CYOLOv8: Improved YOLOv8 for Real-time Detection of Circulating Tumor Cells and Cancer Associated Fibroblasts

Wang Xiandong, Ma Xin

Abstract—**Real-time detection of circulating tumor cells (CTCs) and cancer associated fibroblasts (CAFs) in microscopic images is crucial for determining prognosis, monitoring disease progression, and evaluating treatment effectiveness. While there have been significant strides in deep learning-based While there have been significant strides in deep learning-based cell detection methods, accurately identifying target cells remains a challenging task due to the high density of CTCs and CAFs at intersections in a large number of images. Our study addresses this issue by proposing CYOLOv8, a real-time method for detecting CTCs and CAFs improved YOLOv8. Firstly, the YOLOv8 backbone network was used for the detection of CTCs and CAFs. YOLOv8 backbone network's ability to extract both local and global features from cellular images is enhanced by adding the BOT module, which combines CNN and Transformer benefits. Additionally, by replacing the C2f module with the C2f-SCConv module, the YOLOv8 neck network can maintain its effectiveness. Finally, the EMA attention module is incorporated after the BOT module and the C2f-SCConv module. the C2f-SCConv module, thereby ensnaring the network to focus more on the target feature information and eliminate the hindrance of extraneous information. By conducting experiments on the CAC dataset, the mAP50 and mAP50-95 of CYOLOv8 on the validation set reached an impressive 97.3% and 79.8%, respectively. These results are an improvement of 1.1 and 4.2 percentage points compared to the unaltered YOLOv8 model while reducing the number of parameters and computation by 6% and 3.7%, respectively. The method's validity is confirmed by comparing it with other well-known techniques, indicating that it is an effective means of detecting CTC and CAF in medical images.**

Index Terms*—*Cell detection, Deep learning, YOLOv8, Spatial and Channel Reconstruction Convolution, Muti-Scale Attention.

I. INTRODUCTION

Approximately 90 percent of all cancer-related deaths are due to the process of metastatic spread[1]-[2]. During metastasis, tumor cells detach from the primary tumor and migrate into the vasculature, and these circulating tumor cells (CTCs) spread through the circulatory system to distant organs. In addition to CTCs, circulating cells in the tumor microenvironment also play an important role in metastasis. Cancer associated fibroblasts (CAFs) are a class of active fibroblasts with heterogeneity and plasticity in the tumor microenvironment[3]-[4]. Studies have shown that CAFs can circulate with tumor cells to support tumor cell survival and metastasis formation, which also suggests that CTCs and CAFs may become important biomarkers that contribute to cancer diagnosis and prognosis. Therefore, real-time detection of CTC and CAF in microscopic images is important in determining prognosis, monitoring disease progression, and evaluating treatment efficacy[5].

Traditional detection methods mainly rely on manual detection, which is time-consuming and subjective, while the emergence of deep learning has breathed new life into the field of biomedical images and changed the traditional methods of analyzing biomedical images. Deep learning can be used to detect CTCs and CAFs by analyzing fluorescent images of blood samples enriched with circulating tumor cells. The main advantage of deep learning algorithms over traditional machine learning algorithms is that they reduce the reliance on task-relevant features designed based on expert knowledge and improve feature representation capabilities through end-to-end learning.

Researchers[6]-[7] used convolutional neural networks to identify circulating tumor cells in fluorescence images and microscopy images, respectively. The literature[8] used deep learning techniques, specifically auto-coding convolutional neural networks, to analyze CTCs in the blood of cancer patients, resulting in accurate classification and prediction of patient prognosis. The above articles are all based on deep learning to classify CTC images, while there are currently few studies on the detection and localization of CTCs and CAFs in images based on deep learning. The literature[9] used RetinaNet[10] and Faster R-CNN[11] to detect CTCs and CAFs in images respectively. The detection of CTC is not bad, but the detection effect of CAF still has much room for improvement. Since cell images may contain both CTCs and CAFs, two different algorithms are needed to detect CTCs and CAFs respectively, which lacks uniformity.

