

PM-CResUnet: A Lightweight Multi-Constraint Framework for Cell Segmentation and Counting

Zhuang Wang, Peiwen Yan*

Qingdao University, Qingdao 266071, China

Email: 1692964741@qq.com

How to cite this paper: Wang, Z., & Yan, P. (2026). PM-CResUnet: A Lightweight Multi-Constraint Framework for Cell Segmentation and Counting. *Journal of Computer Science and Frontier Technologies*, 4(2), 31–42. ISSN Print: 3104-4204, ISSN Online: 3104-4212.

<https://doi.org/10.63313/JCSFT.9101>

Published: 2026-09-28

Copyright © 2026 by author(s) and

Erytis Publishing Limited.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Abstract

Cell counting is a common task in the quantitative analysis of fluorescence microscopy images; however, variations in cell size, density, and boundary conditions often make the separation of adjacent cells difficult, thereby degrading counting accuracy. This paper proposes Pyramid-Multitask c-ResUnet (PM-CResUnet). Building upon the lightweight segmentation model c-ResUnet, we introduce pyramidal convolution and boundary peak weight maps, complemented by distance regression and instance-aware auxiliary supervision during training, and adopt complexity-aware adaptive augmentation (C3A) to expand the training data distribution. The model follows a segmentation-post-processing-counting pipeline, in which the auxiliary branches participate only during training and are excluded from the inference path. We systematically evaluate the contribution of each design choice to segmentation and counting through ablation studies on DSB2018. The results show that PM-CResUnet improves instance separation and counting performance while remaining lightweight, and that multi-scale features, boundary constraints, and data diversity act complementarily, with the full model achieving the best overall performance in both instance segmentation and counting.

Keywords

Cell Segmentation; Cell Counting; C-Resunet; Pyramidal Convolution; Multi-Task Learning; Boundary Weighting; Adaptive Augmentation

1. Introduction

Cell segmentation is a critical step in the quantitative analysis of fluorescence microscopy and pathology images. Accurate cell regions provide the basis for downstream analyses such as cell count, morphology, and spatial distribution. In practice, however, cells often exhibit pronounced scale variation, local aggregation, and blurred boundaries. When multiple cells are in contact, even if the foreground region is correctly identified, binary segmentation may still merge adjacent instances into a single connected component, thereby introducing counting bias. For high-throughput analyses, models must also maintain stable performance under

constrained computational resources and limited annotated data.

U-Net and its residual variants provide a mature encoder-decoder framework for medical image segmentation [1], and c-ResUnet further reduces the model scale through a compact architecture [2]. Compared with large instance segmentation models, lightweight networks are better suited to scenarios with limited computational resources; however, their feature representation and instance separation capability are also more susceptible to cell scale variation and adherent boundaries. To this end, this work addresses four aspects, namely model architecture, loss optimization, training supervision, and data distribution: PyConv provides multi-receptive-field feature representations; boundary peak weighting increases the loss contribution at the interfaces between adjacent cells; a distance head and an instance-aware query head supply geometric and instance priors; and C3A expands the effective coverage of the training data through scene-adaptive augmentation.

The goal of this paper is not to re-propose any of the above individual mechanisms, but to examine how they can be combined within a lightweight cell segmentation network and to analyze their actual impact on instance segmentation and counting. The experiments employ component ablation, comparison with classical networks, and counting evaluation, to investigate the relationships among model performance, module contribution, and data conditions.

2. Related Work

U-Net established the classical encoder-decoder paradigm for medical image segmentation [1], recovering fine-grained spatial information through skip connections. Its residual variants improved the optimization of deep networks, and c-ResUnet was further designed in a compact form for the task of fluorescence cell counting [2]. Such models are structurally simple and computationally inexpensive, but their output remains predominantly pixel-level foreground prediction, with limited explicit modeling of mutually contacting instances.

