IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 28 March 2024, accepted 28 April 2024, date of publication 6 May 2024, date of current version 22 May 2024.

*Digital Object Identifier 10.1109/ACCESS.2024.3397686*

RESEARCH ARTICLE

# PGC-Net: A Novel Encoder-Decoder Network With Path Gradient Flow Control for Cell Counting

**JIANKE LI[1,2], JIANGUO WU[3], JING QI[1,2], (Member, IEEE), MINGJUN ZHANG[4], AND ZHENCHAO CUI[1,2]**

[1]School of Cyber Security and Computer, Hebei University, Baoding, Hebei 071002, China
[2]Machine Vision Engineering Research Center, Hebei University, Baoding 071002, China
[3]Department of Laboratory Medicine, Zhejiang Provincial People's Hospital, Hangzhou 310014, China
[4]Department of Laboratory Medicine, People's Hospital of Jiulongpo District, Chongqing 400050, China

Corresponding author: Zhenchao Cui (cuizhenchao@gmail.com)

**ABSTRACT** To carry out cell counting, it is common to use neural network models with an encoder-decoder structure to generate regression density maps. In the encoder-decoder structure, skip connections are usually used to retain detailed features. However, skip connections introduce to the encoder multiple reverse propagation paths; the backward propagation gradients along these paths exhibit significant semantic differences, which affect the encoder's training process and may lead to adverse effects. To remedy this problem, we propose a path-gradient controlling network for cell counting. First, a novel reverse gradient control module is proposed to balance the impact on the encoder of the backward propagation signal from the skip connections. Second, to eliminate noise in the feature maps of the encoder output, the convolutional and channel attention modules are used on the shallowest layer's skip connection. Finally, we utilise depthwise convolution to reduce information loss during the downsampling process, and we use depthwise separable transposed convolution as the upsampling method to mitigate overfitting. Experiments demonstrate that the proposed method outperforms state-of-the-art techniques such as MSCA-UNet, Two-Path Net, SAU-Net, and Cell-Net in terms of the mean absolute error (MAE) metric on four publicly available cell-counting benchmark datasets. Our model performs better on the synthetic bacterial (VGG) dataset (1.9 $\pm$0.1) than does the MSCA-UNet (2.0$\pm$0.2). On the Modified Bone Marrow (MBM) dataset, our model (3.7$\pm$0.2) outperforms SAU-Net (5.7$\pm$1.2). On the human subcutaneous adipose tissue (ADI) dataset, our model, with (8.9$\pm$0.3), surpasses MSCA-UNet with (9.8$\pm$0.7). On the Dublin Cell Counting (DCC) dataset, our model achieves (2.4$\pm$0.2) and outperforms SAU-Net with (3.0$\pm$0.3). The source code of our method is available at https://github.com/mona-aliye/PGC-Net.

**INDEX TERMS** Cell counting, density map, gradient control.

## I. INTRODUCTION

Object counting is an essential task in computer vision, and it includes many sub-tasks, such as crowd counting [1], [2], [3], cell counting [4], [5], [6], and vehicle counting [7], [8], [9]. One of these sub-tasks, cell counting, has attracted the attention of many researchers, as it can aid doctors in diagnosing diseases such as immune granulocytes [10] and COVID-19 [11]. Generally, cell counting methods for microscopic images are divided into detection-based and density map regression-based methods. Traditional approaches in cell counting [12], [13], [14] often employ detection-based methods, wherein considerable features are manually identified to extract individual cells from the microscope images. Due to the limitations of their feature detectors, traditional approaches

The associate editor coordinating the review of this manuscript and approving it for publication was Zhen Ren.

still encounter difficulty in processing images with high cell density. To contend with this problem, density map regression-based methods based on deep learning have been proposed.

Several regression-based methods have been proposed with non-encoder-decoder structures, and other methods with encoder-decoder structures have been proposed. For the methods with non-encoder-decoder structure, Xie et al. [15] proposed a microscopy cell counting and detection method based on a fully convolutional regression network, which marked the first application of fully convolutional networks (FCN) [16] in cell counting and achieved good results. Jiang and Yu [17] built upon the FCN architecture to introduce a foreground mask network for filtering low-level feature maps and conveying valuable information to the decoder. However, in non-encoder-decoder structures, there is a significant loss of spatial detail in high-dimensional semantic feature maps due to the increase in the number of convolutional and downsampling layers. Recent cell counting methods have generally adopted encoder-decoder structures with skip connections to solve this problem. These structures fuse the feature maps outputted from each layer of the encoder with the input feature map of the decoder, which enables the decoder to simultaneously leverage high-dimensional semantic features and spatial details to reconstruct the density map. Jiang and Yu [18] introduced a cell counting network named the Two-Path Network, which featured two encoding pathways: one dedicated to global information extraction and the other focused on local information extraction. Rad et al. [19] proposed a method based on an ensemble of residual dilated U-Nets [20]. Their approach involved adding a residual dilated module at the bottom level of the U-Net, which means stacking convolutional layers having different dilation rates to obtain a larger receptive field, thereby capturing the global features in cell images. Subsequently, they further improved upon this approach by proposing CellNet [5], where the residual dilated module is refined into a residual incremental atrous pyramid structure. Additionally, they introduced a progressive upsampling convolutional decoder based on subpixel convolution. Guo et al. [21] added a spatial self-attention module to the bottom encoding and decoding paths of the U-Net, which enabled the learning of spatial features within a global context. Additionally, they introduced online batch normalization (BN) to alleviate the generalisation gap caused by data augmentation in small cell-counting datasets. Jiang and Yu [22] introduced a weighted channel module called the inter-channel attention (ICA) block to capture the inter-channel correlations in feature maps. This channel attention module is designed with non-parametric features, which makes it suitable for cell datasets. Although the proposed methods have achieved excellent results, they did not fully address the following problems: First, because skip connections construct paths at various depths of the network, there is a noticeable semantic gap in the backward propagation signals received by the encoder modules along different paths. Second, shallow encoder feature maps in the cascaded encoder contain noise, and the forwarding of maps directly to the decoder via skip connections can impact the decoder's output. Lastly, the cell counting task relies heavily on detailed information, and the traditional downsampling methods that lead to the loss of detailed features may bring about errors. A limited number of samples in a cell counting task makes traditional upsampling methods with many trainable parameters prone to under-fitting.

