **b** and **e** The comparison of the proposed algorithm (Ours) and NMS by changing the parameter λ . The baseline method is the algorithm presented by Arteta et al. [3]. **c** and **f** The comparison of detection accuracies among IRV, ITCN (different parameters), and our algorithm. The best results of IRV and ITCN are marked with stars

Second, the proposed cell detection algorithm is compared with three detection algorithms: (1) Iterative radial voting (IRV) [1] with different cell diameter parameter $\{19, 22, 25, 28, 31\}$ for NET and $\{17, 20, 23, 26, 29\}$ for Lung dataset; (2) Image-based tool for counting nuclei (ITCN) [2] with diameter parameter set as $\{20, 23, 26, 29, 32\}$ for NET and $\{17, 20, 23, 26, 29\}$ for Lung dataset; (3) A learning-based cell detection algorithm (Arteta et al. [3]) that does not require the parameter selection once a structured supported vector machine is learned on the training images. Both algorithms (1) and (2) will generate a pool of detection candidates, and we will evaluate whether the proposed algorithm is capable of finding the best subset that outperforms each individual algorithm. Please note that we use IRV+OURS and ITCN+OURS to denote the proposed algorithm using the detection results of IRV and ITCN as candidates for best subset selection, respectively. The experimental results are shown in Fig. 5.3. The first row denotes the testing results using the NET dataset, and the second row presents the testing results using the lung cancer dataset. The detailed comparative results are explained below.

The comparative results of IRV, IRV+OURS, ITCN, ITCN+OURS with respect to different parameters are shown in Fig. 5.3c, f. As one can tell, whether or not IRV and ITCN can provide satisfactory results heavily depend on proper parameter selections, which is not always feasible or convenient during runtime. When the parameter is not selected correctly, the performance will deteriorate significantly as illustrated in (c) and (f) (red and blue dotted lines). However, our proposed algorithm

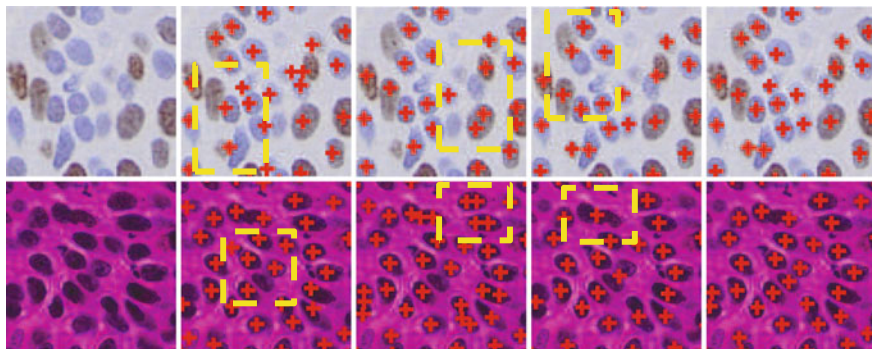


Fig. 5.4 Qualitative cell detection results using different algorithms. The first row denotes the cell detection results using NET and the second row denotes the cell detection results using the Lung cancer dataset. From *left to right*, the columns denote: cropped image patch, cell detection results of [1–3], and the proposed algorithm. The detection errors are labeled with *dotted yellow rectangles*

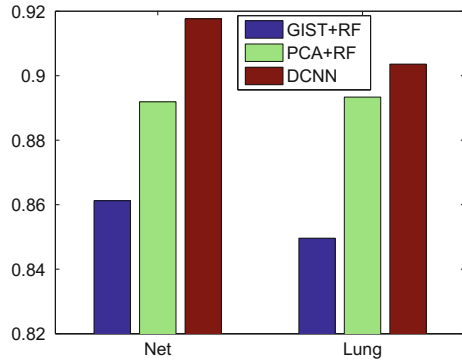
Table 5.2 Comparison of cell detection accuracy