Multi-scale feature extraction is a common strategy for handling scale variation. Pyramid Scene Parsing Network enhances scene representation through multi-scale context aggregation [3], whereas PyConv constructs different receptive fields by employing convolution kernels of different sizes within the same layer [4]. Drawing on the basic idea of PyConv, we place parallel 3×3 and 5×5 convolution branches in selected encoder and decoder modules of c-ResUnet; the purpose is multi-scale feature enhancement within a lightweight backbone, rather than a redefinition of PyConv itself.

Research on cell instance segmentation has gradually shifted from pure region prediction toward geometric or instance representations. Mask R-CNN performs instance prediction through an independent mask branch [5], StarDist describes cell shapes using star-convex polygons [6], Cellpose emphasizes generalization across

cell types and imaging conditions [7], and HoVer-Net assists the separation of clustered nuclei using horizontal and vertical distance maps [8]. These works indicate that additional geometric or instance supervision can compensate for the information missing from binary masks. In this work, we adopt auxiliary prediction of the distance to cell boundaries together with instance supervision based on learnable queries and Hungarian matching, although the auxiliary branches are used only during training.

Boundary-related losses primarily target the narrow regions between adjacent objects that are easily overlooked by global pixel-wise losses. Boundary Loss demonstrated that boundary information can complement region-based losses [9]. The boundary peak weighting adopted in this work shares this motivation, but instead of directly optimizing a contour distance function, it constructs Gaussian weights from instance boundaries and applies them to the pixel-wise loss. In addition, the data distribution of microscopy images often varies considerably, and the diversity of training data directly affects the generalization of the model to unseen conditions; we therefore further introduce C3A, which selects augmentation strategies according to image complexity.

Taken together, the challenges faced by lightweight cell segmentation models are not confined to a single network architecture: multi-scale representation, difficult boundary optimization, instance supervision, and training data diversity correspond to distinct sources of error. PM-CResUnet incorporates these directions into a single lightweight framework and analyzes their complementarity through ablation experiments, rather than treating any existing mechanism as an independent novel method.

3. Datasets

The main experiments use the 2018 Data Science Bowl nuclei dataset [10], cropped to 256×256 inputs following the current pipeline, comprising 509 training images and 89 validation images; during training, C3A is further applied to expand the effective sample size. This dataset covers diverse cell types and imaging conditions, providing a basis for evaluating lightweight models under variations in cell scale and density.

4. Methods

4.1. Model Architecture

As shown in **Figure 1**, PM-CResUnet adopts a U-Net-style encoder-bridge-decoder structure, with the stages named following the layer numbering of c-ResUnet. The input first passes through a 1×1 convolution (denoted c0) that performs a learned mapping from RGB to a single channel, followed by a first 3×3 convolution (c1) that extracts 32-channel features; the two residual stages of the encoder (enc2 and enc3) progressively raise the feature dimensionality to 64 and

128 channels while reducing the resolution to 1/2 and 1/4 of the input; the bridge consists of two 5×5 residual blocks (bridge4 and bridge5) operating at 1/8 of the input resolution with 256 channels; the three decoder stages (dec6, dec7, and dec8) progressively restore the resolution through transposed convolutions and fuse the spatial information of the corresponding encoder stages via skip connections, with 128, 64, and 32 channels, respectively; and a final 1×1 convolution (outc) outputs the foreground and background logits. PyConv is inserted into enc2, enc3, dec6, dec7, and dec8, distributing output channels between the 3×3 and 5×5 convolution branches to capture both local detail and a larger receptive field.