To contend with the above problems in cell counting tasks, we propose a novel method: a path-gradient controlling network (PGC-Net) based on an encoder-decoder structure. To address poor feature map quality in the encoder, we introduce the reverse gradient control (RGC) module to regulate the backward propagation gradients from paths of different depths to guide the encoder to focus more on the gradient signals from specific paths and to alleviate the semantic gap that occurs when backward gradient signals are propagated along different paths. Second, to address the problem of noise in the feature maps of the shallow encoder outputs, we design a lightweight denoising module called the Lite Detail Refiner, which effectively removes foreground and background noise. Finally, we use depthwise convolution to reduce information loss during the downsampling process, and we use depthwise separable transposed convolution as the upsampling method to mitigate overfitting.

In summary, our main contributions can be summarised as follows:

1) To contend with the semantic gap caused by the propagation of backward gradient signals along different paths, we propose a novel network architecture, PGC-Net, that introduces backward gradient control modules into the encoder-decoder structure of neural networks.
2) To address the effect of noise on the decoder's learning of spatial details, we propose a lightweight denoising module named Lite Detail Refiner. It utilises a convolutional block and a channel attention module on the skip connection branch to remove noise from the feature maps of the shallowest encoder output.
3) To remedy the sampling methods for cell counting tasks, we propose using depthwise convolution and transposed depthwise separable convolution [23] for downsampling and upsampling.

The remaining sections of the paper are organised as follows: Section II introduces the proposed method. Section III provides a brief description of the public benchmark datasets and experimental configurations used in this study, comparisons with state-of-the-art methods on each benchmark dataset, and an analysis of the results from ablation experiments. Finally, Section IV concludes the paper with a discussion and a future outlook.

## II. METHOD

In this section, we introduce the proposed network. We first present the overall architecture, and we then explain the submodules in sequence.

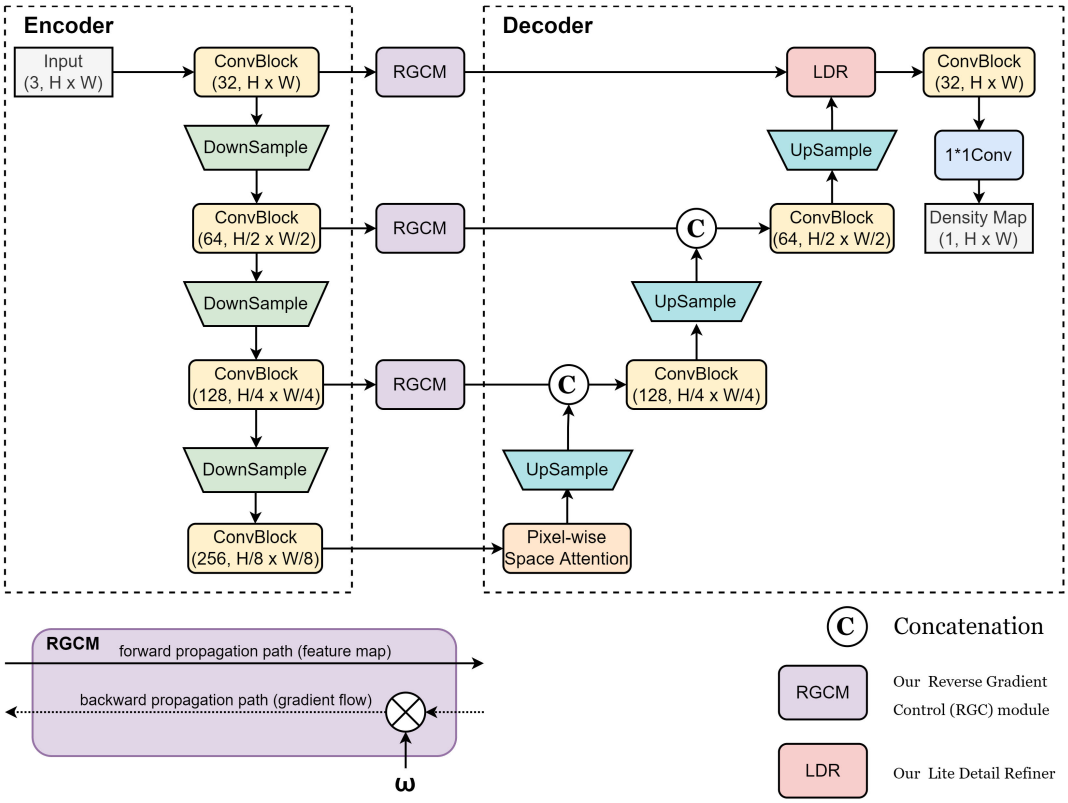



**FIGURE 1.** Overall structure of PGC-Net. PGC-Net is essentially SAU-Net with additional gradient control modules, noise reduction modules, and modifications to the upsampling and downsampling methods. In the gradient control module, we can give a weight to control the gradient flow along this backpropagation path.

### A. ARCHITECTURE

The proposed cell counting network, referred to as PGC-Net, is illustrated in Figure 1.

As shown in Figure 1, there are four layers in the network. Except for the fourth layer of the decoder, which utilises the spatial self-attention module from SAU-Net [21], each layer of the decoder and encoder contains a convolutional block. Each convolutional block comprises two identical modules, each of which is composed of a $3\times3$ convolutional layer, a BN layer, and a ReLU layer. The feature maps produced by each encoder layer are downsampled and then fed into the next encoder layer. The downsampling layer employs depthwise convolution with a stride of 2, and the upsampling layer utilises depthwise separable transposed convolution. Specifically, it involves a transposed depthwise convolution with a stride of 2, followed by a point-wise convolution, i.e., a $1\times1$ convolutional layer, for the reduction of channel dimensions. The RGC modules are placed at the output ends of the encoder skip connection to control the backward propagation gradient flow along each skip connection. Furthermore, our Lite Detail Refiner (LDR) is added to the first layer's skip connection branch. The LDR comprises a $3\times3$ convolutional layer, a BN layer (BN), a ReLU layer, and a channel-wise attention module without parameters, i.e., an ICA module [22]. The convolutional layers use a stride of 1 if not specified otherwise. Specifically, we use regular BN [24] layers instead of the online BN layer proposed in the SAU-Net paper.

