Mathematical Biosciences and Engineering

MBE, 20(2): 3970–3982.
DOI: 10.3934/mbe.2023185
Received: 09 August 2022
Revised: 22 November 2022
Accepted: 27 November 2022
Published: 14 December 2022

http://www.aimspress.com/journal/MBE

Research article

Research on cell detection method for microfluidic single cell dispensing

Junjing Cai[1,2,†], Qiwei Wang[1,†], Ce Wang[1,*] and Yu Deng[2,*]

[1] Jihua Institute of Biomedical Engineering Technology, Jihua Laboratory, Foshan 528200, China
[2] School of Electromechanical Engineering, Guangdong University of Technology, Guangzhou 510006, China

† These authors contributed equally to this work.

* **Correspondence:** Email: 15190010156 @163.com, yu.deng@gdut.edu.cn; Tel: +15190010156, +18680507030.

Abstract: Single cell dispensing techniques mainly include limiting dilution, fluorescent-activated cell sorting (FACS) and microfluidic approaches. Limiting dilution process is complicated by statistical analysis of clonally derived cell lines. Flow cytometry and conventional microfluidic chip methods utilize excitation fluorescence signals for detection, potentially causing a non-negligible effect on cell activity. In this paper, we implement a nearly non-destructive single-cell dispensing method based on object detection algorithm. To realize single cell detection, we have built automated image acquisition system and then employed PP-YOLO neural network model as detection framework. Through architecture comparison and parameter optimization, we select ResNet-18vd as backbone for feature extraction. We train and evaluate the flow cell detection model on train and test set consisting of 4076 and 453 annotated images respectively. Experiments show that the model inference an image of 320 × 320 pixels at least 0.9 ms with the precision of 98.6% on a NVidia A100 GPU, achieving a good balance of detection speed and accuracy.

Keywords: single cell dispensing; label-free cell detection; PP-YOLO

1. Introduction

Recently, extensive efforts have been made to advance the development of methodologies for the

quantification of cellular features at single-cell resolution [1,2]. This greatly facilitates multiple fields of single-cell-based research, such as single-cell genomics, transcriptomics, proteomics and even drug evaluation [3,4]. It is reported that more than 50 monoclonal antibodies have been introduced to treat cancer based on new understanding of single-cell genetics and transcriptomics, generating nearly $100 billion in economic benefits annually [5]. Therefore, an automated single-cell dispensing method that obtain the desired pure and viable cells for high cloning efficiency is critical.

Several different approaches can be used to isolate cells from their populations, such as limiting dilution [6], FACS [7] and single-cell dispensing [8]. FACS method requires complex optical systems and fluorescent labeling, which usually leads to expensive manufacturing cost and inconvenient use. In practice, FACS uses fluorescent dyes to label cells and lasers to excite fluorescence, having an impact on their viability or proliferation. Limiting dilution requires complex and laborious operations, but is still troubled by cell recovery rates, cell clumps and the statistical analysis used to support the assurance of a clonally-derived cell line [6]. Recently, some scholars have employed convolution neural network (CNN) to imaging flow cytometry for cell classification [9,10]. Riba et al. [11] reported a trained CNN model deployed on a c.sight single-cell printer for real-time CHO-K1 cell sorting. Wang et al. [12] designed a microfluidic robot for rare cell sorting based on imaging identification and multi-step sorting strategy, achieving the performance of 4000 cells/s scanning speed and more than 90% sorting purity. It is not hard to see that using microfluidic chip to imaging cells in bright field to achieve single cell distribution can avoid fluorescence labeling and cell vitality damage. This gentle and label-free cell sorting method is important for subsequent single cell research.

Generally, some difficulties will be encountered in dealing with images captured from microfluidic chip, such as low lighting condition and complex image background caused by laser etching traces or residual substrate fragments inside microfluidic chip, cell overlap and artefact occlusion brought by flow cell or sample configuration, image blurring owing to out of focal or mismatch between camera exposure time and the cell flow speed. In this work, we try to dispense single cell by image recognition technology in a label-free manner. More specifically, we focus on the identification of single cells with low quality images brought by the microfluidic flow environment to solve which cells need to be distributed.

In this study, we employed object detection method to realize the single cell dispensing decision. As shown in Figure 1, the main technical process includes cell image from microfluidic channel, region of interest (ROI) preprocessing, single cell inference engineer based on PP-YOLO [13] and final sorting manipulation.

We summarize the main potential contributions of this work as follows:

(1) A nearly non-destructive single-cell dispensing method based on object detection algorithm was implemented to detect label-free cells. Isolation of single cells just by bright-field image can avoid the effects of fluorescent markers and laser radiation on cell viability and would help to provide expected samples for subsequent single cell culture or gene sequencing.

(2) A flow cell image acquisition system was built. It was configured with CMOS camera, adjustable optical magnification module, supplementary light sources and microfluidic chip with driver module.

(3) Flow cell object detection model based on PP-YOLO was trained and deployed. There existed some problems in flow single cell recognition, such as different size, irregular shape, mutual occlusion, stacking, jitter and low illuminance, which would affect single cell detection accuracy. We introduced PP-YOLO method to achieve the desired tradeoff between the accuracy performance and inference efficiency.

