Medical Image Analysis 63 (2020) 101720

Contents lists available at ScienceDirect

Medical Image Analysis

journal homepage: www.elsevier.com/locate/media

DeepDistance: A multi-task deep regression model for cell detection in inverted microscopy images

Can Fahrettin Koyuncu[a], Gozde Nur Gunesli[a], Rengul Cetin-Atalay[b], Cigdem Gunduz-Demir[a,c,*]

[a] Department of Computer Engineering, Bilkent University, Ankara TR-06800, Turkey
[b] CanSyL,Graduate School of Informatics, Middle East Technical University, Ankara TR-06800, Turkey
[c] Neuroscience Graduate Program, Bilkent University, Ankara TR-06800, Turkey

ARTICLE INFO

Article history:
Received 13 November 2018
Revised 28 February 2020
Accepted 4 May 2020
Available online 11 May 2020

Keywords:
Multi-task learning
Feature learning
Fully convolutional network
Cell detection
Cell segmentation
Inverted microscopy image analysis

ABSTRACT

This paper presents a new deep regression model, which we call *DeepDistance*, for cell detection in images acquired with inverted microscopy. This model considers cell detection as a task of finding most probable locations that suggest cell centers in an image. It represents this main task with a regression task of learning an *inner distance* metric. However, different than the previously reported regression based methods, the *DeepDistance* model proposes to approach its learning as a multi-task regression problem where multiple tasks are learned by using shared feature representations. To this end, it defines a secondary metric, *normalized outer distance*, to represent a different aspect of the problem and proposes to define its learning as complementary to the main cell detection task. In order to learn these two complementary tasks more effectively, the *DeepDistance* model designs a fully convolutional network (FCN) with a shared encoder path and end-to-end trains this FCN to concurrently learn the tasks in parallel. For further performance improvement on the main task, this paper also presents an extended version of the *DeepDistance* model that includes an auxiliary classification task and learns it in parallel to the two regression tasks by also sharing feature representations with them. *DeepDistance* uses the inner distances estimated by these FCNs in a detection algorithm to locate individual cells in a given image. In addition to this detection algorithm, this paper also suggests a cell segmentation algorithm that employs the estimated maps to find cell boundaries. Our experiments on three different human cell lines reveal that the proposed multi-task learning models, the *DeepDistance* model and its extended version, successfully identify the locations of cell as well as delineate their boundaries, even for the cell line that was not used in training, and improve the results of its counterparts.

1. Introduction

Automating the analysis of live cell morphology is critical for high throughput screening as this facilitates fast and reproducible measurements under inverted microscopy. The crucial step of this automation is to correctly identify cell morphology and distribution on culture plates. This requires detecting the cell locations whose difficulty lies along a wide range, from easy to very challenging, depending on visual characteristics of the cells. This step becomes difficult when cells appear in varying colors, brightness, and irregular shapes. The difficulty further increases when they grow in overlayers, and as a result, appear as cell clumps.

Before the advances in deep learning, the traditional approach for cell detection/segmentation is to employ low-level handcrafted features, reflecting color, edge, and shape characteristics of cells. This approach has given promising results when the features are defined properly, as a good representation of the visual cell characteristics. On the other hand, these characteristics may change from one cell type to another (see Fig. 1) and new features need to be defined to meet the cell characteristics of a new type. Additionally, when there exists heterogeneity in the visual characteristics of the same cell type, using a single model may not be sufficient to detect all cells of this type, particularly for cancer cells which are exploited more in high throughput screening.

Methods based on deep learning have responded to these issues by having the ability of learning high-level features from image data automatically and reducing the required effort to obtain a generalizable model as a consequence. The majority of the previously reported methods consider cell detection/segmentation as a

* Corresponding author. Fax: [+90] 312 266 4047.
E-mail addresses: koyuncu@bilkent.edu.tr (C.F. Koyuncu), nur.gunesli@bilkent.edu.tr (G.N. Gunesli), rengul@metu.edu.tr (R. Cetin-Atalay), gunduz@cs.bilkent.edu.tr (C. Gunduz-Demir).

https://doi.org/10.1016/j.media.2020.101720
1361-8415/

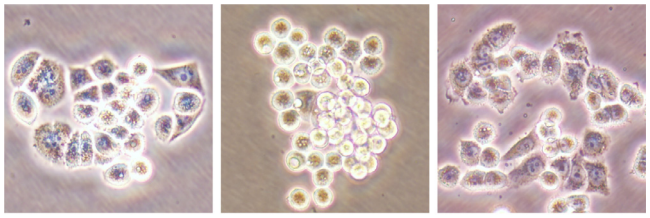

Fig. 1. Example subimages from the cell lines used in our experiments: (a) CAMA-1, (b) MDA-MB-453, and (c) MDA-MB-468 breast cancer cell lines. As seen in these examples, visual characteristics show differences from one cell type to another. Moreover, cells of the same type may appear in different looks. For instance, in (b), there are mostly near-circular cells, which sometimes contain mostly bright pixels but sometimes contain dark pixels inside and bright ones outside. However, in (a) and (c), there are near-circular as well as non-circular cells. For such images, it would not be easy to use a single model to detect cells of all these different looks.

classification problem, in which a deep classifier is trained to differentiate cell pixels from those of the background. Since their focus is the classification of cell pixels, these methods treat the pixels taken from the annotated cells in the same way, regardless of their relative positions within the cell, while training their classifier (Xie et al., 2015a; Song et al., 2017). On the other hand, the position of a pixel relative to a cell center (or to a cell boundary) may bring about additional information. There exist only a few studies that take this information into account by constructing a regression model (Xie et al., 2015b; Sirinukunwattana et al., 2016; Chen et al., 2016; Xie et al., 2018a, 2018b). They approach cell detection/segmentation as a *single-task regression problem* where they learn a single distance output for each pixel. On the other hand, it may be difficult to define a single distance metric that comprehends different aspects of the problem and to learn this single distance by a single model.

In response to these issues, this paper introduces a new multi-task learning framework, which we call *DeepDistance*, for the detection of live cells in inverted microscopy images. This *DeepDistance* framework proposes to concurrently learn two distance metrics for each pixel, where the primary one is learned in regard to the main cell detection task and the secondary distance is defined to stress the variability in morphological cell characteristics and learned for the purpose of increasing the generalization ability of the main task. To this end, this paper constructs a fully convolutional network and end-to-end learns two distance maps at the same time, sharing high-level feature representations at the various layers of this network (layers of its shared encoder path), in the context of multi-task learning. Then, for a given image, it achieves cell detection by generating the primary distance map with the trained network and finding its regional maxima. Furthermore, this paper also suggests a cell segmentation algorithm that makes use of the maps estimated by the trained network. Our experiments on three different cell lines reveal that this proposed multi-task learning framework successfully identifies the locations of cells as well as delineates their boundaries, even for the cell line that was not used in training, and improves the results of its counterparts.

The contributions of this paper are summarized below:

- It takes advantage of the multi-task learning approach, in which shared feature representations are used to learn multiple tasks at the same time. This is different than the previously reported regression-based cell detection studies, which do not use such shared representations for learning a regression task. The multi-task approach used by our study is known to be successful for many domains, leveraging the contribution of different tasks to the feature representation learning process (Caruana, 1997). Concurrent learning of two related tasks with shared representations increases the performance of our model, by better helping it avoid local optimal solutions.
- It defines a distance metric, *normalized outer distance*, that calculates the normalized distance from each cell pixel to the closest cell boundary. As opposed to the *inner distance*, which is calculated with respect to the cell centers and as a result imposes a one-sized circular shape on the cells, this definition does not have shape and size impositions since it uses the boundary annotations. The normalized outer distance better preserves the shape characteristics of the cells whereas the inner distance better suggests the cell centers. Thus, the proposed model defines inner distance estimation as the main task and considers normalized outer distance estimation as complementary to this main task. It learns these two complementary tasks in parallel by forcing them to share feature representations. This improves the performance of each task, and thus, leads to more successful results.
- It shows that one can also include an additional classification task to the proposed multi-task regression network to further increase the performance of the main task. To this end, this study implements another version of the proposed framework where the task of cell pixel classification is added as a parallel task to the regression network. This additional task, which is to be concurrently learned with the two distance maps, aims to construct a classification map from the shared features while learning the regression output maps with a minimum error. This additional task is effective to better learn the regression tasks.