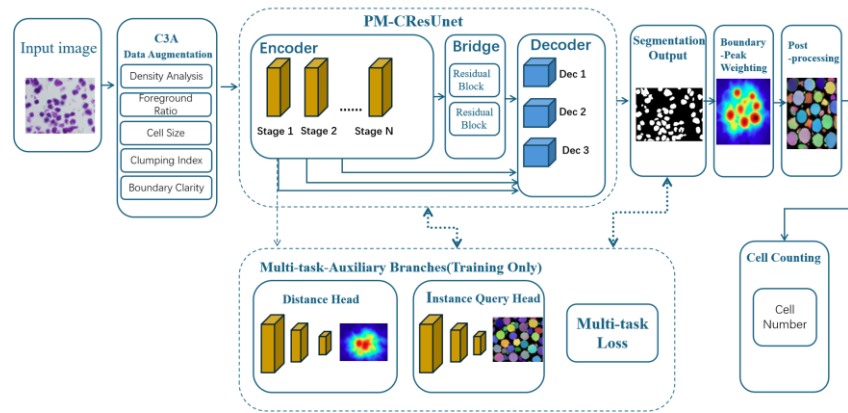


Figure 1. Overall architecture of PM-CResUnet.

4.2. Pyramidal Convolution

To address cell size variation, we adopt a two-branch PyConv [4] in enc2, enc3, dec6, dec7, and dec8. The 3×3 and 5×5 convolutions provide smaller and larger receptive fields, respectively, and the output channels are allocated in inverse proportion to the kernel area. For an output channel number C_{out} , the number of channels in the k -th branch is given by:

$$C_k = \frac{A_{k'}}{A_3 + A_5} \cdot C_{out} \approx \begin{cases} 0.735C_{out}, k = 3 \\ 0.265C_{out}, k = 5 \end{cases}$$

Accordingly, the 3×3 and 5×5 branches account for approximately 73.5% and 26.5% of the output channels, respectively. Since the branch with the larger kernel occupies fewer channels, the total number of parameters and the FLOPs remain essentially comparable to those of a standard convolution, while a dual-scale representation, namely fine local detail from the small kernel and global context from the large kernel, is obtained simultaneously. The original 5×5 residual blocks in the bridge already provide a relatively large receptive field and are therefore not replaced.

4.3. Loss Function and Boundary Peak Weighting

To suppress erroneous connections between contacting cells, we compute, for each instance, the Euclidean distance from every pixel to its cell boundary and accumulate the Gaussian responses of all instances. The weight of foreground (intra-cellular) pixels is set to $w(x) = 1$, while the weight of background pixels is:

$$w(x) = \begin{cases} 1 & x \in \text{foreground} \\ \sum_{k=1}^K \exp\left(-\frac{d_k(x)^2}{2\sigma^2}\right) & \text{otherwise} \end{cases}, \quad \sigma = 25$$

The main segmentation loss consists of a weighted cross-entropy loss and a Dice loss:

$$\mathcal{L}_{\text{seg}} = \mathcal{L}_{\text{CE}}^w + \mathcal{L}_{\text{Dice}}$$

This weight map does not alter the amount of training data; it only increases the loss contribution of difficult boundaries, especially clumping gaps, during optimization.

4.4. Multi-Task Supervision during Training

Conventional segmentation tasks provide only per-pixel category labels, so the model is unaware that cells are closed regions with internal structure and that pixels belong to distinct cells, leaving the judgment of adherent boundaries without a sound basis. To address this difficulty, two lightweight auxiliary heads are connected in parallel to the encoder features:

Distance head (DistanceHead) [11]: fuses features from multiple layers and upsamples them to regress a per-pixel distance map $\hat{d}(x)$ to the nearest cell boundary, supervised by the ground-truth distance map, thereby providing geometric shape supervision;

Instance-aware query head (IDFLHead): $N_q = 80$ learnable query vectors retrieve candidate instances from the features via cross-attention and output instance masks together with confidence scores; during training, the Hungarian algorithm is used to obtain an optimal one-to-one matching between predictions and ground-truth instances [12], and mask and classification losses are applied to the matched pairs.

