OSA CONTINUUM

Multi-column network for cell counting

NI JIANG AND FEIHONG YU[*]

College of Optical Science and Engineering, Zhejiang University, Hangzhou, 310027, China
[*]*feihong@zju.edu.cn*

Abstract: Cell counting is a fundamental but crucial task for microscopic analysis. In this paper, we present a method that can count cells automatically and achieves good accuracy. The algorithm extends the U-net from the single-column to the multi-column to capture the features of cells with various sizes. The general convolutional layers in the U-net body are replaced by residual blocks to help the network converge better. Furthermore, a region-based loss function is designed to guide the model to slide into the proper local minima and avoid overfitting. Experimental results on three public datasets show that the proposed method can handle different kinds of images with promising accuracy. Compared with other state-of-the-art approaches, the proposed approach performs superiorly.

1. Introduction

Counting objects is a basic workflow for various vital applications of interdisciplinary research, for video surveillance, pathological analysis, and diagnosis, etc. In recent years, object counting has drawn high attention from researches. Benefit from the development of hardware, many methods based on the convolutional neural networks (CNNs) have come forward. In general, training CNNs requires lots of images and annotations. But in microscopy, the cell image datasets with annotations are scarce, and it's hard for the non-professionals to construct a new dataset. In this paper, we propose a method that can be trained with a small dataset and achieves good performance on testing sets.

For cell detection, the concavity-based algorithm [1] or the distance transform-based algorithm [2] were the most common ways several years ago, but they can not handle irregular cells. Then, the supervised learning model is introduced to count cells without explicit image processing. It learns the image features to predict a density map [3–9]. The density-based algorithm [3] casts the counting problem as a regression problem. The predicted pixel-level density indicates the count of cells, which means the integral over a region is the corresponding number of cells. It only requires a dot to label a cell and applies a density function to transfer the discrete counting to regression summation. The mapping from the image features to the density map is learned by a model automatically. For machine learning models, the handcrafted features are closely related to the cell type and the model performance depends on the features heavily. Hence, the specificity of features limits the models' application. Recently, the CNN-based model [4,7] draws high attention from researchers. With the convolutional layers, many implicit or explicit semantic features are extracted automatically and adaptively during the forward propagation of input data, and all the parameters of the network will be updated by minimizing the loss function. The CNN-based model makes the counting task entirely automatic and is more robust than the machine learning models, like support vector machine (SVM), random forest (RF), etc.

Base on the notion of the density function [3], we propose a multi-column and multi-resolution network (MM-Net), which adopts U-net [10] as the backbone and places residual blocks in the encoder path and the decoder path. To capture cells or cell clusters with different scales, we use different convolutional kernels to capture receptive fields with different sizes in three column networks. The labeled dots are changed to continuous densities by the convolution with the

#396603 https://doi.org/10.1364/OSAC.396603
Journal © 2020 Received 5 May 2020; revised 19 Jun 2020; accepted 19 Jun 2020; published 29 Jun 2020

density function, so we design a loss function based on the mask of the density map to make cell images with heavily overlapping cells more important, which helps optimize the network.

The rest of the paper is organized as follows. Section 2 reviews the objects counting methods for several decades, especially cell counting. Section 3 illustrates the network architecture and implementation in detail. The experiments on different datasets will be discussed in Section 4. Section 5 draws a conclusion of this paper.

2. Related works

The traditional counting methods segment the objects from the background and then count the objects, e.g., level set [11–13]. Though the level set methods can find the contours effectively, they can't separate overlapping cells that are low-contrast or edge-blurring. To detect individual objects, researchers try to find the concave points on the edge where objects overlap [1]. Based on the shape information, the distance transform is also used to count cells [2,14].

Following the density-based model [3], Fiaschi et al. proposed to learn the mapping with an RF and structured labels, and estimated the average counting by patch-wise prediction [5]. Besides that, Arteta et al. built a tree structure of a set of nested candidate cell-like regions and learned the formulation of the relation between leaf nodes and the assigned labels by structured SVM [15–17]. To count interactively, Arteta et al. developed an interactive counting system [18] based on Lempitsky's previous work [3] and changed the loss function from Maximum Excess over SubArrays distance to the ridge regression, which is the smooth version of the Euclidean distance between predictions and ground truth.

Recently, Xie et al. used CNN to learn the mapping between images and density maps [7] and obtained a good result on the real image. Rather than integrating on the density map, Zhu et al. used the Non-Maximum Suppression (NMS) algorithm to detect cells on the density map [19]. Considering cells with various sizes, Pan et al. took two receptive fields with different sizes at the first convolution layer to obtain more local information [20]. Also, to enlarge the receptive field, Rad et al. stacked a set of dilated convolution layers [21] at the bridge part of networks without parameters increasing significantly [22]. Walach et al. boosted several CNNs layers, the next CNN learns the bias between the ground truth and prediction of the previous CNN [23]. Cohen et al. predicted count by redundant counting based on patches [4], but the predicted density maps aren't intuitive. Rad et al. proposed a novel pyramid network to extract more global information [6], which performs well on embryonic cells. The density-based model is good at counting the crowded cells and the segmentation-based model performs better on the cells that can be distinguished by clear edges. Akram et al. combined two CNNs to detect and segment cells [24]. The first CNN proposes cell bounding boxes and the second CNN predicts the individual cell mask. Kong et al. designed a parallel U-Net to segment the cell regions and the cell centers, and the obtained cell centers will be the seeds of the watershed algorithm to segment the whole cells [25].

