Interactive Dual Network With Adaptive Density Map for Automatic Cell Counting

Rui Liu, Yudi Zhu, Cong Wu, Hao Guo, Wei Dai, *Graduate Student Member, IEEE*, Tianyi Wu, Min Wang, *Graduate Student Member, IEEE*, Wen Jung Li, *Fellow, IEEE*, and Jun Liu, *Member, IEEE*

***Abstract*— Cell counting is an essential step in a wide variety of biomedical applications, such as blood examination, semen assessment, and cancer diagnosis. However, microscopic cell counting is conventionally labor-intensive and error-prone for experts, and most of the existing automatic approaches are confined to a specific image type. To address these challenges, we propose a new interactive dual-network framework for automatic and generic cell counting. In this framework, one deep learning model (counter) is trained to regress a density map from a given microscope image. The number of cells in that image can be estimated by performing integration over the regressed density map. Another network (ground truth generator) is employed to dynamically generate suitable ground truth based on the cell samples and the dot annotations to serve as the supervision for training the counter. The interactive process to obtain the optimal model is achieved by jointly training the counter and ground truth generator iteratively. Moreover, we design a hierarchical multi-scale attention-based architecture to act as the counter in the proposed framework. This architecture is crafted to efficiently and effectively process multi-level features, enabling accurate regression of high-quality density maps. Evaluation experiments on three public cell counting datasets demonstrate the superiority of our method.**

***Note to Practitioners*—This paper is motivated by the need for advanced healthcare in the deep learning era. As a routine assessment procedure in healthcare settings, cell counting usually suffers from poor accuracy and inefficiency. We provide a solution to ameliorate the situation by developing a deep learning-based framework for automatic cell counting. After being trained in an end-to-end manner, the dual-network system is able to estimate the number of cells from the given microscopic images more accurately than existing methods. Additionally, this method is robust in various scenarios, such as calculating cell populations in suspension and cells in tissues. In the future, the presented pipeline has the potential to be implemented by biomedical practitioners who are non-expert in programming via wrapping it into a graphical user interface.**

***Index Terms*— Automatic cell counting, healthcare automation, deep learning in healthcare, interactive dual network, density map.**

I. Introduction

CELL counting in microscopic analysis can provide a critical indicator for medical diagnosis and treatment. For instance, in the Kleihauer–Betke test, fetal-maternal hemorrhage is quantitated by counting the fetal and maternal red blood cells [1]. However, counting cells manually under the microscope is tedious, labor-intensive, and prone to subjective errors, especially in cases of high cell density, occlusions in microscopic images, and large inter-individual morphological variation of cells. Automated cell counting methods have been developed to reduce human involvement. However, most of the conventional automated counting methods are tailored to specific cell images, and the accuracy is also hindered by the inherent drawbacks in crowded cell samples [1], [2], [3]. Therefore, a generic cell counting system is needed to offer a computer-aided diagnosis with sufficient accuracy.

In recent years, medical diagnostics has witnessed significant advancements with the development of artificial intelligence technology. Deep learning-based methods have emerged as powerful tools in various medical applications, including lesion detection in dental care [4], identification of breast cancer [5], and malaria detection [6]. Deep learning technology has also demonstrated potential superiority for cell counting [7], [8], [9], [10].

Based on their working principle, the cell counting methods can be grouped into detection-based and regression-based categories. The number of cells in a microscopic image can be obtained by detecting each cell instance in detection-based counting, as illustrated in Fig. 1(a). However, object detection in dense images is a nontrivial task, and the accumulative errors can significantly damage the accuracy of cell counting. Moreover, most deep learning-based object detection

Manuscript received 1 June 2023; revised 1 September 2023; accepted 11 October 2023. Date of publication 9 November 2023; date of current version 16 October 2024. This article was recommended for publication by Associate Editor X. Liu and Editor L. Zhang upon evaluation of the reviewers' comments. This work was supported in part by the Research Grant Council (RGC) of Hong Kong under Grant 11212321, Grant 11217922, and Grant ECS-21212720; in part by the Basic and Applied Basic Research Foundation of Guangdong Province Fund Project under Grant 2019A1515110175; and in part by the Science, Technology and Innovation Committee of Shenzhen under Grant SGDX20210823104001011. *(Corresponding author: Jun Liu.)*

Rui Liu, Yudi Zhu, Wei Dai, Tianyi Wu, Min Wang, and Jun Liu are with the Department of Mechanical Engineering and the Centre for Robotics and Automation, City University of Hong Kong (CityU), Hong Kong, SAR, China, and also with the CAS–CityU Joint Laboratory for Robotics Research, City University of Hong Kong Shenzhen Research Institute, Shenzhen 518057, China (e-mail: rui.liu@my.cityu.edu.hk; jun.liu@cityu.edu.hk).

Cong Wu and Wen Jung Li are with the Department of Mechanical Engineering and the Hong Kong Centre for Cerebro-Cardiovascular Health Engineering (COCHE), City University of Hong Kong, Hong Kong, SAR, China, and also with the CAS–CityU Joint Laboratory for Robotics Research, City University of Hong Kong Shenzhen Research Institute, Shenzhen 518057, China.

Hao Guo is with the School of Software Engineering, University of Science and Technology of China, Hefei 230026, China.

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TASE.2023.3329973.

Digital Object Identifier 10.1109/TASE.2023.3329973

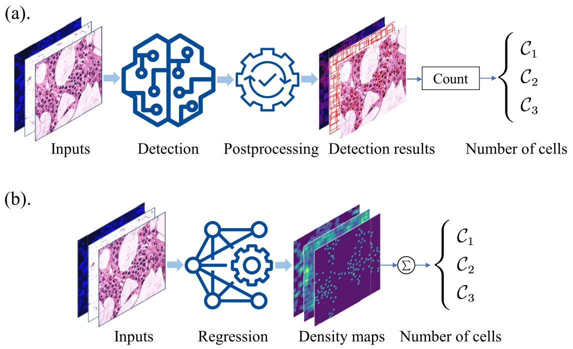


Fig. 1. The workflow of automatic cell counting. (a) Detection-based method. This pipeline acquires the cell count by identifying the position of individual cell instances, which can be intricate, especially for counting cells with a high density. (b) Regression-based method. In this paradigm, the network regresses a density map for each image, with each pixel indicating the probability of a cell's presence. The summation of these pixel values in the density map yields the cell count.

algorithms are computationally expensive and require time-consuming annotations (e.g., labeling with bounding boxes) [11], [12]. On the other hand, regression-based counting does not require prior detection or segmentation of individual objects in a crowded cell image. Instead, it casts the counting task as regressing a spatial density map to output the integral as the number of instances in the given image, as shown in Fig. 1(b). Furthermore, the regression-based pipeline only needs weak annotation, usually a dot or a blob for each object, which is much more cost-efficient.

Owing to its superior performance, density map-based counting has emerged as the mainstream direction over the past few years, especially for counting dense cells. Since the density map is an intermediate representation, it needs to generate the ground truths for supervised learning from the given dot annotations. In previous studies, the ground truths were usually handcrafted by blurring the dots with a fixed Gaussian filter. Although existing deep learning-based methods have brought automatic cell counting a big step forward, the use of handcrafted ground truths in these pipelines limits the potential for further reduction of counting errors.

In this paper, we present an adaptive density map-based framework to improve the accuracy of automatic cell counting in microscopic analysis. Our method highlights the importance of the density map patterns during the training of the counter, which was overlooked in previous approaches. We update the ground truth generation paradigm for density-based cell counting using a deep learning-based generator. Furthermore, a novel counting model is developed to regress high-quality density maps for the given cell images. In summary, the main contributions of this study are as follows:

1) We propose an interactive dual-network framework for automatic cell counting. In this framework, one network acts as the cell counter, while the other is employed to generate the ground truth density maps for the supervised learning of the counter. Through joint training with the counting network, the ground truth (GT) generator can adjust the ground truth maps based on the feature of the input images. Therefore, the density representations with the minimum cell count error can be obtained. To the best of our knowledge, this is the first foray into reducing cell count error through adaptive density maps.
2) We design a hierarchical multi-scale attention-based counter. In this architecture, multi-level features can be processed effectively and efficiently for accurate density map regression.
3) We develop a GT generator based on a convolutional neural network (CNN). The generator can produce suitable ground truth maps based on the features extracted from the cell samples.
4) Our method outperforms the state-of-the-art on two standard public benchmarks, i.e., the VGG [3] and MBM datasets [13], and is on par with competitors on the ADI image set [13], verifying its effectiveness.

The remainder of this article is organized as follows. Section II reviews the related works. The proposed cell counting framework is presented in Section III. The experimental setup and results are reported in Section IV and Section V, respectively, followed by a discussion in Section VI. Finally, Section VII concludes this work.

II. Related Work

This section reviews the recent advances in cell counting methods, followed by a brief discussion of the attention mechanism.