### B. RGC MODULE

The skip connections in the encoder-decoder structure provide forward pathways for the decoder to access spatial details, which effectively enhances the decoder's ability to recognise edges. In regressing to generate density maps, spatial information primarily guides the decoder in utilising semantic information. However, the skip connections simultaneously provide a backward gradient propagation path for the encoder and influence the encoder during training. Specifically, the gradient of the first layer encoder is a summation of gradients from four paths. The gradient from the main path tends to give the shallowest layer of the encoder a more fine-grained spatial feature extraction capability. Conversely, the backward propagation gradient from the shallowest layer tends to enable it to learn coarser features, i.e. more abstract semantic features, as shown in Equation (1). We omit the noise removal module and the upsampling/downsampling layers from the formulas for simplicity. We label the convolutional blocks in the encoder, from shallow to deep, as $E_1$ to $E_4$, and in the decoder, from deep to shallow, as $D_3$ to $D_1$. $D_4$ represents the spatial self-attention module that connects $E_4$ and $D_3$, and the finally connected output $1\times1$ convolutional layer is considered to

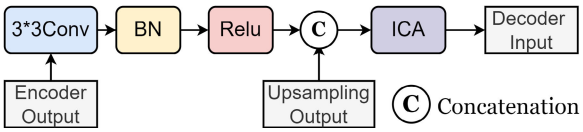



**FIGURE 2.** Structure of LDR. Within the denoising module, initially, a convolutional layer is utilised to establish local element correlations. Following the nonlinear ReLU and BN layers, an ICA module is added to construct global channel correlations.

be part of $D_1$. Let X be the input and let the model output be $\overline{Y}$=f(X). The loss function is L=Y-$\overline{Y}$, where Y is the true label. G(j) represents the effect of the j-th layer's path on the gradient at $E_1$. When the subscript of the product symbol is greater than the superscript, we define it as an empty product, and its value is 1. The gradient at $E_1$ is given by Equation (1):

$$\begin{cases} \dfrac{\partial L}{\partial E_1} = \sum_{j=1}^{l} G(j) \\ G(j) = \dfrac{\partial L}{\partial D_1} \cdot \dfrac{\partial D_j}{\partial E_j} \prod_{k=1}^{j-1} \left( \dfrac{\partial D_k}{\partial D_{k+1}} \cdot \dfrac{\partial E_{k+1}}{\partial E_k} \right) \end{cases} \tag{1}$$

where $l$ represents the number of layers in the symmetric encoder-decoder structure model, and in this case, $l = 4$. We can observe that the backpropagation gradient from the shallow skip connection and the backpropagation gradient from the main pathway are superimposed on $E_1$. We add an RGC module for each layer's skip connection. The weights for the RGC modules from the first to the third layer are denoted as $\omega_j$, where $1 \leq j \leq 3$. In particular, as the main path located at the fourth layer is the deepest path, we assume its backward-propagating gradient to impact $E_1$ positively. Therefore, we do not add an RGC module to the main path. For simplicity, we set $\omega_4 = 1$. The gradient at $E_1$ is then given by the following Equation (2):

$$\begin{cases} \dfrac{\partial L}{\partial E_1} = \sum_{j=1}^{l} \omega_j G(j) \\ G(j) = \dfrac{\partial L}{\partial D_1} \cdot \dfrac{\partial D_j}{\partial E_j} \prod_{k=1}^{j-1} \left( \dfrac{\partial D_k}{\partial D_{k+1}} \cdot \dfrac{\partial E_{k+1}}{\partial E_k} \right) \end{cases} \tag{2}$$

The problems present at $E_1$ also exist at $E_2$ and $E_3$, as their gradients result from the summation of backward propagation gradients from multiple paths. Therefore, the analyses and formulas presented for $E_1$ are equally applicable to $E_2$ and $E_3$, and the details will not be reiterated here.

### C. LDR

The skip connection provides a forward propagation path by which the decoder can obtain detailed spatial features from the shallowest encoder layer. However, for the convolutional block to capture fine-grained spatial details, it often outputs feature maps that contain harmful noise. This noise is not conducive to guiding the decoder to use high-level semantic information for density map estimation. To address this issue, we add a denoising module, namely LDR, to the shallowest branch of the skip connection, as shown in Figure 2. The denoising module first employs a convolutional layer to establish local element correlations. Following the BN and nonlinear layers (ReLU), an ICA module is added to build global channel correlations. In Figure 3, we present a comparative illustration of the effect of the LDR. After the denoising, foreground and background noise are effectively removed, and the feature map is left with primarily valid spatial information.

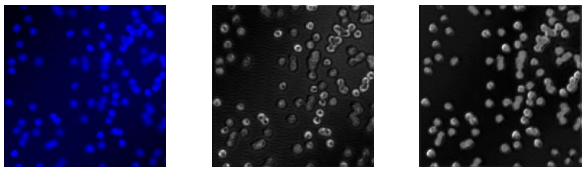



**FIGURE 3.** (a) Original input image, (b) feature map input to the decoder without denoising, and (c) feature map input to the decoder after denoising.

### D. SAMPLING METHOD

Downsampling and upsampling are widely used in encoder-decoder structures. Downsampling reduces the size of the feature maps while removing high-frequency spatial details. Upsampling aims to restore high-dimensional semantic information to the original image space. The commonly used downsampling methods are max pooling and average pooling. The advantage of these methods lies in their simplicity. However, both pooling methods are algorithmically designed to take in prior knowledge, and they apply the same computational pattern to feature maps that contain different features, which leads to the loss of valuable information. For this reason, we propose using depthwise convolution for downsampling operations. With almost no increase in the number of parameters or computational complexity, this approach adaptively generates a computational pattern for each channel of each feature map. It helps reduce information loss during the downsampling process. During the upsampling process, transpose convolution is commonly used, which involves elevating a pixel to map to nearby pixels in both the spatial and channel dimensions. However, due to the small size of cell datasets, such complex relationships can easily lead to overfitting. Therefore, we adopt a transposed depthwise separable convolution strategy to decouple spatial and channel relationships. In the upsampling module, depthwise transposed convolution only needs to learn the spatial relationships between pixels mapped to the same channel's neighbouring pixels. A $1 \times 1$ convolution is then used to learn channel relationships and perform dimensionality reduction, which reduces the risk of overfitting.

### E. LOSS FUNCTION

We use pixel-wise L2 loss, which is also known as mean squared error (MSE) loss, to train our model, as shown in

Equation (3).

$$\text{MSE} = \frac{1}{mn}\sum_{i,j}(\hat{P}_{i,j} - P_{i,j})^2 \tag{3}$$

where the subscripts $i, j$ represent the pixel positions, and $1 \leq i \leq m$ and $1 \leq j \leq n$, $m$ and $n$ denote the maximum pixel values for the height and width of the image, respectively. $\hat{P}_{i,j}$ represents the value predicted by the neural network at position $i, j$, and $P_{i,j}$ represents the ground truth value at position $i, j$.