Apart from cell counting, there are also some excellent works for pedestrian counting. Zhang et al. introduced a deep CNNs with two loss functions switchable to learn, the one focuses on the loss about density map and the other one focuses on the loss about count number [26]. Zhang et al. proposed a Multi-Column Convolution Neural Network (MCNN) to count the crowd, the receptive field size of each column is different and multi-scale features can be learned [27]. Di Kang et al. used a classic CNN to predict the density map pixel by pixel and designed a fully convolutional neural network with skip connections [28]. Yang et al. trained the proposed multi-column CNNs with multi-tasks, these tasks estimating crowd density level, background/foreground mask, and the density map are parallel [29].

3. Method

To better demonstrate the proposed method, how to generate the density map is described first. Given a cell image, each cell is labeled by a dot, as shown in Fig. 1(b). However, it's difficult to count cells by estimating the sparse dots. Following the previous works [3,5], we take a normalized Gaussian function G to smooth the dots and the cell labels become continuous which is the density map in Fig. 1(c). Therefore, the density map $F(x)$ is defined as:

$$F(x) = \sum_{i=1}^{N} G(x_i; \sigma), \tag{1}$$

where x_i is the i^{th} cell dot label, σ controls how to smooth the dots, $G(x_i, \sigma)$ represents the normalized Gaussian function at x_i with σ.

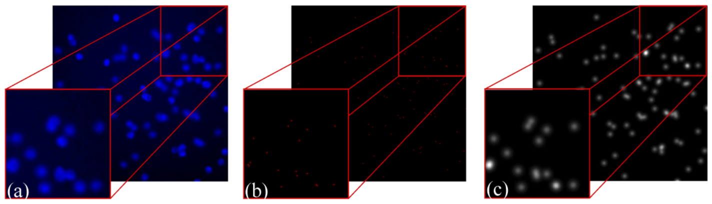

Fig. 1. (a) The cell image. (b) The cell labels. (c) The density map.

The density map indicates how many cells per pixel contains, which means the integral over the whole image corresponds to the numbers of cells.

3.1. Network architecture

The proposed MM-Net is based on the U-Net. The symmetry of the network architecture makes the model pixel-to-pixel, and the regressed density map can not only provide the number of cells but also the spatial density. In order to be adaptive to cells or cell clusters with various sizes, convolutional kernels with different receptive fields are applied at the multi-column. Different from the U-Net, the residual connections are used to optimize the proposed network. Additionally, we leverage the characteristic of the density map to design a region-based loss function, which can highlight the images with heavily overlapped cells.

The overview of MM-Net is shown in Fig. 2 where the number in each block represents the feature channel. Firstly, the input image is resized twice to different resolutions by down-sampling. The resized images (including the original image) are respectively encoded by 5×5, 3×3 and 1×1 kernels on three parallel branches. Then feed these feature maps to different stages of the contracting path according to their spatial dimension. The general convolution blocks of U-Net are replaced by the residual blocks [30,31] whose superiority has been experimentally verified. In Fig. 2, we define different levels according to the resolution sizes and the feature maps with the same resolution size will be at the same level. The input image is encoded with pooling from level 1 to level 4 and then is decoded with up-sampling from level 4 to level 1, so the spatial dimension of feature maps is recovered and the network achieves a pixel-to-pixel prediction. As is known to all, the feature maps from shallow layers preserve spatial information but extract low-level features, and the feature maps from deep layers contain high semantic information but lack spatial information. The skip connections between the encoder and the decoder can merge the two information complementarily to enhance the locations of cells, which is important in density maps. The feature maps from the contracting path are concatenated with the feature maps in the expansive path at the same resolution and then convolve with a 3×3 kernel to reduce the channels to 64 and learn the local context. Both in the contracting path and the expansive path, all the kernel sizes are $3 \times 3 \times 64$, except the kernels of the first residual block are $5 \times 5 \times 64$. At the last layer of MM-Net, a $1 \times 1 \times 1$ kernel is applied to transform the feature maps to the predicted density maps.

