Supplementary information

A novel one-stage deep learning based method for automatic analysis of droplet PCR images

Yuanyang Yao,[a] Shuhao Zhao,[a] Yan Liang,[a] Fei Hu*[a] and Niancai Peng*[a]

[a] State Key Laboratory for Manufacturing Systems Engineering, Xi'an Jiaotong University, Xi'an, 710054, Shaanxi, China

Fei Hu*-- E-mail: hufei0701@xjtu.edu.cn.
Niancai Peng*-- E-mail: ncpeng@email.xjtu.edu.cn. Phone: +86 29 86670656. Fax: +86 29 82216680.

Contents of Supporting Information

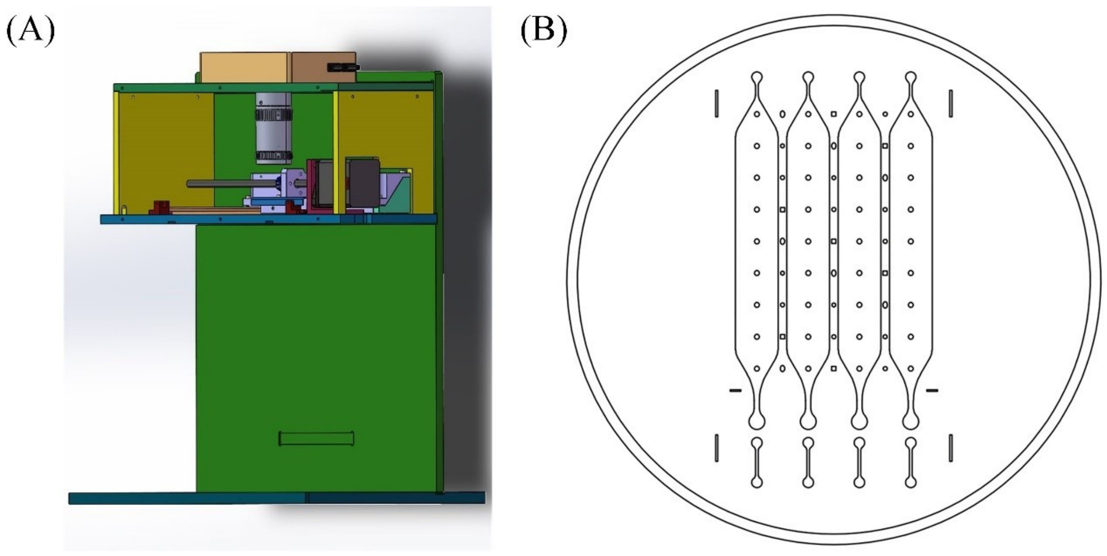

Fig. S1. The schematic of our system. (A) Assembly diagram of optical imaging system. (B) Design diagram of ddPCR chip.