A. Cell Counting by Detection

Detection-based approaches rely on acquiring the location of each cell instance in the microscopic image for counting. Xing et al. [14] reported a Ki-67 counting algorithm based on detection and online dictionary learning to automate the grading of neuroendocrine tumors. Chowdhury et al. [6] modified the YOLO framework [11] to complete blood cell count and malaria detection. Arteta et al. [15] used a tree-shape model to overcome the challenge of detecting and counting cells in overlapping scenarios. Kassim et al. [16] designed a cascaded pipeline for red blood cell detection and counting. In their approach, a U-Net [17] was first employed to produce clusters, proceeding with the Faster R-CNN [18] performing the cell detection within the connected components. Alam et al. [19] proposed a method to count blood cells based on YOLO [11] and then alleviated the repeated counting problem using the K-nearest neighbors and the intersection of the union. The counting accuracy of the above methods depends largely on the cell detection results. However, the cell morphology varies significantly, affecting cell detection accuracy. Moreover, the image acquisition conditions can also change the image features, posing further challenges to the detection. Therefore, the detector needs to be carefully designed and this is not a cost-effective and reliable way to count.

B. Cell Counting by Regression

Lempitsky and Zisserman [3] conducted pioneering research for counting cells by regression. They developed a flexible learning system to accurately and efficiently count objects in various domains by exploiting spatial features. Similarly, Arteta et al. [20] reported an interactive framework to rapidly count cell instances in crowded scenarios by leveraging ridge regression. Fiaschi et al. [21] estimated the density map by averaging the predictive patches, which were output by the regression forest based on the input image. In [22], several random forest algorithms were trained in parallel to determine probability maps individually, and then the probability maps were averaged to obtain the final estimated map and cell count.

More recently, deep CNN-based algorithms have been developed to enhance the quality of the predicted density map in an end-to-end manner. In [23], a regression framework using different trendy models, such as AlexNet [24] and ResNet [25], as backbone was proposed to infer the number of cells in a given microscopic image. Xie et al. [26] designed and compared two alternative CNNs implemented end-to-end for probability map regression. Their methods surpassed the previous counterparts in synthetic images regarding counting performance and could be directly transferred to real cellular data with a slight loss in accuracy. Subsequently, He et al. [9] improved the fully convolution model proposed in [26] by concatenating the spatial feature encoded in the shallow layers to the decoding phase and then deeply guided the training of the network with hierarchical loss to reduce the counting error.

To minimize the error, Paul et al. [13] proposed a redundant counting approach that derived the accurate count by averaging the results obtained by the sliding windows. Rodriguez-Vazquez et al. [7] utilized the adversarial training technique to help the generator regress high-quality probability maps and then achieved positioning and counting on the resulting maps with the Laplacian of Gaussian operator. While other systems focused on 2D images, SAU-Net [10], a universal framework, was proposed for both 2D and 3D cell counting. In addition, a custom batch normalization block was embedded into the SAU-Net to enhance its performance on small datasets. Although the refinement of density maps has yet to be investigated in cell counting, the pattern of density maps has been proved critical in counting crowded populations [27].

C. Attention Mechanism

The attention mechanism is a technique that mimics the principle of biological cognition to focus the algorithms' attention on the essential parts of the data. In computer vision, this technology has been applied in various visual recognition tasks, including segmentation, detection, and generation [28], [29], [30]. A typical implementation of the attention mechanism is the squeeze-and-excitation module [31], which attempts to learn a group of channel-wise weights to refine the corresponding feature maps. Woo et al. [32] introduced an attention module to polish the feature maps in both channel and spatial dimensions. Transformer [33], [34] based on the self-attention mechanism has also achieved impressive results in visual representation recently. However, these transformer-based pipelines often face challenges such as a lack of sufficiently large datasets and high computational costs.

III. Methodology

In this section, we present the novel cell counting framework in which the GT generator and cell counter are trained jointly.

A. Problem Formulation

Given an input image $X \in \mathbb{R}^{H\times W\times C}$, the corresponding density map $\widehat{Y} \in \mathbb{R}^{H\times W}$ (where $\mathbb{R}^{H\times W}$ is the abbreviation of $\mathbb{R}^{H\times W\times 1}$) can be obtained by the regression function and represented as:

$$\widehat{Y} = \mathcal{F}_c(\Theta; \Psi(X)), \tag{1}$$

where Θ is the parameter vector of the mapping function $\mathcal{F}_c(\cdot)$ and $\Psi(X)$ is the local features of the input image. The number of cells in the image can be calculated by summing up $\widehat{Y}(h, w)$, which indicates the object existence probability in the $X(h, w)$. In the deep learning pipeline, the density is directly regressed from the image. Therefore, Equation (1) would be simplified to:

$$\widehat{Y} = \mathcal{F}_c(\Theta; X), \tag{2}$$

The critical factor for the success of this framework is to train a deep learning-based mapping function $\mathcal{F}_c(\Theta)$ to produce a correct density map $\widehat{Y}$ for a given region.

To lower the labor cost, the center of cell instances is annotated with a dot to form a dot map. The density map, namely a heat map of the cell distribution, works as an intermediary ground truth for supervised learning since it is hard for the deep learning model to recognize cells directly from a set of sparse dots. The previous study usually generated ground truths $Y \in \mathbb{R}^{H\times W}$ by convolving the dot map $D \in \mathbb{R}^{H\times W}$ with a Gaussian kernel:

$$Y = D * \mathcal{G}(x, y), \tag{3}$$

where $\mathcal{G}(x, y)$ is a bivariate Gaussian kernel with a handcrafted bandwidth. Under the supervision of the handcrafted ground truths, it can effectively train a deep learning-based density map estimator. However, directly computing a constant map from the dot map overlooks the importance of ground truth, which could significantly impact the counting performance. This is because the background and texture of cell images can vary greatly depending on the acquisition scenario. Therefore, we would take a step towards exploiting the properties of samples and attempt to generate a sample-based ground truth map. In practice, a deep learning network would act as the generator $\mathcal{F}_g(\cdot)$ to produce the ground truth:

$$Y = \mathcal{F}_g(\Phi; X, D), \tag{4}$$

where Φ is the parameters of $\mathcal{F}_g(\cdot)$. The cell samples and the corresponding dot maps are the input of the generator. This ground truth generator is jointly trained with the cell counter in the dual-network framework. Therefore, the ground truth density map is dynamically adjusted according to the input cell samples to reduce the counting error interactively.

B. Overall Framework

Based on the conception mentioned above and inspired by [27], we propose an interactive dual-network framework for automatic cell counting, as shown in Fig. 2. In this system, one network called counter (inside the blue dashed box) acts as a mapping function $\mathcal{F}_c(\Theta)$ to regress a density map from the input image for count estimation. The other one (inside the

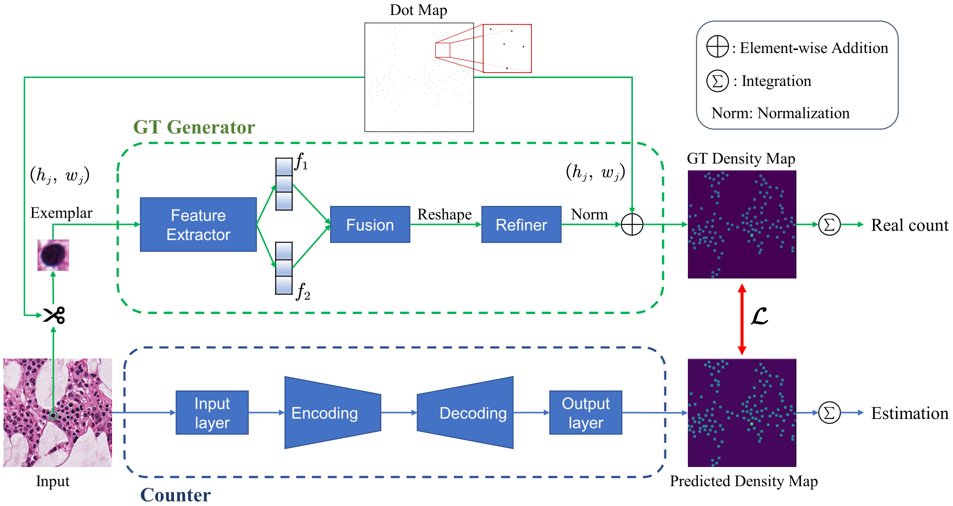


Fig. 2. The proposed interactive dual-network framework for automatic cell counting. In the training stage, the cell samples are input to the counter network to regress the estimated density map. Concurrently, the corresponding cell exemplars are extracted and fed to the GT generator to adjust the ground truth density map dynamically. The trained counter is employed to infer density maps for unseen inputs in the inference phase.

green dashed box) is used as GT generator $\mathcal{F}_g(\Phi)$ to modify ground truth iteratively.

In this pipeline, the role of the GT generator is to aid in the supervised learning of the counter since it can adjust the ground truth density maps according to the feature of input cell samples. Throughout the whole training phase, the counter and the GT generator interact iteratively by minimizing the element-wise mean squared error loss $\mathcal{L}$ between the estimated density maps $\{\widehat{Y}_i\}_1^N$ and the corresponding ground truth density maps $\{Y_i\}_1^N$. The loss function is expressed as:

$$\begin{aligned}\mathcal{L} &= \sum_{i=1}^{N} \left\| \widehat{Y}_i - Y_i \right\|^2 + \lambda(\|\Theta\|^2 + \|\Phi\|^2) \\ &= \sum_{i=1}^{N} \left\| \mathcal{F}_c(\Theta; X_i) - \mathcal{F}_g(\Phi; S_i, D_i) \right\|^2 \\ &\quad + \lambda(\|\Theta\|^2 + \|\Phi\|^2), \end{aligned} \tag{5}$$

where $S \in \mathbb{R}^{M\times M\times C}$ denotes the cell exemplar, and λ is a coefficient that modulates the l_2 penalty to alleviate overfitting.