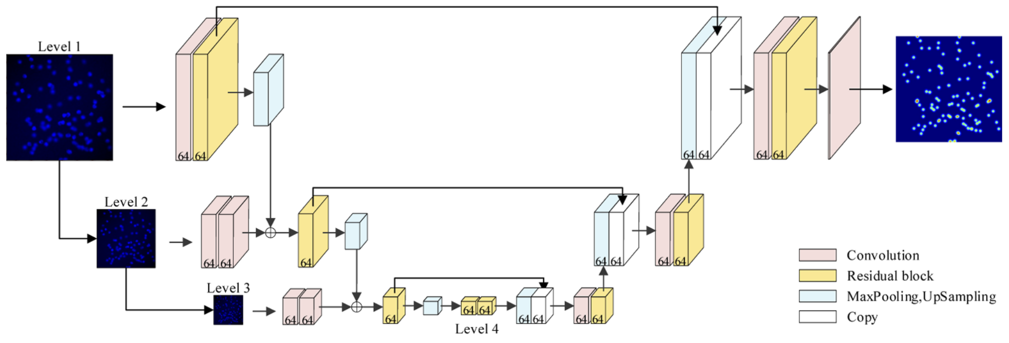

Fig. 2. The overview of the MM-Net

3.2. Residual block

To improve the accuracy of counting, we build the MM-net with the residual block, which has been experimentally observed that it can ease the optimization of network and improve regularization. The residual block is shown in Fig. 3. The shortcut between input and output makes sure that the information from the current unit can be forward propagated to the next unit, which is similar to the skip connection between the contracting path and the expansive path. The output of the residual block is calculated as

$$x_{l+1} = x_l + F(x_l, W_l), \tag{2}$$

where x_l is the input and x_{l+1} is the output, $F(\cdot)$ is the function and W is the parameter set of the residual block.

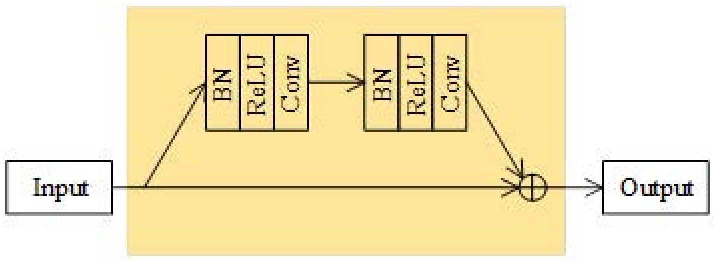

Fig. 3. The residual block

Because the branch contains batch normalization-RELU-Convolution (i.e. BN-RELU-Conv), the BN-RELU is placed before the last convolution layer of MM-Net.

3.3. Multi-column branches

We use a multi-column and multi-resolution architecture to capture the various size of cells or cell clusters, which is shown in Fig. 4. There are three networks for the input image, which are defined as branches A, B, and C respectively. In Fig. 4, the box of CB only means the convolution operations and it has nothing to do with the parameters, the details about these CBs can be found in Table 1. The green boxes 1 and 2 are the layers where the feature maps from different branches merged and different receptive fields can be obtained. The CBs consist of Conv-BN-RELU and the resolution block. Because there has been a BN layer and RELU in the residual block, a single convolution layer is set before the resolution block.

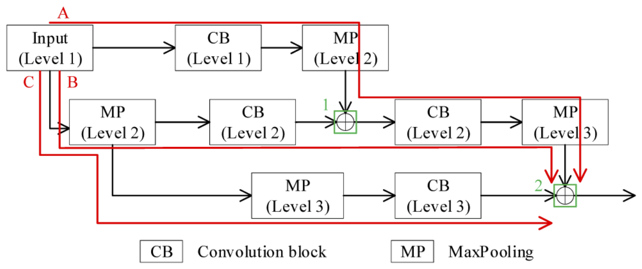

Fig. 4. The multi-column and multi-resolution architecture

Table 1. The sizes of receptive fields on the three branches

Branch A					Branch B					Branch C				
	Layer	K	S	RF		Layer	K	S	RF		Layer	K	S	RF
0	Input			1	0	Input			1	0	Input			1
1	Conv	5×5	1	5	1	MP	2×2	2	2	1	MP	2×2	2	2
2	Conv	5×5	1	9	2	Conv	3×3	1	6	2	MP	2×2	2	4
3	Conv	5×5	1	13	3	Conv	3×3	1	10	3	Conv	1×1	1	4
4	MP	2×2	2	14	4	Conv	3×3	1	14	4	Conv	1×1	1	4
5	Conv	3×3	1	18	5	Conv	3×3	1	18					
6	Conv	3×3	1	22	6	MP	2×2	2	20					
7	MP	2×2	2	24										

Table 1 displays the layers of three branches (ignoring the BN layers and the activation layers), which are related to the receptive fields. K is the kernel size of the convolution layer or the pool size of the MaxPooling layer, S is the stride, RF is the size of receptive field. The size of receptive field is calculated by

$$RF_i = RF_{i-1} + (K-1) \times \prod_{m=1}^{i-1} S_m, \tag{3}$$

where RF_i means the size of receptive field on the i^{th} layer and $RF_0 = 1$.