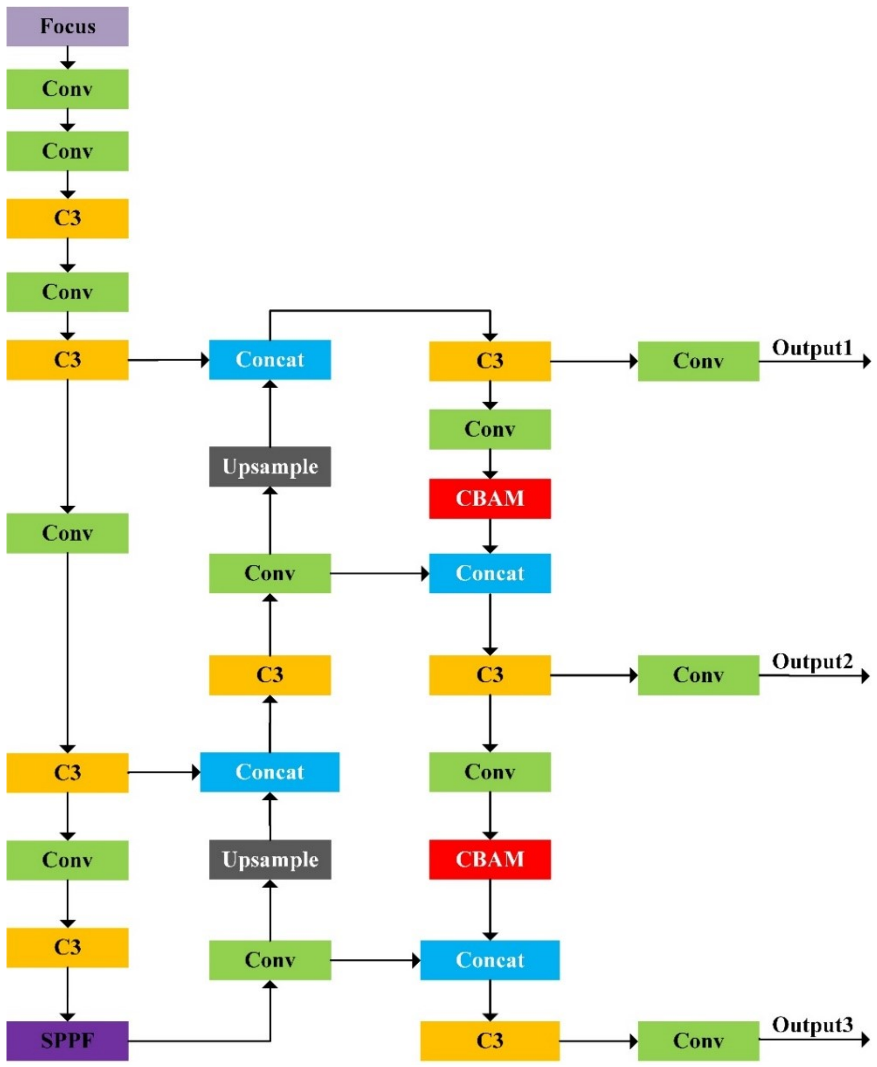

Fig. S2. Structure of the improved YOLOv5 network.

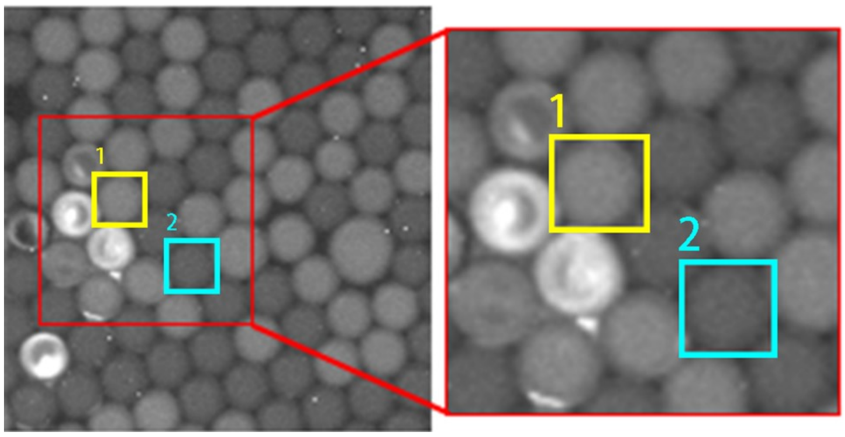

Fig. S3. An image labeled by Darklabel. Droplets in the yellow and blue boxes are positive and negative droplets, respectively.

The YOLOv5 series algorithms extract features from images using several convolutional kernels of different sizes. However, distinguishing the class of the center droplet correctly requires using brightness information from the surrounding droplets. Therefore, we labelled part of the environment around the droplet as target information during labelling (as shown in Fig. S3). This facilitates the network in learning about the environment around the droplet, enabling it to more accurately determine the class of the droplet.

Text S1. Calculation of attentional mechanisms.

Attention mechanisms were initially developed in the field of machine translation, but are now widely used to improve the performance of models for small target detection. The principle of the attention mechanism module is to allocate different weights to the outputs of the network nodes and adjust the weights through continuous training to highlight some key information and suppressing noises.