The current deep learning-based target detection frameworks are divided into two major schools of thought: one-stage methods and two-stage methods. One-stage methods prioritize inference speed, which utilizes regression analysis ideas, omit the candidate box region generation stage, and directly acquire target categories and information, and the classic models include YOLO series[12], RetinaNet, etc. Two-stage methods prioritize accuracy, first generating candidate regions (region proposals) through an independent step, and then performing classification and location localization based on these candidate regions, classic models include Faster R-CNN and Mask R-CNN[13], etc. The two-stage approach is not optimal for cellular detection

Manuscript received October 16, 2023
Wang Xiandong, School of Computer Science and Technology, Tiangong University, Tianjin, China
Ma Xin, School of Computer Science and Technology, Tiangong University, Tianjin, China

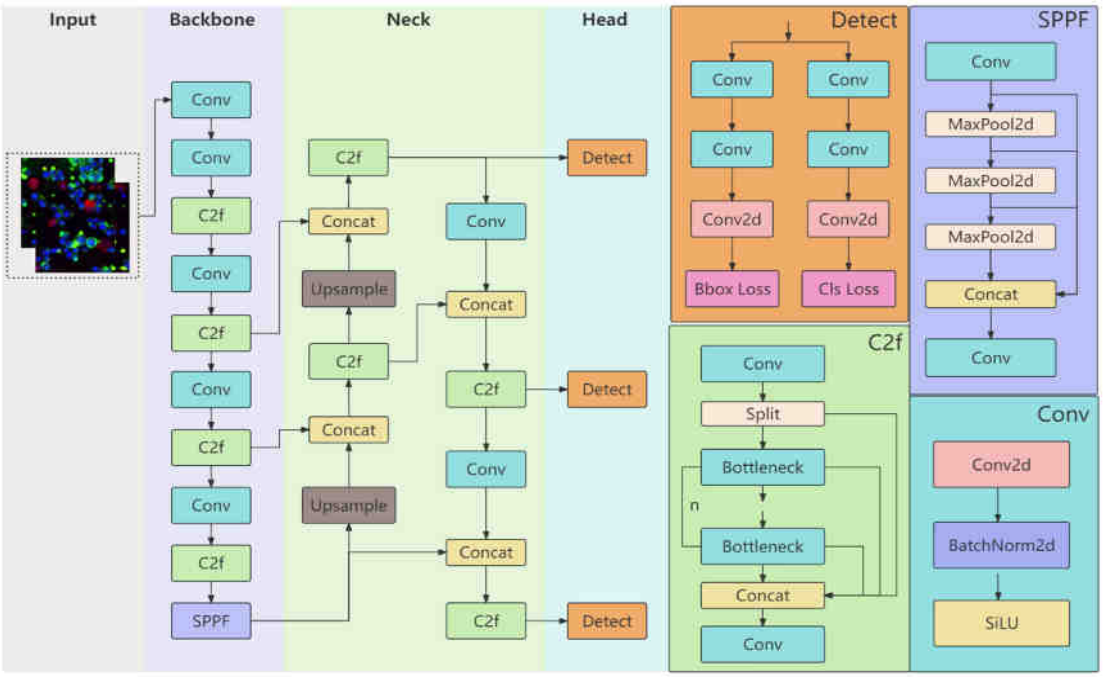


Fig. 1. YOLOv8 network structure

because it is more computationally intensive with higher accuracy and longer model training and inference time.YOLOv8[14] is the latest model of the YOLO series, which has achieved a good balance between detection accuracy and inference speed, and is superior to other detection models. Therefore, this study chose YOLOv8 as the improvement target and proposed CYOLOv8 to detect CTCs and CAFs in cell images, and achieved good results. The Contribution points of this paper are as follows:

1) Introduction of Bottleneck Transformer[15], the BoT module is designed to preserve the local and global features in the image and improve the detection of crossed and overlapped cells in the image.
2) Add Efficient Muti-Scale Attention Module(EMA)[16] to enhance local features and ignore interference from irrelevant cell information, so that the fused feature map contains more effective information and improves the detection accuracy of the algorithm.
3) In the YOLOv8 neck network, the C2f module is replaced by the C2f-SCConv module[17], which allows the model to reduce model parameters without losing performance and speed up feature fusion.