At green box 1, the layer 4 of branch A and the layer 3 of branch B are merged, the receptive fields $[A_1|B_1]$ are [14|10] which has a slight difference. The layer 7 of branch A and the layer 6 of branch B and the layer 4 of branch C meet at the green box 2, where the receptive fields $[A_2|B_2|C_2]$ are [24|20|4]. The difference of receptive fields is further expanded, which is helpful to learn the small cells and large cells. Meanwhile, the depths of the three branches are different. Each time feature maps merged, the depths of different branches are not consistent. Therefore, it also promotes to fuse the feature maps from different levels.

3.4. Loss function

The mean square error (MSE) is the regular loss function for the regression task and it measures the Euclidean distance between the estimated density map and the ground truth density map. It is defined as

$$L = \frac{1}{N}\sum_{i=1}^{N} ||F(X_i) - F_i||_2^2, \tag{4}$$

where $F(X_i)$ and F_i respectively are the estimated density map and ground truth density map of the input image X_i, N is the number of training images.

However, when training with the regular MSE loss function we found that the image with denser cells has a higher loss, which is prone to induce the network to fall into the local minima. In order to decrease the losses of the images with dense cells, we define a loss function based on the cell regions and the background of the density map.

How can we obtain the mask of cell regions in a cell image? In the counting task, the goal is to predict the density map which is defined by a Gaussian kernel. It is natural to binarize the density map by a threshold to obtain the mask. Figure 5 shows the mask of a cell image. The white regions represent the cell region and the black regions represent the background.

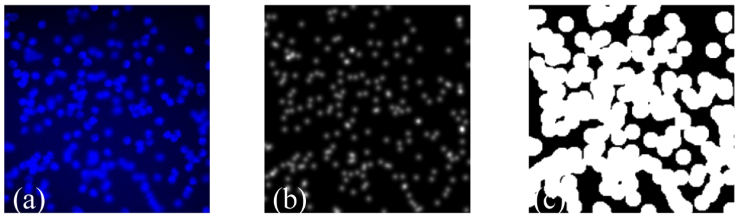

Fig. 5. (a) The cell image. (b) The density of ground truth (c) The mask based on (b)

To avoid the loss of background dominates the optimization, the whole loss L consists of the cell regions' loss L_{fg} and the background's loss L_{bg}. They are formulated as

$$L_{fg} = \frac{\sum_{j\in R_{fg}} (\hat{y}_j^i - y_j^i)^2}{N_{fg}^i}, \tag{5}$$

$$L_{bg} = \frac{\sum_{j\in R_{bg}} (\hat{y}_j^i - y_j^i)^2}{N_{bg}^i}, \tag{6}$$

$$L = \frac{1}{N}\sum_{i=1}^{N} \frac{\lambda}{\omega_i}(L_{fg} + L_{bg}), \tag{7}$$

$$\omega = \frac{N_{fg}}{N_{cell}}, \tag{8}$$

where R_{fg} represents the cell regions and R_{bg} represents the background. $\hat{y}_j^i$ and are respectively the predicted density and the ground truth of j^{th} pixel belonging to the i^{th} image. N_{fg}^i is the pixel number of cell regions and N_{bg}^i is the pixel number of background. ω measures the average area per cell occupies, and it makes the image with high density learned better. N_{cell} is the number of cells in an image. λ is a constant factor whose impact on the accuracy can be neglected, and it is set to 80 in our experiments.

3.5. Implementation Details

For training, a dot is placed near the cell center to annotate. To predict cell count by regression, we apply the normalized Gaussian kernel to transfer the dot label to continuous density value.

The densities near the center are high and decrease gradually along the radial. The sum of a density patch is the corresponding number of cells. Limited by the few images with annotations, data augmentation is necessary.

3.5.1. Data augmentation

Without plenty of annotation datasets, it's necessary to apply data augmentation to generate more training images, such as rotation, shift, rescale, and flip. To overcome the different distribution of data, data standardization is applied.

3.5.2. Optimization

To search the minimum of the loss function, many optimizers are proposed. Compared to other optimizers, Adam [32] can converge faster. However, it may sink in the local minima which might impede the convergence of the network. Cyclical learning rate [33] (CLR) can jump out of the local minima easily to alleviate this problem. Therefore, we combine Adam with CLR to optimize. In our experiment, the base learning rate is set to $5e^{-06}$ and the max learning rate is set to $5e^{-04}$, the step size is set to 80 or 40 depending on the batch size, the learning rate policy chooses the triangular cycle. During training, we first train for 300 epochs to save the well pre-trained parameters. Next, the network is initialized by the pre-trained parameters and then is trained for 300 epochs again.

4. Experiments

To validate the proposed model, three cell datasets are introduced in this section. Then, the evaluation metric about the model performance will be described. Next, the comparisons between the proposed method and the state-of-the-art methods will be discussed. The ablation study will also be conducted.