2. Related work

Traditional cell detection/segmentation studies employ low-level handcrafted features, which are extracted either pixel- or subregion-wise. A large group of pixel-wise studies use intensities to obtain a binary mask by thresholding or clustering (Dima et al., 2011). They then use this mask either directly to locate isolated cells or as an input to shape-based methods to split cell clumps. These methods include the use of distance transforms (Jung and Kim, 2010), concavity detection algorithms (Chang et al., 2013), and morphological erosion operators (Yang et al., 2006). Although it is very common to calculate distance transforms on the obtained binary mask, it is also possible to learn them directly from the handcrafted features (Gao et al., 2014). Another group of pixel-wise studies employ pixel gradients to obtain a feature map, on which regional maxima/minima are identified as cell locations. These studies directly use the gradients to define their feature maps (Koyuncu et al., 2016) or alternatively get pixels voted along their gradient directions and use the votes the pixels take to define their maps (Xing et al., 2014). The subregion-wise studies first partition an image into over-segmented subregions (e.g., superpixels), extract handcrafted color, gradient, and shape features from these subregions, and merge them based on their extracted features to obtain cell locations (Genctav et al., 2012; Su et al., 2013; Koyuncu et al., 2018).

To reduce the required efforts for manual feature definition, deep learning based methods learn high-level features from image data. These methods, especially convolutional neural networks, have shown significant success in many tasks related to medical image analysis (Litjens et al., 2017) also including cell detection/segmentation. Earlier studies train their deep models on small patches cropped around individual pixels to generate an output for each pixel separately. More recently, with the implementations of fully convolutional networks (Long et al., 2015) and the U-net model (Ronneberger et al., 2015), studies have started end-to-end training their models to learn the outputs of all pixels at once.

Most of these studies consider cell detection/segmentation as a classification problem and train a classifier to differentiate cell and background pixels. Then, for an image, they may obtain a binary mask by estimating the class labels of its pixels with the trained classifier and use this mask as an input to the shape-based methods (Song et al., 2015; 2017). Alternatively, they may use the class posteriors of the pixels and identify cell locations on this posterior map by either thresholding (Xu et al., 2016) or clustering (Su et al., 2015) but mostly finding regional maxima (Ciresan et al., 2013; Dong et al., 2015; Xie et al., 2015a; Sadanandan et al., 2017).

There exist relatively few studies that consider cell detection/segmentation as a regression problem (Kainz et al., 2015; Xie et al., 2015b; Sirinukunwattana et al., 2016; Chen et al., 2016; Xie et al., 2018a; 2018b). These studies define their outputs with regard to the Euclidean distance between a pixel and its closest annotated cell center. Most of them calculate this inner distance using only the dot annotations on cell centers without using any boundary (segmentation) information. Thus, they use a threshold to decide pixels for which the distance will be zero (i.e., determine pixels belonging to the background). This thresholding together with the inner distance definition itself impose a one-sized circular shape on the cells, which may not be true for all cell types. These studies approach the learning of this inner distance as a *single-task regression problem*. Different than all these studies, our proposed *DeepDistance* model defines a secondary distance metric that better preserves the morphological characteristics of cells and considers its learning as a complementary task to the main task of cell detection. Additionally, it proposes a *multi-task regression framework* that uses shared feature representations to concurrently learn these two tasks.

There are only a few studies that use a cascaded network architecture for cell detection/segmentation. Ram et al. (2018) propose a network that sequentially learns a classification mask on an image and then regresses a density map on this classification mask for cell detection in 3D microscopy images. Kechyn (2018) uses the architecture proposed by Bai and Urtasun (2017) for cell segmentation. This architecture is mainly designed to learn an energy function to be used in a watershed algorithm for the purpose of splitting a map of under-segmented components into their corresponding objects. Thus, it requires obtaining the segmentation map of an image beforehand and takes it as an input together with the image. It first learns a gradient map of a distance transform from these inputs and then learns a map of energy levels from the gradients. As opposed to our proposed multi-task framework, both of these networks cascade their tasks in serial and learns them without sharing any representations. On the other hand, our model proposes to learn two regression maps in parallel, in the context of multi-task learning, which forces these tasks to use shared feature representations. The latter approach is known to be more effective to avoid local optima, and as a result, to obtain a more generalizable model (Caruana, 1997).

There exists another study that also uses a multi-task framework to detect glands and nuclei in histopathological images. This framework concurrently learns two classification maps, where the first one is the map of gland/nucleus pixels and the other is that of their boundaries. It then combines the two classification maps with a simple fusion function (Chen et al., 2017). However, different than our proposed multi-task regression model, this existing study neither considers detection as a regression problem nor learns regression and classification tasks in a single multi-task network. Additionally, its goal is to locate glands/nuclei in fixed and stained histopathological images whereas our aim is to detect cells in inverted microscopy images which is used for high throughput and real-time cell screening.

3. Methodology

The proposed *DeepDistance* model relies on formulating cell detection as a regression problem, in which a metric map is estimated to express the degree of pixels suggesting a cell center, and identifying regional maxima on this map as cell locations. This model uses *inner distance* as the primary metric and estimates it by a fully convolutional network (FCN), considering the learning of this metric as the main task in regard to the cell detection problem. On the other hand, as opposed to the previous studies, the *DeepDistance* model proposes to approach this learning as a multi-task regression problem, in which multiple regression tasks are learned using shared feature representations. To this end, this model defines a secondary metric, *normalized outer distance*, and considers its learning as a complementary task that represents a different aspect of the problem. The proposed *DeepDistance* model learns this new task in parallel to the main task by constructing and end-to-end training an FCN with a shared encoder path, which forces these multiple tasks to learn shared feature representations at various abstraction levels.

The following subsections give the details of the proposed *DeepDistance* model. Section 3.1 mathematically formulates the distance metrics used to define the tasks. It then gives the architecture of the FCN used for learning these tasks and provides the details of its training. Section 3.2 discusses how to extend the proposed multi-task regression network to cover an additional task(s), by giving the details of another version of the proposed model where cell pixel classification is considered as the additional task. Section 3.3 presents the detection algorithm that uses inner distances estimated by the FCN to locate individual cells in a given image. Finally, Section 3.4 suggests a segmentation algorithm that makes use of all of the estimated maps to delineate the cell boundaries.

3.1. Multi-task FCN for distance learning

The proposed *DeepDistance* model uses two distance metrics for each pixel q. The first one is *inner distance* $d_{inner}(q)$ that is calculated similar to the previous studies. Its learning is considered as the main task; cell locations are detected on the estimated inner distance map of the pixels (see Section 3.3). The second metric is *normalized outer distance* $d_{outer}(q)$ that is defined by this current study in order to better quantify morphological cell characteristics. Learning this outer distance is considered as a complementary task, which is used to improve the performance of the main task.

These distances are defined in Eqs. (1) and (2), respectively, when annotations are provided. The annotations are, of course, not available for images whose cells are supposed to be automatically detected. Thus, our *DeepDistance* model proposes to estimate the distances by an FCN that will be trained on the pixels of annotated images.