II. METHODS

A. *YOLOv8*

The YOLO algorithm was proposed by Redmon in 2016[12]. Its main idea is to segment the image into multiple grid cells, predict the bounding box within each grid cell and the class of objects it contains, and eliminate overlapping bounding boxes using a non-maximal suppression (NMS) algorithm. YOLOv1 is relatively fast, but it can only detect up to two objects of the same class in a grid cell, limiting the ability to predict nearby objects. YOLOv2[18] uses Darknet-19 as a feature extraction network, uses Batch Normalization to help the network converge, and enhances the network's prediction ability by using anchor boxes to predict bounding boxes. YOLOv3[19] added Feature Pyramid(FPN) and Spatial Pyramid Pooling(SPP) modules to enhance the multi-scale detection capability and the processing capability of feature maps of different scales and improve the detection effect of small targets. YOLOv4[20] introduced the Mish activation function to improve the model detection. YOLOv5[21] Introduce the C3 module and SPPF module to improve feature perturbation, accelerate network computation, and improve detection capability. YOLOv7[22] introduced some strategies such as E-ELAN (Extended efficient layer aggregation network), tandem model scaling, and reparameterised convolution, which enhances the learning ability of the network and the expression ability of semantic information, and further optimizes the target detection effect.

In January 2023, Ultralytics, the authors of YOLOv5, released YOLOv8[14], the latest version of the YOLO series. The most important features of YOLOv8 compared to YOLOv5 are the use of a new backbone network, a new decoupling header and new loss functions, and the abandonment of the anchor box, which is more in line with the design philosophy of advanced detection frameworks. YOLOv8 enables real-time target detection through a single forward propagation process, which provides a good balance of speed and accuracy and is ideal for cellular detection. The YOLOv8 network structure is divided into three parts: Backbone network, Neck network, and Detect Head, which is the mainstream architecture of today's mainstream target detection models, where the Backbone is used to extract features, Neck further fuses the extracted features, and the Head makes the final prediction based on the features of different scales fused by Neck. Fig. 1 shows the detailed architecture of YOLOv8.

The Backbone of YOLOv8 adopts the CSPDarnet-53 structure, which consists of a series of Conv blocks and C2f modules, the Conv block consists of standard convolution, Batch Normalization, and SiLU activation function for extracting the image features. The C2f module consists of 2 Conv blocks and several Bottleneck structures and contains

jump-join and split operations to reduce memory and computational cost, C2f combines high-level features with contextual information to improve detection accuracy[23]. At the top of Backbone, YOLOv8 uses the SPPF structure to pool the features of different scales mapped to a fixed-size feature map to speed up the computation of the network. The Neck mainly includes some C2f modules and Conv blocks, and through upsample and concatenation operations, the feature information between different feature layers is fused and output to the Detect Head to predict results. Neck outputs three different sizes of feature maps to Detect Head, which are used to predict large, medium, and small targets respectively. In the Head, YOLOv8 uses an Anchor-free model with a Decoupled Head to handle the classification and regression tasks, outputting Bbox prediction and Cls prediction. For the design of the loss functions, YOLOv8 uses the CIoU and DFL loss functions to represent the Bbox regression loss and the BCE to represent the Cls classification loss, which improve the target detection performance, especially when dealing with smaller objects.

The network model of YOLOv8 consists of five versions (n, s, m, l, x), and the complexity of the five versions of the model increases sequentially, with more complexity representing a higher number of parameters as well as computational effort, and the difficulty of training increases sequentially, but at the same time, higher accuracy can be obtained. In this study, we use the YOLOv8n model, which obtains the best balance between speed and accuracy and improves it to detect CTCs and CAFs.