C. Multi-Scale Attention-Based Counter for Cell Counting

In order to obtain an accurate heat map of the distribution of cell instances, a superior density map estimator should be carefully designed to encode the spatial information. The encoder block constructed with convolution units is reported to be a powerful feature extractor in various tasks [24], [35]. On the other hand, the encoded information needs to be resolved to acquire the full-size map in the decoding process [17], [36]. Therefore, a deep CNN-based encoder-decoder structure is employed to be the basic backbone of the counter. As shown in Fig. 3, the features of different reception fields are captured in each encoder via a convolution operation and a dilated one and then concatenated along the channel dimension. Compared with ordinary convolution, dilated convolution has a larger receptive field without extra computational cost [37]. Thus, spatial features from different receptive fields can be encoded in a computation-efficient manner. Each decoder is composed of two concatenated convolution-normalization-ReLU operations.

A straightforward implementation of an autoencoder-based counter is to connect Encoder 1 and Decoder 1 directly. However, in the bare encoding-decoding pipeline, the decoder receives and processes all the feature units from the encoder indiscriminately, which is inefficient and unsatisfactory. For the decoder to better interpret the feature representation, a channel-spatial-attention module is built by cascading a squeeze-and-excitation network [31] and a spatial attention component. The attention module is applied to refine the encoded features by assigning a learnable weight to each element, as illustrated in Fig. 4.

The counter consisting of only one encoder-decoder can only capture local low-level features of the input image, which is insufficient to regress a high-quality density map. Therefore, Multiple encoder-decoder pairs are employed to process multi-scale feature maps in the proposed counter to exploit the multi-level information fully. These encoders are cascaded via max pooling, with transposed convolution operations as upsampling between corresponding decoders. After multiple max pooling processes, the size of the feature map would be reduced, but each element would perceive more information. Therefore, in this hierarchical autoencoder-based structure, the shallow layer (encoder1-decoder1) captures spatial context, while the deeper counterpart (encoder3-decoder3) learns high-level semantic features. The counter can estimate a relatively precise density map by fusing these multi-level features. The overall architecture of the proposed multi-scale attention-based counting pipeline is depicted in Fig. 3.

D. Dynamic Adaptive Density Map Generation

Cells exhibit diverse morphologies. As a result, the ground truth that reflects the characteristics of a given microscopic image is preferred to provide better supervision for training the counting network. Therefore, our proposed framework

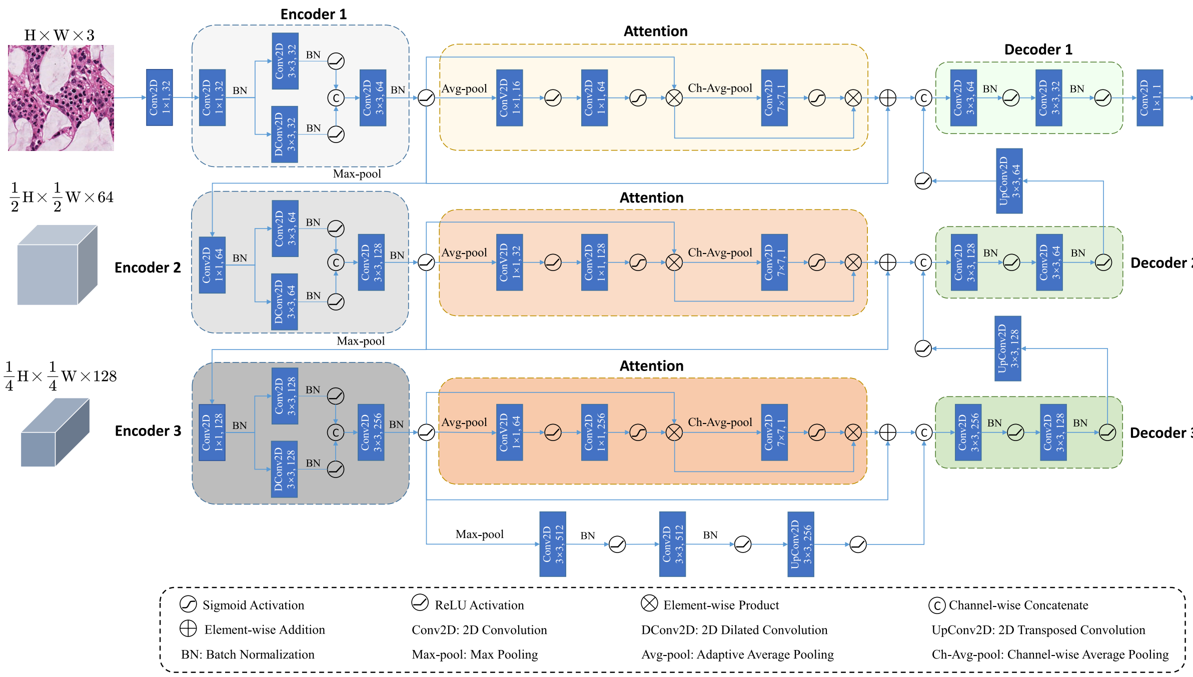


Fig. 3. The proposed multi-scale attention-based encoding-decoding network for density map regression. [Conv2D, $k \times k$, N] represents a 2D convolution operation with the output channel number of N and the kernel size of $k \times k$. The dilated convolution operation (with a dilation of 2) and the transposed convolution operation also have the same description. The stride size of the convolution and dilated convolution is set to 1, while the stride size of the transposed convolution operations is 2. The kernel size and stride of the max pooling in our framework are 2×2 and 2, respectively. The padding operation is used to adjust the output of the convolutions to the desired size.

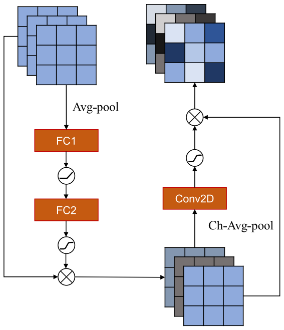


Fig. 4. The schematic diagram of the channel-spatial-attention module. FC refers to the Fully Connected layer, which is implemented by the convolution operation with a kernel of 1×1 in the counting network.

employs a deep CNN as the GT generator, benefiting from its superior nonlinear representation capabilities. Specifically, the CNN extracts features from the input images and produces corresponding ground truth density maps to supervise the training of the cell counter. The generator includes three components: the feature extractor, the fusion block, and the refiner, as shown in Fig. 2. The feature extractor captures the spatial information of the input cell samples and then outputs the corresponding feature vectors. Inspired by [38], these vectors are then modulated by a normal distribution in the fusion module to refine feature vectors by leveraging a re-parameterization loss $\mathcal{L}_r$, which is defined as:

$$\mathcal{L}_r = \frac{\alpha}{2}\sum_{i=1}^{d}((f_1^i)^2 + (f_2^i)^2 - \ln(f_2^i)^2 - 1), \qquad (6)$$

where d is the dimensionality of the feature vectors, f_1^i and f_2^i represent the i-th component of the mean vector and logarithmic variance vector, respectively. The parameter α is the weight of the loss and is set to 0.05. Next, the processed feature representations are fed into the refinement module to dynamically generate a refined kernel. Inspired by the kernel-based strategy [27], the refined kernel is normalized and added to a new blank map, serving as the target for the training of the counting network. The normalization process is defined as follows:

$$\widetilde{k}_{i,j} = \frac{k_{i,j} - \min\left\{(k_{0,0}), \cdots, (k_{P-1,P-1})\right\} + \varepsilon}{\sum\limits_{i=0,j=0}^{i=P-1,j=P-1}\left[k_{i,j} - \min\left\{(k_{0,0}), \cdots, (k_{P-1,P-1})\right\} + \varepsilon\right]}, \qquad (7)$$

where $k_{i,j}$ and $\widetilde{k}_{i,j}$ represent the values of the position (i, j) before and after normalization, respectively, in a kernel of size $P \times P$. ε is a minimal value used to prevent the occurrence of outliers. Each kernel on the ground truth corresponds one-to-one with a cell in the microscope image, ensuring that the integral of the ground truth equals the number of real instances. The centroid coordinates from the dot maps are retrieved to center the kernels at the accurate locations, as shown in Fig. 2. Given an image and dot map pair $\{X, D\}$ with T_0 annotated coordinates $\left\{C_j\right\}_1^{T_0}$ in D, the corresponding ground truth density map Y, can be expressed as follows:

$$Y = \mathcal{F}_g(\Phi; S, D) = \sum_{h_1,w_1}^{h_{T_0},w_{T_0}} \widetilde{K}_{h_j,w_j}, \quad (h_j, w_j) \in \left\{C_j\right\}_1^{T_0}, \qquad (8)$$

TABLE I
THE ARCHITECTURE OF THE GT GENERATOR

Components	Operations	Input Channels	Output Channels
Feature Extraction	Conv2D-ReLU	3	32
	Max-pool	32	32
	Conv2D-ReLU	32	64
	Max-pool	64	64
	Conv2D-ReLU	64	128
	Max-pool	128	128
	Flatten	-	-
	FC1 FC2	-	-
Fusion Block	Fusion	-	-
	Reshape	-	-
Refiner	Conv2D-ReLU	1	16
	Conv2D-ReLU	16	128
	Conv2D-ReLU	128	16
	Conv2D-ReLU	16	1

where $\widetilde{K}_{h_j,w_j}$ is the normalized kernel centered on the coordinate (h_j, w_j) on the ground truth map. The ground truth density map is updated to minimize the mean squared error between the output of the generator and that of the counter, as calculated by Equation (5). This is achieved through the interactive training process between the two networks, where the generator produces a refined ground truth, and the counter predicts an estimated density map based on that updated ground truth.