4.1. Datasets

The VGG Cells dataset is one of the most common datasets which is introduced by Lempitsky et al. [3]. This dataset contains 200 images with an average of 171 ± 64 cells. The synthetic dataset is similar to the real cell images in the overlapping, defocus, and intensity variability, which remains challenges for counting. The MBM dataset derives from the original bone marrow (BM) dataset that is introduced by Kainz et al [34]. It contains 11 images with a 1200×1200 resolution. The background is full of staining impurities, which makes the counting task tougher. To correct several wrong annotations of foreground and ambiguous objects, the dataset is modified by Cohen et al. and is named MBM (Modified BM) [4]. By cropping, the MBM dataset contains 44 images with 126 ± 33 cells. The PhC-HeLa dataset contains 22 phase-contrast images of cervical cancer cell colonies of the HeLa cells, with 2228 dot annotations [15]. Cells in this dataset are very closed to each other and have various shapes and sizes.

4.2. Evaluation metric

Follow the previous works [3–7,23], the mean absolute error (MAE) is used to evaluate models, which is defined as

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|P^i - T^i|, \tag{9}$$

where N is the number of the test images, P^i is the predicted number of cells in the i^{th} image, and T^i is the true number of cells in the i^{th} image. MAE measures the average bias between predictions and ground truth.

For each dataset, we repeat the experiment 10 times with random training images. The final result is the average and standard deviation of the MAEs. As an additional benefit, cells can be located roughly by taking the local maxima on the predicted maps.

4.3. Comparison with the state-of-the-art

4.3.1. VGG Cells dataset

For VGG Cells dataset, we split the dataset into the training set, validation set, and testing set. VGG Cells dataset consists of 200 images, and we take the first 100 images for training and the rest for testing. Same as the state-of-the-art methods, N training images and N valid images are randomly sampled from the training set. Table 2 presents the comparison results on VGG Cells dataset. The Count-ception model counts cells redundantly and has a low MAE, but it can't locate cells. The proposed method does not only have higher accuracy but also more specific locations. With the multi-column architecture, the proposed model performs slightly better than the Cell-Net model when $N = 50$.

Table 2. The comparison results on VGG Cells dataset with state-of-the-art works.

Method	$N = 16$	$N = 32$	$N = 50$
Lempitsky et al. [3]	3.8 ± 0.2	3.5 ± 0.2	N/A
Fiaschi et al. [5]	N/A	3.2 ± 0.1	N/A
Arteta Carlos et al. [18]	3.8 ± 0.3	3.5 ± 0.1	N/A
FCRN-A, Xie [7]	3.4 ± 0.2	2.9 ± 0.2	2.9 ± 0.2
Count-ception, Cohen [4]	2.9 ± 0.5	2.4 ± 0.4	2.3 ± 0.4
Cell-Net, Rad [6]	2.7 ± 0.6	N/A	2.2 ± 0.5
The proposed method	2.9 ± 0.2	2.3 ± 0.1	2.1 ± 0.1

Figure 6 shows the result that the proposed model tests on a testing image. The overlap, defocus blur and the non-uniform illumination in the test image are the main challenge to count. Figure 6(b) is the estimated density map that has been normalized to be shown. Figures 6(c) and (d) are the annotated cells and the detected cells respectively. The blue dots represent the ground truth and the yellow dots represent the locations. The ground truth is 194 cells and the proposed method predicts 196 cells. Except for the extremely close cells, most cells are detected.

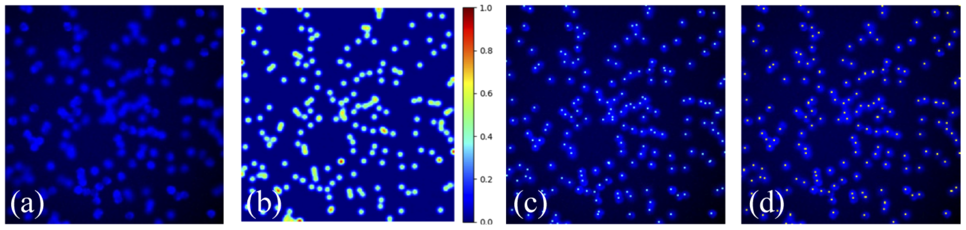

Fig. 6. The proposed model's performance on VGG Cells dataset. (a)The input image. (b) The predicted density map. (c) The annotation. (d)The detection.

4.3.2. MBM Cells dataset

As for MBM Cells, we follow Cohen's strategy [4] and take 30 images as the training set, and the rest as the testing set. The comparison results in Table 3 reveal that the proposed method surpasses the other compared methods. Figure 7 shows the regression counting and detection. The stained image has a complex background which is disturbing to count. Not only that but also the various size and the non-uniform dyeing are the obstacles in counting. The ground truth is 109 cells and the estimated counting is 114 cells. As indicated in Fig. 7(d), the cells of various shapes and sizes also are detected successfully.

