



OPEN Deliod a lightweight detection model for intestinal organoids based on deep learning

Yu Sun^{1,3}, Hanwen Zhang^{2,3}, Fengliang Huang¹, Qin Gao¹, Peng Li², Dong Li²✉ & Gangyin Luo²✉

Intestinal organoids are indispensable tools for exploring intestinal disorders. Deep learning methodologies are often employed in morphological analysis to evaluate the condition of these organoids. Nonetheless, prevailing analytical techniques face obstacles such as many organisational overlaps and tiny targets lead to a high incidence of errors and limited applicability. This paper presents Deliod, a streamlined intestinal organoid detection model founded on YOLOv8 and designed to automate the identification of organoid morphology. Deliod performed excellently compared to leading detection models when applied to an intestinal organoid dataset, attaining an mAP50 of 87.5%. Ablation experiments verified the module's efficacy in improving detection performance. Furthermore, Deliod features a modest parameter count of 5.41 M and a computational load of 16.6 GFLOPs, facilitating the broader application of the detection model in the realm of intestinal organoid image recognition. This streamlined model not only enables efficient and accurate recognition of organoid morphology but also minimizes hardware deployment requirements, broadening its range of potential applications.

Keywords Intestinal organoids, Deep learning, YOLOv8s, Morphological analysis, Lightweight

Organoids are self-organizing three-dimensional cultures that possess the remarkable ability to replicate the cell types, structures, biological complexity, and essential functions of various tissues¹. This makes them valuable tools for disease research, drug screening, and precision medicine. Intestinal organoids, which are derived from intestinal stem cells or induced pluripotent stem cells through three-dimensional culture technology, closely mimic real intestinal tissue with a variety of differentiated cell types and physiological functions². Despite their structural and functional similarities to real tissue, challenges remain in the image analysis of intestinal organoids, particularly in the identification of cell categories. The intricate three-dimensional structures, cell distributions, and diverse cell types present in intestinal organoids pose significant difficulties for image recognition. Moreover, the need to generate large-scale, high-throughput data in organoid research adds layers of complexity to the analysis owing to the high dimensionality and inherent noise of the data and the complex biological processes involved³. To overcome these challenges, researchers are actively exploring and applying artificial intelligence technology, particularly deep learning algorithms, to improve the accuracy and efficiency of cell categorization and distribution identification in intestinal organoid images.

The widespread application of artificial intelligence in the field of organoid development has led to significant innovation in biological research and medicine. In organoid image analysis, deep learning and other algorithms are used for efficient image recognition and feature extraction, improving analysis accuracy and efficiency^{4–6}. In the realm of organoid omics research, artificial intelligence enables researchers to rapidly mine and analyze extensive biomolecular datasets^{7–9}. Within the domain of disease model construction, artificial intelligence integrates multisource data (like genomics, transcriptomics, proteomics) and uses deep learning and machine learning to automatically identify biomarker networks or signaling pathways related to specific diseases. In drug research and development, artificial intelligence simulates drug - biomolecule interactions, predicts drug efficacy, safety, and metabolic pathways, reducing R&D cycle and costs¹⁰. Additionally, artificial intelligence accurately analyzes signal/spectrum changes to assess organoid function and maturity, important for in vitro culture, organ regeneration simulation, and tissue engineering^{11–13}. Furthermore, in drug screening and toxicity

¹College of Electrical and Automation Engineering, Nanjing Normal University, Nanjing 210023, P. R. China.

²Engineering Laboratory of Advanced In Vitro Diagnostic Technology, Suzhou Institute of Biomedical Engineering and Technology Chinese Academy of Sciences, Chinese Academy of Sciences, Suzhou 215163, P. R. China. ³Yu Sun and Hanwen Zhang contributed to the work equally and should be regarded as co-first authors. ✉email: lid@sibet.ac.cn; luogy@sibet.ac.cn

assessments, the use of artificial intelligence has accelerated the drug development process, leading to improved accuracy in screening and success rates^{14–16}.

Deep learning methods are increasingly recognized as valuable tools for analyzing intestinal organoids. Compared with traditional manual methods, deep learning offers notable advantages by enhancing the accuracy of analysis and significantly accelerating processing speeds. Within the realm of intestinal organoid analysis, deep learning methods can be effectively utilized across various domains, including but not limited to classification, target detection, image segmentation, and tracking. Within the domain of classification, CNN, Xception, U-Net and other deep learning models are used to identify organoid morphological features, evaluate vitality and maturity, and distinguish pathological types^{17–19}. For image segmentation, U-Net and its variants perform well in complex background and high-density cell images, achieving high-precision organoid boundary segmentation^{20–22}. In the tracking field, deep learning has been combined with optimization algorithms enabling unmarked, automated organoid sample segmentation and tracking, providing tools for cell fate and development research^{23,24}.

This research is dedicated to the analysis of organoid images and employs deep learning for the detection and categorization of features in intestinal organoid images. While some tools exist for quantification, their capabilities for classification are limited. The deep convolutional neural network OrgaQuant, developed by Kassis et al.²⁵, accurately measures the size of intestinal spheroids. However, it lacks morphological analysis and category determination features, preventing broader applicability. Ostrop et al.²⁶ designed an organoid classification process during epigenetic research, but it requires manual feature selection and category setting, complicating operations. Domènech-Moreno et al.²⁷ developed the Tellu tool with YOLOv5, achieving a 79% classification accuracy for intestinal organoid morphology, which needs improvement. Leng et al.²⁸ developed an improved lightweight Deep-Orga model based on YOLOX to increase the inference speed and performance. Introducing a balanced feature pyramid module and using a balanced L1 loss improved detection performance and addressed the subjectivity and time-consuming nature of intestinal organoid morphological assessment.

Considering the aforementioned limitations and distinctive features of intestinal organoid data, as well as their practical implications, the primary challenges associated with cell categorization and identification within intestinal organoid images can be broadly classified into two areas. First, the imaging process involves intestinal organoids being suspended in a three-dimensional space, which leads to issues such as occlusion, out-of-focus images, and significant variability in size and shape. The presence of numerous small targets within the field of view further complicates organoid cell identification²⁵. Furthermore, the current demands for intestinal organoid image recognition algorithms include not only accuracy but also efficient and lightweight monitoring networks that require low computing power and can be deployed on end-to-side devices. This would significantly increase the practical utility of organoid image analysis algorithms. To address these challenges, this study leverages multicategory intestinal organoid data for training and proposes a lightweight model for detecting intestinal organoids based on YOLOv8s.

This paper presents the following key contributions:

A new lightweight module, DRBNCSPPELAN, is proposed to replace C2f. This module effectively broadens the receptive field, improves the processing capabilities for complex spatial data, accelerates the inference speed, and improves performance. This solves the problem of intestinal organoids being suspended in three-dimensional space and often having considerable heterogeneity.

In view of the problems associated with many small targets, a lightweight ED-FPN module is designed to replace the original neck module. This module optimizes feature selection and integration in the captured image details, fully mixing the higher level of context information to effectively enhance the perception of the intestinal class of small target organs and improve the detection performance of the model.

The original classification loss function is replaced with EMA-SlideLoss, which addresses the issue of imbalanced organoid samples through the SlideLoss function. Additionally, an exponential moving average calculation is employed to mitigate training fluctuations and noise, further enhancing the detection accuracy and stability.

The rest of this paper is organized as follows. Section 2 details the proposed theory and methods. Section 3 briefly introduces the experimental details and analyzes the experimental results. Section 4 discusses and summarizes the key points of the paper.

Methodology

We begin by providing a concise overview of the methodologies and theories addressed in this paper.

Proposed model

YOLOv8, a new addition to the YOLO family released by Ultralytics in 2023, significantly enhances reasoning speed and detection accuracy. Its core architecture comprises a backbone, neck, and head²⁹. The backbone incorporates the CSPDarknet concept and a lightweight C2f module. The neck utilizes PAFPN combined with C2f modules for fusing multiscale features. The head introduces a decoupled head structure to separate classification and detection tasks, optimizing them with different loss functions to accelerate convergence and improve detection performance.

The proposed Deliod model is based on YOLOv8 and incorporates several improvements to its backbone, neck, and loss functions. The discussion below begins with a concise introduction to these four improved components and concludes with an overview of Deliod.

The DRBNCSPELAN module

In deep learning architecture, the introduction of new modules is crucial for enhancing model performance. These modules typically learn multiple related but distinct feature representations to address complex problems. However, such intricate designs also increase the number of parameters and the computational demands of the model³⁰. In the YOLOv8 c2f module, it is necessary to integrate feature maps from different levels, which undoubtedly increases the computational complexity and poses challenges in model design and application. Therefore, in this article, we propose a novel feature extraction module, DRBNCSPELAN, based on the GELAN architecture. Specifically, we replace the C2f module in the YOLOv8 backbone with the DRBNCSPELAN model to produce a more lightweight architecture with enhanced performance. The network structure diagram of DRBNCSPELAN is presented in Fig. 1.

GELAN is a novel network architecture introduced in YOLOv9³¹. It integrates CSPNet³² and ELAN³³, two neural network architectures developed via gradient path planning, to create a generalized efficient layer aggregation network (GELAN) that emphasizes lightweight design, inference speed, and accuracy. CSPNet facilitates the acquisition of richer gradient fusion information while simultaneously reducing computational complexity and enhancing inference speed and accuracy. This technique divides the tensor of the base layer into two segments, which are then merged through a cross-stage hierarchical method. By separating gradient flows, these flows can propagate along distinct network paths. ELAN addresses the issue of the gradual deterioration in deep model convergence that occurs during model scaling. Experimental results obtained by varying ELAN block depths and CSP block depths in GELAN of different sizes indicate that increasing either the ELAN or CSP depth consistently results in a linear relationship between the number of parameters, computational load, and accuracy. This suggests that GELAN is not sensitive to depth, allowing users to freely combine its components to design network architectures that yield models with stable performance.