Auxiliary losses: $\mathcal{L}_{\text{dist}} = \text{MSE}(d(x), \hat{d}(x))$, where $\mathcal{L}_{\text{inst}}$ denotes the matched instance-level loss. The overall training objective is:

$$\mathcal{L} = \mathcal{L}_{\text{seg}} + \lambda_d \mathcal{L}_{\text{dist}} + \lambda_{\text{inst}} \mathcal{L}_{\text{inst}}$$

Side-branch design and inference-time detachment: the outputs of the two auxiliary heads do not enter the main segmentation prediction; they constrain the shared backbone representation only through back-propagation. At inference, only the backbone forward pass is retained, and the auxiliary heads are not involved in computation at all. This design has two properties: (i) zero additional inference cost; and (ii) the auxiliary supervision and architectural modifications to the backbone itself (PyConv) are orthogonally superimposed at the gradient level and do not interfere with each other.

4.5. C3A: Complexity-Aware Adaptive Augmentation

Fixed-pattern augmentation such as flipping and cropping cannot supplement difficult samples on demand when annotations are insufficient. C3A first characterizes each training image using five complexity dimensions, namely density, foreground ratio, mean area, clumping index, and boundary sharpness, and classifies it into low-density, clumping, blurry, or normal scenes; it then applies tailored augmentation according to the scene: elastic deformation for clumping scenes, multi-scale blur and noise for blurry scenes, and Copy-Paste for low-density scenes, in which real cells are transplanted from high-density images to artificially increase density, with the instance masks updated synchronously. The training data are ultimately expanded from 509 to 2781 high-quality augmented samples.

4.6. Model Training and Inference

The model uses the AdamW optimizer with a learning rate of 1×10^{-3} (5×10^{-4} for configurations containing the multi-task auxiliary heads), a cosine annealing schedule, a maximum of 300 epochs, and early stopping with a patience of 15 epochs. All experiments were conducted on a single NVIDIA RTX 3080 Ti (12 GB). At inference, the foreground probability map is first thresholded, and instances are then separated by a distance-guided watershed algorithm. Segmentation is evaluated using Dice, PQ, and AJI, and counting is evaluated using MAE, MRE, and detection F1; the definitions of these metrics and the post-processing parameters are detailed in Section 4.7.

4.7. Evaluation Metrics and Post-Processing Protocol

Segmentation evaluation is primarily based on instance-level metrics. The output of cell segmentation is one instance per cell, and the result is usable only when adjacent cells are correctly separated; pixel-level metrics such as Dice cannot reflect this, because if two adherent cells are predicted as a single connected region, the pixel-level foreground overlap is barely penalized, whereas instance-level evaluation counts this as a missed detection. PQ (Panoptic Quality) [13] matches predicted and ground-truth instances one-to-one at $\text{IoU} > 0.5$ and decomposes the evaluation into the product of recognition quality (RQ, which penalizes missed and false detections) and segmentation quality (SQ, the mean IoU of matched instances); missed detections, false detections, and imprecise masks all reduce it, with each instance weighted equally. AJI (Aggregated Jaccard Index) [14] accumulates intersection and union areas in an aggregated manner, and predictions not matched to any ground-truth instance enter the denominator union directly, which strongly penalizes over-segmentation; it is the official ranking metric adopted by the 2018 Data Science Bowl. The two metrics are complementary: PQ emphasizes strict matching, whereas AJI emphasizes overall

overlap and penalization of over-segmentation; Dice serves as a pixel-level supplementary metric to align with segmentation literature that reports only pixel-level measures.

The biological goal of cell analysis is to determine how many cells are present in a given image (c-ResUnet likewise adopts detection F1 and MAE as its core metrics [2]), and our experiments show that segmentation and counting are partially decoupled, with segmentation gains not propagating 1:1 to counting; counting must therefore be evaluated independently. The counting metrics serve distinct purposes: MAE directly reports the average number of miscounted cells per image; MRE is normalized by the ground-truth cell count so that images of different densities can be averaged fairly; RMSE assigns squared weight to large errors, penalizing catastrophic missed detections; and det-F1 measures whether cells are detected via centroid matching.