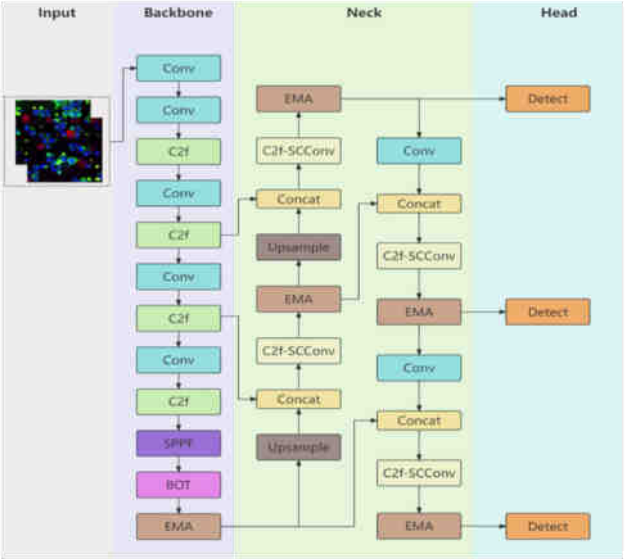


Fig. 2. CYOLOv8 network structure

B. *CYOLOv8*

To further balance the speed and accuracy of cell detection, and to make YOLOv8n more lightweight without loss of accuracy, an algorithmic model CYOLOv8 is proposed in this experiment to improve YOLOv8n, and its network architecture is shown in Fig. 2. The BoT module, which integrates CNN and Transformer, is added after the SPPF module in the backbone part, and the EMA attention module is added after the BoT module. In the neck of YOLOv8, the original C2f module is replaced with the C2f-SCConv module, and an EMA module is added after each C2f-SCConv module. The BoT module enhances the processing capability of feature information, the EMA module aims to enhance the feature extraction capability by highlighting the key information of the cells, and the C2f-SCConv module is useful in reducing the number of parameters and computation of the model while enhancing the detection effect.

1) *BoT Module*

CNN extracts features by sharing convolutional kernels, which reduces the number of parameters and improves the efficiency of the model, CNN also has translation invariance, which means that the network can detect features no matter where they have been moved to in the image. However, the sensory field of CNN is small and is not good for capturing global features. The emergence of the Vision Transformer[23] brought a great change in the field of computer vision. ViT can capture the global information of an image and establish contextual relationships, which can outperform CNNs in many visual tasks, but the Transformer lacks the translation invariance and local correlation of CNNs, which leads to the need for a large amount of data to outperform the CNNs when training Visual Transformer. After the emergence of visual Transformer, many excellent algorithms have combined CNN and Transformer to reduce the computational effort while preserving the local and global features of the image.

BoTNet[15] is a structure that combines the respective strengths of CNNs and Transformers, and by using this structure, the accuracy and efficiency of the model can be steadily improved. BoTNet does this by using Multiple Headed Self Attention (MHSA) instead of ResNet Bottleneck[24] the 3×3 convolution in ResNet to form a new network structure called Bottleneck Transformer(BoT), which improves the performance in tasks such as classification, and target detection, compared to pure CNN's such as ResNet.

In this study, by combining the C2f module with the BoT to form a new BoT module, the overall accuracy of the model is improved at the cost of only a slight amount of computation. Due to the large number of CTCs in the image and the existence of a large number of overlapping and intersecting situations, there may be missed detections, and the detection of CTCs can be improved by adding the BoT module at the end of the trunk of YOLOv8.The structure of the BoT and the BoT module is shown in Fig. 3.

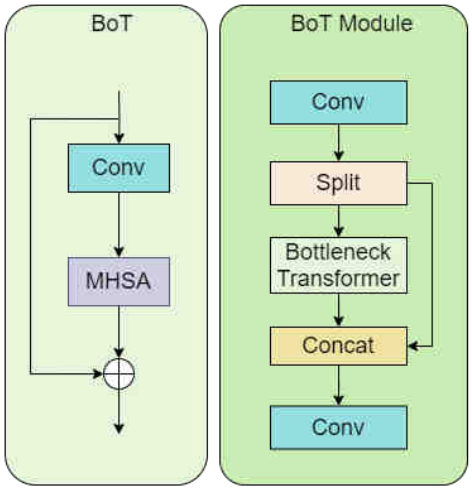


Fig. 3. BoT and BoT module structure

2) EMA Module

Since the image also contains cells other than CTCs and CAFs, to avoid the interference of these irrelevant cells and make the network focus on CTCs and CAFs, the EMA module is added for local feature enhancement between every two feature fusions at the end of the backbone and the neck of YOLOv8, so that the network ignores the interference of irrelevant information, and so that the feature maps after the fusion contain more valid information.

EMA is an efficient multi-scale attention module, it aims to preserve the information on each channel and reduce the computational overhead by reshaping some of the channels into batch dimensions and grouping the channel dimensions into multiple sub-features so that the spatial semantic features are uniformly distributed in each feature set[16]. The structure is shown in Fig. 4, where the EMA divides the input X into G sub-features across the channel dimensions for learning different semantics, and then extracts the attention weights of the grouped feature maps via three parallel paths. Two of the parallel paths use 1 × 1 convolution to encode the channels along the spatial horizontal and vertical directions, respectively, and finally converge to maintain the long-range dependency between the two directions and embed the precise location information into the EMA. The third path uses only 3 × 3 convolution to capture multi-scale features and preserve spatial information. To allow the channels and spatial information the ability to establish interdependencies, the EMA finally achieves richer feature aggregation in different spatial dimensional directions through a cross-spatial learning approach.