Let $\mathcal{A} = \{a_i\}$ be the set of annotated cells in an image, $\mathcal{P}(a_i) = \{p_{ik}\}$ be the set of pixels belonging to an annotated cell a_i, $\mathcal{B}(a_i) = \{b_{ik}\}$ be the set of its boundary pixels, and $\mathcal{C}(a_i)$ be its centroid pixel. For pixel q,

$$d_{inner}(q) = \begin{cases} \dfrac{1}{1 + \alpha \min\limits_{a_i \in \mathcal{A}} \|q - \mathcal{C}(a_i)\|^2} & \text{if } q \in \mathcal{P}(a_i) \\ 0 & \text{if } q \in \text{background} \end{cases} \quad (1)$$

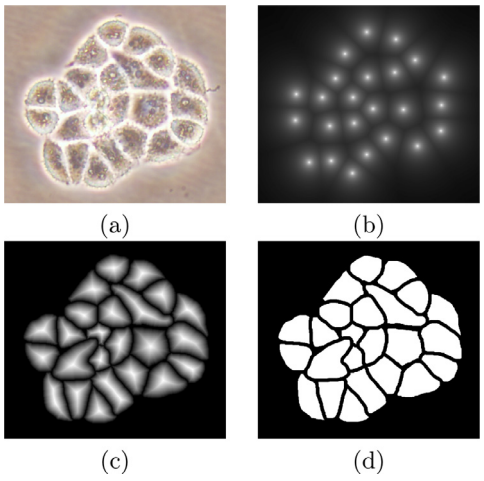

Fig. 2. (a) Original subimage, (b) inner distance map that uses distances from pixels to their closest cell centers, (c) normalized outer distance map that uses distances from cell pixels to their closest boundary annotations, and (d) cell pixel annotations.

$$d_{outer}(q) = \begin{cases} \dfrac{\min\limits_{b_{ik}\in\mathcal{B}(a_i)} \|q - b_{ik}\|^2}{\max\limits_{r\in\mathcal{P}(a_i)} \min\limits_{b_{ik}\in\mathcal{B}(a_i)} \|r - b_{ik}\|^2} & \text{if } q \in \mathcal{P}(a_i) \\ 0 & \text{if } q \in \text{background} \end{cases} \quad (2)$$

where α in Eq. (1) is the decay ratio that is empirically selected as 0.1, similar to the previous studies. The denominator in Eq. (2) corresponds to the maximum distance in annotated cell a_i, which is used as a normalization factor. This normalization is effective to obtain similar distances for cells of different sizes, which will drive the FCN to make better generalizations regardless of the cell size.

For an example subimage given in Fig. 2a, these distance definitions are illustrated in Figs. 2b and 2 c, respectively. The inner distance definition well indicates the cell centers since it uses the Euclidean distances from pixels to their closest cell centers. However, as it uses the centers as the reference point, the distance decrease from a center to its boundaries is the same for all directions and for all cells. Thus, when it is used alone, this definition imposes a circular and one-sized shape on the cells, as also seen in Fig. 2b. On the contrary, since the normalized outer distance is calculated with a reference to a cell boundary, this decrease may differ from one direction to another as well as from one cell to another, depending on the shape and size of the cell. Thus, it better preserves the morphological characteristics of cells, as seen in Fig. 2c.

3.1.1. Network architecture

For multi-task learning of these two distance maps, our *DeepDistance* model constructs an FCN architecture consisting of a shared encoder path and two decoder paths (see Fig. 3). The encoder path is shared by the two tasks to extract shared feature representations from an RGB image, whose pixels are normalized across the image, at various abstraction layers. The two decoder paths, with symmetric connections to the features in the encoder path (shown with concatenation operators in the figure), are used to separately construct the distance maps from these extracted shared features. This architecture has the convolution layers with 3×3 filters and uses the rectified linear unit (ReLU) activation function. Its pooling/upsampling layers use 2×2 filters. The number of the layers and the number of the feature maps used in each convolution layer are depicted in Fig. 3. Note that these numbers are selected by inspiring with the U-net model (Ronneberger et al., 2015). The original U-net model has a single decoder path designed for single-task learning. On the contrary, the *DeepDistance* model has two decoder paths, with symmetric connections to the shared features, for multi-task learning of the two distance maps.

This FCN is end-to-end trained on 512×512 tiles cropped out of the training images. This tile size makes maximum memory use on the GPU that we used (GeForce GTX 1080 Ti) when the batch size is selected as 1. The tiles are cropped by a sliding window approach with an increment of 256 pixels. The selection of this increment size ensures that regions stay on the borders of one tile will be close to the central area of another tile.

3.1.2. Network training

The FCN is implemented in Python using the Keras deep learning library. It is trained from scratch with the backpropagation algorithm that uses the mean squared error as its loss function. It follows an early stopping approach based on the loss calculated for the tiles cropped out of the validation images. The contributions of both tasks to the loss function are the unit weight. The batch size is 1 and the drop-out factor is 0.2. The learning rate and the momentum value are adaptively adjusted using the AdaDelta optimizer (Zeiler, 2012). The source codes of this implementation are available at http://www.cs.bilkent.edu.tr/~gunduz/downloads/DeepDistance.

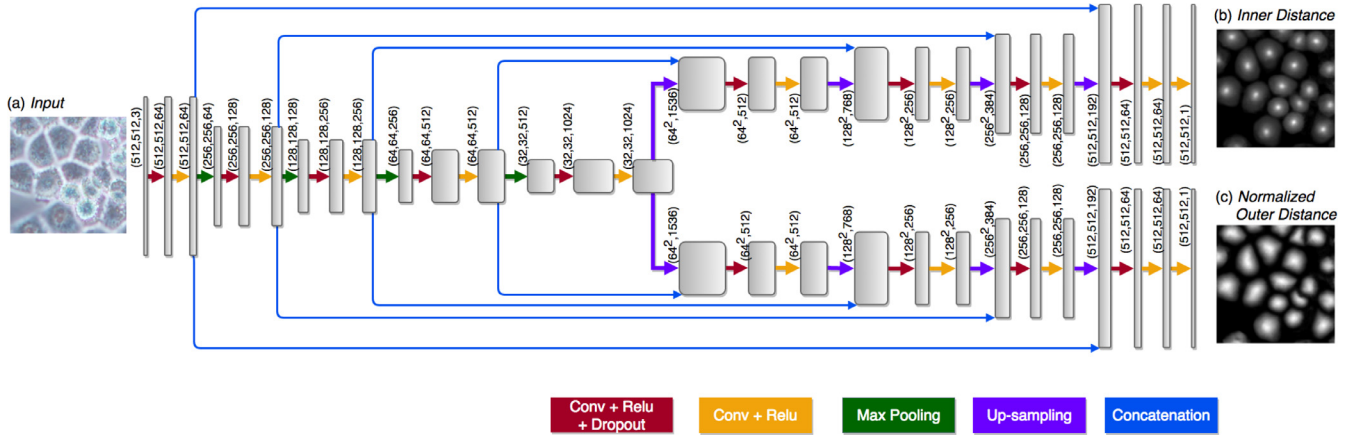

Fig. 3. Architecture of the FCN used for multi-task regression of two distance maps, along with an example tile (a) as the input and the estimated (b) *inner distance* map and (c) *normalized outer distance* map as the outputs. Note that the tile used in this figure is not a part of the training set used in our experiments. Each box represents a multi-channel feature map with its dimensions and number of channels being indicated in order on the left side of the box. Each arrow corresponds to an operation which is distinguishable by its color.

3.2. Extending the FCN for additional tasks

The proposed *DeepDistance* model considers cell detection as a multi-task regression problem that estimates two distance maps from the RGB image, one for formulating the main task of cell detection and the other as an auxiliary task with the motivation of more effectively learning the main task. The FCN architecture given in the previous section is designed to learn these two regression tasks at the same time. This section discusses how this model can be extended to cover more auxiliary tasks, concurrent learning of which may further increase the performance of the main task. For this purpose, this section implements an extended version of the *DeepDistance* model that comprises an additional task of cell pixel classification. This additional task aims to construct a classification map (as shown in Fig. 2d) from the shared features of the encoder path[1] Note that here, instead of defining another regression problem as the additional task, we use a classification problem in order to demonstrate that the model can easily be extended to cover the auxiliary tasks related with regression as well as classification.

The extended version of the *DeepDistance* model uses the FCN that has still one shared encoder path but one extra decoder path, defined for the new classification task. The architecture of this new decoder path is the same with those of the two decoder paths, defined for regressing the distance maps, except that its last convolution layer uses the sigmoid function instead of ReLU. Other than this, it has the same convolution and upsampling layers and uses the same symmetric connections to the features in the encoder path (uses the same concatenation operators). Training of this extended FCN follows the same procedure explained in the previous section, with only a difference of loss calculation. This extended FCN still uses the mean squared error as its loss function and the regression tasks still equally contribute to this loss with a unit weight, but the loss contribution weight of the new classification task is 0.1. The rationality behind using a reduced weight is as follows. Both of the distance outputs, calculated as defined in Eqs. (1) and (2), are in the range between 0 and 1. However, these distances reach the maximum value of 1 for only a few cell pixels whereas they yield much smaller values for the rest of them. On the other hand, the output of the classification task is always 1 for the cell pixels, which results in calculating a larger mean squared error for this task. Since all tasks are learned at the same time by sharing the same features, to avoid creating an unfair bias towards the learning of the classification task, we reduce its loss contribution weight to 0.1.