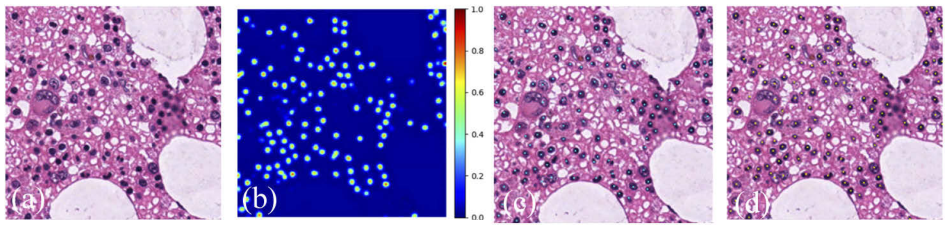

Fig. 7. The proposed model's performance on MBM dataset. (a)The input image. (b)The predicted density map. (c) The annotation. (d)The detection.

Table 3. The comparison results on MBM dataset with state-of-the-art works

Method	$N = 5$	$N = 10$	$N = 15$
FCRN-A, Xie [7]	28.9 ± 22.6	22.2 ± 11.6	21.3 ± 9.4
Count-ception, Cohen [4]	12.6 ± 3.0	10.7 ± 2.5	8.8 ± 2.3
Cell-Net, Rad [6]	11.3 ± 4.8	9.8 ± 3.2	N/A
The proposed method	9.9 ± 0.9	9.2 ± 0.8	7.5 ± 0.7

4.3.3. PhC-HeLa dataset

PhC-HeLa dataset contains 11 images for training and 11 images for testing. Since the size of the dataset is limited, we take 5 images for the train set and 5 images for the validation set. Table 4 displays the comparison results. The proposed model performs better than GMN [35] which is also a neural network, worse than the singletons [17]. The singletons [17] counts cells by detection and the proposed method count cells by regression. Actually, the cells are adjacent closely and the various shape and illumination of cells make the regression task more difficult. Nevertheless, the background is pure as shown in Figs. 8(a) and (b). Counting by detection may perform better than by regression, so we also detect cells by finding the local maxima on the predicted density maps. To suppress the noise, the predicted density maps are smoothed by a Gaussian function and segmented by a threshold. In our experiments, the sigma and threshold are experimentally set to 1.0 and 70. Table 4 indicates that the proposed method becomes the state-of-the art on the PhC_HeLa dataset. As shown in Fig. 8(a), the ground truth is 85 cells, and the prediction by regression in Fig. 8(b) is 80 cells. However, Fig. 8(d) indicates that all the cells are detected by taking the local maxima, and the detecting result is 85 cells, the same as ground truth.

Table 4. The comparison results on PhC-Hela dataset

Method	MAE
Full system w/o surface [17]	3.84 ± 1.44
GMN, Lu et al. [35]	3.53 ± 0.18
Singletons [17]	2.36 ± 0.67
The proposed method	3.47 ± 0.88
The proposed method (detection)	1.61 ± 0.48

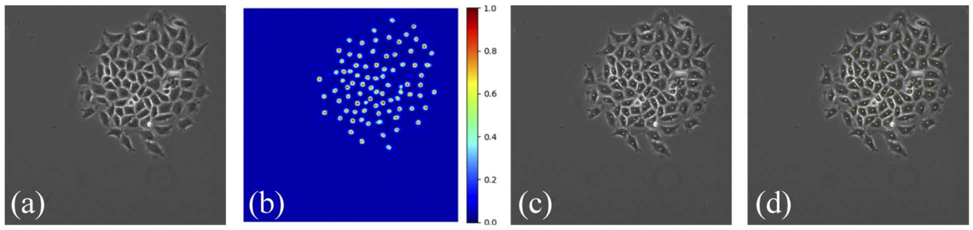

Fig. 8. The proposed model's performance on PhC-HeLa dataset. (a)The input image. (b)The predicted density map. (c) The annotation. (d)The detection.

4.4. Ablation study

To prove the effectiveness of the multi-column branches and residual blocks and the proposed loss function, we conducted the ablation study on the VGG Cells. Table 5 displays the results on five different models:

- U-Net baseline: It consists of the contracting path and the expansive path, as the original U-Net. The Conv-BN-RELU is the basic block and the baseline contains the same convolution kernels as the residual blocks of MM-Net;
- U-Net + MM: The combination of U-Net baseline and the multi-column branches;
- U-Net + RB: The combination of U-Net baseline and the residual blocks (RB);
- MM-Net + MSE Loss: The proposed network trained with the regular MSE loss function;
- MM-Net: the proposed model.