Explicitly, the architecture details of the GT generator are summarized in Table I. Conv2D-ReLU refers to the 2D convolution operation followed by ReLU activation. If not specified, the kernel size and stride of the Conv2D are 3×3 and 1, respectively. FC1 and FC2 have an identical input, which is the flattening of the output of the previous layer. The output of FC1 is assumed to be the mean vector of the feature space, while that of FC2 represents the variance vector. The number of neurons in the output layers of both the FC1 and FC2 is set to P^2. The fusion operation is defined as $f_1 + e^{f_2}$, where f_1 and f_2 are the outputs of FC1 and FC2, respectively. The output of the fusion operation is reshaped into the shape of the kernel and then input to the refiner for refinement.

E. The Training and Inference Processes

During training, the microscopic images are fed into the counter, while the corresponding cell exemplars are cropped and inputted to the GT generator. In this procedure, the centroid coordinates of the cell instances provided by the dot maps are used to generate the ground truths and aid the system in extracting cell samples automatically and accurately from the full-size image. Once training is complete, the generator is stopped, and the trained counter is executed to predict a high-quality density map for an unseen input cell image in the inference stage. Integration over the predicted density map is then applied to approximate the number of cells.

The joint training process and inference procedure of the proposed system are encapsulated in Algorithm 1.

Algorithm 1 The Joint Training Process and Inference Procedure of the Proposed Framework

```
Input: The training cell images $\{X_i\}_1^N$ and the corresponding dot maps $\{D_i\}_1^N$; the new set of microscopic images $\{X_i'\}_1^{N'}$.
Output: The trained cell counter $\mathcal{F}_c(\Theta)$ and the trained GT generator $\mathcal{F}_g(\Phi)$; the inferred density map $\left\{\widehat{Y_i}'\right\}_1^{N'}$ and estimated count $\left\{\widehat{c_i}'\right\}_1^{N'}$.

# The Joint Training Process
Initialize the counter $\mathcal{F}_c(\Theta)$ and the GT generator $\mathcal{F}_g(\Phi)$.

for all $epoch = \{1, \ldots, epoch_{max}\}$ do
    for all $i \in (1, N)$ do
        Crop cell exemplar $S_i$ from the $X_i$.
        Feed $S_i$ and $D_i$ into $\mathcal{F}_g(\Phi)$.
        Feed $X_i$ into $\mathcal{F}_c(\Theta)$.
        $Y_i \longleftarrow \mathcal{F}_g(\Phi; S_i, D_i)$.
        $\widehat{Y_i} \longleftarrow \mathcal{F}_c(\Theta; X_i)$.
        Update $\mathcal{F}_c(\Theta)$ with the loss calculated by Equation (5).
        Update $\mathcal{F}_g(\Phi)$ with the loss calculated by Equation (5) and update the feature extraction part of $\mathcal{F}_g(\Phi)$ with the loss computed by Equation (6).
    end for
end for
return the trained $\mathcal{F}_c(\Theta)$ and $\mathcal{F}_g(\Phi)$.

# The Inference Process
Initialize the $\mathcal{F}_c(\Theta)$ with the trained parameters.
for all $i \in \left(1, N'\right)$ do
    Feed $X_i'$ into $\mathcal{F}_c(\Theta)$.
    $\widehat{Y_i}' \longleftarrow \mathcal{F}_c\left(\Theta; X_i'\right)$.
    Obtain $\widehat{c_i}'$ by summing up the elements in $\widehat{Y_i}'$ .
end for
return $\left\{\widehat{Y_i}'\right\}_1^{N'}$, $\left\{\widehat{c_i}'\right\}_1^{N'}$.
```

IV. EXPERIMENTAL SETUP

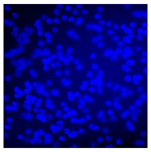
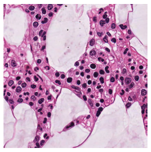
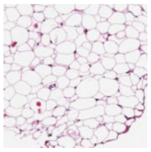
In this section, we provide a comprehensive description of the datasets used for the experimental evaluation of the proposed framework, as well as the implementation details.

A. Dataset Description

The proposed method are evaluated on three publicly available benchmarks widely used for automatic cell counting: the synthetic bacterial cell dataset, the bone marrow cell image set, and the human subcutaneous adipose tissue database. Details of these datasets are provided in Table II.

1) Synthetic Bacterial Cell Dataset (VGG): the synthetic bacterial cell dataset was produced by Lempitsky et al. [3] using a simulation platform developed by Lehmussola et al. [39]. The provider of this database is the Visual Geometry Group of the University of Oxford. This dataset is also called VGG. This dataset consists of 200 synthetic RGB

TABLE II
DATASET DETAILS

Datasets	VGG	MBM	ADI
Scenarios	Synthetic	Real	Real
Image size	256×256×3	600×600×3	150×150×3
Min. count	74	65	48
Max. count	317	195	299
Avg. count	176	126	148
# of images	200	44	200
Examples			

fluorescent microscopic images of size 256×256, containing 35192 objects to simulate bacterial cells. The synthetic bacterial cell dataset is designed to increase the difficulty of cell counting by generating clustered and overlapping bacteria, as well as simulating images with varying focal distances.

2) Bone Marrow Cell Dataset (MBM): the bone marrow cell dataset was introduced by Paul et al. [13] by modifying the original version reported in [40]. These Hematoxylin and Eosin stained samples were collected from healthy human bone marrow. This database has 44 RGB images of 600×600 pixels, accommodating 5553 cells. These cells are not easy to recognize due to the inhomogeneous background in these microscopic images.

3) Human Subcutaneous Adipose Tissue Dataset (ADI): the samples in the human subcutaneous adipose tissue dataset were acquired from the Genotype-Tissue Expression Consortium (GTEx) [41] and then down-sampled to 150×150 pixels by Paul et al. [13]. This sample set has 200 RGB images with 29684 instances in total. In addition to significant intra-class variation, the cells in these images are tightly packed, making the counting extremely challenging.

B. Implement Details

For a fair comparison with the competitors, 50 samples were randomly selected from the VGG and ADI datasets for training in each experiment, and the rest were used as the test set. Similarly, only 15 samples from the MBM were utilized for training the networks, and the remaining were used for testing. During the training, multiple processes were performed to augment the data to avoid overfitting and lower the counting errors. Firstly, each input image was randomly cropped to 87.5% of its original size. In this process, the height and width of the obtained new sample were rounded down to be divisible by eight because there were three cascaded max pooling operations with a kernel size of two in the counting network. Subsequently, the cropped images were randomly flipped horizontally and vertically, proceeding with being rotated with an angle of $N \times 90^\circ (N = 1, 2, 3, 4)$. The same procedures were carried out in the corresponding dot map. The size of the cell exemplar was set to 24 × 24 pixels for VGG images and 32 × 32 pixels for MBM and ADI datasets. The kernel size of each instance on the ground truth was 15 × 15 pixels. In addition, to accommodate the counter while maintaining the real number of cells, each image-dot-map-pair in the test set of ADI was padded to 152×152 pixels, respectively.

According to the size of the training set in each dataset, the batch size was set to 32 for VGG and ADI, respectively, and 8 for MBM. The Adam approach was utilized to optimize the models [42]. The weight decay for the optimizer was set to 0.0001 to alleviate overfitting. Inspired by the previous study [43], we employed a learning rate scheduler that combined cosine annealing with warming up for joint training. The initial learning rates were set to 0.0005 for the counter and 1e-6 for the GT generator. We trained for 2000 epochs, including a warming-up epoch of 10. Each set of experiments was repeated five times, and the average testing results were taken as the performance of the method. We conducted a grid search to find optimal hyperparameter settings based on performance on the validation datasets. The proposed pipeline was implemented in PyTorch and supported by Python. The training was conducted on one Nvidia RTX3090 GPU with 24GB memory and an Intel Xeon Platinum 8375C CPU.

V. EXPERIMENTAL RESULTS

The experimental results evaluating the proposed framework are reported in this section. We first specify the evaluation metrics. Then, we compare the proposed approach with state-of-the-art methods and detection-based counterparts regarding cell counting accuracy. Ablation studies are also employed to verify the effectiveness of the designs of our system.

A. Evaluation Metrics

The mean absolute error (MAE) between the real count and the estimated cell number was measured to evaluate the performance of the methods. Explicitly, the MAE can be defined as:

$$MAE = \frac{1}{N'}\sum_{i=1}^{N'}\left|\widehat{c_i}' - c_i'\right|, \tag{9}$$

where N' is the number of samples in the test set. $\widehat{c_i}'$ and c_i' refer to the estimated cell count and real cell count of the i-th sample, respectively. The estimated cell count $\widehat{c_i}'$ is calculated as follows:

$$\widehat{c_i}' = \sum_{h=0}^{H-1}\sum_{w=0}^{W-1}(\widehat{Y_i}')_{h,w}, \tag{10}$$

where $(\widehat{Y_i}')_{h,w}$ is the element value at position $(h,\ w)$ of the i-th predicted density map. Similarly, the real cell count c_i' can be computed by:

$$c_i' = \sum_{h=0}^{H-1}\sum_{w=0}^{W-1}(Y_i')_{h,w}, \tag{11}$$

where $(Y_i')_{h,w}$ is the value at position $(h,\ w)$ of the i-th ground truth density map.