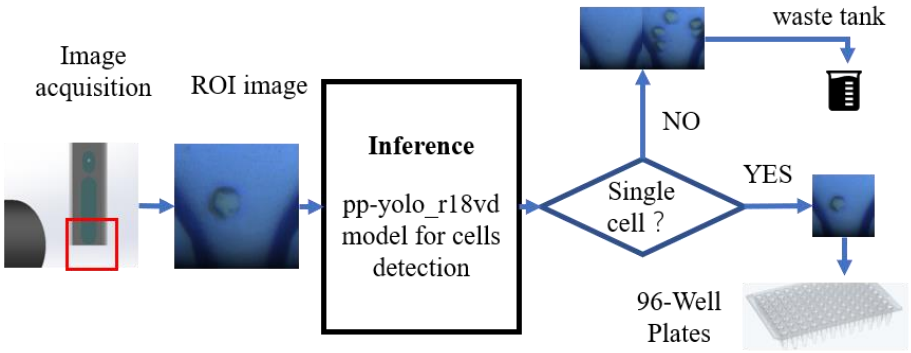

Figure 1. Flow cell image detection process.

2. Materials and methods

2.1. Sample preparation and digital image acquisition

2.1.1. Sample preparation

The sample used in our experiment was the suspension prepared by self-cultured smooth muscle cells and PBS buffer (PH 7.4), where the concentration was 8 × 105 cells/ml. Our microfluidic chip was customized design and processing, which used silicon chip as the base material, adopted the processing method of laser etching runner and made glass chip as the cover plate, as shown in Figure 1(c).

2.1.2. Image acquisition system

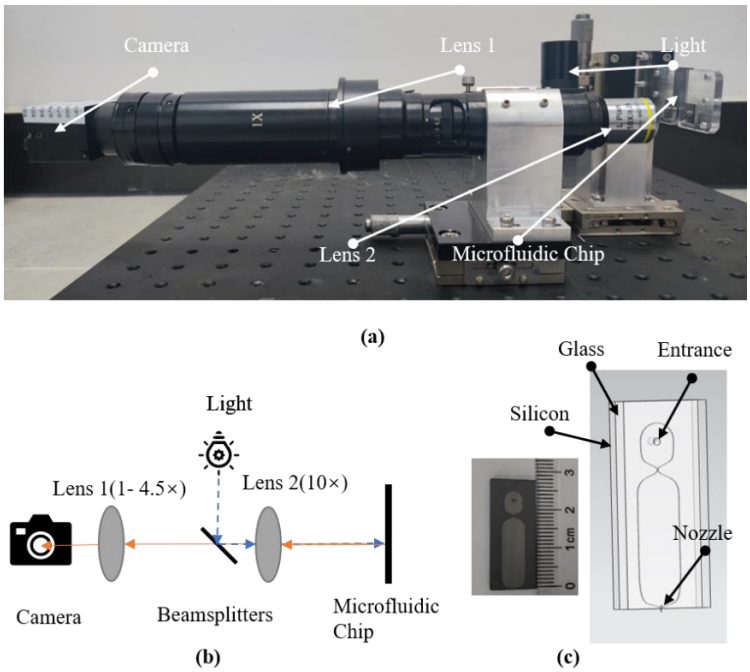

Figure 2. Image acquisition system. (a) Hardware components, (b) Imaging optical path, (c) Microfluidic chip.

The image acquisition system took CMOS camera (MV-SUF401GC-T-CL, MindVision, China) as the core device. It mainly contained the camera itself, illumination, 10 × Plan Apochromatic objective (0.25 N.A., 10.60 mm W.D., Olympus, Japan) and lens group with adjustable magnification (1~4.5 ×). The hardware components and principle are shown in Figure 2(a) and Figure 2(b) respectively.

Flow cell imaging optical path was demonstrated in Figure 2(b). The light from the light source horizontally went into the objective lens through the beam splitters, and the 10 × objective lens (Lens 2) converged the light to illuminate cells that flowed through the observation area of the microfluidic chip. Then, the reflected light entered into camera through the objective lens and adjustable magnification module in order to generate an image.

2.1.3. Region of interest

The ROI area was determined by the design of the microfluidic chip and the volume of droplets dispensed from a single squeeze. After several experiments, the volume of a droplet (V_{drop}) ejected from the microfluidic chip was about 60 pL. We assume that the observation area of the cell was rectangular. Given the width of the microfluidic nozzle of 40 μm and the depth (d_m) of 40 μm, we could calculate that the other side length of the imaging area is 39.11 μm. Considering the chip was squeezed to cause liquid shock, the side length of the square was enlarged to 110%, i.e., 43.0, and rounded it to 40 μm. According to the imaging final magnification (45 ×) and pixel size (5.5 × 5.5 μm^2), the image resolution could be calculated to be 320 × 320 pixels, shown in Figure 3. The mathematical calculation of this process followed the below formula.