Table 5. The results of ablation study on VGG Cells

Method	MAE	MSE
U-Net baseline	2.4 ± 0.2	9.9 ± 1.9
U-Net + MM	2.4 ± 0.2	10.0 ± 1.5
U-Net + RB	2.2 ± 0.2	8.6 ± 1.6
MM-Net + MSE Loss	2.3 ± 0.1	9.1 ± 1.1
MM-Net	2.1 ± 0.1	7.4 ± 0.8

Table 5 shows that the multi-column branch seems to be invalid for the U-Net baseline to improve the performance. By contrast, the residual blocks do more help to achieve higher accuracy. The MM-Net, with both multi-column branches and residual blocks, can improve performance further. It can be explained that the residual blocks ease the optimization and that is the reason why MM-Net outperforms the U-Net + MM. To validate the proposed region-based loss function, we also trained the MM-Net with regular MSE loss function. It can be inferred from Table 5 that the proposed region-based loss function is beneficial to perform with lower MAE and MSE.

Even though the performances on the whole test images have been demonstrated, we want to explore the performances of the MM-Net and the U-Net + RB in some test images. We compare the performance on the images with many cells that are hard to count. The images are arranged by the number of cells from high to low, and the top 20 images are tested. Figure 9 shows the prediction of the top 20 images. The orange dots are predictions of the MM-Net and the green dots are predictions of the U-Net + RB, and the blue line indicates where the ground truth is supposed to be. It can be observed the orange dots are closer to the blue line in all, which means the MM-Net surpasses the U-Net + RB.

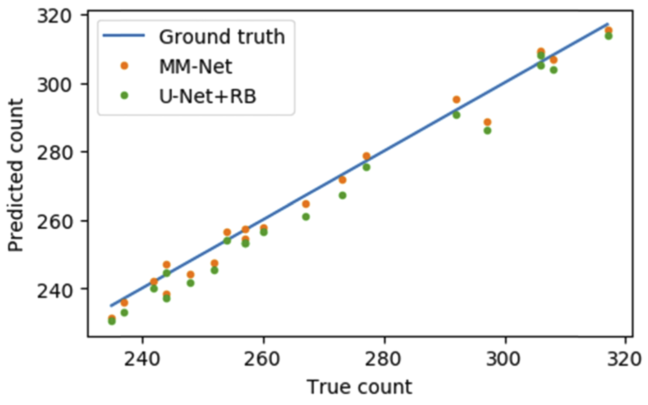

Fig. 9. The predictions of U-Net + RB and the MM-Net on some test images

5. Conclusion

In this paper, a multi-column multi-resolution network based on U-net is proposed to count cells automatically. With the various receptive fields, more details of images can be captured and the network is adaptive to learn features of the cells with various sizes. The residual blocks on MM-Net make the network converge better. To improve the performance, a region-based loss function is proposed to decrease the overall loss. With the pooling and up-sampling layers, the proposed model is pixel-to-pixel and the output has the same resolution as the input. We can not only get the estimated count result but also the positions of cells, especially for the less-crowded cells. Experiments on three various datasets confirm that the proposed method works well for cell counting, and the detection based on regression as an auxiliary task also provides more position information and even performs better than the regression under some circumstance.

Acknowledgments

Sincere thanks to Lempitsky, Zisserman, Cohen and Arteta for providing the datasets we used in this paper.

Disclosures

The authors declare no conflicts of interest.

References

1. S. Kothari, Q. Chaudry, and M. D. Wang, "Automated cell counting and cluster segmentation using concavity detection and ellipse fitting techniques," in *2009 IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, (IEEE, 2009), 795–798.
2. C. Zhang, C. Sun, R. Su, and T. D. Pham, "Segmentation of clustered nuclei based on curvature weighting," in *Proceedings of the 27th Conference on Image and Vision Computing New Zealand*, 2012, 49–54.
3. V. Lempitsky and A. Zisserman, "Learning to count objects in images," in *Neural Information Processing Systems 2010*, 2010, 1324–1332.
4. J. Paul Cohen, G. Boucher, C. A. Glastonbury, H. Z. Lo, and Y. Bengio, "Count-ception: Counting by fully convolutional redundant counting," in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2017), 18–26.
5. L. Fiaschi, U. Köthe, R. Nair, and F. A. Hamprecht, "Learning to count with regression forest and structured labels," in *Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012)*, (IEEE, 2012), 2685–2688.