Attentional mechanisms can be broadly classified into two categories. In the first category, the channel attention mechanism assigns weights to different channels and suppresses unimportant information. In the other category, the spatial attention mechanism assigns weights to different locations of the feature map in every channel. Generally, the channel attention mechanism is more focused on the question of what the target is, whereas the spatial attention mechanism is more focused on the location of the target.

The Convolutional Block Attention Module (CBAM) combines the advantages of both the channel and spatial attention mechanisms, as shown in Fig. S4. The whole process can be expressed by the following equations:

$$\begin{cases} W_1 = \sigma\left(Conv^*\left(AvgPool(I)\right) + Conv^*\left(MaxPool(I)\right)\right) \\ I' = W_1 \times Input \\ W_2 = \sigma(Conv(Concat(AvgPool(I'), MaxPool(I')))) \\ Output = W_2 \times I' \end{cases}$$

The channel attention module performs maximum pooling and global average pooling on the input I with a dimension of H×W×C to obtain two feature vectors with dimensions of 1×1×C. Two convolution operations are then performed on these vectors, and the output feature is activated by the activation function to obtain the weight coefficient, W_1. The input is then fed to the spatial attention module after the channel weight W_1 was applied. Subsequently, the spatial attention module performs maximum pooling and global average pooling on the channel dimension of the input. The resulting features are then concatenated, and the weight W_2 is obtained by convolution and activation operations.

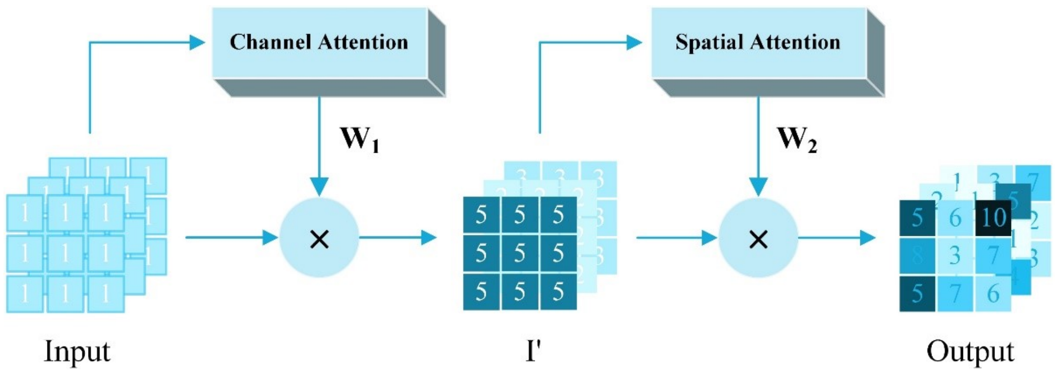

Fig. S4. Process of CBAM attention module.

Text S2. Calculation of the loss.

During training a deep learning model, the value of the loss function is used to evaluate the difference between the predicted results of the current parameters and the true results in the training set. The value for the loss function is continuously varied throughout the training cycle to guide the model parameters to the appropriate direction.

The Complete-IoU loss (CIoU) used by YOLOv5 was improved from IoU, and three factors were simultaneously considered: shape cost, distance cost and IoU. It is calculated by the following equations:

$$\begin{cases} CIoU = IoU - \left(\dfrac{\rho^2(b,b^{gt})}{c^2} + \alpha v\right) \\ v = \dfrac{4}{\pi^2}\left(\arctan\dfrac{w^{gt}}{h^{gt}} - \arctan\dfrac{w}{h}\right)^2 \\ \alpha = \dfrac{v}{(1 - IoU) + v} \\ Loss_{CIoU} = 1 - CIoU \end{cases}$$

where α is a weighting factor, ν is the shape cost, ρ is the Euclidean distance between the centre points of the two boxes, and c is the diagonal length of the smallest rectangular region that can contain both boxes. CIoU is used to solve the problem that the result of Distance-IoU (DIoU) loss is the same as that of IoU loss when the centre points of the prediction box coincide with that of the real box. However, the models using CIoU loss cannot be optimized when two prediction boxes have the same aspect ratio and their centres overlapped with those of the real box.