$$H_{ROI} = 1.1\sqrt{(V_{drop}/d_m)}. \quad (1)$$

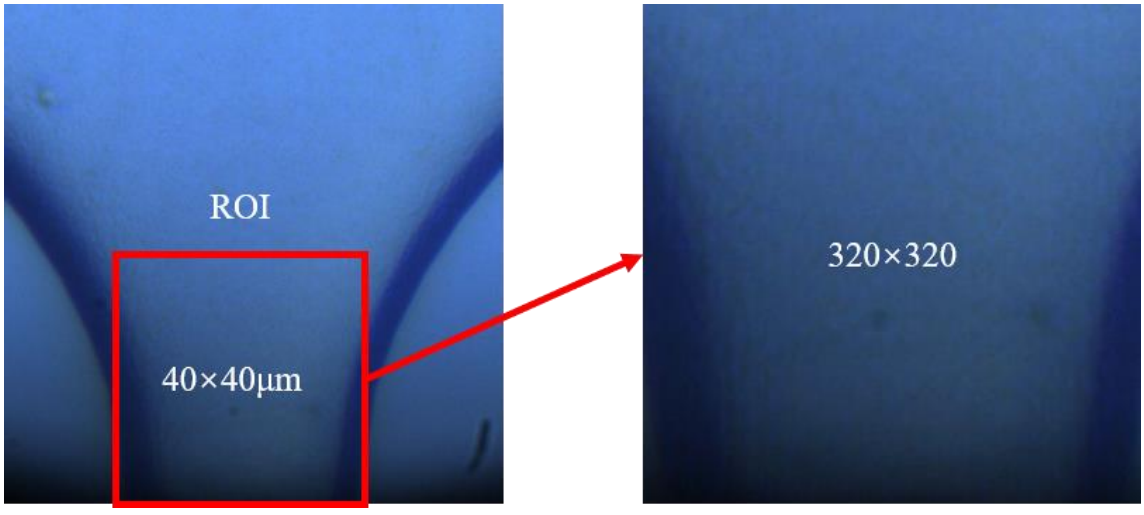

Figure 3. ROI area and image resolution.

2.1.4. Dataset

We took over 20,000 images and found the following problems. First, the exit of the microfluidic chip was funnel-shaped with the sides converging to the middle, and the laser etching marks made

the exit darker than other areas. For instance, Figure 4(a), (b) and (c) were ROI area, nozzle shift up 20 μm and 40 μm, respectively. Secondly, some cells flowing in the chip would stick together, causing images to appear with cells obscuring each other, incomplete cell outlines and missing edges, as shown in Figure 4(d). Finally, impurities, residual silicon slag from microfluidic chip processing and broken cell debris would interfere with the assay, as shown in Figure 4(e).

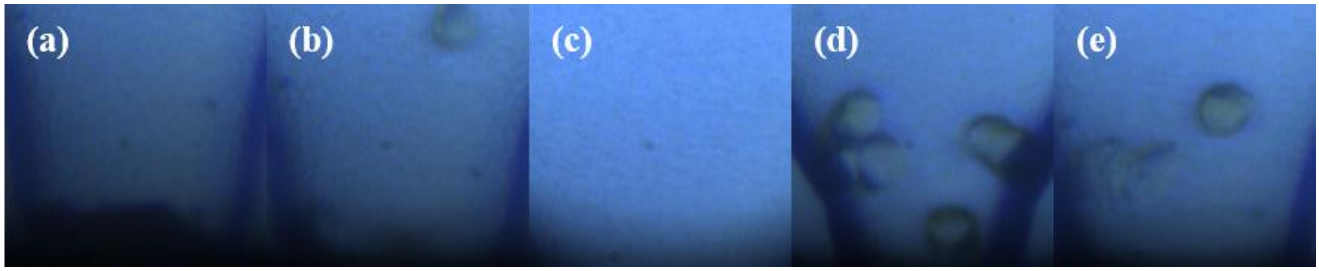

Figure 4. Dataset from different condition. (a) ROI area, (b) Nozzle shift up 20 μm, (c) Nozzle shift up 40 μm, (d) Cell stick, (e) Non-cell impurity.

We filtered and labeled 4529 images from over 20,000 images that were suitable for training (as shown in Figure 5). All images were RGB ones with resolution of 320 × 320 pixels. For the sake of diversity of the data set, we deliberately selected some chips with impurities or obvious processing traces for filming. Meanwhile, we added a small amount of 8 μm microspheres to the smooth muscle cell suspension, with the aim of enhancing the final robustness of the model.