On this basis, the dilated residual block (DRB) module³⁴ was utilized as a computational block in conjunction with CSPNet and GELAN, leading to the design of the feature extraction module DRBNCSPELAN presented in this article. This integration is based on the original lightweight and accurate GELAN framework. The inclusion of the DRB module effectively expands the model's receptive field while maintaining a low computational cost, thereby enhancing its ability to process data with complex spatial structures and further improving the model's performance. The DRB is a neural network module that combines dilation and reparameterization techniques. It utilizes a nondilated small kernel along with multiple dilated small kernel layers to enhance a nondilated large kernel convolutional layer. Its hyperparameters include the size of the large kernel K , the size of the parallel convolutional layer k , and the expansion rate r . Ignoring input pixels in dilated convolution is similar to inserting additional zero terms into the convolution kernel; as a result, a dilated convolution layer with a small convolution kernel can be equivalently transformed into a transposed convolution with stride r , using noninflated convolution layers with sparse larger kernels.

The original convolution kernel is:

$$W \in R^{k \times k} \quad (1)$$

After inserting zeros, it becomes:

$$W' \in R^{((k-1)r+1) \times ((k-1)r+1)} \quad (2)$$

The transposed convolution with stride r is:

$$W' = \text{conv_transpose2d}(W, I, \text{stride} = r) \quad (3)$$

In the layout depicted in Fig. 2, the depthwise separable convolution block (DRB) comprises four parallel branches, characterized by the parameters $K=9$, a kernel size list $k = (5, 5, 3, 3)$, and an expansion rate list $r = (1, 2, 3, 4)$. For scenarios requiring larger receptive fields (i.e., larger K values), this can be accomplished through

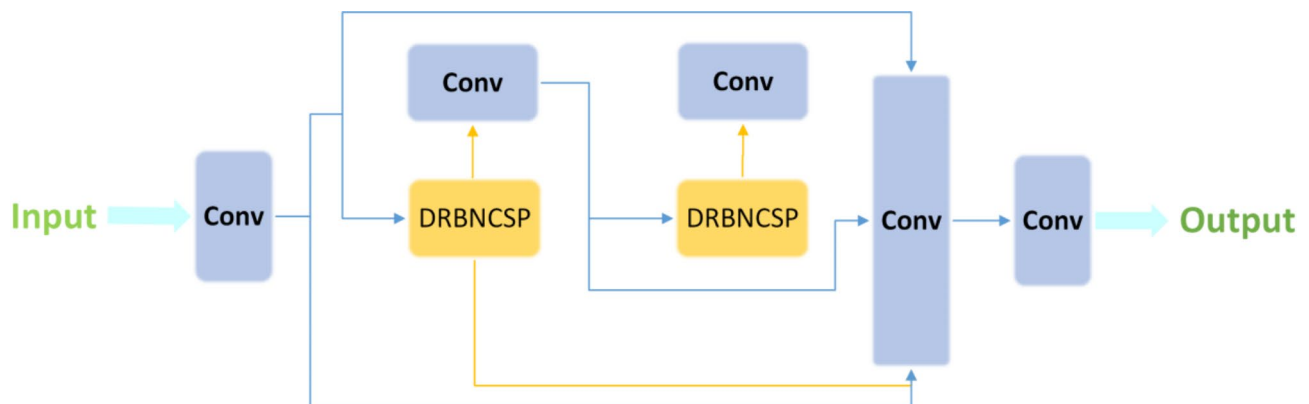


Fig. 1. The network structure diagram of DRBNCSPELAN.

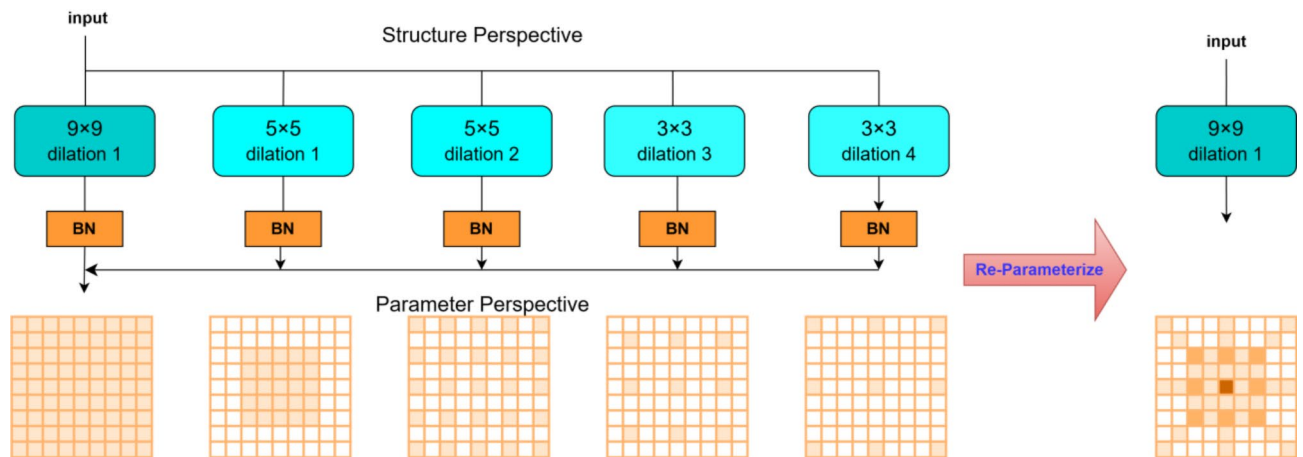


Fig. 2. The structure of the DRB module.

either an increase in the number of parallel branches or layers with larger kernel sizes or expansion rates. The kernel sizes and expansion rates of these parallel branches can be adjusted flexibly; the only constraint that must be satisfied is $(k-1) \times r + 1 \leq K$, which ensures that the resulting large kernel convolutional layer is appropriate. The process of converting a DRB into a large-kernel convolutional layer suitable for inference involves the following steps. First, the batch normalization (BN) layer in each branch is merged with its corresponding preceding convolution (conv) layer. Then, each layer with a dilation rate of $r > 1$ is converted into an equivalent standard convolutional layer via Eq. (3). This layer is transformed into an equivalent standard convolutional layer. This conversion procedure converts the dilated convolutional layer into a standard convolutional layer with sparse large kernels. After all the convolutional layers with dilation rates are converted, all the resulting kernels are summed. Given that the kernel sizes and expansion rates of different branches can vary, some kernels need to be appropriately zero-padded before addition to guarantee that they are aligned in the spatial dimension. Importantly, the logical order of parallel branches in the original DRB remains unchanged throughout the conversion process to ensure that the converted model retains the same functionality and performance as the original DRB. Upon completing these conversion steps, a large-kernel convolutional layer suitable for inference is obtained, which offers enhanced inference speed and optimization potential while retaining the original DRB functionality.

ED-FPN module

In the intestinal organoid dataset, the organoid detection task is confronted by multiscale issues and the challenge of extracting and identifying small target features, which make it difficult for the model to detect organoids precisely. This issue emerges because there are frequent disparities in diameter and morphology among intestinal organoids of different origins, even when the same type of intestinal organoid is imaged under distinct microscopes.

To address this problem, we draw inspiration from³⁵ and design an ED-FPN module based on a hierarchical scale feature pyramid network (HS-FPN) to achieve multiscale feature fusion. This design allows the model to capture a more comprehensive understanding of intestinal organoid characteristics while maintaining a lightweight architecture. The structure of the ED-FPN is illustrated in Fig. 3. The ED-FPN consists of two primary components: the feature selection module and the feature fusion module. Different proportions of feature maps are screened in the feature selection module; subsequently, the high-level and low-level information communication feature fusion modules jointly integrate these features. This integration maximizes the utilization of both local and global contextual information, resulting in rich semantic features that encapsulate detailed characteristics of intestinal organoids across multiple scales. These enhanced semantic features significantly improve the model's ability to detect and accurately localize small target features in microscopy images of intestinal organoids^{36,37}, thereby improving the model's detection capabilities.

The ELA module³⁸ is a crucial component of the feature selection module. It initially processes the input feature graph $f_{in} \in \mathbb{R}^{C \times H \times W}$, where C represents the number of channels, H represents the height of the feature graph, and W represents the width of the feature graph. Two average pooling layers (Avg Pooling) are first applied to the feature map to capture the global sensory field and precise positional information. Subsequently, 1D convolution is applied to enhance the positional information in both the horizontal and vertical directions. Group normalization (GN) is then used to process the enhanced positional information, and this is followed by the application of the sigmoid activation function to determine the weight values for each channel. Finally, the weights for each channel, denoted as $f_{ELA} \in \mathbb{R}^{C \times 1 \times W}$, $f_{ELA} \in \mathbb{R}^{C \times 1 \times H}$, are obtained. The introduction of average pooling enables the network to efficiently acquire information across its wide range of receptive fields and to generate feature maps that contain horizontal and vertical coordinate information within the spatial dimension. The use of one-dimensional convolution captures long-distance dependencies and mitigates the effects of irrelevant areas on label prediction while maintaining a lightweight architecture. Therefore, the integration of these methods within the ELA module allows it to effectively capture the location information of the area of