It is worth noting that, in contrast to the MESA loss [25], our model aims to learn the probability density distribution at a single pixel by regressing from neighbouring pixels, rather than by directly obtaining global metrics. Therefore, we have chosen the per-pixel loss calculation approach.

## III. EXPERIMENTS

In this section, we first introduce the dataset, the details of the experimental setup, and the evaluation metrics. We then compare the proposed cell counting network with other state-of-the-art methods. Finally, we represent the ablation experimental results, and we compare them with the baseline model to validate the effectiveness of the proposed approach. Additionally, we discuss cell detection results in the context of our proposed approach.

We used Python 3.9 as the programming language, PyTorch 1.12 as the deep learning framework, and NVIDIA V100 GPU with CUDA 11.3 for training and validation. The experiments were conducted on Ubuntu 20.04.

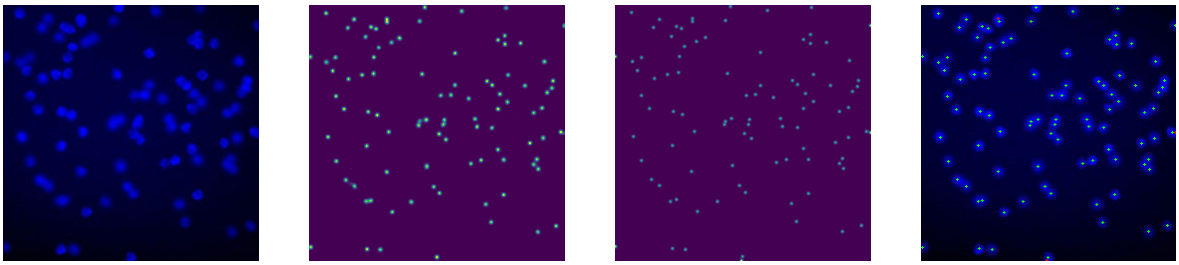
### A. DATASETS

We conduct experiments on four datasets: the synthetic bacterial (VGG) dataset [25], Modified Bone Marrow(MBM) dataset [26], human subcutaneous adipose tissue(ADI) dataset [27] and Dublin Cell Counting(DCC) dataset [4]. Additionally, to validate the practicality of the proposed backward gradient control module in models with skip connections, we test the effectiveness of the backward gradient control module on the classic ResNet20 [28] model using the CIFAR-10 [29] dataset. **VGG**: Lempitsky and Zisserman [25] created the VGG dataset. This dataset simulated bacterial cells observed under a fluorescence optical microscope at different focal lengths. **MBM**: Cohen et al. [26], based on the dataset initially released by Kainz et al. [30], created the MBM dataset. This dataset contains real images of various cell types in the human bone marrow stained in blue. **ADI**: The ADI dataset [27] was constructed by the Genotype-Tissue Expression Consortium. **DCC**: The DCC dataset was created by Marsden et al. [4] to represent various cells, including embryonic mouse stem cells, human lung adenocarcinoma cells, human mononuclear cells, and others. The image sizes range from 306×322 to 798×788 to enhance the variability of the dataset. **CIFAR-10**: CIFAR-10 [29] is a classic dataset that is widely used for computer vision tasks. It consists of coloured images from ten different classes. Each class contains 6000 RGB images with a resolution of 32×32 pixels for a total of 60,000 images.

### B. DETAILS OF EXPERIMENTAL CONFIGURATION

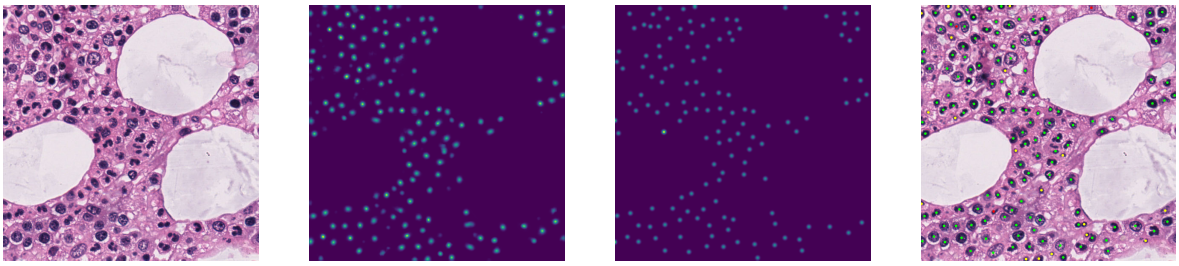
In this section, we primarily focus on the experimental configuration and the implementation details of the training process. For training on the VGG dataset, a batch size of 16 is employed with a total of 200 samples. The training set consists of 64 samples, and the remaining samples are allocated to a validation set. On the MBM dataset, a batch size of 5 is used, and the dataset comprises 44 samples. Of these, 15 samples are used for training and the rest are allocated to a validation set. On the ADI dataset, a batch size of 16 is utilized, and the dataset contains a total of 150 samples: 50 samples are used for training, and the remainder are reserved for validation. As for the DCC dataset, the batch size is set to 16, and the dataset comprises 176 samples. In this case, 100 samples are utilized for training and the rest are allocated to a validation set. Additionally, on the VGG [25], MBM [26], ADI [27], and DCC [4] datasets, the label scaling factors are set to 100,1000,100, and 300, respectively. These scaling factors divide the predicted density maps for the computation of the mean absolute error (MAE) metric. This adjustment ensures that the initial loss values are within a reasonable range. We use the Adam optimiser with decoupled weight decay [31], and the weight decay is set to 0.001. The network weights are initialised using He initialisation [32]. For the learning rate, we employ cosine annealing with warm restarts [33], which gradually decreases the learning rate according to a cosine decay function. The initial value of the schedule is set to 0.01 with a restart step of 50 and a multiplier of 2. We run each experiment for 350 iterations. **Preprocessing**: Previous related works [18], [21], and [34] have stated that it is necessary to preprocess images to adjust their dimensions to multiples of 8 to accommodate downsampling in the encoder. We adopt the same preprocessing method for better comparison with the baseline SAU-Net [21] model. Specifically, we pad the edges of the ADI dataset images to resize them from 150 × 150 to 152 × 152, and we downsample both the images and labels in the DCC dataset to 256 × 256. **Data Augmentation**: During training, images from the VGG [25], MBM [26], ADI [27] and DCC [4] datasets are randomly cropped to sizes of 224×224,512×512,128×128, and 224×224, respectively. Random horizontal flips, vertical flips, and 90-degree rotations are implemented.