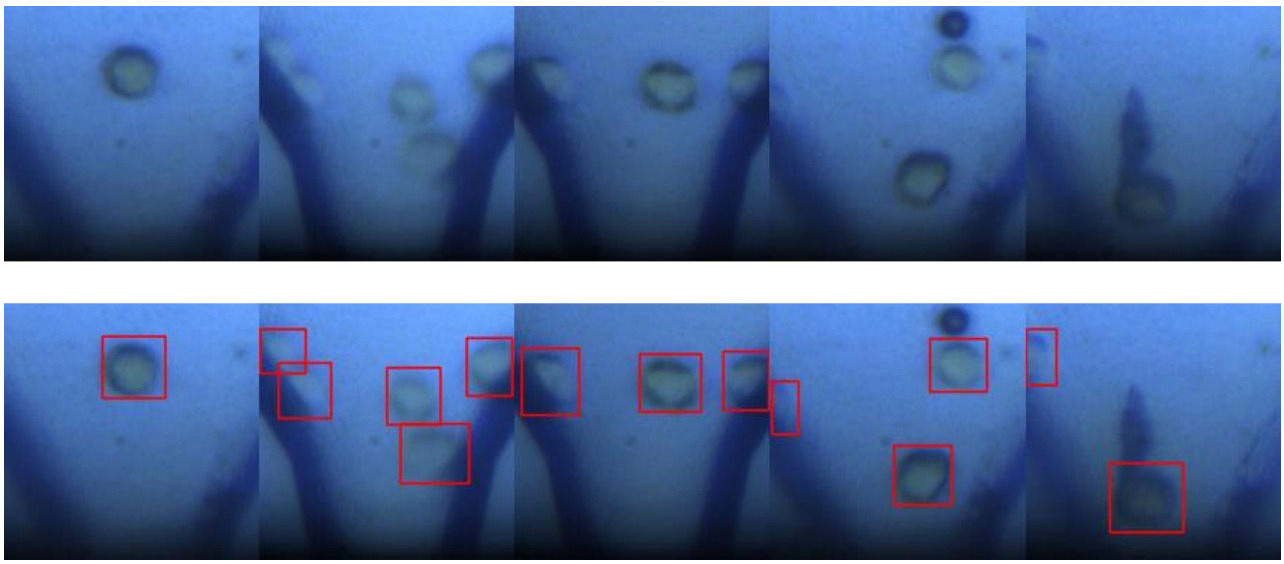

Figure 5. Image dataset marked by manual.

Among them, the training set and validation set (90%) were utilized to train model to determine the network parameters of the detection model, and the test set (10%) was used to test the recognition rate and generalization of the trained detection model (Table 1).

Table 1. The number of images of various data sets.

Training/Validation Data	Test Set	Total (Labeled Images)
4076 (90%)	453 (10%)	4529

2.2. Object detection model

PP-YOLO is a One-stage object detector and widely used in industrial applications due to its good balance of speed and accuracy. It can select feature extraction network flexibly according to the demand and realizes end-to-end model training. Therefore, the PP-YOLO network was employed as the basic framework in this experiment to construct an end-to-end cell detection model.

2.2.1. Feature extraction network

PP-YOLO is an object detection model proposed by Baidu team in 2020 based on YOLOv3 [14] that balances prediction speed and accuracy. It does not focus on the improvement of backbone or the exploration of network structure, but uses more tricks that are reasonable flexibly, achieving a performance beyond YOLOv3 and YOLOv4 [15]. PP-YOLO architecture was drawn in Figure 6.

A One-stage object detector is usually composed of a feature extraction network, a neck (such as feature pyramid networks) and a head (for classification and localization). ResNet50-vd-dcn was initially chosen as the backbone network to replace Darknet53 in the YOLOv3. However, we found that the cell detection task using ResNet50 was too complex, wasting computational resources and causing slow inference in our experiments.

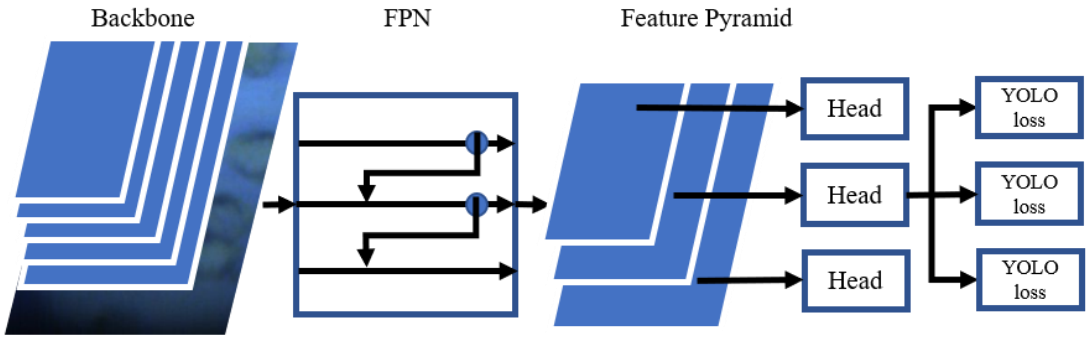

Figure 6. Architecture of PP-YOLO.

Officials have given multiple backbone versions of PP-YOLO, and in this study, we selected ResNet-18vd as backbone. ResNet_vd was first proposed in [16], and in the original ResNet [17] (Figure 7(a)), the first convolutional layer of Path A with stride = 2 caused 3/4 of the information to be lost. ResNet-vd swapped the stride of the first and second convolutional layer, as shown in Figure 7(b). In front of the convolution layer of Path B, a pooling layer (2 × 2, stride = 2) was added to advance the down sampling, which would avoid the loss of 3/4 information and reduces the number of parameters significantly.