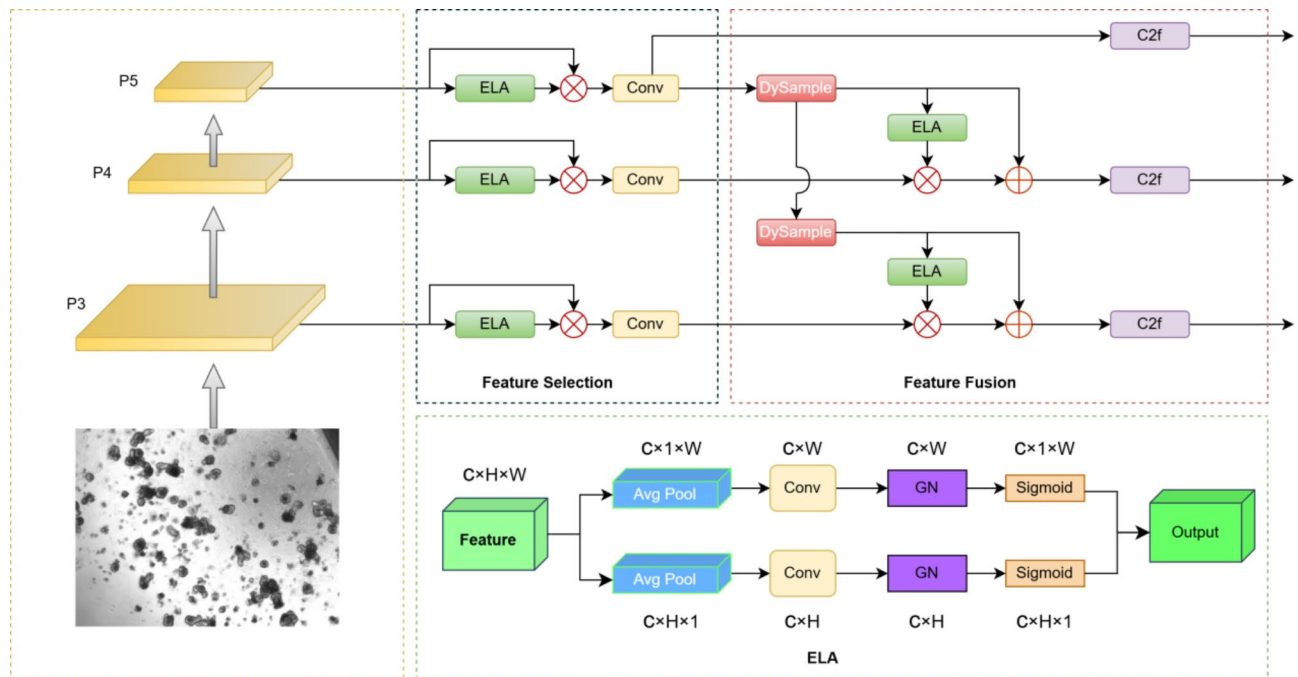


Fig. 3. The structure of the ED-FPN module.

interest, enhances the model's ability to locate the target accurately, and preserves the channel dimension of the input feature map. This preservation avoids the negative impact that a reduction in the channel dimension could have on attention generation. The filtered feature map is then produced by multiplying the weight information by the corresponding scaled feature map. Prior to feature fusion, it is essential to perform size matching on feature maps of different scales, as they have different numbers of channels. A 1×1 convolution is employed for this purpose, reducing the number of channels in each scale feature map to 256.

The original HS-FPN feature fusion module enhances high-level features by employing a transposed convolution (T-Conv) with a stride of 2 and a 3×3 convolution kernel. This is followed by bilinear interpolation for upsampling or downsampling the high-level features. Although this approach is straightforward and efficient, it may inadequately capture detail and small amounts of target information in intestinal organoid images, leading to the loss of critical information. To mitigate this effect, we introduce an ultralightweight and effective dynamic upsampler, the DySample module³⁹, whose structural diagram is presented in Fig. 4. The DySample module significantly enhances the model's performance for low-resolution images and improves its detection capabilities for smaller organoid targets while minimizing false positives and false negatives. DySample employs a point-based sampling method and a learning-based sampling perspective for upsampling, completely avoiding time-consuming dynamic convolution operations and additional subnetworks. This leads to reduced computational resource requirements, enabling increased image resolution without imposing additional burdens, thereby enhancing the model's efficiency and performance at minimal computational cost. After the DySample module produces its output, high-level features are transformed into corresponding attention weights through the ELA module, which filters low-level features. Finally, the filtered low-level features are fused with high-level features via the C2f module.

Enhancement of the YOLOv8 loss function

Intestinal organoids face the challenge of sample imbalance in target detection, which is characterized by a predominance of easy samples over difficult samples. Easy samples are defined by high intersection over union (IoU) values between the predicted bounding box and the actual bounding box, which indicate that the model has accurately positioned the object. In contrast, difficult samples have low IoU values, which may result in inaccurate positioning due to factors such as small target sizes, occlusions, irregular shapes, or complex backgrounds. This issue may be attributed to the binary cross-entropy (BCE) classification loss function employed by the YOLOv8 model, which prioritizes object category prediction while neglecting positional and scale information. Consequently, we propose the introduction of the EMA-SlideLoss loss function⁴⁰. The EMA-SlideLoss function integrates an exponential moving average (EXPMA) with the SlideLoss loss function. In the SlideLoss framework⁴¹, the average IoU values of all bounding boxes are used to establish a threshold denoted as μ . Samples with IoU values below μ are classified as negative samples, and those exceeding μ are classified as positive samples. This approach allows for the adaptive learning of threshold parameters μ for both easy and difficult samples and the assigning of higher weights to difficult samples and lower weights to easy samples. This weighting strategy ensures that the loss function emphasizes difficult samples, enhancing its ability to accurately perform classification for objects across various locations and scales, ultimately improving the accuracy of target detection. The weighting function $f(x)$ is defined as follows:

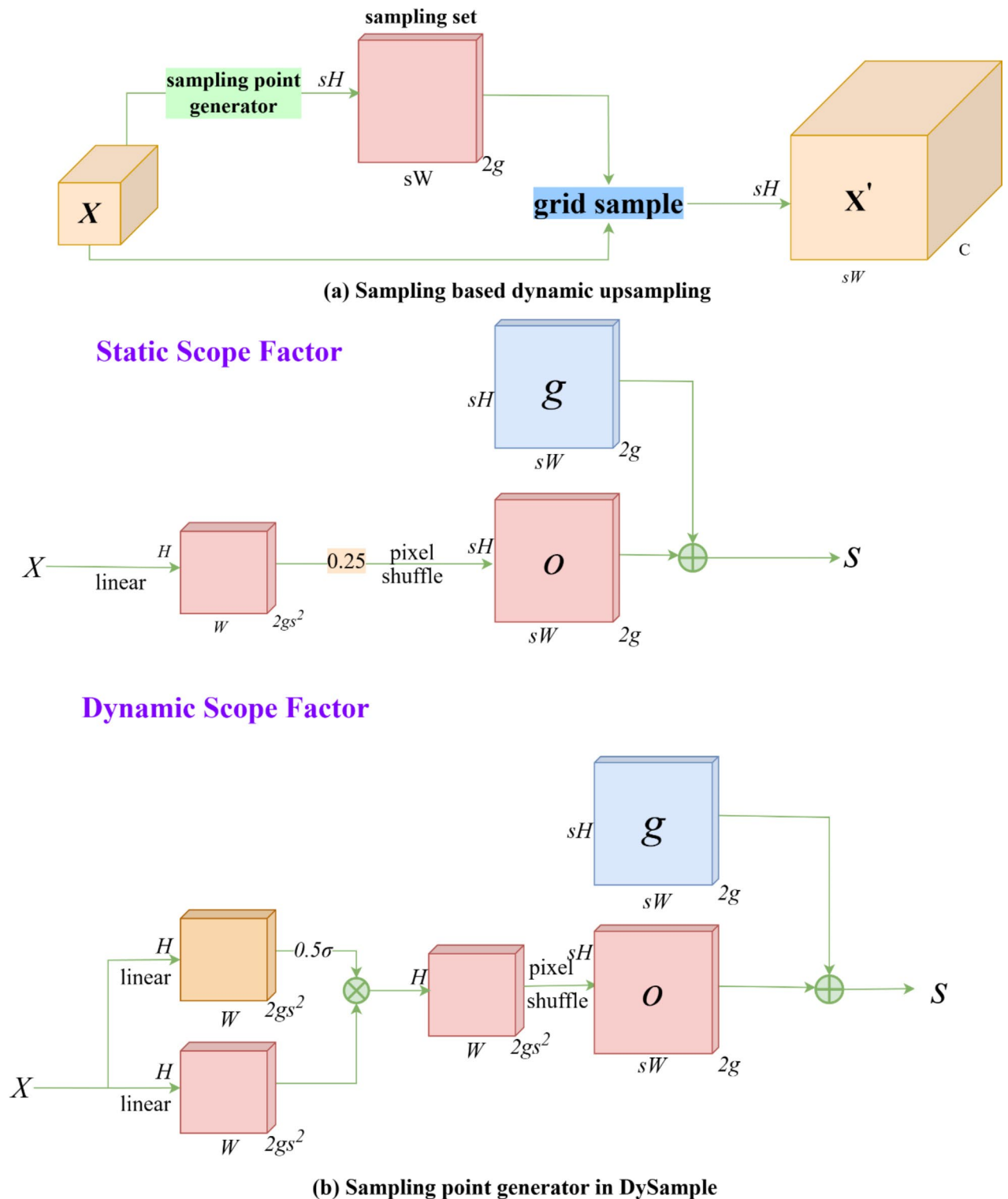


Fig. 4. The sampling-based dynamic upsampling and module designs for DySample. **(a)** The sampling set is generated by the sampling point generator and used to resample the input feature with the grid_sample function. **(b)** In the generator, the sampling set is the sum of the generated offset and the original grid position.

$$f(x) = \begin{cases} 1, & x \leq \mu - 0.1 \\ e^{1-\mu}, & \mu - 0.1 < x < \mu \\ e^{1-x}, & x \geq \mu \end{cases} \quad (4)$$

where x denotes the intersection over union (IoU) between the predicted boxes and the ground truth, and μ represents the weight threshold.

However, as a result of the limited number of difficult samples, directly utilizing μ as the weight value would lead to nonsmooth generated values in the graph and could even introduce noticeable fluctuations and noise due to occasional outliers. In this scenario, we employ an exponential moving average (EXPMA) approach⁴² to optimize the parameter in the model. The specific calculation formula is as follows:

$$\mu_t = \beta \times \mu_{t-1} + (1 - \beta) \times \theta_t \quad (5)$$

where θ_t represents the weights of all the parameters obtained in update t , μ_t represents the moving average of all the parameters in update t , and β denotes the weight parameter.

During training, EMA-SlideLoss calculates the exponential moving average of the SlideLoss loss value to obtain a smoothed loss value. This helps reduce fluctuations and noise in the training process, improving the stability and generalizability of the model.