TABLE III

COMPARING WITH STATE-OF-THE-ART METHODS ON STANDARD CELL COUNTING DATASETS

Methods	VGG		MBM		ADI	
	MAE↓	N_{train}	MAE↓	N_{train}	MAE↓	N_{train}
ResNet-152 (R), Xue et al. (2016) [23]	7.5±2.2	100	-	-	-	-
GMN, Lu et al. (2019) [44]	3.56±0.27	32	-	-	-	-
Arteta et al. (2014) [20]	3.5±0.1	32	-	-	-	-
Marsden et al. (2018) [45]	-	-	20.5±3.5	15	-	-
Adiposoft, Galarraga et al. (2012) [46]	-	-	-	-	14.8±13.6†	50
FCRN-A, Xie et al. (2018) [26]	2.9±0.2	64	21.3±9.4‡	15	-	-
Cell-Net, Rad et al. (2019) [47]	2.2±0.5	100	9.8±3.2	20	-	-
C-FCRN+Aux, He et al. (2021) [9]	2.37±2.27	160	6.55±5.26	32	-	-
Jiang and Yu (2020) [48]	2.1±0.1	50	7.5±0.7	15	-	-
CSRNet, Li et al. (2018) [49]*	3.3±0.3	50	9.2±1.1	15	15.7±0.9	50
SGANet, Wang and Breckon (2022) [50]*	2.7±0.6	50	5.7±0.6	15	12.6±0.5	50
SAU-Net, Guo et al. (2021) [10]	2.6±0.4	64	5.7±1.2	15	14.2±1.6	50
CCF, Jiang et al. (2020) [22]	2.6±0.1	50	8.6±0.3	15	14.5±0.4	50
GauNet (ResNet-50), Cheng et al. (2022) [51]*	2.5±0.3	50	5.8±0.8	15	13.6±0.9	50
Ciampi et al. (2022) [8]	2.5±0.1	50	5.7±0.9	15	**8.7±0.8**	50
Count-Ception, Paul et al. (2017) [13]	2.3±0.4	50	8.3±2.3	15	19.4±2.2	50
Jiang and Yu (2021) [52]	2.2±0.2	50	6.0±0.6	15	10.6±0.3	50
Rodriguez-Vazquez et al. (2022) [7]	2.2±0.5	50	4.2±2.4	15	17.3±3.6	50
The proposed method	**1.9±0.1**	50	**4.0±0.8**	15	11.1±0.4	50

↓ indicates that the smaller the MAE is preferred for better performance. † The experiments are implemented by Jiang et al. [22]. ‡ The experiments are implemented by Paul et al. [13]. * The experiments are implemented by us and the baseline of GauNet is ResNet-50. The best performance is highlighted in blod. The underline and double underline indicate the second and the third best results, respectively.

B. Comparison With the State-of-the-Art Methods

The proposed method is compared with state-of-the-art approaches in terms of cell counting accuracy on three public datasets: VGG, MBM, and ADI. The mean and standard deviation of the evaluation results are summarized in Table III. Overall, the proposed framework outperforms the previous methods on both the VGG and MBM datasets in terms of MAE and achieves competitive results on the ADI dataset. On the VGG dataset, the proposed pipeline surpasses the leading competitor by a significant margin (9.5%) in counting errors (1.9 vs. 2.1). Likewise, we renewed the record for the MBM dataset (4.0 vs. 4.2), setting an advanced state-of-the-art. As for the ADI dataset, although our approach doesn't obtain the best performance, it still stays at the forefront among all the competitors.

The one that achieves the lowest counting error (8.7) on the ADI dataset is a two-stage counting pipeline presented by Ciampi et al. [8]. However, this method performs relatively poorly on the other two benchmarks, lagging far behind our approach in terms of MAE (2.5 vs. 1.9 on VGG and 5.7 vs. 4.0 on MBM). Furthermore, in order to attain optimal performance for each dataset, Ciampi and his collaborators utilized three different architectures in the first stage of their pipeline to extract the necessary features for counting in the second stage. A similar situation in terms of the MAE is observed in [52], where the reported approach slightly outperforms our framework on the ADI dataset (10.6 vs. 11.1) and also produces inferior results on VGG (2.2 vs. 1.9) and MBM (6.0 vs. 4.0) datasets. These experimental results bear out the superiority of our method.

TABLE IV

COMPARING WITH POPULAR OBJECT DETECTION ALGORITHMS ON STANDARD CELL COUNTING DATASETS

Methods	VGG	MBM	ADI
	(MAE ↓)	(MAE ↓)	(MAE ↓)
YOLOv4 [53]	40.0±3.0	33.4±4.1	78.4±4.0
YOLOv4-Tiny [54]	35.7±3.4	22.6±1.3	71.6±4.7
RetinaNet [55]	35.3±6.2	56.2±7.7	47.2±13.1
QueryDet [56]	38.8±1.4	29.7±3.5	66.9±15.3
PP-YOLOE-SOD [57]	46.2±10.3	15.2±3.4	56.5±14.5
Our method	**1.9±0.1**	**4.0±0.8**	**11.1±0.4**

Further comparing with the state-of-the-art algorithms [49], [50], [51] in crowd counting, our method consistently performs better. This may be attributed to the fact that the techniques tailored to handle the complexity of crowd scenes can be excessively intricate and redundant for cell counting.

C. Comparison With the Object Detection Algorithms

We also compare the proposed dual-network framework with popular object detection algorithms. The ground truths for training the detectors were produced based on the dot annotations to ensure identical cell count. In the comparison experiments, the number of training images is set the same for all the methods. As summarized in Table IV, our method surpasses the detection systems by a large margin in terms of MAE on all three datasets. Different object detection algorithms are suitable for a particular type of microscopic image. RetinaNet achieves higher counting accuracy on VGG and

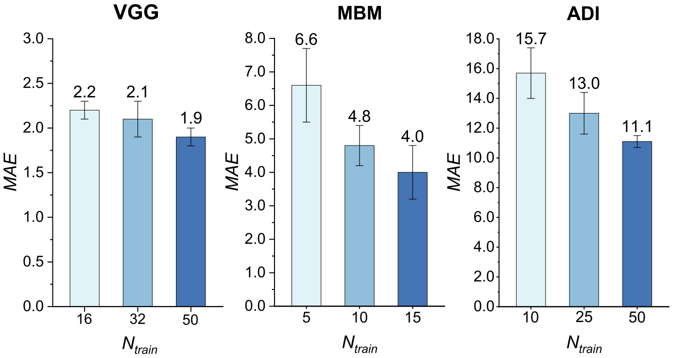


Fig. 5. The change of MAE with the volume of the training set.

ADI datasets, while PP-YOLOE-SOD has better performance on MBM images. Interestingly, the tiny version of YOLOv4 obtains lower errors than its original counterpart, implying that the larger model might be redundant in cell counting applications. These larger models are typically hungry for data and may not be a suitable choice for processing a limited number of microscopic images.

D. Analysis of the Size of the Training Set

We conducted extended experiments to investigate the influence of the training set's volume on the performance of the proposed framework. The experimental results are displayed in Fig. 5. It can be seen that the MAE decreases significantly as the number of training samples increases, implying that the proposed method is likely to produce satisfactory counting accuracy as long as there are enough samples. Surprisingly, the presented pipeline can still yield state-of-the-art results on the VGG dataset with an MAE of 2.1 when trained with fewer images than its competitors (32 vs. 50). Similarly, good performance with an MAE of 4.8 is also achieved on the MBM dataset with only ten training samples. These results reveal that our method is compelling in representation learning and has the potential to be a reliable choice for small-sample scenarios.

E. Ablation Analysis

The ablation study is presented here to examine the rationality of our proposed approach. In the ablation experiments, all other settings are the same except for the configuration of the framework. The experimental results are summarized in Table V.

We compare the MAE with different configurations over the three datasets to justify the design choice of the proposed counter. In the first trial, we evaluated three different network architectures as cell counters trained without adaptive ground truth density maps: a bare encoder-decoder, an encoder-decoder with an attention block, and a multi-scale hierarchical attention-based counter. It can be seen from Table V that the naive encoder-decoder performs extremely poorly, for example, MAE of 23.1 on the ADI dataset. With the attention block, this count error is reduced to 19.0. The most gratifying change is brought by the multi-scale attention-based strategy, and the MAE on the ADI dataset dropped sharply to 12.3. Similar trends of MAE are also observed on the other two datasets, from 18.0 to 14.3 on the MBM and from 4.9 to 2.9 on the VGG dataset. These results confirm the effectiveness of the proposed counting network.

We next demonstrate the benefits of the utilization of adaptive ground truth. As shown in Table V, in the second trial, we trained the above three different counters under the supervision of adaptive density maps. The counting accuracy of all the counters is greatly improved on the three datasets. The most significant improvement is observed on the MBM dataset by the multi-scale attention-based counter in terms of MAE (from 14.3 to 4.0). Notably, the simple encoder-decoder trained with the adaptive density maps can achieve lower counting errors than its multi-scale attention-based counterpart trained with handcrafted ground truth on the VGG (2.6 vs. 2.9) and MBM (11.4 vs. 14.3) datasets. The considerable advancement in the performance of counters proves the effectiveness of the adaptive ground truth.