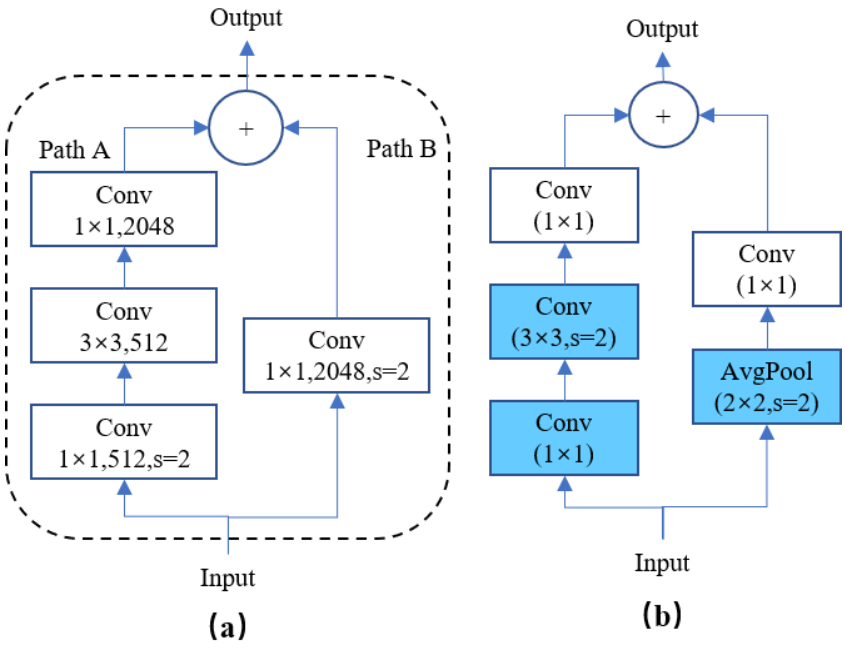

Figure 7. ResNet tweaks. (a) ResNet, (b) ResNet-vd.

2.2.2. Hyper parameters

Learning rate:

The model was iteratively optimized in the training set for 300 epochs, with batch size of 64 and initial learning rate of 0.02. The mathematical representation of this process was as follows:

$$lr_{new} = \frac{lr_{default} * (batch\_size_{new} * GPU\_num_{new})}{batch\_size_{default} * GPU\_num_{default}}. \quad (2)$$

The number of training GPU and the batch size were determined by the lab equipment. In addition, the best model for training was evaluated by average precision IoU = 0.5:0.95. The IoU value between the regression box and the real box was set as the background when it was less than 0.5. To highlight the effectiveness of PP-YOLO-r18vd, we trained several related models simultaneously.

Batch random resize:

Since "batch random resize" function was turned on during training, for each time a batch of training images would be resized to a random size in the "target_size list", followed the below formula:

$$target_{size} = 320 + n * 32,\ n \in Z\ on\ [-2, 9]. \quad (3)$$

Anchor:

In PaddleDetection, anchors could be obtained for our own dataset. Its principle was to use the k-means clustering method to learn different anchors from the training set, but in practice, we found the performance was not as desired in training. Therefore, according to the characteristics of our own data

set, we gave a set of customized parameters.

To sum up, the hyper parameters for PP-YOLO as listed in Tabel 2.

Table 2. Hyper parameters for PP-YOLO.

Hyper parameters	detail
Epoch	300
Batch_size	64
Learning_rate	0.02
Target_size list	256, 288, 320, 352, 384, 416
Anchors	[10, 14], [23, 27], [37, 58], [81, 82], [135, 169]

3. Experiment

3.1. Experimental environment

This experiment was conducted on the paddlepaddle deep learning framework, and we used PaddleDetection 2.4. Besides, NVidia A100, CUDA 11.2, TensorRT 8.2.1 and Ubuntu 18.04 operation system were also configured.

3.2. Experimental strategy

By analyzing the current task, we found that predicting the results did not require determining the exact anchor frame location for the cells, but only the number of cells within the picture. The output of non-maximum suppression (NMS) [18] was shaped as [N, 6], where N is the number of predictor boxes, and 6 is [*class_id*, *score*, x_1, y_1, x_2, y_2]. When the output is *bbox_num*, only the number of prediction boxes for each image is output. For example, if batch size is 2, the output is [N1, N2], which means the first image contains N1 prediction boxes, and the second image contains N2. The total number of prediction boxes is the same as the first dimension N of NMS output. In addition, the inference time did not include the cost of data reading and post-processing.

3.3. Experimental results and analysis

3.3.1. Precision

The mean average precision (mAP) is an indicator to evaluate the accuracy of the model, which is the sum of the average precision of all object. However, in our task, there is only one class of target class in this study. Hence, mAP is the AP of a single class. The training convergence curve of the model is shown in the Figure 8(a). It is easy to see the loss function converges quickly, which is attributed to the effectiveness of the ResNet structure. Meanwhile, we can get from Figure 8(b) that after 200 epochs, the bbox-map stabilizes.

In order to compare the inference performance of the model, we conducted the speed test (s-FP32, s-FP16) did contain or not contain the data reading and post-processing (NMS) cost respectively. All tests were accelerated by TensorRT.

As can be seen from Figure 9(a) and Table 3, the AP of PP-YOLO _r18vd was 0.986 and 0.938

when IoU = 0.5 and IoU = 0.75, respectively. It was the same as the other models in the control group.