The proposed Deliod model

The proposed Deliod model retains the core architecture and overall structure of the YOLOv8 model while incorporating the specific structure shown in Fig. 5. To enhance the model, we initially replaced the original YOLOv8 model's backbone feature extraction module C2f with DRBNCSPeLAN. This new module integrates DRB, CSPNet, and GELAN to maintain the portability and accuracy of the original model while effectively expanding the receptive field, enabling better handling of complex spatial structures in organoid data and thereby improving overall performance.

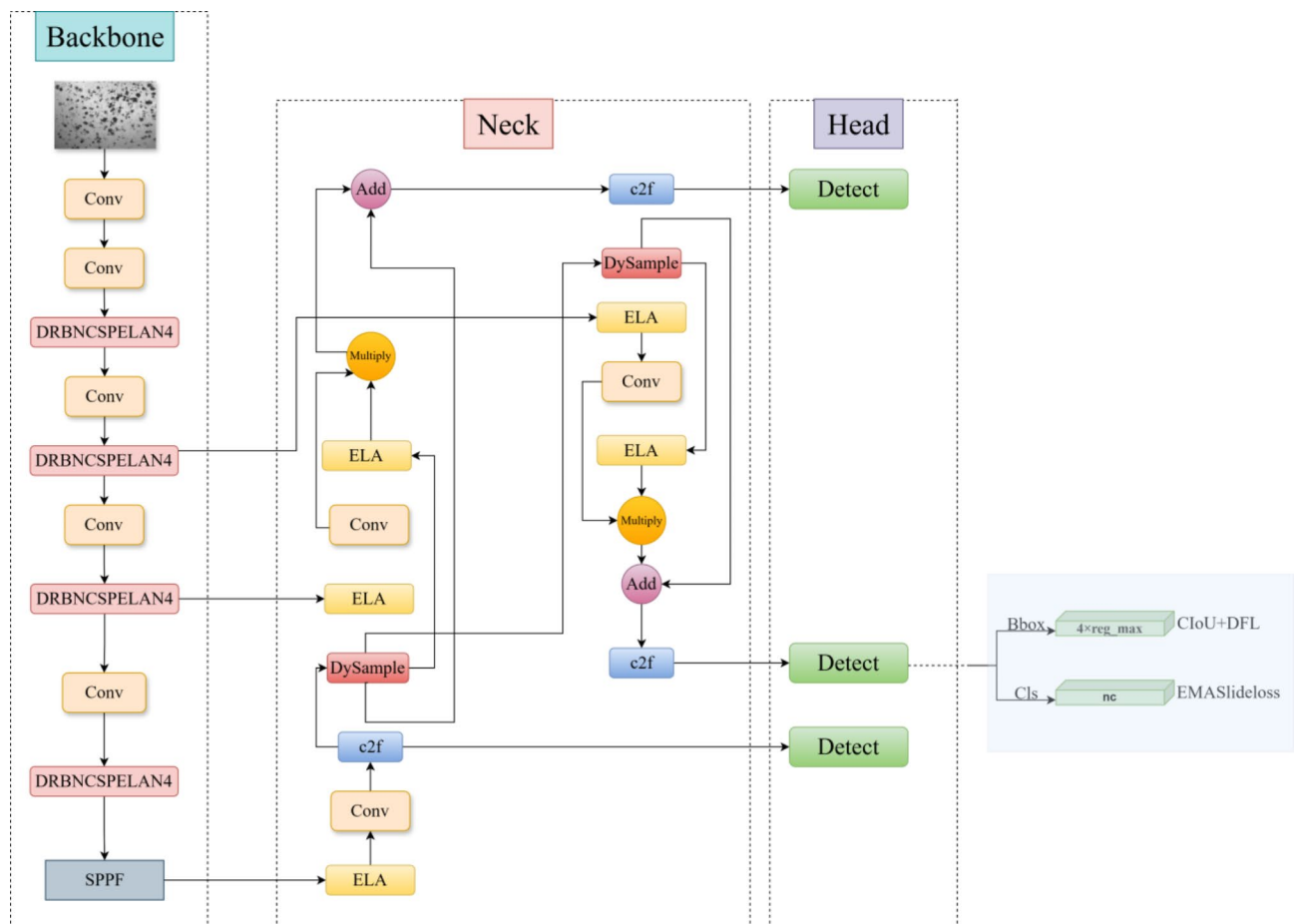


Fig. 5. The framework of the proposed Deliod model pipeline.

Additionally, to address small organoids and multiple targets, we adopt a more lightweight ED-FPN module to replace the original model's neck component. This module generates features containing rich semantic information through feature selection and fusion modules. These features not only capture image details but also incorporate higher-level context information, significantly enhancing detection performance.

Finally, in the model header, we replace the original classification loss function with EMA-SlideLoss. This change not only addresses uneven organoid samples but also reduces training fluctuations and noise through exponential moving average calculations for improved accuracy and stability.

Experiments and results

Datasets

The dataset utilized in this study is derived from previous literature²⁷. The dataset was obtained by using an EVOS FL microscope with a 4× objective lens to acquire a total of 840 transmitted light microscope images of intestinal organoids grown at different stages of culture and at different densities. Culturing of organoids consisted of isolation of crypts from mouse small intestine and inoculation into 48-well plates for culture in a specific medium containing several growth factors, mixed with Matrigel, with each well usually between 12 and 15 mm in diameter and approximately 16–18 mm in depth, with the medium being changed periodically during the culture process and the combination of growth factors being adjusted according to experimental needs. The number of days of culture depended on the experimental design and was not specified, but usually involved multiple passaging cycles. Images were acquired to cover the entire area of the culture well to record growth and morphological changes of the organoids, and the acquisition field was large enough to encompass the entire organoids within the culture well. The images were manually classified into four categories based on the morphological characteristics of the intestinal organoids: non-budding cystic organoids labeled “Cq”, early budding organoids with one or two crypt units labeled “Zq”, late-stage organoids with three or more buds labeled “Wq”, and “spheroids” that lack crypt structures but possess thin walls and larger diameters labeled “Qt”.

The entire dataset encompasses 840 images with a total of 23,066 annotations, resulting in an average of 28.2 annotations per image. Within the dataset, non-budding cystic organoids (Cq) constituted the largest proportion, accounting for 52% with 11,922 instances, followed by early-stage organoids (Zq) with one or two crypt units, which made up 24% with 5,510 instances, spheroids (Qt) representing 14% with 3,368 instances, and finally, late organoids (Wq), comprising 10% with 2,266 instances. For experimental purposes, the dataset was randomly divided into a training set (90%) and a test set (10%). Figure 6, presented below, showcases typical microscopy images from the dataset.

Training strategies

The computer system utilized in the experiment runs on the Windows 11 operating system. The model was constructed using the PyTorch 2.0.0 framework. The CPU configuration features an Intel i7-13700k, while the GPU configuration includes one set of NVIDIA GeForce RTX4070Ti. The initial learning rate of the model is 0.01, with a weight decay coefficient of 0.0001. Employing a warm-up training strategy, we set the number of warmup epochs to 30 and the warmup momentum to 0.8. To reduce memory usage and increase the training speed, we adopt a mixed precision training strategy. Additionally, mosaic image enhancement is disabled in the final 10 epochs for improved performance and efficiency. Tables 1 and 2 provide details of the experimental environment and hyperparameter settings.

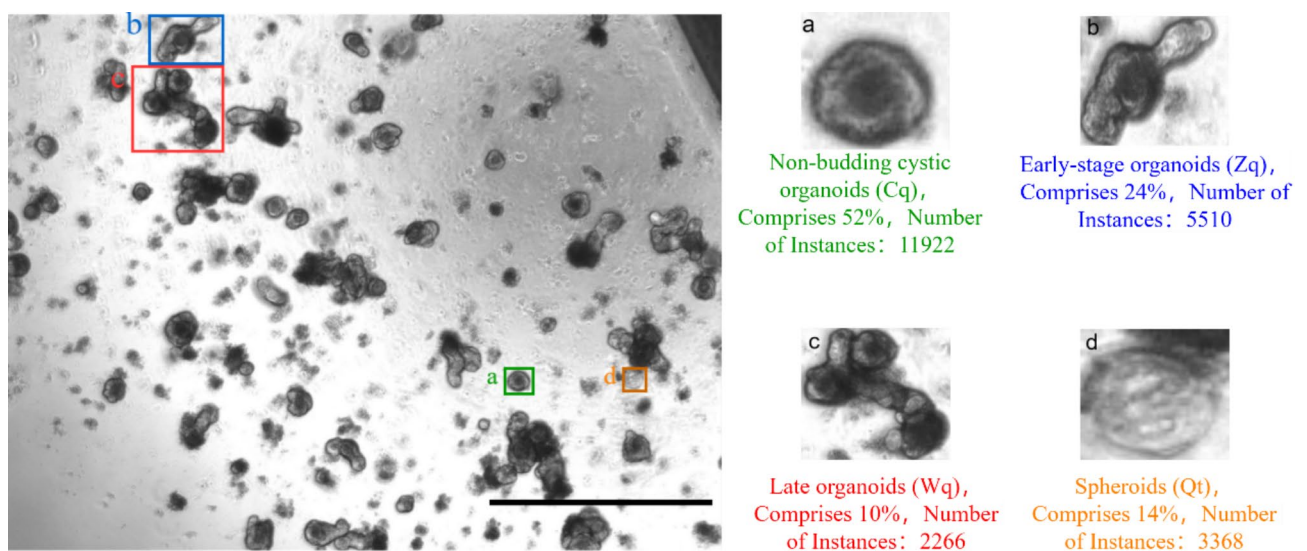


Fig. 6. Representative image showing the four organoid classes. (The entire dataset was randomly divided into a training set (90%) and a test set (10%), boxed areas are shown magnified (x3) on the right. The arrowhead indicates a developing crypt. Scale bar: 1000 μm.)

Software Environment	Hardware Environment
Operating system: Windows 11	CPU: Intel i7-13700k
Programming language: Python 3.9	GPU: RTX 4070ti(12GB)×1
Deep learning framework: PyTorch 2.0.0	
Accelerated environment: CUDA 11.8	

Table 1. The experimental environment.