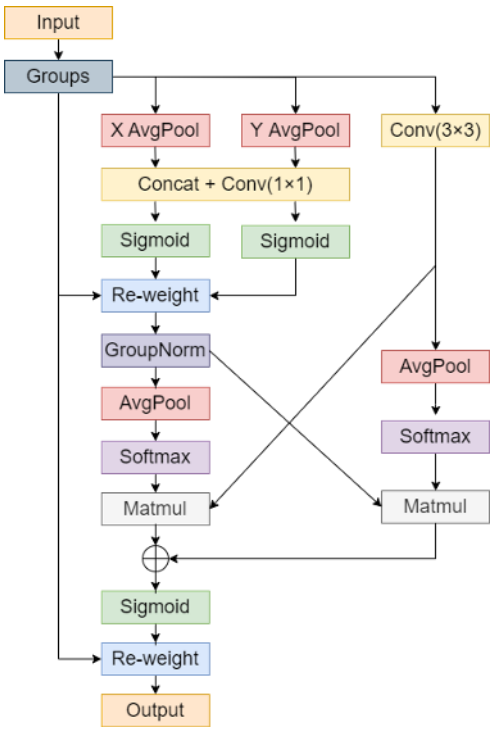


Fig. 4. EMA structure

3) C2f-SCConv Module

Standard convolution suffers from feature redundancy and consumes computational resources when extracting features, and most of the work addresses this problem using model compression strategies and careful design of the network structure. Li[17] proposed a novel CNN compression method to jointly reduce the spatial and channel redundancy in the convolutional layer, called Spatial and Channel Reconstruction Convolution(SCConv), which aims to reduce the number of parameters and computation without loss of performance. The number of parameters in SCConv is about 1/5 of the standard convolution, but better performance than the standard convolution is obtained. The structure of SCConv is shown in Fig. 5, and it can be seen that SCConv consists of two structural units, the Spatial Reconstruction Unit (SRU) and the Channel Reconstruction Unit (CRU). The input feature X is mapped to the spatial reconstruction feature Xw after passing through the SRU, and Xw is mapped to the spatial reconstruction feature *Y* after passing through the CRU, and *Y* is the standard output of SCConv.

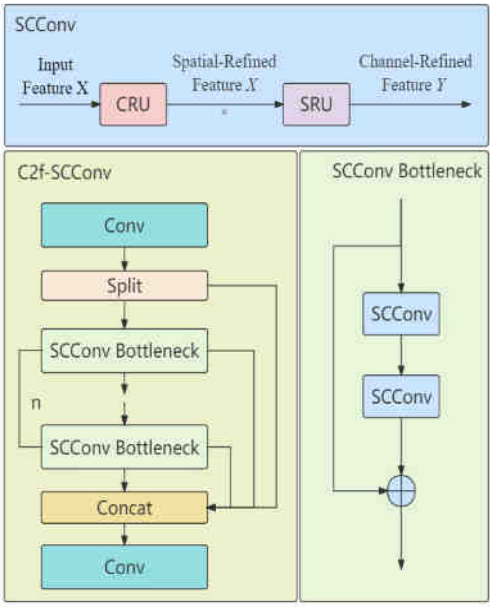


Fig. 5. The structure of SCConv, SCConv Bottleneck and C2f-SCConv

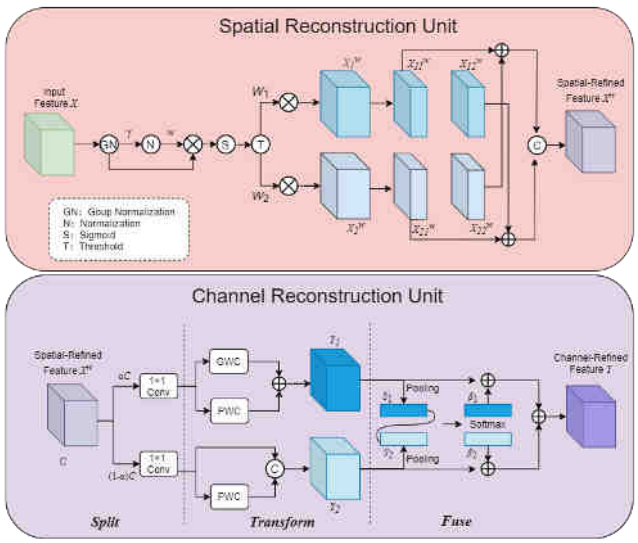


Fig. 6. SRU and CRU structure

The detailed structure of SRU and CRU is shown in Fig. 6, SRU adopts the separation reconstruction method to suppress the spatial redundancy, the input feature *X* first goes through the scaling factor in Group Norm to evaluate the information content in different feature maps and outputs γ. The richer the spatial information is, the larger the γ is and