It is also worth mentioning that the GT generator is not engaged in the inference phase, which means that it does not bring the extra computational cost to the actual counting. What's more, it has the potential to help lighten the counter, for example, tailoring an appropriate GT generator to ease the learning complexity of the counting network.

VI. Discussion

The prediction results of the proposed framework and those of the detection algorithm (YOLOv4-tiny) are visualized in Fig. 6 to visually demonstrate the superiority of our method. The predictive density maps estimated by the proposed dual-network system are very close to the ground truth maps. The fundamental reason behind this is that the GT generator can adjust the ground truth according to the specific properties of the input microscopic images, making it much easier for the counter to regress accurate predictions. In contrast, the YOLOv4-tiny tends to underestimate the number of cells in images. It can be seen from the estimated results that the YOLOv4-tiny is unable to recognize cell instances without conspicuous texture information. Fundamentally, this is because this deep learning-based detector is much more data-reliable, and there are not enough cell samples to improve the detection accuracy. The least accurate case is observed on the ADI images, where the cells are highly morphology-variable and cluster irregularly. Thus, it is particularly hard for a detection-based counter to learn effective feature representations for the subcutaneous adipose cells with a small number of samples.

The correlation plots are also employed to measure the disparity between the proposed method and a perfect counter. We quantify the effectiveness of the counter with the coefficient of determination (R^2), which is expressed as:

$$R^2 = 1 - \frac{\sum_{i=1}^{N'} (\widehat{c_i}' - c_i')^2}{\sum_{i=1}^{N'} (c_i' - \overline{c'})^2}, \tag{12}$$

where $\overline{c'}$ is the mean of the real cell count.

TABLE V

THE EXPERIMENTAL RESULTS OF ABLATION STUDY

Configurations				**VGG**	**MBM**	**ADI**
Encoder-decoder	Attention	Multi-scale	Adaptive ground truth	$(MAE \downarrow)$	$(MAE \downarrow)$	$(MAE \downarrow)$
✓	-	-	-	4.7± 0.4	19.3± 7.8	23.1±3.1
✓	✓	-	-	4.9 ±0.8	18.0± 3.0	19.0 ± 2.0
✓	✓	✓	-	2.9±0.3	14.3±5.7	12.3± 0.8
✓	-	-	✓	2.6±0.2	11.4±1.4	16.4±1.0
✓	✓	-	✓	2.4±0.3	11.3±1.3	15.2±1.7
✓	✓	✓	✓	**1.9±0.1**	**4.0±0.8**	**11.1±0.4**

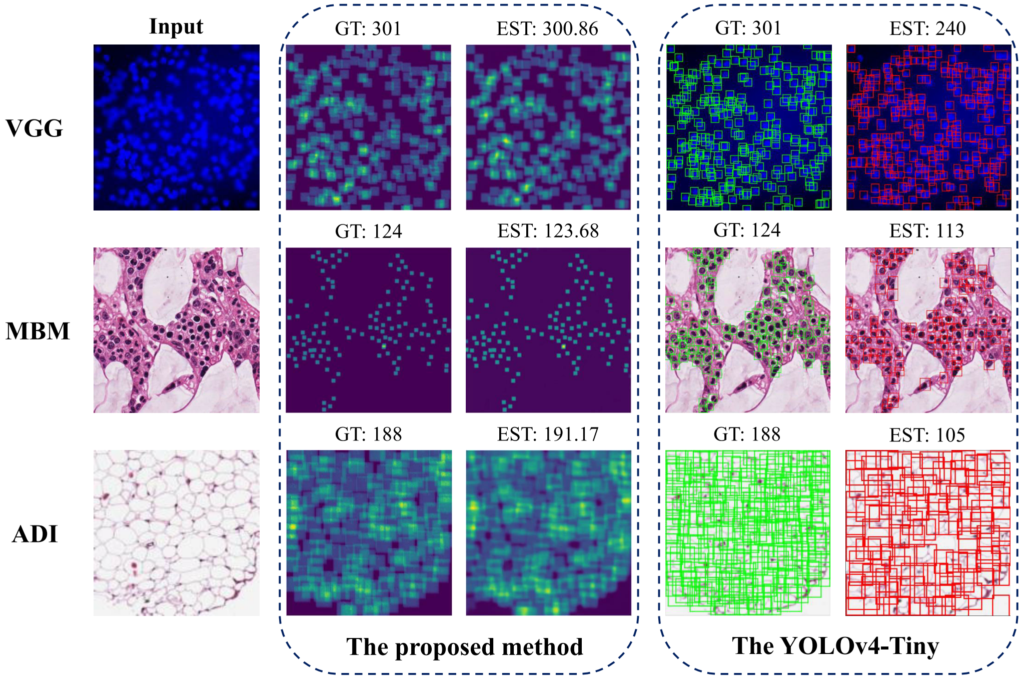


Fig. 6. Comparing the prediction results of our method with those of the detection algorithm. EST and GT refer to the estimated number of cells and the real count, respectively. The second column shows the ground truth density maps, followed by the corresponding estimated density maps regressed by the proposed counter. The ground truths and predictions of the YOLOv4-tiny [54] are in the fourth and fifth columns, respectively.

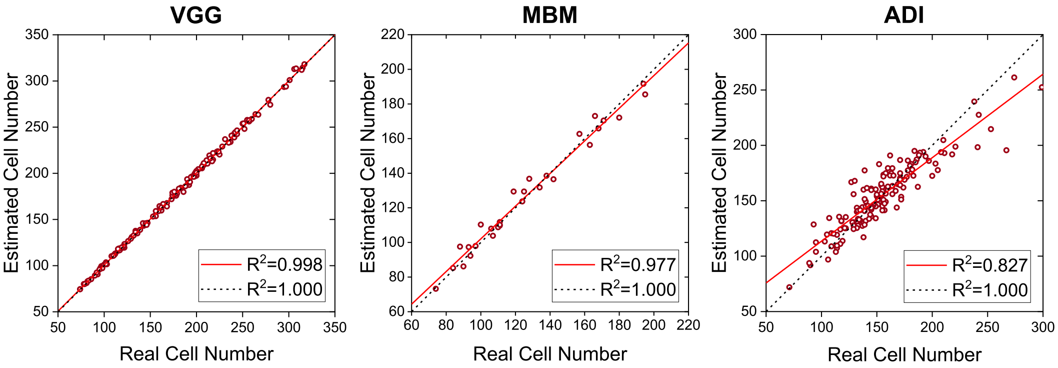


Fig. 7. The R^2-plots of the proposed framework on the test images of the three datasets. Each circle denotes the number of cells in one microscope image, with the x-value being the real count and the y-value being the count estimated by our method. The dotted line depicts the perfect counter, while the proposed counter is represented by the solid line.

Fig. 7 displays the R^2-plots computed using the test images from each dataset in one experiment. The dashed line (R^2=1) represents an ideal counter, which is a diagonal line with a slope of one, indicating that the estimates are perfectly accurate and equal to the real count. The presented approach achieves an R^2 larger than 0.82 on all the three datasets, demonstrating its practicability and robustness. Explicitly, our framework is nearly foolproof for the VGG images with an

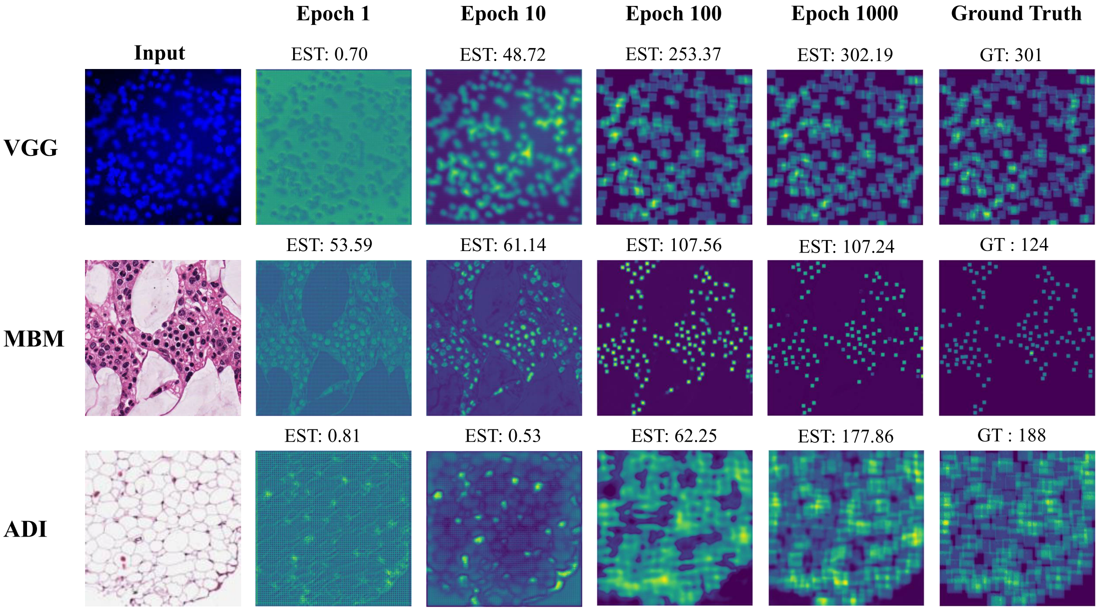


Fig. 8. Visualization of the predicted density maps at $Epoch1$, $Epoch10$, $Epoch100$, and $Epoch1000$ of the training process. The brightness simply represents the relative value of the pixel in individual feature maps for clear visualization.