Parameters	Values
lr0	0.01
lrf	0.001
Momentum	0.937
Weight decay	0.0005
Warmup epochs	30.0
Warmup momentum	0.8
Batch size	4
Epochs	300
Workers	6

Table 2. The hyperparameter settings.

Models	optimizer	Initial learning rate	final learning rate	Batch size	Epoch number
YOLOv8s	SGD	0.01	0.001	4	300
YOLOv7	Adam	0.01	0.1	4	300
YOLOv5s	SGD	0.01	0.001	4	300
Faster R-CNN	SGD	0.01	0.01	4	300
RTDERT-r18	AdamW	0.0001	1.0	4	300
Deliod	SGD	0.01	0.001	4	300

Table 3. Training parameters for each model.

The trained models include Deliod, YOLOv8s, YOLOv7, YOLOv5s, Faster R-CNN, and RTDERT-r18. Among these, YOLOv7 is a prominent model in the YOLO series, while YOLOv5s serves as the fundamental model for the Tellu detector, as described in the literature²⁷. All of these are single-stage detection models. Faster R-CNN represents a typical two-stage detection model with CNN structures, while RTDERT-r18 is a target detection model featuring a transformer structure. Upon completion of training, we save the best model weight file and load it into the corresponding detection image to display the organoid detection results. The parameters used for each model training are detailed in Table 3 below.

Evaluation metrics

To better assess the performance of the model, the analysis indices of COCO data were uniformly adopted in this experiment: the mean average precision (mAP), model parameters (Params), Gigabit floating point operations per second (GFLOPs), and frames per second (FPS). The specific calculation formulae are as follows:

$$P = \frac{TP}{TP + FP} \times 100\%$$
 (6)

$$R = \frac{TP}{TP + FN} \times 100\%$$
 (7)

$$AP = \sum_{i=1}^N p_i \triangle r_i$$
 (8)

$$mAP = \frac{1}{n} \sum_{i=1}^n AP_i$$
 (9)

where N represents the number of targets within the same category, n denotes the total number of target categories, P represents the precision rate, R represents the recall rate, π represents the accuracy function, p_i represents the recall rate function, and TP denotes the number of correct detections. FP (false positives) refers to instances where the model incorrectly identifies a region of one type of organoid as another type. FN (false negatives) refers to cases where the model fails to correctly identify actual organoids. AP represents the accuracy of a specific category. The mAP in this article is calculated as the average accuracy using an intersection ratio threshold of 0.50.

The TIDE method⁴³ is used to conduct an error analysis on the target detection model. This method calculates each error type individually and directly from the original mAP , thereby circumventing the complexity of step-by-step calculations. By defining two IoU values, namely, the foreground threshold f (usually denoted as t_f) and the background threshold b (usually denoted as t_b), IoU_{max} is utilized to represent the overlap of the maximum IoU of the FP with the ground truth (GT) for a specific category. The TIDE method categorizes error prediction in target detection models into six distinct categories:

- (1) Classification Error (Cls): Located correctly but misclassified; $IoU_{max} \geq t_f$ for boxes in the wrong category.
- (2) Localization Error (Loc): The category is correct but the location is incorrect; $t_b \leq IoU_{max} \leq t_f$ indicates the box of the correct category.
- (3) Classification + Localization Error (Cls+Loc): A description classification error or location error; $t_b \leq IoU_{max} \leq t_f$ for misclassified boxes.
- (4) Duplicate detection error (Dupe): $IoU_{max} \geq t_f$; the box is assigned to the correct category, but another detection with a higher score has been associated with the ground truth.
- (5) Background Error (Bkg): the background is detected as a target, and the predicted bounding box with GT satisfies $IoU_{max} \leq t_b$.
- (6) Missed GT Error (Miss): all undetected GTs that do not belong to the classification or localization error.

Experimental results and analysis

The experiment consists of three components: (1) a comparative analysis of Deliod and other current models, (2) an ablation study, and (3) a comparison of the effectiveness of the proposed method for different datasets.

Experimental results and comparative analysis of different models

In this section, we conduct a comprehensive comparison of various models, focusing on five key aspects: detection performance, error weights, computational load, number of parameters, and actual detection efficacy. We begin by comparing the performance of each model in organoid detection and then analyze the error weights to identify the primary sources of error. We subsequently provide a detailed comparison of the computational load, number of parameters, and number of frames processed per second to assess the calculation efficiency, performance, and resource utilization, ensuring that each model achieves an optimal balance between performance and computational efficiency. Finally, we visually demonstrate the actual effectiveness of each model in organoid detection through real test comparisons.

To comprehensively assess the detection performance of each model, we partition the dataset into 10 folds, using one fold for testing and averaging the results over 10 iterations. We adopt evaluation criteria from the COCO dataset and categorize the organoids into three categories based on object size: large (area>96*96 pixels²), medium (32*32 pixels² < area<96*96 pixels²), and small (area<32*32 pixels²). Upon comparison, the Deliod model showed the best detection performance for all types of organoids, followed by YOLOv8s, YOLOv7, YOLOv5s, and RTDERT-r18, respectively, whereas the Faster R-CNN model had the worst detection performance. Additionally, we found that all the models performed better at detecting larger objects. However, not all of the models provided satisfactory performance when dealing with early-stage organoids (Zq) and spheroids (Qt).

To provide a quantitative representation of the detection performance of each model, we calculate the average mAP value of the 10 folds for each model. The mean squared error (MSE) of each model's 10-fold cross-validation is less than 0.001. The detailed detection results are shown in Table 4. The proposed Deliod model exhibits optimal detection performance for the four periods, namely, Cq, Zq, Wq, Qt, and medium, small, and all objects.

We calculated and quantified the absolute error weights for each model, and the results are presented in Table 5. YOLOv5s, Faster R-CNN, and RTDERT-r18 yield higher weights for classification errors, This is consistent with the poorer detection performance observed in the previous experiment. Among all the models,

Model	Average mAP							
	Cq %	Zq %	Wq %	Qt %	Large object %	Medium object %	Small object %	All object%
YOLOv8s	89.0	80.5	86.7	83.2	86.1	84.6	50.8	84.4
YOLOv7	89.2	79.6	85.9	83.2	87.1	83.3	54.2	84.6
YOLOv5s	88.4	80.4	85.7	82.3	85.7	83.5	49.0	83.5
Faster R-CNN	84.7	77.5	80.6	77.7	72.3	80.4	51.3	79.2
RTDERT-r18	87.3	77.6	80.6	77.0	78.7	79.4	55.4	79.9
Deliod	90.7	82.4	87.7	85.2	86.8	86.6	59.5	86.8

Table 4. A performance comparison of various network models.

Model	Cls	Loc	Both	Dupe	Bkg	Miss
YOLOv8s	7.56	0.79	0.51	0.14	1.43	2.55
YOLOv7	8.26	0.86	0.41	0.03	1.58	2.42
YOLOv5s	8.56	0.86	0.40	0.12	1.45	2.86
Faster R-CNN	8.96	1.38	0.46	0.01	1.38	3.85
RTDERT-r18	12.60	0.63	0.46	0.13	1.36	2.10
Deliod	7.26	0.55	0.40	0.23	1.33	2.22

Table 5. A comparison of the absolute weights of each model error.

the Deliod model proposed in this study shows the lowest performance across all the error types, except for repeated detection errors, demonstrating its effectiveness in enhancing detection accuracy and minimizing errors. Notably, Deliod excels particularly in addressing positioning and classification errors. This is attributed to the ELA module, which enhances the model’s positioning capabilities, and the EMA-SlideLoss function, which addresses the challenge of imbalanced organoid samples.

Figure 7 offers a more intuitive comparison of the absolute weights of the errors for each model. It can be observed from the figure that while balancing various errors, the overall performance of the Deliod model is excellent, with the classification error (Cls) and localisation error (Loc) having the lowest weights, compared to other models, such as YOLOv8S and Faster R-CNN, which show some ability in target classification, but have relatively high weights for the omission detection error, YOLOv7 vs. YOLOv5s are higher in localisation accuracy, but still lower than Deliod’s. In addition, although models such as RTDERT-r18 also show some competitiveness, the advantages of Deliod’s model appear to be particularly prominent after comprehensive consideration. This suggests that the model improvements are effective.

We compare the number of parameters and the computational requirements of each model for an input image size of 1088 × 1088 in Table 6. Models such as Faster R-CNN, YOLOv7, and RTDERT-r18 have higher computational demands and parameter counts, whereas Deliod has the lowest values in both categories. Additionally, Table 6 provides the frames per second (FPS) results for each model. A comprehensive comparison reveals that the proposed Deliod model has clear advantages in the target detection task. While Deliod’s FPS is slightly better than those of the YOLOv8s and YOLOv5s models, it still significantly outperforms the other models. Specifically, Deliod’s inference speed is 2–5 times faster than that of the other models.

A more intuitive comparison of the relationships between the detection performance, computational cost, and number of parameters across various models is presented in Fig. 8. The figure illustrates that the Deliod model proposed herein effectively enhances the detection performance while simultaneously reducing both computational demands and the number of parameters, resulting in superior performance among several models.

To evaluate the actual detection effects of the models, we selected representative images to test them. The results show that the Deliod model achieved the best detection results in all cases. As shown in Fig. 9, the Deliod model detects organoids with higher confidence than the other models do. While YOLOv8s, YOLOv7, YOLOv5s, Faster R-CNN, and RTDERT-r18 perform well when detecting large objects, they do not perform well when detecting densely overlapping and small objects, and there are many missed detections and false detections.

Ablation experiment analysis

To evaluate the effectiveness of the Deliod algorithm, we conducted ablation experiments using the original YOLOv8s model. The ablation experiments are structured as follows: A: The original YOLOv8 is used. B: The C2f module in the backbone is replaced with DRBNCSPPELAN. C: The lightweight ED-FPN module is substituted for the neck component of the original model. D: The classification loss function is replaced with EMA-SlideLoss. The results of these experiments are presented in Table 7.

After the C2f module in the backbone was replaced with DRBNCSPPELAN, the model’s accuracy significantly improved in all aspects, particularly for small target objects. Concurrently, the model’s computational requirements and the number of parameters were reduced by 1.91 M and 5.5G, respectively.