3.3. Cell detection

This step is to detect cells in an unannotated image. For that, it feeds the tiles cropped out of the image to the trained FCN and identifies cell locations on the inner distance maps estimated by this FCN. Since pixels belonging to a cell center are expected to have higher estimated values, the *DeepDistance* model identifies regional maxima on the inner distance maps as the cell centers. To suppress possible noise in the estimated maps, the model applies the h-maxima transform beforehand and suppresses the maxima whose height is less than the value of h.

This step may result in poor estimations for regions close to tile edges. As a solution to this problem, our model estimates the inner distance maps for overlapping tiles and then averages all distances calculated for the same pixel. The overlapping tiles are obtained by sliding a window over the image with an increment of 64 pixels. Considering the 512 $\times$ 512 tile size used by the FCN, this increment size is small enough to ensure that the regions close to the edges of one tile will be close to the central region of some others. It is also large enough to cause only negligible speed-down in the computational time. Note that the same sliding-window approach is used for all comparison algorithms.

3.4. Cell segmentation

The *DeepDistance* models are primarily designed to identify individual cells by locating their centers without delineating their boundaries. This goal is to address the problem of identifying cells for the purpose of counting (or tracking), which is a very common practice for cell culture research. For instance, cell counting can be used to determine the number of cells in a tissue culture in-real-time within predefined intervals for examining the cell growth under the effects of a cytotoxic treatment.

Nevertheless, it is also possible to design an algorithm that exploits the outputs of *DeepDistance* for cell segmentation. To this end, this section presents a simple marker-controlled region growing algorithm that works on the estimated maps. Let $\mathcal{M}_I$ and $\mathcal{M}_O$ be the estimated maps of the inner and normalized outer distances, respectively, and $\mathcal{M}_C$ be the estimated classification map. The algorithm first considers the cell locations identified on $\mathcal{M}_I$ by the cell detection algorithm (Section 3.3) as markers. It then iteratively grows these markers onto foreground pixels in $\mathcal{M}_C$ with respect to the distances in $\mathcal{M}_O$. That is, it grows the markers pixel by pixel starting from the pixel with the highest $\mathcal{M}_O$ to the one with the smallest $\mathcal{M}_O$.

At the end, it applies three-step postprocessing on the grown markers. First, it eliminates the holes in the grown markers and those smaller than an area threshold a_{thr}. Then, it dilates the remaining grown markers using a structuring element with a size of f_{dilate}. Here it is important to note that, in training, cell boundaries in the annotated maps are widened and subtracted from the classification maps to take overlapping cells apart, and thus, to train the networks to better learn the boundaries between overlapping cells. Thus, the grown markers are dilated to add these boundary pixels back to the segmented cells since the growing process grows the markers on the estimated classification maps. Finally, it smoothes the boundaries of the dilated markers by applying a majority filter with a size of $f_{majority}$. Note that this postprocessing is also used for the comparison algorithms. Its parameters (namely, a_{thr}, f_{dilate}, and $f_{majority}$) are selected for each algorithm separately using the grid search on the training and validation images (see Section 4.3).

4. Experiments

4.1. Datasets

We test our *DeepDistance* model on three datasets, each of which consists of live cell images of a different cell line. They are the CAMA-1, MDA-MB-453, and MDA-MB-468 human breast cancer cell lines. The images in all datasets were acquired at 20 $\times$ magnification and 3096 $\times$ 4140 pixel resolution. An example image from each dataset is shown in Fig. 1. As seen in this figure, cells might be visually different within and across different cell lines.

Three images are randomly selected from each of the CAMA-1 and MDA-MB-453 cell lines and are used for training the FCN as well as for selecting the parameters of the cell detection and segmentation steps. While training the FCN, the tiles cropped out of four of these six images are used to learn the weights of the FCN and those of the remaining two are used as validation tiles for early stopping. The cells in the rest of the images in these two cell lines are used for testing. In our experiments, these test cells, which belong to CAMA-1 and MDA-MB-453, are considered

[1] To take overlapping cells apart, and hence to obtain an improved map, cell boundaries are widened and subtracted from the classification map. This improved map is also used in the comparison methods to make fair comparisons.

Table 1
For each cell line, the number of images and the number of cells in its training, validation, and test sets.

	Training		Validation		Test	
	Image	Cell	Image	Cell	Image	Cell
CAMA-1	2	752	1	84	6	1254
MDA-MB-453	2	522	1	137	4	765
MDA-MB-468	-	-	-	-	8	1679
Total	4	1274	2	221	18	3698

as dependent test samples since other images/cells of the same cell lines are used for training. To assess the success of our model on an unseen cell line, none of the images of MDA-MB-468 are used for training the FCN or for parameter selection. Thus, the cells of this MDA-MB-468 cell line are considered as independent test samples. For each cell line, the number of images and the number of cells in the training, validation, and test sets are presented in Table 1. The cells in these sets are annotated by manually drawing their boundaries.

4.2. Evaluation

The results are quantitatively evaluated on the test cells. For cell detection, the cell-level F-score metric is calculated on the detected cells (markers). In this calculation, a detected cell D is considered as true positive if the following three conditions are satisfied: (i) at least 50 percent of the regional maxima $\texttt{rmax}_D$ corresponding to D should be inside an annotated cell A, (ii) at most 10 percent of $\texttt{rmax}_D$ could be inside another annotated cell, and (iii) the annotated cell A should not contain more than 50 percent of the regional maxima corresponding to another detected cell. Note that the second condition is used especially to exclude the under-segmented cells from the true positives. Then, the precision and recall metrics are obtained on the true positives, and the F-score is calculated as their harmonic mean.

For cell segmentation, the F-score metric is calculated based on the intersection-over-union (IoU) scores. For that, for each pair of overlapping segmented cell S and annotated cell A, the IoU score is calculated as $\text{IoU}(S, A) = |S \cap A|/|S \cup A|$. Then, for a given IoU threshold τ, a segmented cell S is considered as true positive if $\text{IoU}(S, A) > \tau$ for an annotated cell A. Likewise, the precision, recall, and F-score metrics are calculated on these true positives. In our experiments, the F-scores are calculated for different IoU thresholds $\tau = \{0.50, 0.55, \ldots, 0.90\}$ and their average is used for quantitative evaluation.

4.3. Parameter selection

The *DeepDistance* model uses one external parameter for cell detection. This parameter is the h value used by the h-maxima transform to suppress possible noise in the estimated inner distance map. The value of this parameter is selected on the six images belonging to the training and validation sets of the CAMA-1 and MDA-MB-453 cell lines. For that, the following values of $h = \{0.1, 0.2, 0.3, 0.4, 0.5\}$ are considered and the one that yields the highest cell-level F-score metric for the cells in these six images is selected. The selected value is $h = 0.2$.

The model uses four external parameters for cell segmentation. They are used by the marker-controlled region growing algorithm. The first one is the h value of the h-maxima transform to detect the markers. The other three are those of the postprocessing step: the area threshold a_{thr} to eliminate small grown markers, the structuring element size f_{dilate} to dilate the remaining grown markers, and the majority filter size $f_{majority}$ to smooth their boundaries. In the experiments, the grid search is used to select the values of these parameters. For that, all combinations of the following sets $h = \{0.1, 0.2, 0.3, 0.4, 0.5\}$, $a_{thr} = \{500, 1000, 1500, 2000\}$, $f_{dilate} = \{5, 7, 9, 11\}$, and $f_{majority} = \{11, 15, 21, 25\}$ are considered. Then, the one that gives the highest average IoU-based F-score on the training and validation images is selected. Note that for each combination, this average is taken on the F-scores calculated using different IoU thresholds. The selected values are $h = 0.2$, $a_{thr} = 2000$, $f_{dilate} = 11$, and $f_{majority} = 15$. Also note that the parameter values are selected similarly for the comparison methods.