Note that we attempted to align our choice of the sample size, preprocessing methods and hyperparameters as closely as possible with the approach outlined in the SAU-Net [21] work. Therefore, differences in the samples used for training, preprocessing methods and hyperparameters with some techniques may introduce bias.



(a) Input image (b) Predicted density map (c) Ground truth density map (d) Cell detection

**FIGURE 4.** Sample predicted density map on the test set from the VGG dataset. Ground truth cell count: 96; predicted: 97.6.



(a) Input image (b) Predicted density map (c) Ground truth density map (d) Cell detection

**FIGURE 5.** Sample predicted density map on the test set from the MBM dataset. Ground truth cell count: 138; predicted: 141.7.

**TABLE 1.** Comparison results on the VGG dataset.

| Method | MAE | Ntrain[a] |
|---|---|---|
| Xue et al. [35] | $7.5 \pm 2.2$ | 100 |
| Lu et al. [36] | $3.6 \pm 0.3$ | 32 |
| Lempitsky et al. [25] | $3.5 \pm 0.2$ | 32 |
| Fiaschi et al. [37] | $3.2 \pm 0.1$ | 32 |
| FCRN-A [15] | $2.9 \pm 0.2$ | 64 |
| SAU-Net [21] | $2.6 \pm 0.4$ | 64 |
| Count-ception [26] | $2.3 \pm 0.4$ | 50 |
| Cell-Net [5] | $2.2 \pm 0.5$ | 50 |
| Two-Path Net [18] | $2.2 \pm 0.2$ | 50 |
| MSCA-UNet [34] | $2.0 \pm 0.2$ | 32 |
| PCG-Net(proposed) | $\mathbf{1.9 \pm 0.1}$ | 64 |

[a] Ntrain refers to the number of samples used as training samples

**TABLE 2.** Comparison results on the MBM dataset.

| Method | MAE | Ntrain |
|---|---|---|
| FCRN-A [15] | $21.3 \pm 9.4$ | 15 |
| Marsden et al. [4] | $20.5 \pm 3.5$ | 15 |
| Cell-Net [5] | $9.8 \pm 3.2$ | 10 |
| Count-ception [26] | $8.8 \pm 2.3$ | 15 |
| Two-Path Net [18] | $6.0 \pm 0.6$ | 15 |
| SAU-Net [21] | $5.7 \pm 1.2$ | 15 |
| MSCA-UNet [34] | $5.8 \pm 0.7$ | 15 |
| PCG-Net(proposed) | $\mathbf{3.7 \pm 0.2}$ | 15 |

**TABLE 3.** Comparison results on the ADI dataset.

| Method | MAE | Ntrain |
|---|---|---|
| Count-ception [26] | $19.4 \pm 2.2$ | 50 |
| SAU-Net [21] | $14.2 \pm 1.6$ | 50 |
| Two-Path Net [18] | $10.6 \pm 0.3$ | 50 |
| MSCA-UNet [34] | $9.8 \pm 0.7$ | 50 |
| PCG-Net(proposed) | $\mathbf{8.9 \pm 0.3}$ | 50 |

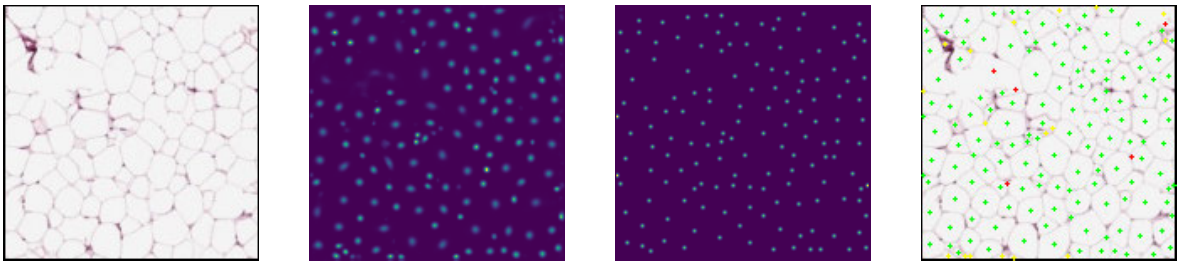
**TABLE 4.** Comparison results on the DCC dataset.

| Method | MAE | Ntrain |
|---|---|---|
| Marsden et al. [4] | 8.4 | 100 |
| SAU-Net [21] | $3.0 \pm 0.3$ | 100 |
| PCG-Net(proposed) | $\mathbf{2.4 \pm 0.2}$ | 100 |

### C. COMPARISONS WITH STATE-OF-THE-ART MODELS

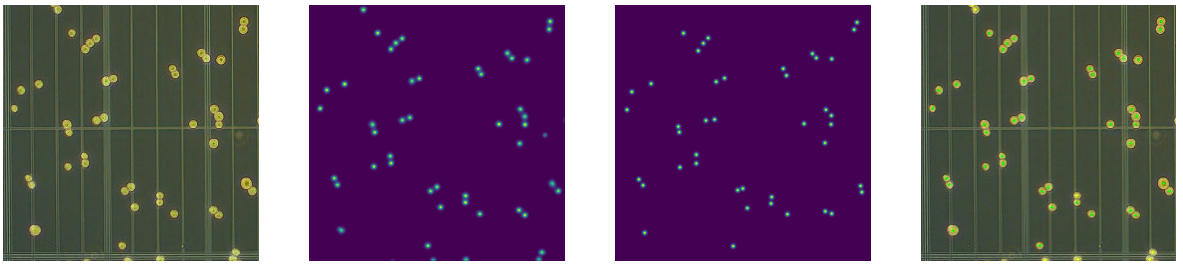
We use the MAE as the evaluation metric, which calculates the absolute difference between the ground truth and the predicted counts for each image. In each experiment, the training and validation splits for each dataset are randomly selected. The experiments are repeated ten times, and the mean and standard deviation of the MAE are computed. We compare our method with state-of-the-art approaches.

On the VGG [25] dataset, our proposed method achieves the best experimental results. The experimental outcomes and sample predictions are presented in Table 1 and Figure 4. On the MBM dataset [26], our method demonstrates good performance on real cell images, as depicted in Table 2 and Figure 5. On the ADI dataset [27], our proposed approach exhibits the best experimental outcomes, and the results and sample predictions are shown in Table 3 and Figure 6. On the DCC dataset [4], which consists of various real cell images, our method obtains the best experimental results, as presented in Table 4 and Figure 7. In summary, our proposed approach demonstrates superior performance in achieving a low MAE and small standard deviation across all datasets relative to the other cell counting methods. This result reflects the generalisation and robustness of our method on cell counting benchmark datasets. To validate the utility of the proposed RGC module on other models with skip



(a) Input image (b) Predicted density map (c) Ground truth density map (d) Cell detection

**FIGURE 6.** Sample predicted density map on the test set from the ADI dataset. Ground truth cell count: 145; predicted: 140.8.