In SCYLLA-IoU (SIoU), the angle between the prediction box and the real box was employed to accelerate the convergence of the model to redefine the distance cost as follows:

$$\begin{cases} \Delta = \sum\limits_{t = x,y} \left(1 - e^{-\gamma\rho_t}\right) \\ \rho_x = \left(\dfrac{b_{cx}^{gt} - b_{cx}}{c_w}\right)^2, \rho_y = \left(\dfrac{b_{cy}^{gt} - b_{cy}}{c_h}\right)^2 \\ \gamma = 2 - \Lambda \\ \Lambda = \sin(2\alpha) \end{cases}$$

where Δ is the distance cost, α is an angle shown in Fig. S5, γ is the weight associated with α, and b_{cx}^{i} and b_{cy}^{i} are the horizontal and vertical coordinates of the centre point of the corresponding box, respectively. The shape loss was also redefined as follows:

$$\begin{cases} \Omega = \sum_{t=w,h} \left(1 - e^{-\omega_t}\right)^{\theta} \\ \omega_w = \dfrac{\left|w - w^{gt}\right|}{max\left(w,w^{gt}\right)}, \omega_h = \dfrac{\left|h - h^{gt}\right|}{max\left(h,h^{gt}\right)} \end{cases}$$

where Ω is the shape cost, and w and h are the width and height of the corresponding boxes, respectively. The size of θ indicates the attention of the model to shape cost, and its value ranged between 2 and 6.

The SIoU loss is calculated by the following equation:

$$\begin{cases} SIoU = IoU - \dfrac{\Delta + \Omega}{2} \\ Loss_{SIoU} = 1 - SIoU \end{cases}$$

The total loss is calculated by the following equation:

$$L = W_{box}Loss_{SIoU} + W_{cls}Loss_{cls} + W_{obj}Loss_{obj}$$

where W_{box}, W_{obj} and W_{cls} are the weights of box loss, object loss, and class loss, respectively.

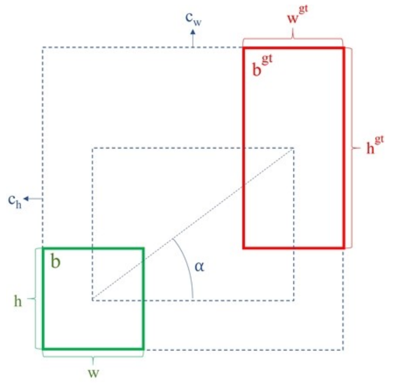

Fig. S5. Diagram representing the physical meaning of parameters required for SIoU loss calculation.

Text S3. Principle of model pruning.

Convolutional neural networks need to determine millions of parameters during the training process. This can make the model more complex and increase the time needed for detection. In addition, having too many parameters can also result in a large model size, which makes it difficult to apply the model to monolithic devices. Network pruning is a technique that can be used to remove unimportant channels and reduce the size of the model while maintaining its accuracy. The steps for network pruning are shown in Fig. S6.

After convolution operation, batch normalization is usually performed to adjust the data distribution. The process involves the introduction of scale factor γ and bias factor βinto the batch normalization process:

$$x_{out} = \frac{x_{in} - \mu_c}{\sqrt{\sigma_c^2 + \varepsilon}}\gamma + \beta$$

where σ_C^2 and μ_C are the variance and mean of the input data x_{in} in a mini-batch, respectively, γ and β are the parameters to be learned, and ε is a very small number that prevents the denominator from reaching zero.

The value of γ reflects the importance of the convolution kernel in a given channel. According to the formula above, when the value of γ approaches zero, the output of the channel becomes constant; that is, the channel contains few or no image features and can be safely removed. However, the general training method seldom produces channels with γ equal to 0. To obtain a sparse model, a new loss function was introduced:

$$L_{new} = L_{ori} + \lambda\sum_{\gamma \in \Gamma} g(\gamma)$$

where L_{ori} is the loss function of the original network, λ is a hyperparameter, and $g(\gamma) = |\gamma|$ is the L1 regularization.