5. Cell detection results

The quantitative results obtained by the proposed multi-task *DeepDistance* models are given in Table 2. This table reveals that our models lead to accurate cell detection results on both dependent and independent test samples. It also shows that extending the model by including the additional task of cell pixel classification further improves the results. Additionally, the visual results obtained on exemplary subimages are presented in Fig. 4. Note that this figure shows only the cells correctly identified by the models; it does not show any incorrectly located cell, which does not match with any annotated cell satisfying the aforementioned three conditions.

In order to understand the effectiveness of the multi-task regression framework used by the *DeepDistance* model, we compare it with three deep learning based methods that use a single-task framework. These methods are designed to separately learn the tasks used by the proposed *DeepDistance* models. In particular, they learn only an inner distance map (*SingleInner*), only a normalized outer distance map (*SingleOuter*), and only a classification map of cell pixels (*SingleClassification*) from the RGB image, respectively. For learning their single-tasks, all these methods use an FCN with a single encoder path, similar to our models, but also only a single decoder path, as opposed to ours. The convolution and pooling/upsampling layers of this single encoder and single decoder path are the same with those specified in Fig. 3. They also end-to-end train their FCNs and the training setups are the same with ours. Obviously, since they have only one task to be learned, none of these methods take advantage of learning the shared feature representations. After learning their FCNs, these methods take the same cell detection steps of our model. These steps include estimating a map by the FCN, suppressing its noise by the h-maxima transform, and finding regional maxima on the resulting map. Here the *SingleInner* and *SingleOuter* methods use their estimated dis-

Table 2
For cell detection, the cell-level F-score metrics obtained on the test sets.

	Dependent test samples		Independent test samples
	CAMA-1	MDA-MB-453	MDA-MB-468
DeepDistance	91.80	90.87	84.48
DeepDistance (extended)	92.07	91.40	85.45
SingleInner	87.67	89.17	82.02
SingleOuter	90.39	88.03	79.67
SingleClassification	81.73	80.41	57.71

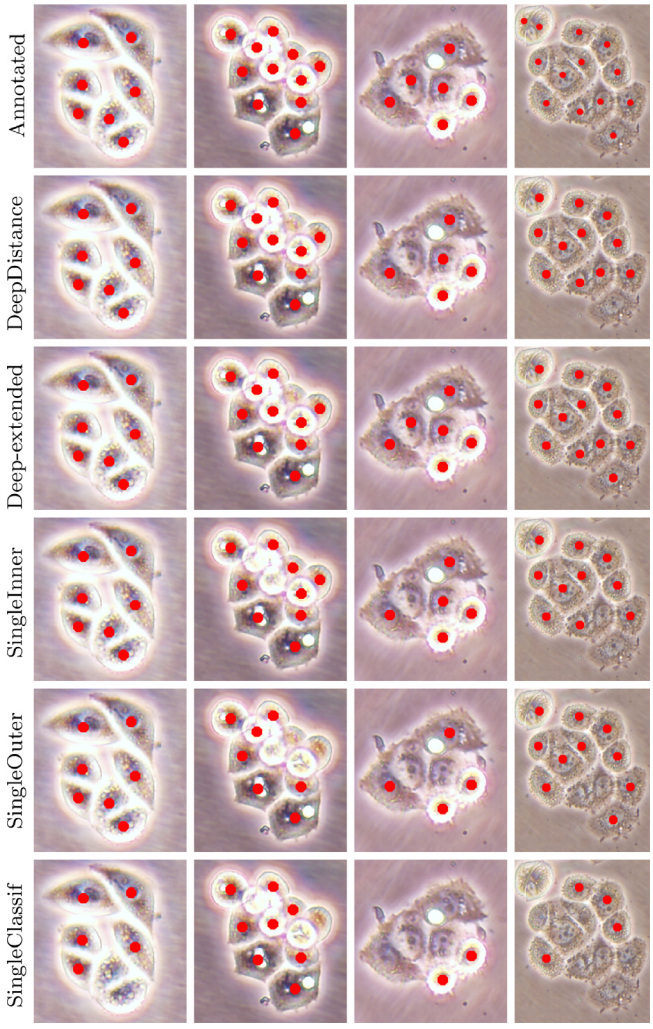

Fig. 4. For cell detection, visual results obtained on illustrative subimages. The first two subimages are taken from the dependent test samples and the last two from the independent test samples, which belong to the cell line that was not used in any part of the training. First row: annotated cells. Second and third rows: cells correctly identified by the proposed *DeepDistance* model and its extended version. Fourth to sixth rows: cells correctly identified by the single-task networks; namely, the *SingleInner, SingleOuter*, and *SingleClassification* methods. Note that this figure does not show any cell incorrectly identified by the algorithms.

tance maps and the *SingleClassification* method uses the estimated posterior map of the cell pixel class. The quantitative results of these methods are given in Table 2 and their visual results on exemplary subimages are presented in Fig. 4. These results reveal that concurrent learning of multiple tasks improves the results of single-task learning. It is worth to noting that this improvement is more evident for the independent test samples.

5.1. Parameter analysis

When it is used for cell detection, *DeepDistance* has one external parameter: the h value used by the h-maxima transform to suppress noise in the estimated inner distance map. Small h values do not sufficiently suppress noise, resulting in false cell detections and oversegmentations. On the other hand, unnecessarily large values suppress too many pixels as noise, causing not to identify many actual cells and leading to undersegmentations. Both of these cases decrease the performance. This is consistent with our experimental results shown in Figs. 5a and 5b, which depict the

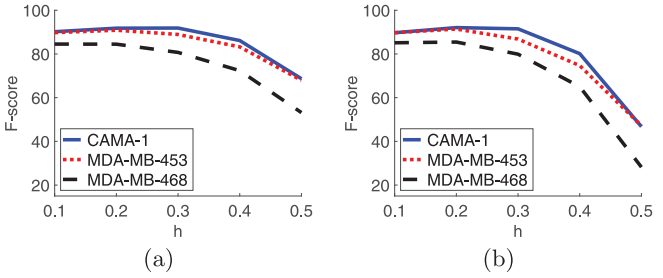

Fig. 5. For cell detection, test set cell-level F-scores as a function of the h parameter for (a) the *DeepDistance* model and (b) its extended version.

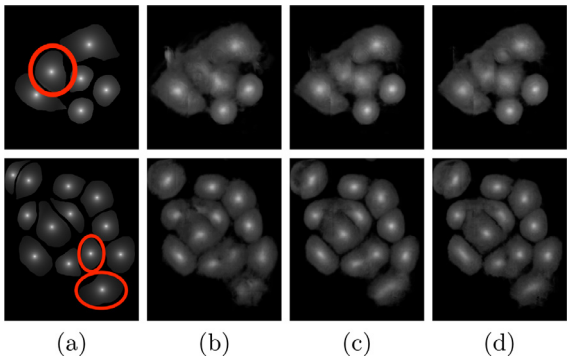

Fig. 6. (a) Maps of calculated inner distances when the ground truths are provided. Inner distance maps estimated by (b) *SingleInner*, (c) *DeepDistance*, and (d) its extended version.

test set cell-level F-scores as a function of the h value for the *DeepDistance* model and its extended version, respectively.

5.2. Multi-task vs. single-task learning

Since the main goal of this work is cell detection, our *DeepDistance* models define the estimation of an inner distance map as the main task and find regional maxima on this estimated map to detect cells. The motivation behind these choices is the fact that the inner distance definition gives sharp increases at cell centers and the locations with these sharp increases can be detected by finding regional maxima. Hence, to obtain accurate detections, one should estimate an inner distance map with distinct differences between the cell centers and their surrounding pixels such that these centers can be identified as regional maxima. That is, one should estimate a map consisting of sharp enough bright regions close to the cell centers. To improve the performance of the task of this inner distance estimation, our models take advantage of multi-task learning approach. This approach helps the models become more robust to avoid overfitting a task, compared to the approach of learning the same task alone with a single-task model (Caruana, 1997). To get more insight in this multi-task learning approach, this section visually analyzes the estimated maps of single-task and multi-task models.