After the lightweight ED-FPN module replaced the neck module of the original model, both the computational burden and the number of parameters significantly decreased by 3.98 M and 5.5 G, respectively. Although there was a decrease in detection accuracy for large targets, the overall accuracy of the model was enhanced.

Furthermore, replacing the classification loss function with EMA-SlideLoss did not increase the computational burden or the number of parameters. However, it addressed the sample imbalance problem and improved the accuracy of the model.

Performance comparison with other methods

To verify the generalizability and performance advantages of the model, we selected a different dataset²⁵ for comparison, which consists of bright-field microscopy images of organoid cultures derived from patient duodenal biopsies. The dataset contains 1,750 images and a total of 14,242 organoids. For the purpose of experimentation, the dataset was randomly split into a training set and a test set, with a ratio of 9:1. Figure 10, presented below, showcases typical microscopic images from the dataset. Table 8 lists the statistical detection performance results for two representative datasets. The methods proposed in this article all demonstrate strong detection performance.

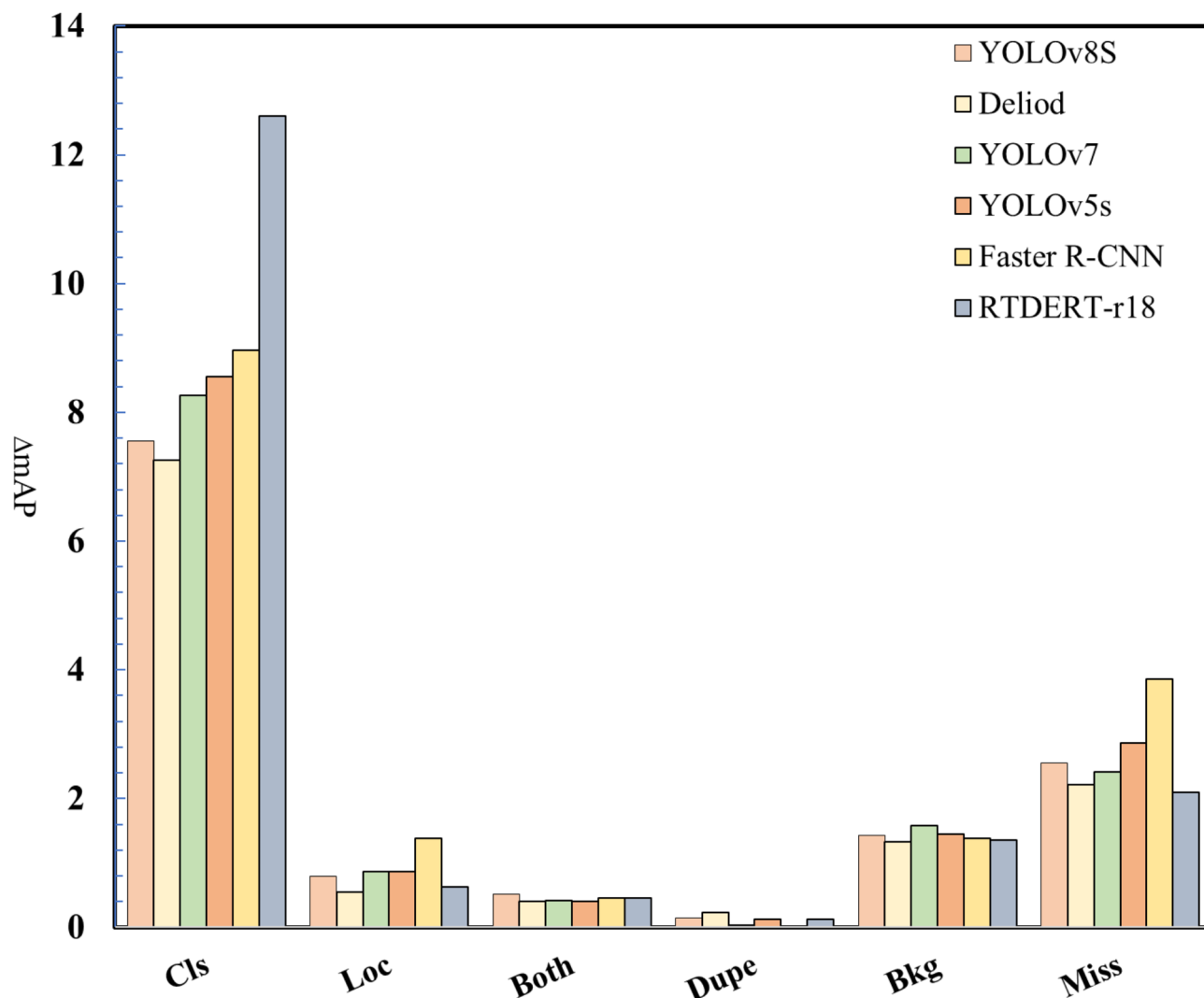


Fig. 7. A comparison of the absolute error weights across different models.(where the y-axis indicates the magnitude of the error weights and different colors represent different models; the x-axis indicates the classification of various errors explanations for each error metric: (1) Classification Error (Cls)Boxes are correctly - located but misclassified, (2) Localization Error (Loc)The category is correct, but location is incorrect, (3) Classification + Localization Error (Cls + Loc)Describes either classification or location error, (4) Duplicate Detection Error (Dupe), the box is assigned to the correct category, but another detection with a higher score has been associated with the ground truth.(5) Background Error (Bkg)Background is detected as a target, (6) Missed Ground Truth(GT) Error (Miss) All undetected GTs not belonging to classification or localization errors.)

Model	Params(M)	GFLOPs(s)	FPS
YOLOv8s	11.13	28.4	166.6
YOLOv7	36.5	103.2	49.2
YOLOv5s	9.11	23.8	165.6
Faster R-CNN	41.364	184	36.9
RTDERT-r18	19.88	57.0	67.1
Deliod	5.41	16.6	163.2

Table 6. A comparison of the number of computations and the number of parameters for each model.

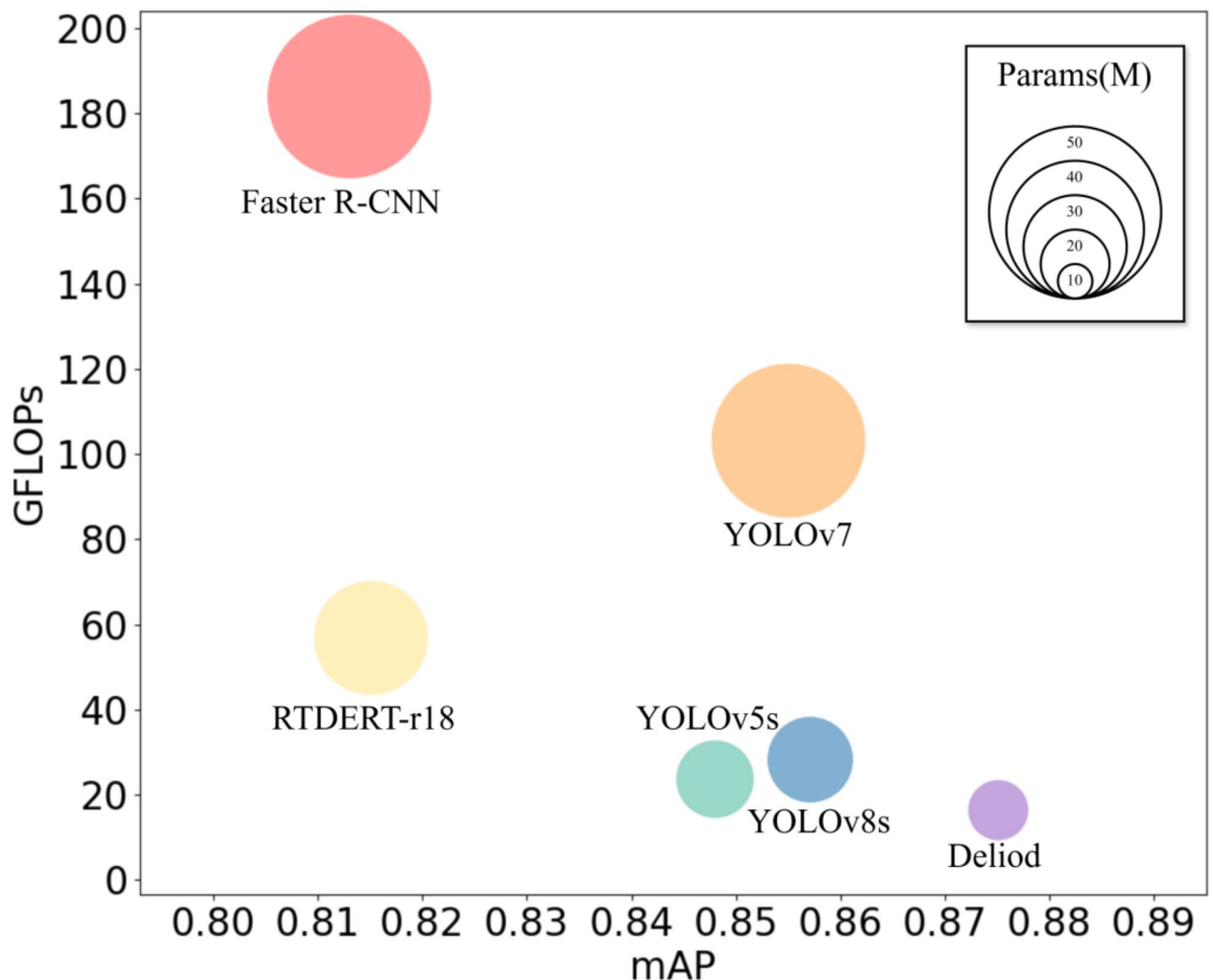


Fig. 8. The detection performance and computational and parametric quantities for each model. The x-axis represents 'mAP' (Mean Average Precision, Mean Average Precision) with values ranging from 0.80 to 0.89, and the y-axis represents 'GFLOPs' (Giga Floating - point Operations per Second, Giga Floating - point Operations per Second) with values ranging from 0 to 200. Giga Floating - point Operations per Second), which ranges from 0 to 200, Params (M)' represents the number of parameters of the model in millions, the smaller the circle the smaller the model is.