Using the new loss function to train the model can cause the value of γ in many batch normalization layers to converge towards 0. The model can then be pruned using the pruning process. After pruning, the numbers of output channels and input channels for each layer are changed, and the new model requires to be retrained to meet accuracy requirements.

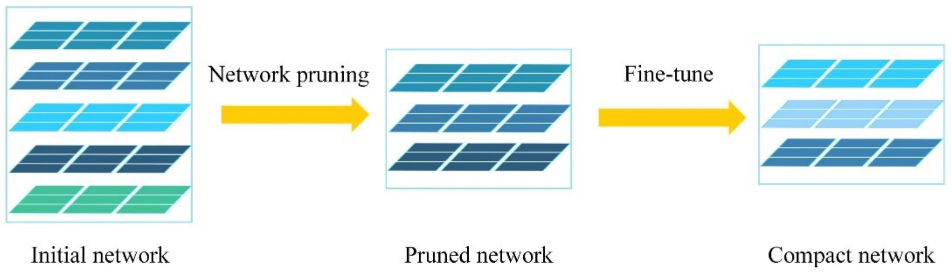

Fig. S6. Schematic diagram of the model pruning process.

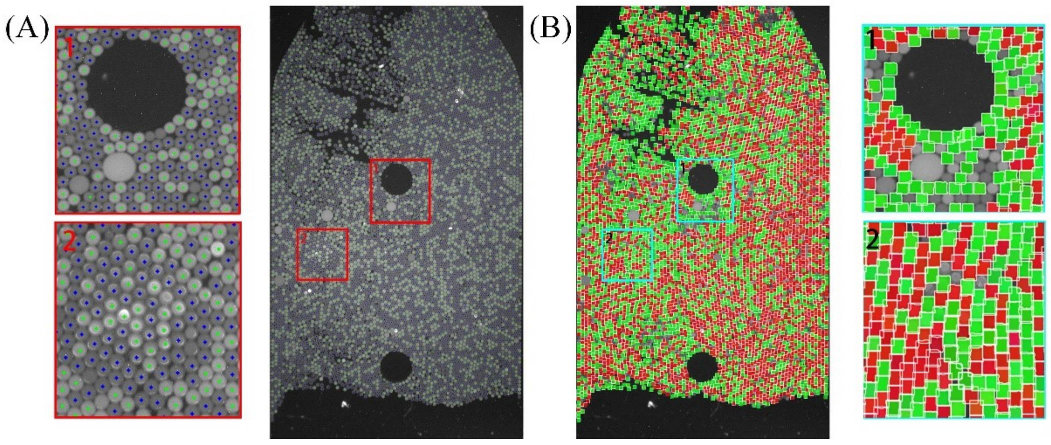

Fig. S7. The images processed using deep learning algorithms.

Our method (A) yielded higher detection rate and effectively avoided interference, while the output image using Mask R-CNN (B) exhibited lower detection rates and lower classification accuracy for droplets influenced by uneven illumination.

Table S1: Hyperparameter used for training model.

Number	Parameter name	Value
1	lr0	0.01
2	lrf	0.1
3	momentum	0.937
4	warmup_epochs	3.0
5	warmup_momentum	0.8
6	warmup_bias_lr	0.1
7	box	0.05
8	cls	0.5
9	cls_pw	1.0
10	obj	1.0
11	obj_pw	1.0
12	iou_t	0.20
13	anchor_t	1.3
14	fl_gamma	0.0
15	hsv_h	0.015
16	hsv_s	0.7
17	hsv_v	0.4
18	degrees	0.0
19	translate	0.1
20	scale	0.5
21	shear	0.0
22	pers[ective	0.0
23	flipud	0.5
24	fliplr	0.5
25	mosaic	1.0
26	mixup	0.001
27	copy_paste	0.01