For the independent test samples given in Fig. 4, Fig. 6a shows the maps of calculated inner distances when the ground truths are given. Figs. 6b, 6c, and 6d illustrate the inner distance maps estimated by the *SingleInner* method, the proposed *DeepDistance* model, and its extended version, respectively. *SingleInner* learns its map as a single-task whereas our models define auxiliary tasks and learn the inner distance map in parallel to these auxiliary tasks, forcing them to learn shared representations with a shared encoder path. The latter type of learning, which is an example of multi-task learning, is known to be effective for increasing the performance of individual tasks for many domains. We also observe

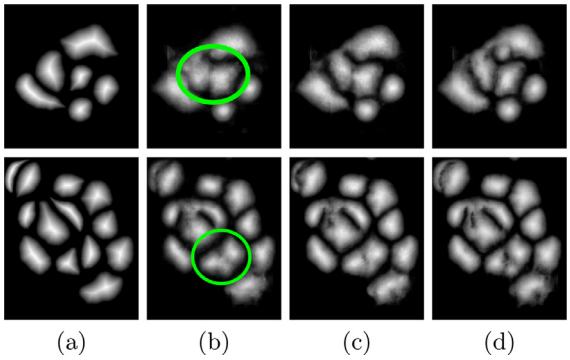

Fig. 7. (a) Maps of calculated outer distances when the ground truths are provided. Outer distance maps estimated by (b) *SingleOuter*, (c) *DeepDistance*, and (d) its extended version.

this performance increase in the estimated maps given in Fig. 6. *SingleInner* cannot successfully detect the three cells shown inside red ellipses since it cannot produce sharp enough bright regions (with distinct enough estimated distances) for these cells. Although *DeepDistance*, which uses one auxiliary task, leads to brighter regions for these cells, they are still not sharp enough for two of them to be identified as regional maxima. The extended version of *DeepDistance*, which uses one more auxiliary task, does better job in inner distance estimations such that they have sharp enough bright regions for all of these three cells. Here it is worth to noting that all methods apply the h-maxima transform to suppress noise, and hence, to prevent over-segmentations and false positives. If it was not applied, *SingleInner* might give regional maxima for some of the three cells even though the distances estimated for their centers were not that distinct (bright). However, that case would also give many over-segmented cells and false positives.

Likewise, Fig. 7a shows the maps of calculated outer distances when the ground truths are given. Figs. 7b, 7c, and 7 d show the outer distance maps estimated by *SingleOuter, DeepDistance*, and its extended version, respectively. It is observed that a single-task *SingleOuter* method is less accurate in estimating outer distances especially for pixels close to cell boundaries. Due to this incorrect estimation, it locates only a single cell for each of the cell pairs shown inside green ellipses, resulting in under-segmentations for these cell pairs. Our multi-task *DeepDistance* models yield better estimations for these boundary pixels. However, it is important to note that our models do not use the estimated outer distances in the detection algorithm but define this estimation as an auxiliary task to represent a different aspect of the problem. Concurrent learning of two related tasks with a multi-task model, which uses shared feature representations, better helps avoid local optima. In other words, when two related tasks share the same representations, it is more difficult to finetune these representations for only one of these tasks. This is effective to obtain better learning performances for individual tasks, as also shown in Figs. 6 and 7.

6. Cell segmentation results

The quantitative results obtained by the *DeepDistance* model for cell segmentation are given in Table 3. Its visual results on exemplary subimages are presented in Fig. 8. These results show that the proposed multi-task framework used by the *DeepDistance* model is effective to obtain better results also for cell segmentation. Note that this section includes the results of the extended version of the *DeepDistance* model since the segmentation algorithm described in Section 3.4 employs an estimated classification map in its region growing process.

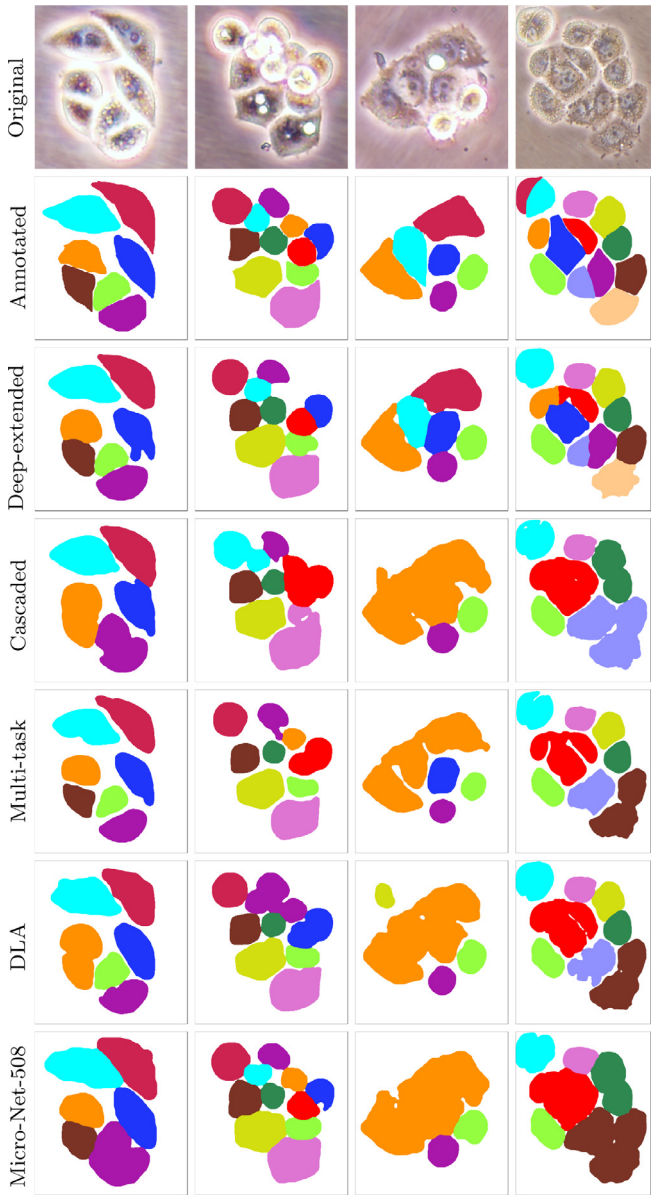

Fig. 8. For cell segmentation, visual results obtained on illustrative subimages. The first two subimages are taken from the dependent test samples and the last two from the independent test samples. First and second rows: original subimages and their annotations. Third row: cells segmented by the proposed *DeepDistance* model. Fourth to seventh rows: cells segmented by the comparison methods; namely, the *CascadedClassification, MultiTaskClassification, DLA*, and *Micro-Net-508* methods.

Next, we compare *DeepDistance* with four cell segmentation methods. The first two of them use more than one task in designing their models. The first one is *CascadedClassification* that uses a cascaded architecture similar to the one proposed by Ram et al. (2018). This cascaded architecture is designed to sequentially learn a classification map of cell pixels from the RGB image and then to regress an inner distance map from the classification map. This method learns these maps in serial by using two serially cascaded FCNs that do not share any feature representation. Each of these serial FCNs has a single encoder and a single decoder path that contains the same convolution and pooling/upsampling layers specified in Fig. 3. Although these two FCNs are learned at the same time by backpropagating the error through the entire network in an end-to-end training fashion, they do not learn any shared feature representation. This method uses the same training

Table 3
For cell segmentation, the average IoU-based F-score metrics obtained on the test sets. Note that this average is taken over the F-scores calculated using different IoU thresholds $\tau = \{0.50, 0.55, \ldots, 0.90\}$.

	Dependent test samples		Independent test samples
	CAMA-1	MDA-MB-453	MDA-MB-468
DeepDistance (extended)	75.35	71.42	63.71
CascadedClassification	61.53	59.87	33.67
MultiTaskClassification	72.62	62.04	55.54
DLA (Kang et al., 2019)	68.73	67.59	49.55
Micro-Net-508 (Raza et al., 2019)	65.22	67.60	43.84

setup with our model and applies the same postprocessing steps on its estimated classification maps to segment the cells. The results given in Table 3 demonstrate that *DeepDistance*, which learns multiple tasks in parallel with shared feature representations, gives more accurate results than *CascadedClassification*, which learns two tasks in serial without sharing any representation. This indicates the effectiveness of learning shared feature representations from multiple tasks, which is indeed known to be effective for many domains (Caruana, 1997). As also seen in the visual results given in Fig. 8, *CascadedClassification* yields many undersegmentations, which is the reason of obtaining low IoU-based F-scores especially for the independent test samples.