6. R. M. Rad, P. Saeedi, J. Au, and J. Havelock, "Cell-Net: Embryonic Cell Counting and Centroid Localization via Residual Incremental Atrous Pyramid and Progressive Upsampling Convolution," IEEE Access **7**, 81945–81955 (2019).
7. W. Xie, J. A. Noble, and A. Zisserman, "Microscopy cell counting and detection with fully convolutional regression networks," Computer methods in biomechanics and biomedical engineering: Imaging & Visualization **6**(3), 283–292 (2018).
8. M. von Borstel, M. Kandemir, P. Schmidt, M. K. Rao, K. Rajamani, and F. A. Hamprecht, "Gaussian process density counting from weak supervision," in *European Conference on Computer Vision*, (Springer, 2016), 365–380.
9. Y. Xue, N. Ray, J. Hugh, and G. Bigras, "Cell counting by regression using convolutional neural network," in *European Conference on Computer Vision*, (Springer, 2016), 274–290.
10. O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*, (Springer, 2015), 234–241.
11. T. Chan and L. Vese, "An active contour model without edges," in *International Conference on Scale-Space Theories in Computer Vision*, (Springer, 1999), 141–151.
12. C. Xu and J. L. Prince, "Snakes, shapes, and gradient vector flow," IEEE Transactions on image processing **7**(3), 359–369 (1998).
13. C. Li, C. Xu, C. Gui, and M. D. Fox, "Distance regularized level set evolution and its application to image segmentation," IEEE transactions on image processing **19**(12), 3243–3254 (2010).
14. C. Jung, C. Kim, S. W. Chae, and S. Oh, "Unsupervised segmentation of overlapped nuclei using Bayesian classification," IEEE Trans. Biomed. Eng. **57**(12), 2825–2832 (2010).
15. C. Arteta, V. Lempitsky, J. A. Noble, and A. Zisserman, "Learning to detect cells using non-overlapping extremal regions," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, (Springer, 2012), 348–356.
16. C. Arteta, V. Lempitsky, J. A. Noble, and A. Zisserman, "Learning to detect partially overlapping instances," in *Proceedings of the IEEE Conference on Computer Vision And Pattern Recognition*, 2013), 3230–3237.
17. C. Arteta, V. Lempitsky, J. A. Noble, and A. Zisserman, "Detecting overlapping instances in microscopy images using extremal region trees," Med. Image Anal. **27**, 3–16 (2016).
18. C. Arteta, V. Lempitsky, J. A. Noble, and A. Zisserman, "Interactive object counting," in European conference on computer vision, (Springer, 2014), 504–518.
19. R. Zhu, D. Sui, H. Qin, and A. Hao, "An extended type cell detection and counting method based on FCN," in *2017 IEEE 17th International Conference on Bioinformatics and Bioengineering (BIBE)*, (IEEE, 2017), 51–56.
20. X. Pan, D. Yang, L. Li, Z. Liu, H. Yang, Z. Cao, Y. He, Z. Ma, and Y. Chen, "Cell detection in pathology and microscopy images with multi-scale fully convolutional neural networks," World Wide Web **21**(6), 1721–1743 (2018).
21. F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," arXiv preprint arXiv:1511.07122 (2015).
22. R. M. Rad, P. Saeedi, J. Au, and J. Havelock, "Blastomere cell counting and centroid localization in microscopic images of human embryo," in *2018 IEEE 20th International Workshop on Multimedia Signal Processing (MMSP)*, (IEEE, 2018), 1–6.
23. E. Walach and L. Wolf, "Learning to count with cnn boosting," in European conference on computer vision, (Springer, 2016), 660–676.
24. S. U. Akram, J. Kannala, L. Eklund, and J. Heikkilä, "Cell segmentation proposal network for microscopy image analysis," in Deep Learning and Data Labeling for Medical Applications (Springer, 2016), pp. 21–29.
25. Y. Kong, H. Li, Y. Ren, G. Z. Genchev, X. Wang, H. Zhao, Z. Xie, and H. Lu, "Automated yeast cells segmentation and counting using a parallel U-Net based two-stage framework," OSA Continuum **3**(4), 982–992 (2020).
26. C. Zhang, H. Li, X. Wang, and X. Yang, "Cross-scene crowd counting via deep convolutional neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, 833–841.
27. Y. Zhang, D. Zhou, S. Chen, S. Gao, and Y. Ma, "Single-image crowd counting via multi-column convolutional neural network," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, 589–597.
28. D. Kang, Z. Ma, and A. B. Chan, "Beyond Counting: Comparisons of Density Maps for Crowd Analysis Tasks—Counting, Detection, and Tracking," IEEE Transactions on Circuits and Systems for Video Technology **29**(5), 1408–1422 (2019).
29. B. Yang, J. Cao, N. Wang, Y. Zhang, and L. Zou, "Counting challenging crowds robustly using a multi-column multi-task convolutional neural network," Signal Processing: Image Communication **64**, 118–129 (2018).
30. K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," in *European conference on computer vision*, (Springer, 2016), 630–645.
31. K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016), 770–778.
32. D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," arXiv preprint arXiv:1412.6980 (2014).
33. L. N. Smith, "Cyclical learning rates for training neural networks," in *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, (IEEE, 2017), 464–472.
34. P. Kainz, M. Urschler, S. Schulter, P. Wohlhart, and V. Lepetit, "You should use regression to detect cells," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, (Springer, 2015), 276–283.

35. E. Lu, W. Xie, and A. Zisserman, “Class-agnostic counting,” in *Asian conference on computer vision*, (Springer, 2018), 669–684.