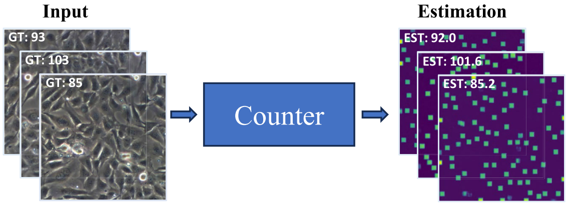


Fig. 9. Application of the proposed method in the quantification of cell growth.

R^2 of 0.998 and also an excellent counter for the MBM data ($R^2 = 0.977$).

We next investigate the reasons for the struggling performance of the algorithm on the ADI dataset. Intuitively, the adipose cells exhibit greater variation in size and shape compared to the synthetic bacteria and bone marrow cells, making it more challenging for the networks to learn the representative features of the samples, as shown in Table II. Moreover, the tight packing of cell instances in adipose tissue and the lack of clear boundaries between individual cells pose additional challenges for the recognition and regression. To gain insights into the underlying mechanism, we visualize the estimated heat maps at different epochs during the training process.

As shown in Fig. 8, at the beginning ($Epoch1$), the counter just stochastically copies the low-level textures related to the input, which have nothing to do with counting. However, at $Epoch10$, different phenomena are observed on different datasets. For the adipose tissue, the algorithm only learns edge and distractors information, which might even have a negative impact on the final estimation. In contrast, the system captures the features of bacteria and bone marrow cell instances. With further training, the estimator has learned the semantic information of samples from both the VGG and MBM datasets at $Epoch100$ and regressed the preliminary density maps. At this stage, the network has just got a rough picture of the spatial distribution of cells in the ADI image, which is still far from the desired instance-level heat map.

Subsequently, at $Epoch1000$ epoch, the proposed learning system can output a fairly accurate density map when inputting a VGG or MBM image. The high-level semantic features of adipose tissue are captured by the system at this stage. The training process also verifies that the proposed framework can effectively extract the semantic information of the various cells, which is central to regressing an instance-level density map for counting. Therefore, our method is optimal for zenithal isotropic cell counting under microscopy and also a decent alternative for polymorphic scenarios.

Cell count is an indicator for the assessment of cell growth. We also evaluate the effectiveness of our method in this practical application by automating the cell counting process based on our previous study [58]. In the experiments, we collected 60 images of cells cultured on Petri dishes with a size of 384 × 384 pixels, as shown in Fig. 9. After training the proposed network with 30 samples, we tested the trained counter with the remaining images and output the cell counts. Our method achieves an MAE of 1.8 on the test set, further demonstrating its generalizability.

The computational efficiency of a deep learning model is an essential consideration in practical applications. Therefore, we measure the average inference time of our method on the test set to evaluate its real-time performance. Specifically, inference times for ADI and VGG datasets are 10.5 ms per image and 11.3 ms per image, respectively. In contrast, the processing time significantly increases to 52.6 ms for each MBM sample due to the increased image size. In summary, our approach demonstrates the capability to perform real-time processing on images of common resolutions, such as ADI and VGG datasets.

VII. Conclusion

In this paper, we propose an interactive dual-network framework for automatic cell counting in microscopic analysis. The GT generator in our framework is able to dynamically adapt the intermediate ground truth to match the properties of the cell samples through joint training with the counter. Thus, the counting network can regress high-qualify density maps under the supervision of the optimized ground truth maps. In addition, a multi-scale attention-based network is developed to act as the counter. By leveraging the attention block and the hierarchical multi-scale encoder-decoder architecture, the counter can effectively and efficiently extract and incorporate multi-level features to further improve the accuracy of the predicted density maps. The presented framework is evaluated on three public cell counting datasets. The new method achieves the best performance on two benchmark datasets (i.e., VGG and MBM) and performs competitively with the state-of-the-art on ADI dataset, demonstrating its effectiveness, universality, and practicability.

Compared to the 2D scenario, 3D cell counting techniques remain largely underexplored. In the future, the proposed framework has the potential to be extended to operate in 3D space and handle the challenges associated with volumetric data. In addition, our work provides inspiration for ground truth refinement in other microscopic analysis tasks such as cell detection.

References

[1] J. Ge et al., "A system for automated counting of fetal and maternal red blood cells in clinical KB test," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2014, pp. 1706–1711.

[2] O. Barinova, V. Lempitsky, and P. Kholi, "On detection of multiple object instances using Hough transforms," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 9, pp. 1773–1784, Sep. 2012.

[3] V. Lempitsky and A. Zisserman, "Learning to count objects in images," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 23, 2010, pp. 1324–1332.

[4] Z. Zheng, H. Yan, F. C. Setzer, K. J. Shi, M. Mupparapu, and J. Li, "Anatomically constrained deep learning for automating dental CBCT segmentation and lesion detection," *IEEE Trans. Autom. Sci. Eng.*, vol. 18, no. 2, pp. 603–614, Apr. 2021.

[5] H. Wu, B. Zhang, J. Pan, and J. Qin, "Multi-level object-aware guidance network for biomedical image segmentation," *IEEE Trans. Autom. Sci. Eng.*, early access, Mar. 31, 2023, doi: 10.1109/TASE.2023.3261344.

[6] A. B. Chowdhury, J. Roberson, A. Hukkoo, S. Bodapati, and D. J. Cappelleri, "Automated complete blood cell count and malaria pathogen detection using convolution neural network," *IEEE Robot. Autom. Lett.*, vol. 5, no. 2, pp. 1047–1054, Apr. 2020.

[7] J. Rodriguez-Vazquez, A. Alvarez-Fernandez, M. Molina, and P. Campoy, "Zenithal isotropic object counting by localization using adversarial training," *Neural Netw.*, vol. 145, pp. 155–163, Jan. 2022.

[8] L. Ciampi et al., "Learning to count biological structures with raters' uncertainty," *Med. Image Anal.*, vol. 80, Aug. 2022, Art. no. 102500.

[9] S. He, K. T. Minn, L. Solnica-Krezel, M. A. Anastasio, and H. Li, "Deeply-supervised density regression for automatic cell counting in microscopy images," *Med. Image Anal.*, vol. 68, Feb. 2021, Art. no. 101892.

[10] Y. Guo, O. Krupa, J. Stein, G. Wu, and A. Krishnamurthy, "SAU-Net: A unified network for cell counting in 2D and 3D microscopy images," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 19, no. 4, pp. 1920–1932, Jul. 2022.

[11] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 779–788.

[12] X. Mai, H. Zhang, and M. Q.-H. Meng, "Faster R-CNN with classifier fusion for small fruit detection," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2018, pp. 7166–7172.

[13] J. P. Cohen, G. Boucher, C. A. Glastonbury, H. Z. Lo, and Y. Bengio, "Count-ception: Counting by fully convolutional redundant counting," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops (ICCVW)*, Oct. 2017, pp. 18–26.

[14] F. Xing, H. Su, J. Neltner, and L. Yang, "Automatic Ki-67 counting using robust cell detection and online dictionary learning," *IEEE Trans. Biomed. Eng.*, vol. 61, no. 3, pp. 859–870, Mar. 2014.

[15] C. Arteta, V. Lempitsky, J. A. Noble, and A. Zisserman, "Detecting overlapping instances in microscopy images using extremal region trees," *Med. Image Anal.*, vol. 27, pp. 3–16, Jan. 2016.

[16] Y. M. Kassim et al., "Clustering-based dual deep learning architecture for detecting red blood cells in malaria diagnostic smears," *IEEE J. Biomed. Health Informat.*, vol. 25, no. 5, pp. 1735–1746, May 2021.

[17] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.* Munich, Germany: Springer, Oct. 2015, pp. 234–241.

[18] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017.

[19] M. M. Alam and M. T. Islam, "Machine learning approach of automatic identification and counting of blood cells," *Healthcare Technol. Lett.*, vol. 6, no. 4, pp. 103–108, Aug. 2019.

[20] C. Arteta, V. Lempitsky, J. A. Noble, and A. Zisserman, "Interactive object counting," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*. Cham, Switzerland: Springer, 2014, pp. 504–518.

[21] L. Fiaschi, U. Koethe, R. Nair, and F. A. Hamprecht, "Learning to count with regression forest and structured labels," in *Proc. 21st Int. Conf. Pattern Recognit. (ICPR)*, Nov. 2012, pp. 2685–2688.

[22] N. Jiang and F. Yu, "A cell counting framework based on random forest and density map," *Appl. Sci.*, vol. 10, no. 23, p. 8346, Nov. 2020.

[23] Y. Xue, N. Ray, J. Hugh, and G. Bigras, "Cell counting by regression using convolutional neural network," in *Proc. Eur. Conf. Comput. Vis.* Amsterdam, The Netherlands: Springer, Oct. 2016, pp. 274–290.

[24] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, May 2017.

[25] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[26] W. Xie, J. A. Noble, and A. Zisserman, "Microscopy cell counting and detection with fully convolutional regression networks," *Comput. Methods Biomech. Biomed. Eng., Imag. Visualizat.*, vol. 6, no. 3, pp. 283–292, May 2018.

[27] J. Wan, Q. Wang, and A. B. Chan, "Kernel-based density map generation for dense object counting," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 3, pp. 1357–1370, Mar. 2022.