The second comparison method is *MultiTaskClassification* that approaches cell segmentation as a multi-task classification problem. Similar to the model proposed by Chen et al. (2017), this method defines two classification tasks, where one is the task of cell pixel classification and the other is the task of cell boundary classification, and learns them in parallel by also using shared feature representations. For learning these tasks, this method uses an FCN whose architecture is the same with the one given in Fig. 3. This FCN is end-to-end trained also using the same training setup. To segment the cells in a given image, *MultiTaskClassification* combines the two classification maps estimated by the FCN with a simple fusion technique that is also used by Chen et al. (2017). For that, the pixels estimated as boundary are subtracted from the estimated cell pixel classification map and the connected components on the resulting map are identified as cells. These identified cells are also postprocessed using the same steps described in Section 3.4. The results of this method are also given in Table 3 and Fig. 8. They show that this method, which approaches cell segmentation as a multi-task classification problem, yields lower accuracies than the proposed *DeepDistance* model. This might be attributed to the regression tasks learned by our model conveying more complementary information, concurrent learning of which may better help the model less overfit on the training samples. As a result, this may give a more generalizable model.

The other two comparison algorithms, namely the *DLA* and *Micro-Net-508* methods, are from the recent studies of cell segmentation. The *DLA* method (Kang et al., 2019) proposes to design a two-stage network by stacking two U-nets. The first stage is to learn a three-class segmentation map, where classes are defined as nucleus, boundary, and background. The second one is to refine the output of the first stage and to obtain the final fine-grained segmentation map. In our experiments, we observe that directly using this final map leads to many undersegmented cells, and thus, greatly lowers the IoU scores. Thus, we subtract the boundary map estimated by the first stage from the final map and postprocess the result applying the same steps used by our model. The results of this method are given in Table 3 and Fig. 8. They reveal that even though the estimated boundaries are subtracted from the final map, it still yields undersegmentations. Here it is worth to noting that the *DLA* method gives relatively better results for the dependent test samples but it yields lower performance for the independent ones. This indicates the better generalizability of our proposed model, which poses cell segmentation as a multi-task regression problem.

The *Micro-Net-508* method (Raza et al., 2019) designs an architecture that trains its network at multiple resolutions of the input image. It also defines connections between the intermediate layers of the networks of different resolutions. Then, it proposes to generate the output using multi-resolution deconvolutional filters. Likewise, in our experiments, we observe that this method generates outputs containing many undersegmentations. Therefore, we add one more step to the postprocessing algorithm. This additional step erodes the generated segmentation map with a structuring element, finds the connected components on this eroded map, and then separately dilates each component using the same structuring element. Note that we use such kind of additional step since this method does not generate an additional boundary map. The quantitative and visual results obtained by the *Micro-Net-508* method are given in Table 3 and Fig. 8. Similar to the *DLA* method, *Micro-Net-508* also leads to many undersegmented cells, especially for the independent test samples, and as a result, gives lower IoU scores.

6.1. Parameter analysis

For segmentation, this study suggests a marker-controlled region growing algorithm that uses four external parameters. We also analyze the effects of these parameters on the segmentation performance. To this end, for each parameter, the selected values of the other three parameters are fixed and the test set performance

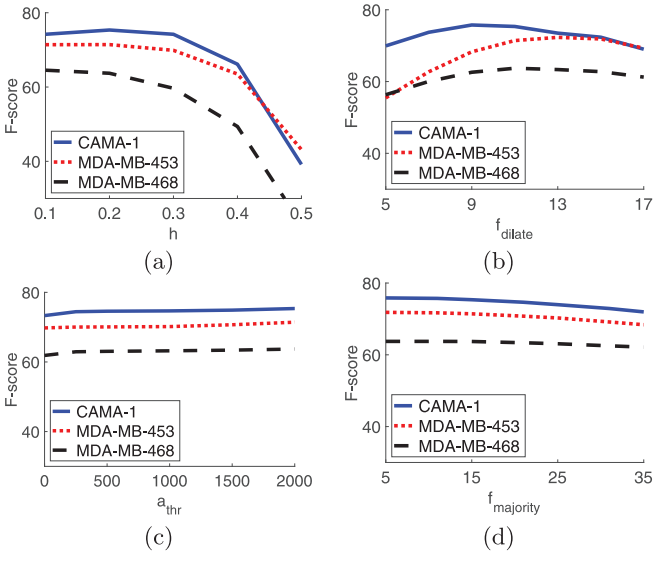

Fig. 9. For cell segmentation, test set average IoU-based F-scores as a function of (a) the parameter h, (b) the dilation structuring element size f_{dilate}, (c) the area threshold a_{thr}, and (d) the majority filter size $f_{majority}$.

is measured as a function of the parameter of interest. These analyses are depicted in Fig. 9.

The first parameter is the h value of the h-maxima transform, which is used to determine the markers. As mentioned in Section 5.1, this parameter should be selected large enough to suppress noise but also small enough not to suppress pixels belonging to cells. Unnecessarily suppressing these pixels results in many false negatives and undersegmentations, leading to huge decreases in the average IoU-based F-scores (Fig. 9a). The other three parameters are used to postprocess the grown markers. Among these three, the size f_{dilate} of the structuring element used to dilate the grown markers affects the performance the most. Remember that, in training, cell boundaries in the annotated maps are widened and the corresponding pixels are subtracted from the classification maps to take overlapping cells apart. Since the network is trained to estimate these boundary pixels as background, the f_{dilate} parameter should be selected large enough to add these pixels back to the segmented cells (Fig. 9b). The other two parameters are the area threshold a_{thr} to eliminate small grown markers and the size $f_{majority}$ of the majority filter to smooth their boundaries. As seen in Figs. 9c and 9d, these parameters slightly affect the average IoU-based F-scores.

7. Conclusion

This paper presents the *DeepDistance* model, which designs a multi-task regression framework for detecting individual cells in microscopy images, and experimentally demonstrates the successful use of this model on cells of three different cell lines. For the cell detection problem, this is the first proposal of a multi-task regression model that learns multiple regression tasks in parallel by using shared feature representations.

The *DeepDistance* model designs this regression framework to concurrently learn two distance metrics for image pixels in the context of multi-task learning. To this end, it defines the *normalized outer distance* metric to represent a different aspect of the problem and proposes to learn it in parallel to the primary inner distance metric, which is defined in regard to the main cell detection task, for the purpose of increasing its generalization ability. For this concurrent learning, the *DeepDistance* model constructs an FCN with a shared encoder path, which forces the two tasks to learn shared feature representations. Such shared representation learning on multiple tasks is indeed known to be more effective to avoid each task to overfit, and as a result, to obtain more generalized models. Our experiments on three different cell lines also reveal that this multi-task learning together with formulating cell detection as a regression problem lead to accurate results, improving the results of the single-task frameworks.

This work mainly focuses on the cell detection problem. Nevertheless, it also suggests a simple region growing algorithm that employs the estimated maps to delineate cell boundaries. The experiments show that this cell segmentation algorithm yields more accurate results than the previous approaches. However, one may design a network that includes additional tasks specifically focused on learning the cell boundaries (e.g., defining cell boundary estimation as the main task) and use the estimated boundaries in the region growing process too. Designing such multi-task networks and developing more sophisticated cell segmentation algorithms are considered as one future research direction. We believe that the cell detection method proposed by this study is not limited to static cell images, but it can easily be adapted and used for real-time live images throughout a time dependent experiment under inverted microscopy. The investigation of the latter use is considered as another future research direction of this study.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

CRediT authorship contribution statement

Can Fahrettin Koyuncu: Conceptualization, Methodology, Software, Writing - original draft. **Gozde Nur Gunesli:** Conceptualization. **Rengul Cetin-Atalay:** Data curation, Formal analysis. **Cigdem Gunduz-Demir:** Conceptualization, Methodology, Investigation, Formal analysis, Writing - original draft, Funding acquisition.