The segmentation metrics are defined as follows: let $P = \{p_i\}$ be the set of predicted instances and $G = \{g_j\}$ the set of ground-truth instances, with $\text{IoU}(p, g) = |p \cap g| / |p \cup g|$. Predictions and ground truths are first matched greedily in descending order of IoU, and only matched pairs with $\text{IoU} > 0.5$ are retained as TP; unmatched predictions are recorded as FP, and unmatched ground truths as FN.

$$\begin{aligned} \text{PQ} &= \frac{\sum_{(p,g) \in \text{TP}} \text{IoU}(p, g)}{|\text{TP}| + \frac{1}{2}|\text{FP}| + \frac{1}{2}|\text{FN}|} = \text{RQ} \times \text{SQ} \\ \text{RQ} &= \frac{|\text{TP}|}{|\text{TP}| + \frac{1}{2}|\text{FP}| + \frac{1}{2}|\text{FN}|} \\ \text{SQ} &= \frac{1}{|\text{TP}|} \sum_{(p,g) \in \text{TP}} \text{IoU}(p, g) \end{aligned}$$

where RQ denotes recognition quality and SQ segmentation quality. AJI aggregates over all ground-truth instances, taking for each g_j the prediction $p(g_j)$ with the highest IoU and applying no IoU threshold, with $|p(g_j) \cap g_j| = 0$ for unmatched ground truths:

$$\text{AJI} = \frac{\sum_{g_j \in G} |p(g_j) \cap g_j|}{\sum_{g_j \in G} |p(g_j) \cup g_j| + \sum_{p \in U} |p|}$$

where U is the set of predictions not matched to any ground truth, and this term incorporates the surplus instances produced by over-segmentation into the denominator. The pixel-level Dice is:

$$\text{Dice} = \frac{2|P \cap G|}{|P| + |G|}$$

The counting metrics are defined as follows: for N images in the validation set, let y_k be the ground-truth cell count and $\hat{y}_k$ the predicted count for the k -th image; we report:

$$\begin{aligned} \text{MAE} &= \frac{1}{N} \sum_{k=1}^N |\hat{y}_k - y_k| \\ \text{MRE} &= \frac{1}{N} \sum_{y_k > 0} \frac{|\hat{y}_k - y_k|}{y_k} \\ \text{RMSE} &= \sqrt{\frac{1}{N} \sum_{k=1}^N (\hat{y}_k - y_k)^2} \\ R^2 &= 1 - \frac{\sum_{k=1}^N (\hat{y}_k - y_k)^2}{\sum_{k=1}^N (y_k - \bar{y})^2} \end{aligned}$$

det-F1 adopts the centroid matching scheme consistent with the original c-ResUnet paper [2]: predicted instance centroids and ground-truth centroids are matched greedily in ascending order of distance, pairs with a distance below the threshold δ are recorded as TP, and unmatched predictions and ground truths as FP and FN, respectively, defined as:

$$\text{F1} = \frac{2\text{TP}}{2\text{TP} + \text{FP} + \text{FN}}$$

The threshold δ is set according to the image scale; in this work $\delta = 20$ px at 256×256 , corresponding to 50 px at 512×512 in the original paper.

Post-processing and unified protocol: at inference, instances are separated by a distance-transform-enhanced watershed (geometric watershed) on the foreground probability map, with a threshold of $\tau = 0.5$. The minimum distance d was selected by valley scanning: scanning d in $\{4, 5, 6, 7, 8, 9, 10\}$ on the full model yields a U-shaped MAE (7.60 / 5.92 / 5.62 / 6.00 / 6.52 / 7.25 / 8.11), and $d = 6$ is precisely the balance point between over-segmentation (for $d < 6$, precision collapses from 0.92 to 0.84) and missed detection (for $d > 6$, MAE deteriorates monotonically), as shown in Figure 2; it is therefore adopted uniformly. All models share the same post-processing pipeline and parameters ($\tau = 0.5$, $d = 6$, $\text{min_area} = 30$), ensuring that comparisons across configurations reflect model differences rather than differences in post-processing tuning.