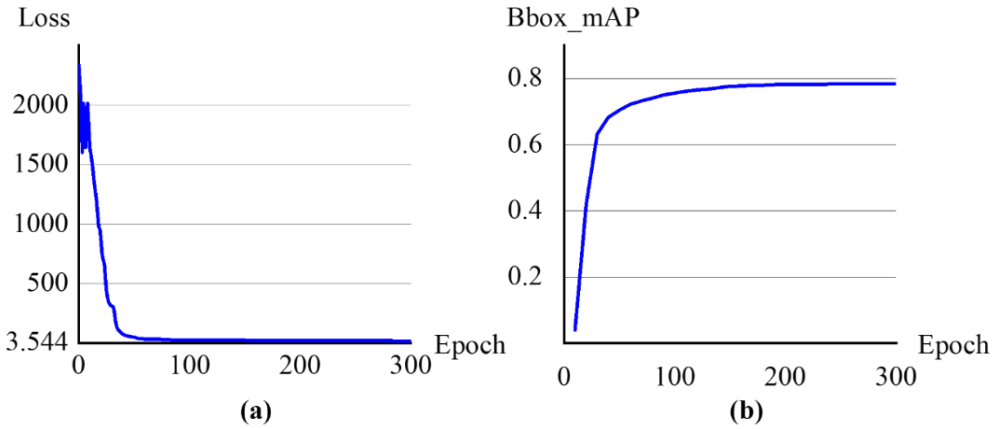

Figure 8. Training process. (a) Loss, (b) mAP.

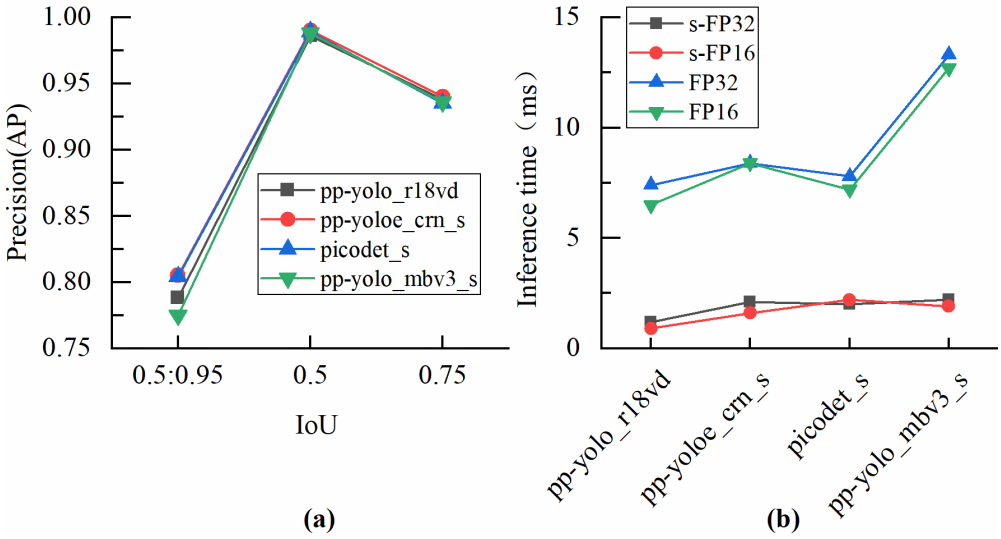

Figure 9. Model performance. (a) Precision, (b) Inference time.

Table 3. Precision.

Model	IoU(0.5:0.95)	IoU(0.5)	IoU(0.75)
PP-YOLO_r18vd	0.788	0.986	0.938
PP-YOLOE_crn_s [19]	0.805	0.990	0.940
PicoDet_s [20]	0.804	0.989	0.935
PP-YOLO_mbv3_s	0.775	0.988	0.936

3.3.2. Inference time

To compare the inference efficiency advantage of PP-YOLO_r18vd, we trained several models with different parameters for a deeper exploration of inference deployment. We tested the inference time of a single image on the exported model. As shown in Figure 9(b) and Table 4, we found PP-YOLO_r18vd object detection model was the best one, only costing 0.9 ms. With the help of TensorRT inference acceleration library and FP16 data quantification, the model could infer an image of 320×320 pixels at 6.5 ms. This means that, using PP-YOLO_r18vd to infer a picture and output the visualization results, the inference efficiency of 153 frame per second (FPS) can be achieved uing TensorRT in GPU computing mode.

Table 4. Inference time.

Model	No Preprocess + No Postprocess		Preprocess + Postprocess	
	FP32(ms)	FP16(ms)	FP32(ms)	FP16(ms)
PP-YOLO_r18vd	1.2	0.9	7.4	6.5
PP-YOLOE_crn_s	2.1	1.6	8.4	8.4
PicoDet_s	2	2.2	7.8	7.2
PP-YOLO_mbv3_s	22	1.9	13.3	12.7

3.3.3. Analysis

From Table 3, we can see that PP-YOLO_r18vd had a certain gap in accuracy compared with PP-YOLOE_crn_s and PicoDet_s. At IoU = 0.5:0.95, the accuracy is about 1.7% lower. In practice, we generally set the threshold value to 0.5, and then the detection accuracy of PP-YOLO_r18vd was about 1.4% lower.

When the slight difference of detection accuracy is acceptable, we paid more attention to whether the predicting speed of model meets our application requirements. PP-YOLO_r18vd had excellent performance in pure inference time without data reading and post-process cost. When the data type was set to FP16, it took 0.7 ms (+77.8%) less than PP-YOLOE and 1.3 ms (+144%) less than PicoDet. PP-YOLO also had advantages in measuring the total inference time. When the data type was set to FP16, it cost 1.9 ms (+29.2%) less than PP-YOLOE and 0.7 ms (+10.8%) less than PicoDet.