obtains the normalized weight Wγ, and then maps the weights to the range of (0,1) through the Sigmoid function, and then a threshold is set to distinguish the acquired information weights, those above the threshold are called W_1 and those below the threshold are W_2, the process of acquiring W can be expressed by (1).

$$W = Gate(Sigmoid(W_\gamma(GN(X)))) \tag{1}$$

Then the input features X *are* multiplied by W_1 and W_2, respectively, to produce 2 weighted features: information-rich X_1^w, information-poor X_2^w, and X_2^w is considered redundant, to reduce spatial redundancy, SCConv uses a cross reconstruction operation to weightedly sum the information-rich features with the information-poor ones and to enhance the flow of information between them as shown in (2).

$$\begin{cases} X_1^w = W_1 \otimes X, \\ X_2^w = W_2 \otimes X, \\ X_{11}^w \oplus X_{22}^w = X^{w1}, \\ X_{21}^w \oplus X_{12}^w = X^{w2}, \\ X^{w1} \cup X^{w2} = X^w. \end{cases} \tag{2}$$

Xw is the output of SRU, where $\otimes$ is the element-by-element multiplication, $\oplus$ is element-by-element summation, and $\cup$ is concatenation. After applying SRU to the intermediate input features X, it not only separates the information-rich features from the less informative ones but also reconstructs them to enhance the representative features and suppress the redundant features in the spatial dimension. However, the spatially refined feature map Xw remains redundant in the channel dimension[17].

CRU then eliminates these channel redundancy features via three operators - Spit, Transform, and Fuse. In the Spit stage, X_w is divided into αX_w and $(1 - \alpha)X_w$ using the splitting ratio α ($0 < \alpha < 1$), and then the channels of the feature map are compressed using 1×1 convolution to obtain spatially refined features X_{up} and X_{low}. In the Transform stage, X_{up} is sent to the upper transformation stage, which serves as a "rich feature extractor". In the Transform stage, X_{up} is fed into the upper transform stage to act as a “rich feature extractor”, which extracts the rich representative features Y_1 on the same feature map X by using a kind of efficient convolution operations GWC and PWC respectively. X_{low} is fed into the lower transform stage, which generates the shallow hidden features by using the PWC operation and then stitches the shallow features with X_{low} to obtain the features Y_2 with supplementary detailed information. In the Fuse stage, Y_1, Y_2 are used to collect the global spatial information S_1, S_2 through global average pooling, and then S_1, S_2 are stacked together to obtain the feature importance vectors β_1 and β_2, and the final output Y can be expressed as(3).

$$Y = \beta_1 Y_1 + \beta_2 Y_2 \tag{3}$$

We redesigned Bottleneck by replacing the 3×3 convolution in the original Bottleneck with SCConv, called SCConv Bottleneck, and then replacing Bottleneck in the C2f module with SCConv Bottleneck, called C2f-SCConv, SCConv Bottleneck, and the structure of C2f-SCConv are shown in Fig. 5. The addition of SCConv can speed up the comprehension of the model, and at the same time effectively reduce the computational and parametric quantities of the original C2f with almost no degradation in accuracy.

III. Experiments And Results

A. Dataset

The images of CTCs and CAFs used in this work are from the open-source CAC dataset[9] The dataset contains a total of 1100 images of size 1000×1000, and each image corresponds to two voc format annotation files, CTC and CAF, respectively. A Python script is written to convert the 2200 voc annotation files into yolo format annotation files and merge the CTC and CAF annotation files together to facilitate the simultaneous detection of CTCs and CAFs in an image. Therefore, the final dataset contains 1100 images and 1100 corresponding yolo format annotation files. The dataset is divided into training and validation sets in the ratio of 8:2.

B. Experimental environment and metric

The development environment for the experiment is Python 3.9, PyTorch 1.13, Windows 11, the acceleration environment is CUDA 11.7, the CPU used is Intel® Xeon® Silver 4214R Processor @ 2.40 GHz, and the GPU is RTX 3080 Ti 12GB.