(a) Input image (b) Predicted density map (c) Ground truth density map (d) Cell detection

**FIGURE 7.** Sample predicted density map on the test set from the DCC dataset. Ground truth cell count: 42; predicted: 43.4.

**TABLE 5.** Ablation study on the impact of the proposed method.

| No Gradient Control [1,1,1] | Gradient Control Weight [0.8,1,1] | Gradient Control weight [0.4,0.8,0.4] | Gradient Control weight [0,0.8,0] | Lite Detail Refiner | Modified Sampling Method | MAE |
|---|---|---|---|---|---|---|
| YES | NO | NO | NO | NO | NO | $2.6 \pm 0.4$ |
| YES | NO | NO | NO | NO | YES | $2.4 \pm 0.3$ |
| YES | NO | NO | NO | YES | NO | $2.2 \pm 0.2$ |
| YES | NO | NO | NO | YES | YES | $2.1 \pm 0.1$ |
| NO | YES | NO | NO | YES | YES | $2.0 \pm 0.1$ |
| NO | NO | YES | NO | YES | YES | $\mathbf{1.9 \pm 0.1}$ |
| NO | NO | NO | YES | YES | YES | $2.0 \pm 0.1$ |

connections, we select the classical ResNet20 [28] model for testing. We add the RGC module to each residual block's residual connection and set the reverse gradient weight to 0. On the CIFAR-10 [29] dataset, the performance improvement is 0.57% relative to the original study [28] (from 91.25% to 91.82%). This result demonstrates the generality of the proposed method.

### D. ABLATION STUDY

In this section, we conduct ablation experiments on the VGG dataset [25] to validate the effectiveness of the RGC module, the denoising module, and the improved sampling method. The results are reported in Table 5 and indicate that incorporating each of the RGC modules, denoising module and improved sampling method one at a time leads to a further improvement in model performance compared with when these components are not used.

It is worth noting that the choice of weights for the RGC module at each layer is a consideration worth exploring. To choose the appropriate weights, we design a search experiment to find the optimal weights with predefined values in the range of [0,0.4,0.8,1]. These values represent a low-weight reverse gradient flow, a high-weight reverse gradient flow, a full reverse gradient flow, and no reverse gradient flow. The detailed results of the search experiment can be found in Appendix A.

We select three sets of weights that achieved excellent results, and we conduct ten repeated experiments for validation. Ultimately, the best weights are determined to be [0.4,0.8,0.4].

### E. CELL DETECTION

We follow the metrics and methods described in the MSCA-UNet [34] study and use the Xie's approach [15] to obtain cell detection results by locating the local maxima. The predicted maps of cell detection include three types of instances: TP (True Positive), FP (False Positive) and FN (False Negative), represented by green, yellow, and red points, respectively (Figures 4, Figure 5, Figure 6, Figure 7). The precision, recall and F1 score are used to evaluate the cell detection performance, as shown below:

$$Precision = \frac{TP}{TP + FP} \tag{4}$$

**TABLE 6.** Results of cell detection on datasets.

| Dataset | Batch size | Precision(%) | Recall(%) | F1-Score |
|---|---|---|---|---|
| VGG | 16 | $98.79 \pm 0.34$ | $95.00 \pm 0.21$ | $0.9684 \pm 0.0012$ |
| MBM | 5 | $86.33 \pm 0.22$ | $96.49 \pm 0.13$ | $0.9106 \pm 0.0014$ |
| ADI | 16 | $84.26 \pm 1.13$ | $96.67 \pm 0.29$ | $0.9004 \pm 0.0057$ |
| DCC | 16 | $83.62 \pm 1.39$ | $97.04 \pm 0.21$ | $0.8946 \pm 0.0081$ |

$$Recall = \frac{TP}{TP + FN} \tag{5}$$

$$F1\text{-}Score = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{6}$$

Precision represents the proportion of true positive instances among all instances predicted as positive. Recall represents the proportion of true positive instances predicted among all instances labelled as positive. The F1 score comprehensively measures both. The cell detection results on the four datasets are shown in Table 6. TP represents true positive predicted instances, where a predicted instance is considered to be a match to a labelled positive instance if it falls within a threshold radius and is the closest predicted instance to the labelled positive instance. FN represents labelled positive instances that are not predicted, meaning there are no predicted instances within the threshold radius of those labelled positive instances. FP represents false positive instances, i.e., predicted positive instances other than TP. The threshold radius we use is 10 pixels, which is the same as in the MSCA-UNet study.

Compared with MSCA-UNet, our model has higher recall but lower precision. These results mean that our model has fewer FN instances but more FP instances. This difference is mainly due to the treatment of local detailed features. Whereas MSCA-UNet uses large convolutional kernels to capture features of largescale structures for density map regression, our model learns predominantly from local detailed features. Regression from more local detailed features to the cell density map introduces more false predictions and predictions that match real cells, which results in higher recall and lower precision.

## IV. CONCLUSION

In this study, our main contribution is to re-examine the pros and cons of skip connections in encoder-decoder structures from a novel perspective. We introduce a plug-and-play RGC module to address the semantic gap introduced by backward gradient signals from different paths. In addition, we improve our model to better suit the characteristics of cell counting tasks, including the denoising of shallow feature maps and the selection of sampling methods.

However, compared with the model employing the RGC modules, the original ResNet20 model demonstrated a faster loss reduction rate when it was trained on the CIFAR dataset. Interestingly, the model that implemented the RGC modules with a weight set to 0 performed better during the final convergence. Intuitively, this can be explained by the residual connection path providing a stronger adjustment signal to the shallow layers of the model. As a result, the loss decreases rapidly during the early stages of training. Fine-tuning the shallow layers with deep signals in the later stages of training yields better results. Therefore, the choice of control path and the specific weight settings, including the temporal (within training epochs) and numerical settings, are all factors worth considering, depending on the different goals and application scenarios.