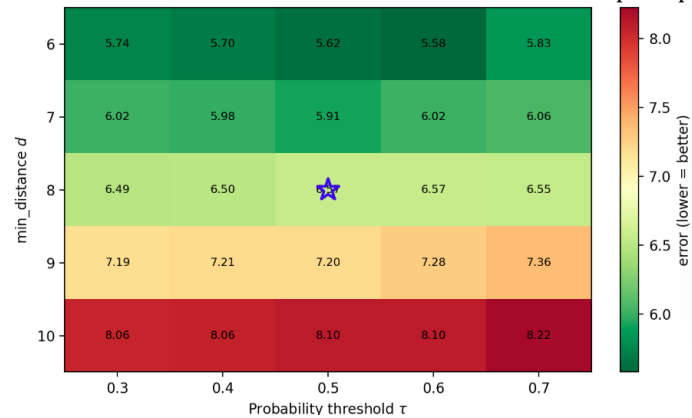


Figure 2. Valley scanning of the minimum distance d .

4.8. Ablation Study

The ablation experiments are organized in the order baseline, PyConv, weight map, C3A, multi-task, and full model. The single-component experiments estimate the marginal change of each mechanism relative to the baseline, while the full model examines the overall performance after joint use. As the current combination is not a complete factorial design, the numerical changes are used to support the analysis of module contributions rather than to interpret each difference as a strictly independent causal effect.

5. Results

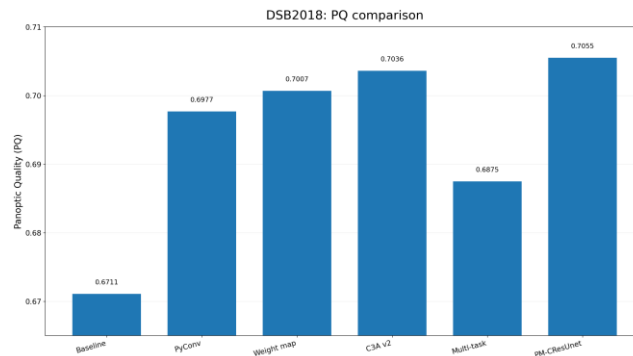
5.1. Segmentation Results on DSB2018

Table 1 presents the component ablation results. The c-ResUnet baseline achieves a PQ of 0.6711; adding PyConv raises it to 0.6977, indicating that multi-scale convolution improves instance-level feature representation; boundary peak weighting further increases it to 0.7007, showing that strengthening the optimization of difficult boundaries helps reduce instance adhesion; and C3A raises the PQ to 0.7036, indicating that expanding the training distribution further improves generalization. The multi-task configuration attains a PQ of 0.6875, a smaller gain than the other components when used alone, whereas the full model reaches 0.7055, indicating that auxiliary supervision remains complementary given sufficient data and the support of other components.

Table 1. Component ablation results on DSB2018.

Configuration	Dice	PQ	AJI	$\Delta PQ(\times 10^{-2})$
Baseline	0.9028	0.6711	0.7385	—
+ PyConv	0.9139	0.6977	0.7588	+2.66
+ Weight map	0.9168	0.7007	0.7619	+2.96
+ C3A	0.9164	0.7036	0.7662	+3.25
+ Multi-task	0.9097	0.6875	0.7485	+1.64
PM-CResUnet	0.9117	0.7055	0.7638	+3.44

Figure 3. PQ comparison of the ablation experiments on DSB2018.