Discussion and conclusion

This study has proposed a lightweight detection model for intestinal organoids named Deliod. The challenges associated with occlusions in intestinal organoids, out-of-focus artifacts, considerable heterogeneity in size and shape, as well as the

effectively addressed. Deliod improves organoid detection performance by optimizing the model architecture of YOLOv8 while significantly reducing its computational requirements and number of parameters. Specifically, we replaced the original C2f module with DRBNCSPELAN and introduced the ED-FPN module in the neck. Additionally, we utilized EMA-SlideLoss to address the dataset class imbalance problem while alleviating the problem of heterogeneity.

Compared with the YOLO-series models, Faster R-CNN, and the transformer-based RT-DETR, Deliod exhibits remarkable competitiveness in terms of detection accuracy, computational load, and parameter count. Notably, when computational resources are limited, Deliod adeptly performs intelligent detection, charting a new course for the image analysis of intestinal organoids.

Moreover, the dataset used in this study has a small data volume and an uneven number of categories, which restricts the performance of the Deliod model to some extent. Through an in-depth exploration of the relationships between datasets and experimental outcomes, we determined that the detection results across different categories are influenced both by the quantity of data and the complexity of the features involved. Although the Deliod model has alleviated these effects to some degree, discrepancies in detection accuracy among categories remain a significant concern. This observation underscores the necessity for future research prioritizing the balance and diversity of datasets. In addition, we have not provided biological analyses and proof

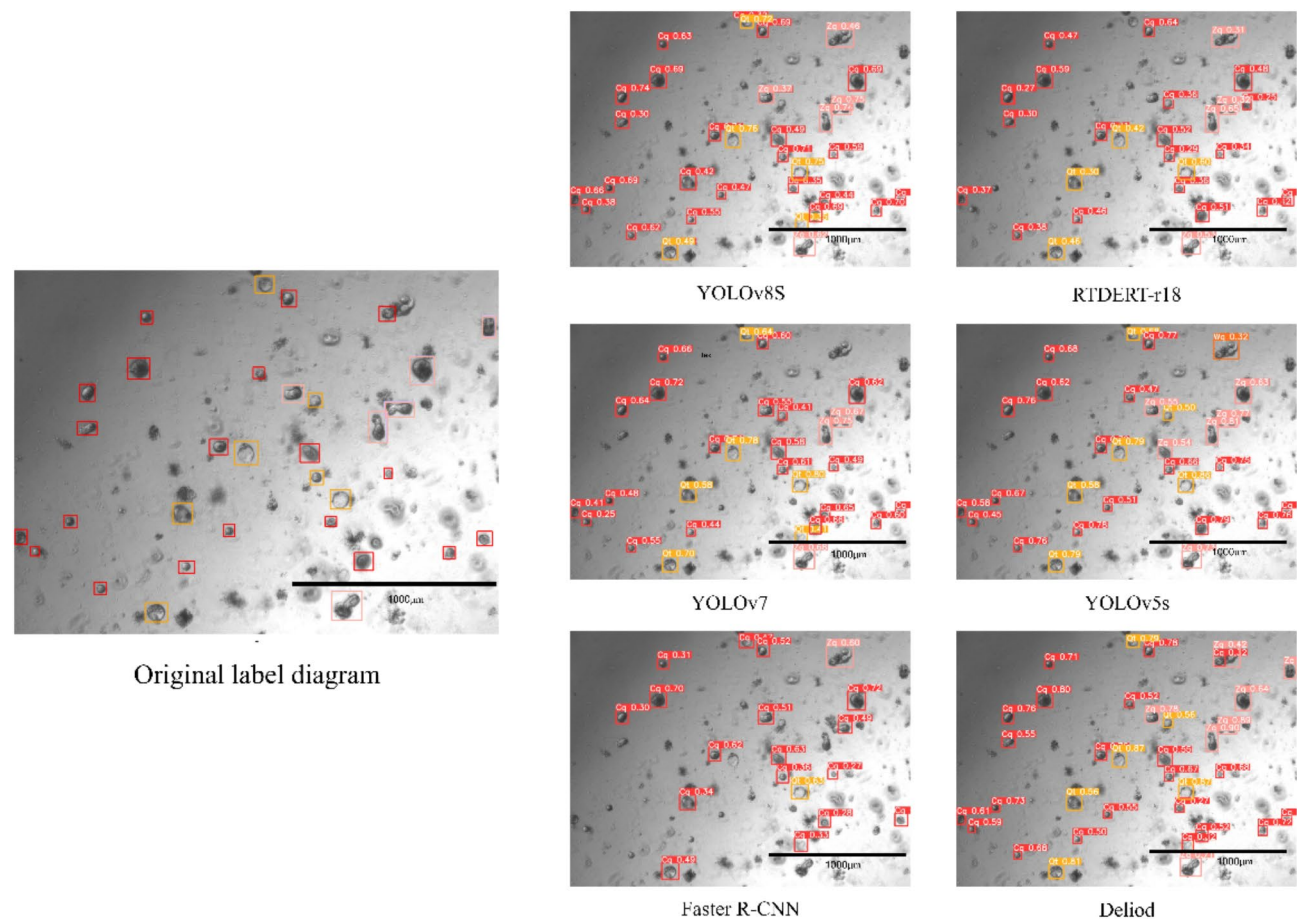


Fig. 9. A comparison of the actual detection effects of each model. The image on the left is the original label anchor frame image, and on the right is the anchor frame image identified by each model, with consistent colours indicating correct identification. The higher confidence level in the upper right corner indicates that the model is more accurate.

A	B	C	D	mAP				
				Cq %	Zq %	Wq %	Qt %	All objects %
√				88.9	82.3	91.1	80.4	85.7
√	√			89.5	83.1	91.5	82.3	86.3
√	√	√		89.9	84.1	90.2	82.9	86.8
√	√	√	√	89.8	85.6	91.8	82.7	87.5
A	B	C	D	mAP				
				Large Objects %	Medium Objects %	Small Objects	Params(M)	GFLOPs
√				81.1	86.0	51.5	11.3	28.8
√	√			83.6	86.5	58.6	9.39	23.3
√	√	√		82.5	86.1	59.7	5.41	16.6
√	√	√	√	81.2	87.7	62.7	5.41	16.6

Table 7. A comparison of ablation experiments with different models.

of the accuracy of the method, due to the limitations of the dataset. Consequently, our future efforts will focus on both expanding and optimizing the dataset and collaborating closely with biologists. On one hand, we will acquire more organoid data while ensuring balanced representation across categories. This will help minimize the influence of data - related factors on model performance and enhance the model's generalizability and detection

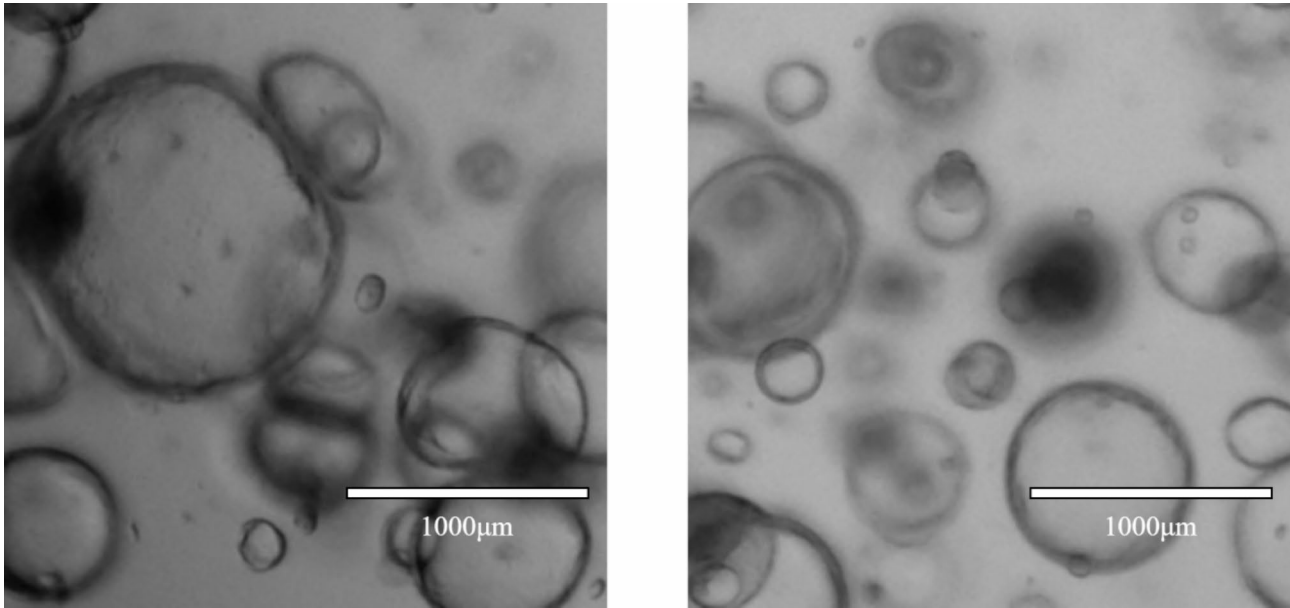


Fig. 10. Typical images and detection results for dataset²⁵. (Bright field microscopy images of tissue organ cultures from patient duodenal biopsies. The dataset contains 1,750 images and 14,242 patients with rheumatoid arthritis, which were randomly divided into training and test sets in a 9:1 ratio in the experiment.)

Dataset	Detection model	mAP
Dataset 1	Tellu ²⁷	79%
	Deep-Orga ²⁸	81%
	Deliod	87.5%
Dataset 2	OrgaQuant ²⁵	80%
	Deep-Orga	90%
	Deliod	92.1%

Table 8. A comparison of the results for two datasets using different methods.

accuracy. On the other hand, we will design experiments to quantify biologically relevant metrics, such as the percentage of organoids forming spherical structures or analyses of morphological differences in biological treatments. By doing so, we can incorporate biological insights into model evaluations, enabling us to gain a more comprehensive understanding of model performance and its impact on biological studies. Furthermore, we have investigated advanced deep learning methods and techniques to further refine the performance of the Deliod model. For example, attention mechanisms, pruning distillation, and other technologies can be employed to increase the capability of the model to extract complex organoid features while controlling the number of parameters and computational load. Additionally, transfer learning, domain adaptation, and other methods can be utilized to improve the model’s generalizability across diverse datasets and to achieve a balanced enhancement in detection accuracy across various categories.