Due to the small difference in accuracy but a large speed advantage, PP-YOLO was selected as the detection model in this work. We tried to compare the mainstream feature extraction backbone between mobilenet_v3_small and resnet_18vd. Experimental results showed that resnet_18vd had an absolute advantage whether in accuracy or speed.

3.3.4. Cell detection result

Figure10 shows some inference results of PP-YOLO_r18vd for our flow single cell detection task. It can be found that the model was robust enough to deal with flow cell recognition in microfluidic scenarios, such as complex background, blurred target boundary, mutual stacking and large brightness change.

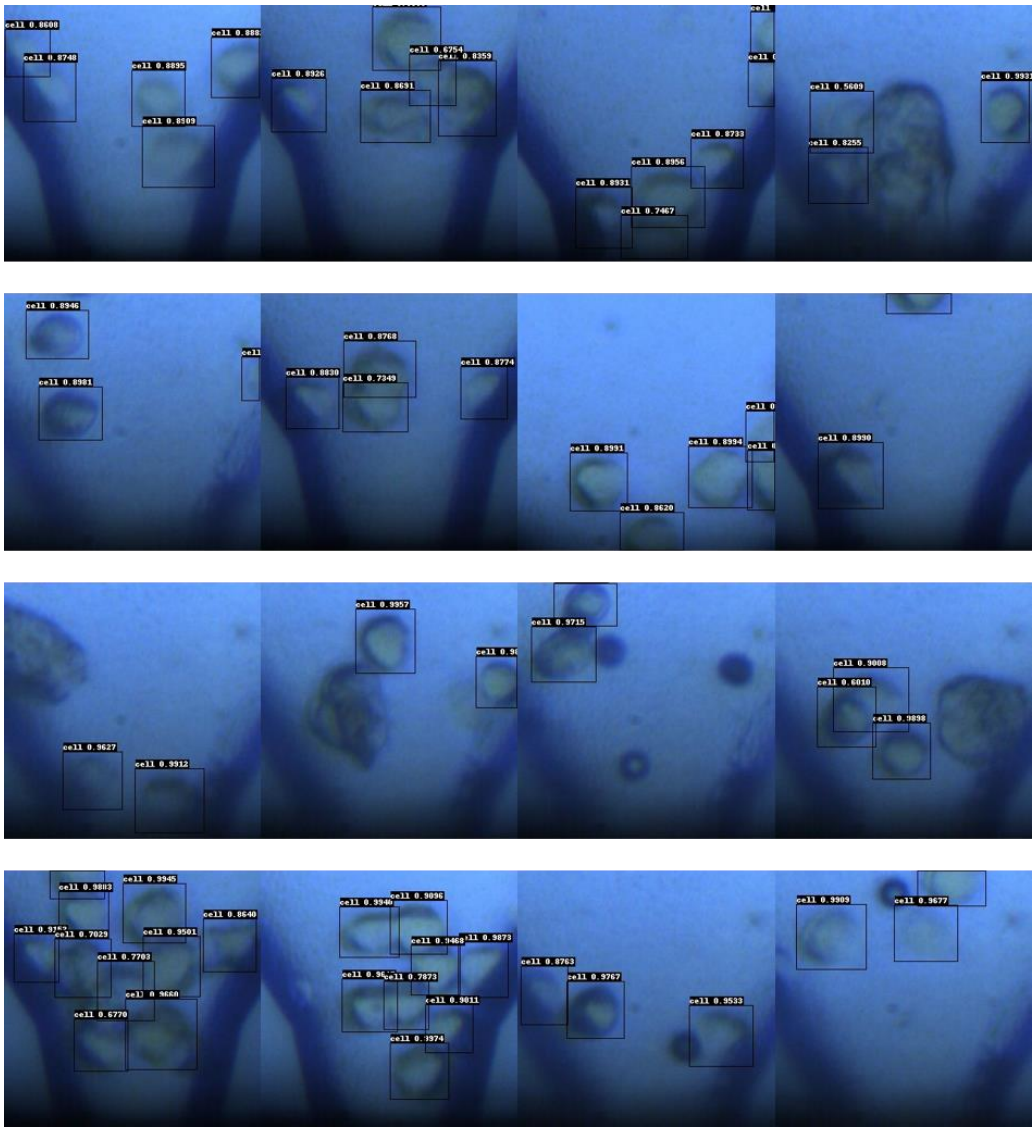

Figure 10. Detection results.

4. Conclusions

This study implements a machine vision based single-cell partitioning method that enables end-to-end real-time detection separation. Experiments show that PP-YOLO_r18vd has relatively good time performance in flow cell detection task, and its detection accuracy can meet our requirements.

Due to experimental conditions, the dataset is not enough in cell types, so that the model may not generalize to all cell assignments. Aim to this problem, if more cell types are added on the basis of existing dataset, the more robustness of the model will be improved. In terms of inference speed, we believe that there is still a lot of room for improvement. In the future, we may further reduce the size of the feature extraction network, remove unnecessary preprocessing, and remove post-processing. As well, we can try to convert the current RGB image to grayscale and reduce the size even further.