5.2. Comparison with Classical and Transformer-Based Baselines

For external reference, we also compare U-Net [1], c-ResUnet [2], and SegFormer-B0 [15]. The results are shown in **Table 2**. U-Net has 7.77M parameters and a PQ of 0.6911; c-ResUnet has approximately 6.9M parameters and a PQ of 0.6711; and PM-CResUnet achieves a PQ of 0.7055 at a comparable parameter scale. SegFormer-B0 has approximately 3.7M parameters and a PQ of 0.7044, but it is initialized with ImageNet pre-training, whereas PM-CResUnet is randomly initialized. This comparison therefore indicates that PM-CResUnet is competitive in instance-level performance, but it should not be interpreted as a fully matched architectural fairness experiment.

Table 2. Comparison with baseline networks.

Model	Initialization	Params	Dice	PQ	AJI
SegFormer-B0	ImageNet pre-trained	3.7M	0.9185	0.7044	0.7656
U-Net	Random init.	7.77M	0.9131	0.6911	0.7589
c-ResUnet	Random init.	6.9M	0.9028	0.6711	0.7385
PM-CResUnet	Random init.	6.9M	0.9117	0.7055	0.7638

5.3. Counting Results

The counting results are shown in Table 3 and Figure 4. With the unified post-processing setting $\tau = 0.5$ and $d = 6$, the boundary weighting configuration achieves the lowest MAE of 5.596, indicating that boundary enhancement improves the separation of adherent instances. The full model attains an MAE of 5.624, together with the lowest MRE of 0.1182 and the highest detection F1 of 0.8883. These results indicate that improvements in pixel-level segmentation propagate to instance recovery and count estimation to a certain extent, but the metrics are not entirely consistent, so counting capability cannot be judged by Dice alone.

Table 3. Cell counting results on DSB2018.

Configuration	MAE ↓	MRE ↓	det-F1 ↑
Baseline	5.748	0.1336	0.8807
+ Multi-task	5.781	0.1270	0.8853
+ PyConv	5.624	0.1248	0.8867
+ C3A	5.663	0.1239	0.8864
+ Weight map	5.596	0.1259	0.8875
PM-CResUnet	5.624	0.1182	0.8883

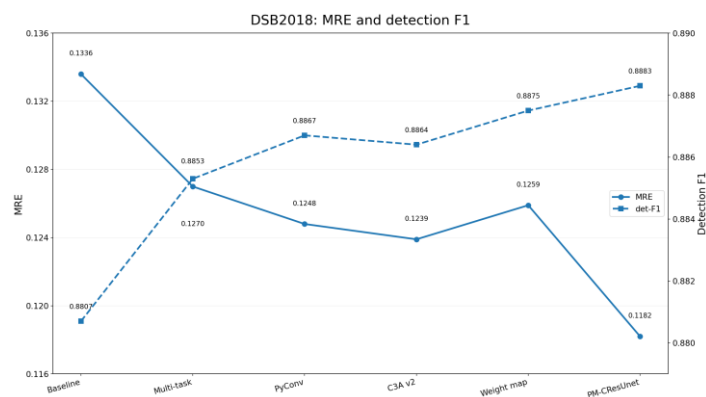


Figure 4. MAE of cell counting on DSB2018.

5.4. Stability and Post-Processing Sensitivity

The multi-task configuration is likewise sensitive to the auxiliary loss weight: an overly large instance loss weight may let the auxiliary gradients weaken the optimization stability of the main segmentation task, whereas adopting $\lambda_{\text{inst}} = 0.5$ markedly narrows the inter-seed variation, indicating that auxiliary supervision in a lightweight backbone requires strength calibration.

6. Conclusion

This paper proposes PM-CResUnet, a multi-constraint framework for lightweight cell segmentation and counting. The model introduces PyConv into c-ResUnet to enhance multi-scale feature representation and combines boundary peak weighting, distance and instance supervision during training, and C3A data augmentation to improve the instance separation and count estimation of adherent cells. The DSB2018 ablation experiments show that the architectural, loss-side, and data-side mechanisms are complementary, and that their joint use achieves the best overall performance in both instance segmentation and counting. Because the auxiliary branches are removed at inference, the model retains a lightweight main segmentation path.