Method	NET			Lung		
	F ₁ -score	Prec.	Rec.	F ₁ -score	Prec.	Rec.
IRV [1]	0.8260	0.7999	0.8539	0.7584	0.6657	0.8812
ITCN [2]	0.7950	0.8277	0.7647	0.7264	0.6183	0.8804
Arteta et al. [3]	0.8328	0.8806	0.7899	0.8118	0.8820	0.7520
[1]+[2]+[3]+OURS	0.9182	0.9003	0.9369	0.9036	0.8843	0.9237

does not require careful selection of parameters as shown in Fig. 5.3b, e. In addition, it consistently outperforms any best individual detection result using IRV and ITCN (red and blue lines) (Fig. 5.4).

In order to justify the accuracy of the assigned weights w using DCNN in Eq. (5.1), we have compared DCNN with a random forest (RF) classifier using different features: (1) Global scene descriptor (GIST) and (2) raw pixel values following by a principle component analysis (PCA) for the dimension reduction. The comparison results can be seen in Fig. 5.5. It is obvious that DCNN consistently provides better results than other methods on both Net and Lung datasets. The quantitative detection results are summarized in Table 5.2. We can see that the proposed algorithm consistently performs better than both the parameter sensitive methods (IRV and ITCN) and the parameter nonsensitive method [3]. Please note that in Table 5.2, we report the best detection results of [1] and [2] using the optimal parameters. Some qualitative automatic cell detection results are shown in Fig. 5.4 using both NET and lung cancer data.

Fig. 5.5 Comparisons of methods to compute w in Eq.(5.1)



5.4 Conclusion

In this chapter, we have proposed a novel cell detection algorithm based on maximum-weight independent set selection that will choose the heaviest subset from a pool of cell detection candidates generated from different algorithms using various parameters. The weights of the graph are computed using a deep convolutional neural network. Our experiments show that this novel algorithm provides ensemble detection results that can boost the accuracy of any individual cell detection algorithm.

References

1. Parvin B, Yang Q, Han J, Chang H, Rydberg B, Barcellos-Hoff MH (2007) Iterative voting for inference of structural saliency and characterization of subcellular events. *TIP* 16(3):615–623
2. Byun J, Verardo MR, Sumengen B, Lewis GP, Manjunath B, Fisher SK (2006) Automated tool for the detection of cell nuclei in digital microscopic images: application to retinal images. *Mol. Vis.* 12:949–960
3. Arteta C, Lempitsky V, Noble JA, Zisserman A (2012) Learning to detect cells using non-overlapping extremal regions. In: *MICCAI*. Springer, Berlin, pp 348–356
4. Matas J, Chum O, Urban M, Pajdla T (2004) Robust wide-baseline stereo from maximally stable extremal regions. *Image Vis Comput* 22(10):761–767
5. LeCun Y, Bottou L, Bengio Y, Haffner P (1998) Gradient-based learning applied to document recognition. *Proc IEEE* 86(11):2278–2324
6. Cireşan DC, Giusti A, Gambardella LM, Schmidhuber J (2013) Mitosis detection in breast cancer histology images with deep neural networks. In: *MICCAI*. Springer, Berlin, pp 411–418
7. Li N, Latecki LJ (2012) Clustering aggregation as maximum-weight independent set. In: *NIPS*, pp 791–799
8. Brendel W, Todorovic S (2010) Segmentation as maximum-weight independent set. In: *NIPS*, pp 307–315

9. Brendel W, Amer M, Todorovic S (2011) Multiobject tracking as maximum weight independent set. In: CVPR. IEEE, New York, pp 1273–1280
10. Gold S, Rangarajan A (1996) A graduated assignment algorithm for graph matching. PAMI 18(4):377–388
11. Ciresan D, Giusti A, Schmidhuber J et al. (2012) Deep neural networks segment neuronal membranes in electron microscopy images. In: NIPS, pp 2852–2860