Acknowledgments

This work was supported in part by the National Key Research and Development Program of

China under Grants 2021YFF0704800, and in part by Jihua Laboratory Program under Grants X190141TD190.

Conflict of interest

The authors declare that there is no conflict of interest.

References

1. X. Zhang, X. Wei, Y. J. Wei, M. L. Chen, J. H. Wang, The up-to-date strategies for the isolation and manipulation of single cells, *Talanta*, **218** (2020), 121147. https://doi.org/10.1016/j.talanta.2020.121147
2. P. Hu, W. H. Zhang, H. B. Xin, G. Deng, Single cell isolation and analysis, *Front. Cell Dev. Biol.*, **4** (2016), 116. https://doi.org/10.3389/fcell.2016.00116
3. M. H. Spitzer, G. P. Nolan, Mass cytometry: Singlecells, many features, *Cell*, **165** (2016), 780–791. https://doi.org/10.1016/j.cell.2016.04.019
4. X. Wei, Y. Lu, X. Zhang, M. L. Chen, J. H. Wang, Recent advances in single-cell ultra-trace analysis, *Trends Anal. Chem.*, **127** (2020), 115886. https://doi.org/10.1016/j.trac.2020.115886
5. A. L. Grilo, A. Mantalaris, The increasingly human and profitable monoclonal antibody market, *Trends Biotechnol.*, **37** (2019), 9–16. https://doi.org/10.1016/j.tibtech.2018.05.014
6. D. Shaw, M. Yim, J. Tsukuda, J. C. Joly, A. Lin, B. Snedecor, et al., Development and characterization of an automated imaging workflow to generate clonally-derived cell lines for therapeutic proteins, *Biotechnol. Prog.*, **34** (2018), 584–592. https://doi.org/10.1002/btpr.2561
7. J. Fieder, P. Schulz, I. Gorr, H. Bradl, T. Wenger, A single-step FACS sorting strategy in conjunction with fluorescent vital dye imaging efficiently assures clonality of biopharmaceutical production cell lines, *Biotechnol. J.*, **12** (2017). https://doi.org/10.1002/biot.201700002
8. M. Yim, D. Shaw, Achieving greater efficiency and higher confidence in single-cell cloning by combining cell printing and plate imaging technologies, *Biotechnol. Prog.*, **34** (2018), 1454–1459. https://doi.org/10.1002/btpr.2698
9. N. Nitta, T. Sugimura, A. Isozaki, H. Mikami, K. Hiraki, S. Sakuma, et al., Intelligent image-activated cell sorting, *Cell*, **175** (2018), 266–276. https://doi.org/10.1016/j.cell.2018.08.028
10. A. Isozaki, H. Mikami, H Tezuka, H. Matsumura, K. Huang, M. Akamine, et al., Intelligent imag e-activated cell sorting 2.0, *Lab Chip*, **20** (2020), 2263–2273. https://doi.org/10.1039/d0lc00080a
11. J. Riba, J. Schoendube, S. Zimmermann, P. Koltay, R. Zengerle, Single-cell dispensing and "real-time" cell classification using convolutional neural networks for higher efficiency in single-cell cloning, *Sci. Rep.*, **10** (2020), 1193. https://doi.org/10.1038/s41598-020-57900-3
12. Y. Wang, D. F. Wang, H. F. Wang, J. W. Wang, J. Z. Pan, X. G. Guo, et al., A microfluidic robot for rare cell sorting based on machine vision identification and multi-step sorting strategy, *Talanta*, **226** (2021), 122136. https://doi.org/10.1016/j.talanta.2021.122136
13. X. Long, K. P. Deng, G. Z. Wang, Y. Zhang, Q. Q. Dang, Y. Gao, et al., PP-YOLO: An effective and efficient implementation of object detector, preprint, arXiv:2007.12099.
14. J. Redmon, A. Farhadi, YOLOv3: An incremental improvement, preprint, arXiv:1804.02767.
15. A. Bochkovskiy, C. Y. Wang, H. Liao, YOLOv4: Optimal speed and accuracy of object detection, preprint, arXiv:2004.10934.

16. T. He, Z. Zhang, H. Zhang, Z. Y. Zhang, J. Y. Xie, M. Li, Bag of tricks for image classification with convolutional neural networks, in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. https://doi.org/10.1109/CVPR.2019.00065
17. K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. https://doi.org/10.1109/CVPR.2016.90
18. X. L. Wang, R. F. Zhang, T. Kong, L. Li, C. H. Shen, SOLOv2: Dynamic and fast instance segmentation, in *Advances in Neural Information Processing Systems 33*, NeurIPS, (2020), 17721–17732.
19. S. L. Xu, X. X. Wang, W. Y. Lv, Q. Y. Chang, C. Cui, K. P. Deng, PP-YOLOE: An evolved version of YOLO, preprint, arXiv:2203.16250.
20. G. H. Yu, Q. Y. Chang, W. Y. Lv, C. Xu, C. Cui, W. Ji, et al., PP-PicoDet: A better real-time object detector on mobile devices, preprint, arXiv:2111.00902.

AIMS Press