Acknowledgments

This work was supported by the Turkish Academy of Sciences under the Distinguished Young Scientist Award Program (TÜBA GEBİP).

References

Bai, M., Urtasun, R., 2017. Deep Watershed Transform for Instance Segmentation. In: Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition, pp. 5221–5229.
Caruana, R., 1997. Multitask learning. Mach. Learn. 28, 41–75.
Chang, H., Han, J., Borowsky, A., Loss, L., Gray, J.W., Spellman, P.T., Parvin, B., 2013. Invariant delineation of nuclear architecture in glioblastoma multiforme for clinical and molecular association. IEEE Trans. Med. Imag. 32, 670–682.
Chen, H., Qi, X., Yu, L., Dou, Q., Qin, J., Heng, P.A., 2017. DCAN: Deep contour-aware networks for object instance segmentation from histology images. Med. Image. Anal. 36, 135–146.
Chen, H., Wang, X., Heng, P.A., 2016. Automated mitosis detection with deep regression networks. In: IEEE 13th Int. Symp. on Biomedical Imaging, pp. 1204–1207.
Ciresan, D.C., Giusti, A., Gambardella, L.M., Schmidhuber, J., 2013. Mitosis detection in breast cancer histology images with deep neural networks. In: Int. Conf. on Medical Image Computing and Computer-Assisted Intervention. Springer, pp. 411–418.
Dima, A.A., Elliott, J.T., Filliben, J.J., Halter, M., Peskin, A., Bernal, J., Kociolek, M., Brady, M.C., Tang, H.C., Plant, A.L., 2011. Comparison of segmentation algorithms for fluorescence microscopy images of cells. Cytometry Part A 79, 545–559.
Dong, B., Shao, L., Da Costa, M., Bandmann, O., Frangi, A.F., 2015. Deep learning for automatic cell detection in wide-field microscopy zebrafish images. In: IEEE 12th Int. Symp. on Biomedical Imaging, pp. 772–776.
Gao, Y., Wang, L., Shao, Y., Shen, D., 2014. Learning distance transform for boundary detection and deformable segmentation in CT prostate images. Int. Workshop Mach. Learn. Med. Imag. 8679, 93–100.
Genctav, A., Aksoy, S., Onder, S., 2012. Unsupervised segmentation and classification of cervical cell images. Pattern Recognit. 45, 4151–4168.
Jung, C., Kim, C., 2010. Segmenting clustered nuclei using h-minima transform-based marker extraction and contour parameterization. IEEE Trans. Biomed. Eng. 57, 2600–2604.
Kainz, P., Urschler, M., Schulter, S., Wohlhart, P., Lepetit, V., 2015. You should use regression to detect cells. In: Int. Conf. on Medical Image Computing and Computer-Assisted Intervention. Springer, pp. 276–283.
Kang, Q., Lao, Q., Fevens, T., 2019. Nuclei segmentation in histopathological images using two-stage learning. In: Int. Conf. on Medical Image Computing and Computer-Assisted Intervention. Springer, pp. 703–711.
Kechyn, G., 2018. Instance segmentation: automatic nucleus detection. Towards Data Science, https://towardsdatascience.com/instance-segmentation-automatic-nucleus-detection-a169b3a99477.
Koyuncu, C.F., Akhan, E., Ersahin, T., Cetin-Atalay, R., Gunduz-Demir, C., 2016. Iterative h-minima-based marker-controlled watershed for cell nucleus segmentation. Cytometry Part A 89, 338–349.
Koyuncu, C.F., Cetin-Atalay, R., Gunduz-Demir, C., 2018. Object-oriented segmentation of cell nuclei in fluorescence microscopy images. Cytometry Part A.
Litjens, G., Kooi, T., Bejnordi, B.E., Setio, A.A.A., Ciompi, F., Ghafoorian, M., van der Laak, J.A., Van Ginneken, B., Sanchez, C.I., 2017. A survey on deep learning in medical image analysis. Med. Image Anal. 42, 60–88.
Long, J., Shelhamer, E., Darrell, T., 2015. Fully convolutional networks for semantic segmentation. In: Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition, pp. 3431–3440.
Ram, S., Nguyen, V.T., Limesand, K.H., Sabuncu, M.R., 2018. Joint cell nuclei detection and segmentation in microscopy images using 3d convolutional networks. arXiv preprint arXiv:1805.02850.
Raza, S.E.A., Cheung, L., Shaban, M., Graham, S., Epstein, D., Pelengaris, S., Khan, M., Rajpoot, N.M., 2019. Micro-net: a unified model for segmentation of various objects in microscopy images. Med. Image Anal. 52, 160–173.

Ronneberger, O., Fischer, P., Brox, T., 2015. U-net: Convolutional networks for biomedical image segmentation. In: Int. Conf. on Medical Image Computing and Computer-Assisted Intervention. Springer, pp. 234–241.
Sadanandan, S.K., Ranefall, P., Le Guyader, S., Wahlby, C., 2017. Automated training of deep convolutional neural networks for cell segmentation. Sci. Rep. 7, 7860.
Sirinukunwattana, K., Raza, S.E.A., Tsang, Y.W., Snead, D.R., Cree, I.A., Rajpoot, N.M., 2016. Locality sensitive deep learning for detection and classification of nuclei in routine colon cancer histology images. IEEE Trans. Med. Imag. 35, 1196–1206.
Song, Y., Tan, E.L., Jiang, X., Cheng, J.Z., Ni, D., Chen, S., Lei, B., Wang, T., 2017. Accurate cervical cell segmentation from overlapping clumps in pap smear images. IEEE Trans. Med. Imag. 36, 288–300.
Song, Y., Zhang, L., Chen, S., Ni, D., Lei, B., Wang, T., 2015. Accurate segmentation of cervical cytoplasm and nuclei based on multiscale convolutional network and graph partitioning. IEEE Trans. Biomed. Eng. 62, 2421–2433.
Su, H., Xing, F., Kong, X., Xie, Y., Zhang, S., Yang, L., 2015. Robust cell detection and segmentation in histopathological images using sparse reconstruction and stacked denoising autoencoders. In: Int. Conf. on Medical Image Computing and Computer-Assisted Intervention. Springer, pp. 383–390.
Su, H., Yin, Z., Huh, S., Kanade, T., 2013. Cell segmentation in phase contrast microscopy images via semi-supervised classification over optics-related features. Med. Image Anal. 17, 746–765.
Xie, W., Noble, J.A., Zisserman, A., 2018a. Microscopy cell counting and detection with fully convolutional regression networks. Comput. Method. Biomech. Biomed. Eng. 6, 283–292.
Xie, Y., Kong, X., Xing, F., Liu, F., Su, H., Yang, L., 2015a. Deep voting: A robust approach toward nucleus localization in microscopy images. In: Int. Conf. on Medical Image Computing and Computer-Assisted Intervention. Springer, pp. 374–382.
Xie, Y., Xing, F., Kong, X., Su, H., Yang, L., 2015b. Beyond classification: Structured regression for robust cell detection using convolutional neural network. In: Int. Conf. on Medical Image Computing and Computer-Assisted Intervention. Springer, pp. 358–365.
Xie, Y., Xing, F., Shi, X., Kong, X., Su, H., Yang, L., 2018b. Efficient and robust cell detection: a structured regression approach. Med. Image Anal. 44, 245–254.
Xing, F., Su, H., Neltner, J., Yang, L., 2014. Automatic ki-67 counting using robust cell detection and online dictionary learning. IEEE Trans. Biomed. Eng. 61, 859–870.
Xu, J., Xiang, L., Liu, Q., Gilmore, H., Wu, J., Tang, J., Madabhushi, A., 2016. Stacked sparse autoencoder (SSAE) for nuclei detection on breast cancer histopathology images. IEEE Trans. Med. Imag. 35, 119–130.
Yang, X., Li, H., Zhou, X., 2006. Nuclei segmentation using marker-controlled watershed, tracking using mean-shift, and kalman filter in time-lapse microscopy. IEEE Trans. Circuits Syst. I Regul. Pap. 53, 2405–2414.
Zeiler, M.D., 2012. Adadelta: an adaptive learning rate method. arXiv preprint arXiv:1212.5701.