**TABLE 7.** Experimental results with different configurations.

| Gradient Control Weight of layer1 | Gradient Control Weight of layer2 | Gradient Control Weight of layer3 | Lite Detail Refiner | Modified Sampling Method | MAE |
|---|---|---|---|---|---|
| 1 | 1 | 1 | YES | YES | 2.12 |
| 0.8 | 1 | 1 | YES | YES | 1.77 |
| 0.4 | 1 | 1 | YES | YES | 1.92 |
| 0 | 1 | 1 | YES | YES | 1.89 |
| 1 | 0.8 | 1 | YES | YES | 2.15 |
| 0.8 | 0.8 | 1 | YES | YES | 2.2 |
| 0.4 | 0.8 | 1 | YES | YES | 2.08 |
| 0 | 0.8 | 1 | YES | YES | 1.88 |
| 1 | 0.4 | 1 | YES | YES | 2.2 |
| 0.8 | 0.4 | 1 | YES | YES | 1.92 |
| 0.4 | 0.4 | 1 | YES | YES | 2.11 |
| 0 | 0.4 | 1 | YES | YES | 1.87 |
| 1 | 0 | 1 | YES | YES | 3.41 |
| 0.8 | 0 | 1 | YES | YES | 3.47 |
| 0.4 | 0 | 1 | YES | YES | 3.12 |
| 0 | 0 | 1 | YES | YES | 2.49 |
| 1 | 1 | 0.8 | YES | YES | 1.99 |
| 0.8 | 1 | 0.8 | YES | YES | 2.06 |
| 0.4 | 1 | 0.8 | YES | YES | 2.27 |
| 0 | 1 | 0.8 | YES | YES | 1.86 |
| 1 | 0.8 | 0.8 | YES | YES | 1.89 |
| 0.8 | 0.8 | 0.8 | YES | YES | 1.92 |
| 0.4 | 0.8 | 0.8 | YES | YES | 1.89 |
| 0 | 0.8 | 0.8 | YES | YES | 1.89 |
| 1 | 0.4 | 0.8 | YES | YES | 2.41 |
| 0.8 | 0.4 | 0.8 | YES | YES | 2.05 |
| 0.4 | 0.4 | 0.8 | YES | YES | 1.94 |
| 0 | 0.4 | 0.8 | YES | YES | 1.88 |
| 1 | 0 | 0.8 | YES | YES | 3.06 |
| 0.8 | 0 | 0.8 | YES | YES | 3.69 |
| 0.4 | 0 | 0.8 | YES | YES | 4.62 |
| 0 | 0 | 0.8 | YES | YES | 2.48 |
| 1 | 1 | 0.4 | YES | YES | 2.15 |
| 0.8 | 1 | 0.4 | YES | YES | 1.92 |
| 0.4 | 1 | 0.4 | YES | YES | 1.93 |
| 0 | 1 | 0.4 | YES | YES | 1.91 |
| 1 | 0.8 | 0.4 | YES | YES | 2.04 |
| 0.8 | 0.8 | 0.4 | YES | YES | 2.05 |
| 0.4 | 0.8 | 0.4 | YES | YES | 1.76 |
| 0 | 0.8 | 0.4 | YES | YES | 1.85 |
| 1 | 0.4 | 0.4 | YES | YES | 2.02 |
| 0.8 | 0.4 | 0.4 | YES | YES | 2.14 |
| 0.4 | 0.4 | 0.4 | YES | YES | 2.12 |
| 0 | 0.4 | 0.4 | YES | YES | 2.07 |
| 1 | 0 | 0.4 | YES | YES | 3.69 |
| 0.8 | 0 | 0.4 | YES | YES | 3.9 |
| 0.4 | 0 | 0.4 | YES | YES | 3.77 |
| 0 | 0 | 0.4 | YES | YES | 3.29 |
| 1 | 1 | 0 | YES | YES | 2.09 |
| 0.8 | 1 | 0 | YES | YES | 1.9 |
| 0.4 | 1 | 0 | YES | YES | 1.97 |
| 0 | 1 | 0 | YES | YES | 1.96 |
| 1 | 0.8 | 0 | YES | YES | 2.09 |
| 0.8 | 0.8 | 0 | YES | YES | 1.88 |
| 0.4 | 0.8 | 0 | YES | YES | 1.89 |
| 0 | 0.8 | 0 | YES | YES | 1.78 |
| 1 | 0.4 | 0 | YES | YES | 3.2 |
| 0.8 | 0.4 | 0 | YES | YES | 2.19 |
| 0.4 | 0.4 | 0 | YES | YES | 1.99 |
| 0 | 0.4 | 0 | YES | YES | 1.92 |
| 1 | 0 | 0 | YES | YES | 5.48 |
| 0.8 | 0 | 0 | YES | YES | 2.94 |
| 0.4 | 0 | 0 | YES | YES | 3.85 |
| 0 | 0 | 0 | YES | YES | 4.24 |


**FIGURE 8.** Ten training and validation loss curves for the datasets.

In future work, we will consider treating the number of training steps as an independent variable and designing a weight adjustment function to introduce it into the RGC module. Our aim is to achieve both faster convergence and improved performance simultaneously.

## APPENDIX A
## PARAMETER SEARCH EXPERIMENTS OF GRADIENT CONTROL WEIGHT

See Table 7.

## APPENDIX B
## TRAIN AND VALIDATE LOSS

See Figure 8.

## REFERENCES

[1] J. Liu, C. Gao, D. Meng, and A. G. Hauptmann, "DecideNet: Counting varying density crowds through attention guided detection and density estimation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 5197–5206.

[2] Z. Zou, Y. Cheng, X. Qu, S. Ji, X. Guo, and P. Zhou, "Attend to count: Crowd counting with adaptive capacity multi-scale CNNs," *Neurocomputing*, vol. 367, pp. 75–83, Nov. 2019.

[3] Z. Zou, Y. Liu, S. Xu, W. Wei, S. Wen, and P. Zhou, "Crowd counting via hierarchical scale recalibration network," 2020, *arXiv:2003.03545*.

[4] 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.

[5] R. Moradi 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.

[6] C. Arteta, V. Lempitsky, J. A. Noble, and A. Zisserman, "Interactive object counting," in *Proc. 13th Eur. Conf. Comput. Vis. (ECCV)*, Zurich, Switzerland. Cham, Switzerland: Springer, Sep. 2014, pp. 504–518.

[7] 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.

[8] S. Zhang, G. Wu, J. P. Costeira, and J. M. Moura, "FCN-rLSTM: Deep spatio-temporal neural networks for vehicle counting in city cameras," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2017, pp. 3667–3676.

[9] S. Aich and I. Stavness, "Improving object counting with heatmap regulation," 2018, *arXiv:1803.05494*.