In conclusion, the Deliod model proposed in this study is a new tool that can solve the challenges of intestinal organoid detection with limited computational resources. It has also shown great potential in practical applications, For example, in drug screening and toxicity assessment, by detecting morphological and functional changes in intestinal organoids, the effects of drugs on the intestine can be predicted, and the success rate of new drug development can be improved; in disease modelling and tissue regeneration research, by using deep learning techniques to detect the growth and differentiation process of intestinal organoids, the pathogenesis of intestinal diseases can be understood in depth, and provide a basis for personalised treatment plan development. In the future, we will continue to optimize the performance of the model to provide more efficient, objective, and accurate analytical methods for organoid research.

Data availability

The dataset utilized in this research is freely accessible and openly available to the public via the Zenodo repository at <https://zenodo.org/records/6768583>.

Received: 13 September 2024; Accepted: 5 February 2025

Published online: 11 February 2025

References

1. Zhao, Z. et al. Organoids. *Nat. Rev. Methods Primers* **2**, (2022).
2. Fordham, R. P. et al. Transplantation of expanded fetal intestinal progenitors contributes to colon regeneration after injury. *Cell. Stem Cell.* **13**, 734–744 (2013).
3. Pham, M. T. et al. Generation of human vascularized brain organoids. *Neuroreport* **29**, 588–593 (2018).
4. Xu, J. et al. Deep learning in cell image analysis. *Intelligent Computing* 9861263 (2022). (2022).
5. Sun, A. et al. 3D in vivo magnetic particle imaging of human stem cell-derived islet organoid transplantation using a machine learning algorithm. *Front. Cell. Dev. Biol.* **9**, 704483 (2021).
6. Liu, X. et al. Advances in Deep Learning-Based Medical Image Analysis. *Health Data Sci* 8786793 (2021). (2021).
7. Feng, W. et al. Computational profiling of hiPSC-derived heart organoids reveals chamber defects associated with NKX2-5 deficiency. *Commun. Biol.* **5**, 399 (2022).
8. He, C. et al. BOMA, a machine-learning framework for comparative gene expression analysis across brains and organoids. *Cell. Rep. Methods.* **3**, 100409 (2023).
9. Devall, M. et al. Novel insights into the molecular mechanisms underlying risk of colorectal cancer from smoking and red/processed meat carcinogens by modeling exposure in normal colon organoids. *Oncotarget* **12**, 1863–1877 (2021).
10. Garcez, P. P. et al. Zika virus impairs growth in human neurospheres and brain organoids. *Science* **352**, 816–818 (2016).
11. Hasib, M., Lybrand, Z., Estevez, V. N., Hsieh, J. & Huang, Y. *Characterizing hESCs organoids from electrical signals with machine learning*. in *IEEE EMBS International Conference on Biomedical & Health Informatics (BHI)*. 2019. IEEE. (2019).
12. Becker, L. et al. Data-Driven identification of biomarkers for in situ monitoring of drug treatment in bladder Cancer Organoids. *Int. J. Mol. Sci.* **23**, (2022).
13. Tubbesing, K. et al. Raman microspectroscopy fingerprinting of organoid differentiation state. *Cell. Mol. Biol. Lett.* **27**, 53 (2022).
14. Leslie, J. L. et al. Persistence and toxin production by *Clostridium difficile* within human intestinal organoids result in disruption of epithelial paracellular barrier function. *Infect. Immun.* **83**, 138–145 (2015).
15. Mittal, R. et al. Organ-on-chip models: implications in drug discovery and clinical applications. *J. Cell. Physiol.* **234**, 8352–8380 (2019).
16. Larsen, B. M. et al. A pan-cancer organoid platform for precision medicine. *Cell. Rep.* **36**, 109429 (2021).
17. Powell, R. T. et al. deepOrganoid: a brightfield cell viability model for screening matrix-embedded organoids. *SLAS Discov.* **27**, 175–184 (2022).
18. Abdul, L. et al. D-CryptO: deep learning-based analysis of colon organoid morphology from brightfield images. *Lab. Chip.* **22**, 4118–4128 (2022).
19. Okamoto, T. et al. Integration of human inspection and artificial intelligence-based morphological typing of patient-derived organoids reveals interpatient heterogeneity of colorectal cancer. *Cancer Sci.* **113**, 2693–2703 (2022).
20. Ronneberger, O., Fischer, P. & Brox, T. U-net: Convolutional networks for biomedical image segmentation. in *Medical image computing and computer-assisted intervention—MICCAI 2015: 18th international conference, Munich, Germany, October 5–9, proceedings, part III* 18. 2015. Springer. (2015).
21. Park, T. et al. Development of a deep learning based image processing tool for enhanced organoid analysis. *Sci. Rep.* **13**, 19841 (2023).
22. de Medeiros, G. et al. Multiscale light-sheet organoid imaging framework. *Nat. Commun.* **13**, 4864 (2022).
23. Matthews, J. M. et al. OrganoID: a versatile deep learning platform for tracking and analysis of single-organoid dynamics. *PLoS Comput. Biol.* **18**, e1010584 (2022).
24. Zheng, X. et al. Organoid cell fate dynamics in space and time. *Sci. Adv.* **9**, eadd6480 (2023).
25. Kassis, T., Hernandez-Gordillo, V., Langer, R. & Griffith, L. G. OrgaQuant: human intestinal organoid localization and quantification using deep convolutional neural networks. *Sci. Rep.* **9**, 12479 (2019).
26. Ostrop, J. et al. A semi-automated Organoid Screening Method demonstrates epigenetic control of intestinal epithelial differentiation. *Front. Cell. Dev. Biol.* **8**, 618552 (2020).
27. Domènech-Moreno, E. et al. Tellu - an object-detector algorithm for automatic classification of intestinal organoids. *Dis. Model. Mech.* **16**, (2023).
28. Leng, B., Jiang, H., Wang, B., Wang, J. & Luo, G. Deep-Orga: an improved deep learning-based lightweight model for intestinal organoid detection. *Comput. Biol. Med.* **169**, 107847 (2024).
29. Varghese, R. & Sambath, M. YOLOv8: A Novel Object Detection Algorithm with Enhanced Performance and Robustness. in *International Conference on Advances in Data Engineering and Intelligent Computing Systems (ADICS)*. 2024. IEEE. (2024).
30. Ma, N., Zhang, X., Zheng, H. T. & Sun, J. ShuffleNet v2: Practical guidelines for efficient cnn architecture design. in *Proceedings of the European conference on computer vision (ECCV)*. (2018).
31. Wang, C. Y., Yeh, I. H. & Liao, H. Y. M. YOLOv9: Learning what you want to learn using programmable gradient information. *arXiv preprint arXiv:2402.13616*, (2024).
32. Wang, C. Y. et al. CSPNet: A new backbone that can enhance learning capability of CNN. in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*. (2020).
33. Wang, C. Y., Liao, H. Y. M. & Yeh, I. H. Designing network design strategies through gradient path analysis. *arXiv preprint arXiv:2211.04800*, (2022).
34. Falahkheirkhah, K., Yeh, K., Confer, M. P. & Bhargava, R. DRB-net: Dilated residual block network for infrared image restoration. in *International Symposium on Visual Computing*. Springer. (2022).
35. Chen, Y. et al. Accurate leukocyte detection based on deformable-DETR and multi-level feature fusion for aiding diagnosis of blood diseases. *Comput. Biol. Med.* **170**, 107917 (2024).
36. Cao, Y., Xu, J., Lin, S., Wei, F. & Hu, H. Global context networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **45**, 6881–6895 (2023).
37. Wang, X., Girshick, R., Gupta, A. & He, K. Non-local neural networks. in *Proceedings of the IEEE conference on computer vision and pattern recognition*. (2018).
38. Xu, W. & Wan, Y. E. L. A. Efficient Local Attention for Deep Convolutional Neural Networks. *arXiv preprint arXiv:2403.01123*, (2024).
39. Liu, W., Lu, H., Fu, H. & Cao, Z. Learning to upsample by learning to sample. in *Proceedings of the IEEE/CVF International Conference on Computer Vision*. (2023).
40. Jiang, T. et al. Improved YOLOv8 model for Lightweight Pigeon Egg Detection. *Anim. (Basel)* **14**, (2024).
41. Yu, Z. et al. Yolo-facev2: a scale and occlusion aware face detector. *Pattern Recogn.* **155**, 110714 (2024).
42. Tarvainen, A. & Valpola, H. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. *Advances in neural information processing systems* 30, (2017).
43. Bolya, D., Foley, S., Hays, J. & Hoffman, J. Tide: A general toolbox for identifying object detection errors. in *Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, Proceedings, Part III* 16. 2020. Springer. (2020).

Acknowledgements

The authors thank the teachers from Suzhou Institute of Biomedical Engineering and Technology, Chinese Academy of Sciences, and the School of Electrical Engineering and Automation, Nanjing Normal University, for their support.

Author contributions

YS, HWZ: Investigation, Formal analysis, Writing - Original Draft. FLH, QG, PL: Writing - review & editing. DL: Investigation, Supervision. GYL: Project administration, Funding acquisition. All authors contributed to the article and approved the submitted version. All authors read and approved the final manuscript.

Funding

This work was supported by Suzhou Key Technology (Research) Project of Critical and Infectious Diseases Precaution and Control (Grant No. GWZX202102); Strategic Priority Research Program of the Chinese Academy of Sciences (Grant No. XDA16021102).

Declarations

Competing interests

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to D.L. or G.L.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Open Access This article is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License, which permits any non-commercial use, sharing, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if you modified the licensed material. You do not have permission under this licence to share adapted material derived from this article or parts of it. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by-nc-nd/4.0/>.

© The Author(s) 2025