[28] A. Howard et al., "Searching for MobileNetV3," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 1314–1324.

[29] X. Yang, Y. Qian, H. Zhu, C. Wang, and M. Yang, "BAANet: Learning bi-directional adaptive attention gates for multispectral pedestrian detection," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, May 2022, pp. 2920–2926.

[30] Q. H. Cap, H. Uga, S. Kagiwada, and H. Iyatomi, "LeafGAN: An effective data augmentation method for practical plant disease diagnosis," *IEEE Trans. Autom. Sci. Eng.*, vol. 19, no. 2, pp. 1258–1267, Apr. 2022.

[31] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7132–7141.

[32] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "CBAM: Convolutional block attention module," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*. Cham, Switzerland: Springer, 2018, pp. 3–19.

[33] Z. Liu et al., "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 9992–10002.

[34] K. Lu, C. Fu, Y. Wang, H. Zuo, G. Zheng, and J. Pan, "Cascaded denoising transformer for UAV nighttime tracking," *IEEE Robot. Autom. Lett.*, vol. 8, no. 6, pp. 3142–3149, Jun. 2023.

[35] Y. Sun, W. Zuo, P. Yun, H. Wang, and M. Liu, "FuseSeg: Semantic segmentation of urban scenes based on RGB and thermal data fusion," *IEEE Trans. Autom. Sci. Eng.*, vol. 18, no. 3, pp. 1000–1011, Jul. 2021.

[36] Y. Sun, W. Zuo, and M. Liu, "RTFNet: RGB-thermal fusion network for semantic segmentation of urban scenes," *IEEE Robot. Autom. Lett.*, vol. 4, no. 3, pp. 2576–2583, Jul. 2019.

[37] F. Yu and V. Koltun, “Multi-scale context aggregation by dilated convolutions,” in *Proc. Int. Conf. Learn. Represent.*, 2016, pp. 1–13.

[38] D. P. Kingma and M. Welling, “Auto-encoding variational Bayes,” 2013, *arXiv:1312.6114*.

[39] A. Lehmussola, P. Ruusuvuori, J. Selinummi, H. Huttunen, and O. Yli-Harja, “Computational framework for simulating fluorescence microscope images with cell populations,” *IEEE Trans. Med. Imag.*, vol. 26, no. 7, pp. 1010–1016, Jul. 2007.

[40] P. Kainz, M. Urschler, S. Schulter, P. Wohlhart, and V. Lepetit, “You should use regression to detect cells,” in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.* Munich, Germany: Springer, Oct. 2015, pp. 276–283.

[41] J. Lonsdale et al., “The genotype-tissue expression (GTEx) project,” *Nature Genet.*, vol. 45, no. 6, pp. 580–585, 2013.

[42] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” in *Proc. Int. Conf. Learn. Represent.*, 2014, pp. 1–15.

[43] I. Loshchilov and F. Hutter, “SGDR: Stochastic gradient descent with warm restarts,” in *Proc. Int. Conf. Learn. Represent.*, 2017, pp. 1–16.

[44] E. Lu, W. Xie, and A. Zisserman, “Class-agnostic counting,” in *Proc. Asian Conf. Comput. Vis.* Perth, WA, Australia: Springer, Dec. 2019, pp. 669–684.

[45] M. Marsden, K. McGuinness, S. Little, C. E. Keogh, and N. E. O’Connor, “People, penguins and Petri dishes: Adapting object counting models to new visual domains and object types without forgetting,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8070–8079.

[46] M. Galarraga et al., “Adiposoft: Automated software for the analysis of white adipose tissue cellularity in histological sections,” *J. Lipid Res.*, vol. 53, no. 12, pp. 2791–2796, 2012.

[47] 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*, vol. 7, pp. 81945–81955, 2019.

[48] N. Jiang and F. Yu, “Multi-column network for cell counting,” *OSA Continuum*, vol. 3, no. 7, pp. 1834–1846, 2020.

[49] Y. Li, X. Zhang, and D. Chen, “CSRNet: Dilated convolutional neural networks for understanding the highly congested scenes,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 1091–1100.

[50] Q. Wang and T. P. Breckon, “Crowd counting via segmentation guided attention networks and curriculum loss,” *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 9, pp. 15233–15243, Sep. 2022.

[51] Z.-Q. Cheng, Q. Dai, H. Li, J. Song, X. Wu, and A. G. Hauptmann, “Rethinking spatial invariance of convolutional networks for object counting,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 19606–19616.

[52] N. Jiang and F. Yu, “A two-path network for cell counting,” *IEEE Access*, vol. 9, pp. 70806–70815, 2021.

[53] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, “YOLOv4: Optimal speed and accuracy of object detection,” 2020, *arXiv:2004.10934*.

[54] C.-Y. Wang, A. Bochkovskiy, and H. M. Liao, “Scaled-YOLOv4: Scaling cross stage partial network,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 13024–13033.

[55] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, “Focal loss for dense object detection,” in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2999–3007.

[56] C. Yang, Z. Huang, and N. Wang, “QueryDet: Cascaded sparse query for accelerating high-resolution small object detection,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 13658–13667.

[57] S. Xu et al., “PP-YOLOE: An evolved version of YOLO,” 2022, *arXiv:2203.16250*.

[58] C. Wu et al., “Rapid nanomolding of nanotopography on flexible substrates to control muscle cell growth with enhanced maturation,” *Microsyst. Nanoeng.*, vol. 7, no. 1, p. 89, Nov. 2021.

Rui Liu received the B.S. degree from Central South University, Changsha, China, in 2013, and the M.S. degree from Zhejiang University, Hangzhou, China, in 2016. He is currently pursuing the Ph.D. degree with the City University of Hong Kong, Hong Kong, SAR, China.

His research interests include automation at micro/nano scale, microscopic image analysis, and deep learning.

Yudi Zhu received the B.S. degree from Southeast University, Nanjing, China, in 2018, and the M.S. degree from the City University of Hong Kong, Hong Kong, in 2023.

Her research interests are computer vision and deep learning.

Cong Wu received the B.Eng. degree from Beihang University, Beijing, China, in 2012, and the M.S. and Ph.D. degrees from the City University of Hong Kong, Hong Kong, SAR, China, in 2013 and 2020, respectively.

Her main research interests include advanced micro/nano manufacturing and its applications in biomimetic cell manipulation.

Hao Guo is currently pursuing the M.S. degree with the University of Science and Technology of China, Hefei, China.

His research interests are software system development and machine learning.

Wei Dai (Graduate Student Member, IEEE) received the B.Eng. degree from the South China University of Technology, Guangzhou, China, in 2021. He is currently pursuing the Ph.D. degree with the City University of Hong Kong, Hong Kong, SAR, China.

He was a Visiting Student at the University of St Andrews, St Andrews, Scotland, in 2019. Currently, he is dedicated to self-supervised learning and domain adaptation. His research interests include medical image analysis and label-efficient learning.

Tianyi Wu received the B.S. and M.S. degrees from the Beijing Institute of Technology, Beijing, China, in 2017 and 2020, respectively. He is currently pursuing the Ph.D. degree with the City University of Hong Kong, Hong Kong, SAR, China.

His current research interests include deep learning, machine vision, and assembly automation.

Min Wang (Graduate Student Member, IEEE) received the B.S. degree from the Hefei University of Technology (HFUT), Hefei, China, in 2017, and the M.S. degree from the Harbin Institute of Technology Shenzhen (HITSZ), Shenzhen, China, in 2020. He is currently pursuing the Ph.D. degree with the City University of Hong Kong, Hong Kong, SAR, China.

His main research interests include actuation and perception for miniature robots.

Wen Jung Li (Fellow, IEEE) received the B.S. and M.S. degrees from the University of Southern California, Los Angeles, CA, USA, in 1987 and 1989, respectively, and the Ph.D. degree from UCLA, Los Angeles, in 1997.

He is currently a Chair Professor with the Department of Mechanical Engineering, the City University of Hong Kong (CityU), Hong Kong. He is also the Director of the CAS–CityU Joint Laboratory for Robotics, City University of Hong Kong Shenzhen Research Institute, Shenzhen, China. Prior to joining CityU, he was with the Department of Mechanical and Automation Engineering, The Chinese University of Hong Kong (CUHK), Hong Kong, from 1997 to 2011. Before joining CUHK, he held Research and Development positions with the NASA/ Caltech Jet Propulsion Laboratory, Pasadena, CA, USA; The Aerospace Corporation, El Segundo, CA, USA; and Silicon Microstructures Inc., Fremont, CA, USA. His current research interests include BioMEMS, AI image processing for scanning super-resolution microscopy, and AIsensors for healthcare.

Jun Liu (Member, IEEE) received the Ph.D. degree from the University of Toronto, Toronto, ON, Canada, in 2016.

From 2017 to 2019, he worked as a Post-Doctoral Fellow with Weill Cornell Medical College, Cornell University, Ithaca, NY, USA. He is currently an Assistant Professor with the Department of Mechanical Engineering, City University of Hong Kong, Hong Kong, SAR, China. His current research interests include micro/nanorobotics, medical imaging, and biomedical instrumentation. His research has been recognized in the field of robotics and automation by winning multiple awards, including the Best Student Paper Award and the Best Medical Robotics Paper Finalist Award from the IEEE International Conference on Robotics and Automation in 2014 and the IEEE TRANSACTIONS ON AUTOMATION SCIENCE AND ENGINEERING Best New Application Paper Award in 2018.