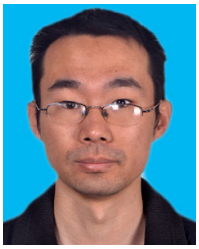
[10] J. D. Farkas, "The complete blood count to diagnose septic shock," *J. Thoracic Disease*, vol. 12, no. S1, pp. S16–S21, Feb. 2020.
[11] R. Khasla, "The role of biomarkers in diagnosis of COVID-19: A systematic review," *Population Med.*, vol. 5, Apr. 2023, Art. no. 117788.
[12] M. Maitra, R. Kumar Gupta, and M. Mukherjee, "Detection and counting of red blood cells in blood cell images using Hough transform," *Int. J. Comput. Appl.*, vol. 53, no. 16, pp. 13–17, Sep. 2012.
[13] C. Jung and C. Kim, "Segmenting clustered nuclei using H-minima transform-based marker extraction and contour parameterization," *IEEE Trans. Biomed. Eng.*, vol. 57, no. 10, pp. 2600–2604, Oct. 2010.
[14] J. M. Sharif, M. Miswan, M. Ngadi, M. S. H. Salam, and M. M. B. A. Jamil, "Red blood cell segmentation using masking and watershed algorithm: A preliminary study," in *Proc. Int. Conf. Biomed. Eng. (ICoBE)*, Feb. 2012, pp. 258–262.
[15] W. Xie, J. A. Noble, and A. Zisserman, "Microscopy cell counting and detection with fully convolutional regression networks," *Comput. Methods Biomechanics Biomed. Eng., Imag. Visualizat.*, vol. 6, no. 3, pp. 283–292, May 2018.
[16] E. Shelhamer, J. Long, and T. Darrell, "Fully convolutional networks for semantic segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 4, pp. 640–651, Apr. 2017.
[17] N. Jiang and F. Yu, "A foreground mask network for cell counting," in *Proc. IEEE 5th Int. Conf. Image, Vis. Comput. (ICIVC)*, Beijing, China, Jul. 2020, pp. 128–132.
[18] N. Jiang and F. Yu, "A two-path network for cell counting," *IEEE Access*, vol. 9, pp. 70806–70815, 2021.
[19] R. M. Rad, P. Saeedi, J. Au, and J. Havelock, "Blastomere cell counting and centroid localization in microscopic images of human embryo," in *Proc. IEEE 20th Int. Workshop Multimedia Signal Process. (MMSP)*, Vancouver, BC, Canada, Aug. 2018, pp. 1–6.
[20] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer Assisted Intervention—MICCAI*. Cham, Switzerland: Springer, Oct. 2015, pp. 234–241.
[21] Y. Guo, J. Stein, G. Wu, and A. Krishnamurthy, "SAU-Net: A universal deep network for cell counting," in *Proc. 10th ACM Int. Conf. Bioinf. Comput. Biol. Health Informat.*, Niagara Falls, NY, USA, Sep. 2019, pp. 299–306.
[22] N. Jiang and F. Yu, "Cell counting with channels attention," in *Proc. IEEE 5th Int. Conf. Signal Image Process. (ICSIP)*, Nanjing, China, Oct. 2020, pp. 494–498.
[23] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "MobileNets: Efficient convolutional neural networks for mobile vision applications," 2017, *arXiv:1704.04861*.
[24] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. Int. Conf. Mach. Learn.*, Lille, France, Jul. 2015, pp. 448–456.
[25] V. Lempitsky and A. Zisserman, "Learning to count objects in images," in *Proc. Annu. Conf. Neural Inf. Process. Syst. (NIPS)*, Vancouver, BC, Canada, 2010, pp. 1324–1332.
[26] J. Paul 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)*, Venice, Italy, Oct. 2017, pp. 18–26.
[27] J. Lonsdale et al., "The genotype-tissue expression (GTEx) project," *Nature Genet.*, vol. 45, no. 6, pp. 580–585, May 2013.
[28] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 770–778.
[29] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," Dept. Comput. Sci., Univ. Toronto, Tech. Rep. 001, Apr. 2009.
[30] P. Kainz, M. Urschler, S. Schulter, P. Wohlhart, and V. Lepetit, "You should use regression to detect cells," in *Medical Image Computing and Computer-Assisted Intervention—MICCAI*. Cham, Switzerland: Springer, Oct. 2015, pp. 276–283.
[31] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," 2017, *arXiv:1711.05101*.
[32] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 1026–1034.
[33] I. Loshchilov and F. Hutter, "SGDR: Stochastic gradient descent with warm restarts," 2016, *arXiv:1608.03983*.
[34] L. Qian, W. Qian, D. Tian, Y. Zhu, H. Zhao, and Y. Yao, "MSCA-UNet: Multi-scale convolutional attention UNet for automatic cell counting using density regression," *IEEE Access*, vol. 11, pp. 85990–86001, 2023.
[35] Y. Xue, N. Ray, J. Hugh, and G. Bigras, "Cell counting by regression using convolutional neural network," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Amsterdam, The Netherlands. Cham, Switzerland: Springer, Oct. 2016, pp. 274–290.
[36] E. Lu, W. Xie, and A. Zisserman, "Class-agnostic counting," in *Proc. 14th Asian Conf. Comput. Vis.*, Perth, WA, Australia. Cham, Switzerland: Springer, Dec. 2018, pp. 669–684.
[37] 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.

**JIANKE LI** is currently pursuing the master's degree with Hebei University, China. His research interests include cell counting and image processing.

**JIANGUO WU** received the master's degree, in 2008, and the Ph.D. degree from Zhejiang University, in 2014. He is currently the Chief Technician with the Department of Laboratory Medicine, Zhejiang Provincial People's Hospital, and serves as a part-time master's Supervisor in clinical medicine at Zhejiang University. His research interest includes AI diagnosis of leukemia morphology.

**JING QI** (Member, IEEE) received the master's degree from Yanshan University, in 2010, and the Ph.D. degree from Beihang University, in 2021. She is currently a Master's Supervisor in computer science with Hebei University. Her research interests include control, perception, and interaction of robots. She is a member of CCF.

**MINGJUN ZHANG** received the master's degree, in 2004, and the Ph.D. degree from Chongqing Medical University, in 2014. She is currently the Director of the Department of Laboratory Medicine, People's Hospital of Jiulongpo District, Chongqing. Her research interest includes cell morphology for clinical examination.



**ZHENCHAO CUI** is currently a Professor with the School of Cyber Security and Computer, Hebei University. His research interests include deep learning and medical imaging processing.