References

- [1] Ronneberger, O., Fischer, P. and Brox, T. (2015) U-Net: Convolutional Networks for Biomedical Image Segmentation. *Lecture Notes in Computer Science*, 9351, 234-241. http://dx.doi.org/10.1007/978-3-319-24574-4_28
- [2] Morelli, R., Clissa, L., Amici, R., Cerri, M., Hitrec, T., Luppi, M., Rinaldi, L., Squarcio, F. and Zoccoli, A. (2021) Automating Cell Counting in Fluorescent Microscopy through Deep Learning with c-ResUnet. *Scientific Reports*, 11, Article No. 22920.
- [3] Zhao, H., Shi, J., Qi, X., Wang, X. and Jia, J. (2017) Pyramid Scene Parsing Network. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2881-2890.
- [4] Duta, I.C., Liu, L., Zhu, F. and Shao, L. (2020) Pyramidal Convolution: Rethinking

- Convolutional Neural Networks for Visual Recognition. arXiv Preprint, arXiv:2006.11538.
- [5] He, K., Gkioxari, G., Dollár, P. and Girshick, R. (2017) Mask R-CNN. Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2961-2969.
 - [6] Schmidt, U., Weigert, M., Broaddus, C. and Myers, G. (2018) Cell Detection with Star-Convex Polygons. Lecture Notes in Computer Science, 11071, 265-273.
 - [7] Stringer, C., Wang, T., Michaelos, M. and Pachitariu, M. (2021) Cellpose: A Generalist Algorithm for Cellular Segmentation. Nature Methods, 18, 100-106.
 - [8] Graham, S., Vu, Q.D., Raza, S.E.A., Azam, A., Tsang, Y.W., Kwak, J.T. and Rajpoot, N. (2019) HoVer-Net: Simultaneous Segmentation and Classification of Nuclei in Multi-Tissue Histology Images. Medical Image Analysis, 58, Article No. 101563.
 - [9] Kervadec, H., Bouchtiba, J., Desrosiers, C., Belaid, A., Dolz, J. and Ben Ayed, I. (2021) Boundary Loss for Highly Unbalanced Segmentation. Medical Image Analysis, 67, Article No. 101851. <http://dx.doi.org/10.1016/j.media.2020.101851>
 - [10] Caicedo, J.C., Goodman, A., Karhohs, K.W., Cimini, B.A., Ackerman, J., Haghighi, M., Heng, C., Becker, T., Doan, M., McQuin, C., Rohban, M., Singh, S. and Carpenter, A.E. (2019) Nucleus Segmentation across Imaging Experiments: The 2018 Data Science Bowl. Nature Methods, 16, 1247-1253. <http://dx.doi.org/10.1038/s41592-019-0612-7>
 - [11] Naylor, P., Laé, M., Rey, F. and Walter, T. (2019) Segmentation of Nuclei in Histopathology Images by Deep Regression of the Distance Map. IEEE Transactions on Medical Imaging, 38, 448-459.
 - [12] Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A. and Zagoruyko, S. (2020) End-to-End Object Detection with Transformers. Lecture Notes in Computer Science, 12346, 213-229.
 - [13] Kirillov, A., He, K., Girshick, R., Rother, C. and Dollár, P. (2019) Panoptic Segmentation. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 9404-9413.
 - [14] Kumar, N., Verma, R., Sharma, S., Bhargava, S., Vahadane, A. and Sethi, A. (2017) A Dataset and a Technique for Generalized Nuclear Segmentation for Computational Pathology. IEEE Transactions on Medical Imaging, 36, 1550-1560.
 - [15] Xie, E., Wang, W., Yu, Z., Anandkumar, A., Alvarez, J.M. and Luo, P. (2021) SegFormer: Simple and Efficient Design for Semantic Segmentation with Transformers. Advances in Neural Information Processing Systems, 34, 12077-12090.