The evaluation metrics used in this experiment include precision (P), recall (R), mean average precision (mAP), parameter counts (Params), and floating point operations per second (FLOPs). The accuracy of the model is measured by the precision rate P and R as the basic metrics, and the mAP is calculated from the precision rate P and R as the final evaluation metrics. The P represents the ratio of correct predictions among all the positively predicted samples, the R represents the ratio of correctly predicted positive samples to the total number of positive samples, and the mAP represents the average accuracy rate of n categories. Among them, mAP is the most commonly used evaluation index in the field of target detection, and the higher mAP represents the better detection performance of the model. P, R, and mAP are calculated as shown in (4).

$$\begin{aligned} P &= \frac{TP}{TP + FP} \\ R &= \frac{TP}{TP + FN} \\ mAP &= \frac{1}{n}\sum_{i=1}^{n}\int_0^1 P(R)d(R) \end{aligned} \tag{4}$$

Where TP denotes the total number of correctly detected targets, FP denotes the total number of incorrectly detected targets, FN denotes the total number of non-detected targets, and n denotes the number of categories. FLOPs are used to measure the complexity of the model or algorithm, while Params denotes the number of parameters of the model. In general, the smaller the Params and FLOPs, the less

computationally intensive the model is and the lower the performance requirements for the hardware.

C. *Experiments and Results*

During training, the image size was uniformly adjusted to 640×640, the batch size was 32, training epochs were 300, the initial learning rate was set to 0.01, AdamW was used as the optimizer, and to prevent the model from overfitting, if the model did not have any performance improvement within 50 epochs, the training was terminated early using the early-stop strategy mentioning the early stopping strategy, and the other hyper-parameter settings and data enhancement strategies were all adopted YOLOv8 default settings and enhancement strategies.

Under the same training environment, YOLOv8 and CYOLOv8 are trained on the CAC dataset to obtain the weights with the best training effect, respectively, and then their respective performances are tested on the validation set, and the results show that the mAP50 and mAP50-95 of CYOLOv8 are 97.3% and 79.8%, respectively, which are comparable to that of the original model. mAP50 and mAP50-95 improved by 1.1 and 4.2 percentage points, respectively, while the number of parameters and computational effort were reduced by 6% and 3.7%, respectively, indicating that the improved algorithms achieved very good results in CTC and CAF detection.

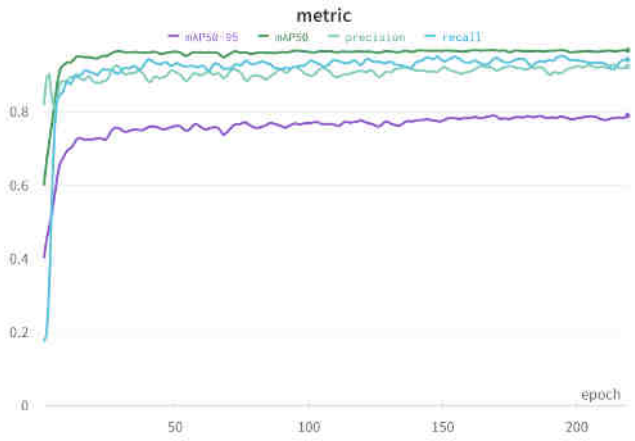


Fig. 7. Improved algorithm detection metrics

Fig. 7 shows the accuracy rate, check-all rate, and mAP of the weights obtained by CYOLOv8 on the validation set at the end of each training round, and it can be seen that all the indicators are in an upward trend as the number of training rounds increases until it tends to stabilize in obtaining the best weights. Due to the early stopping strategy, the improved algorithm obtains the best weights in the 169th round, and the performance is not improved in the following 50 rounds of training, so the training is terminated in the 219th epoch. Fig. 8 shows the comparison of the detection effect of YOLOv8 and the improved algorithm, it can be seen that due to the dense distribution of CTCs and high overlap, the unimproved algorithm is prone to false detection phenomenon in detection, after the improvement, this problem has been significantly improved, the detection effect of the algorithm is significantly improved, and the detection accuracy of the third image can even reach 100 percent. For some unlabelled cells in the image, the improved algorithm can also correctly detect them, which is enough to prove the superiority of the improved algorithm.

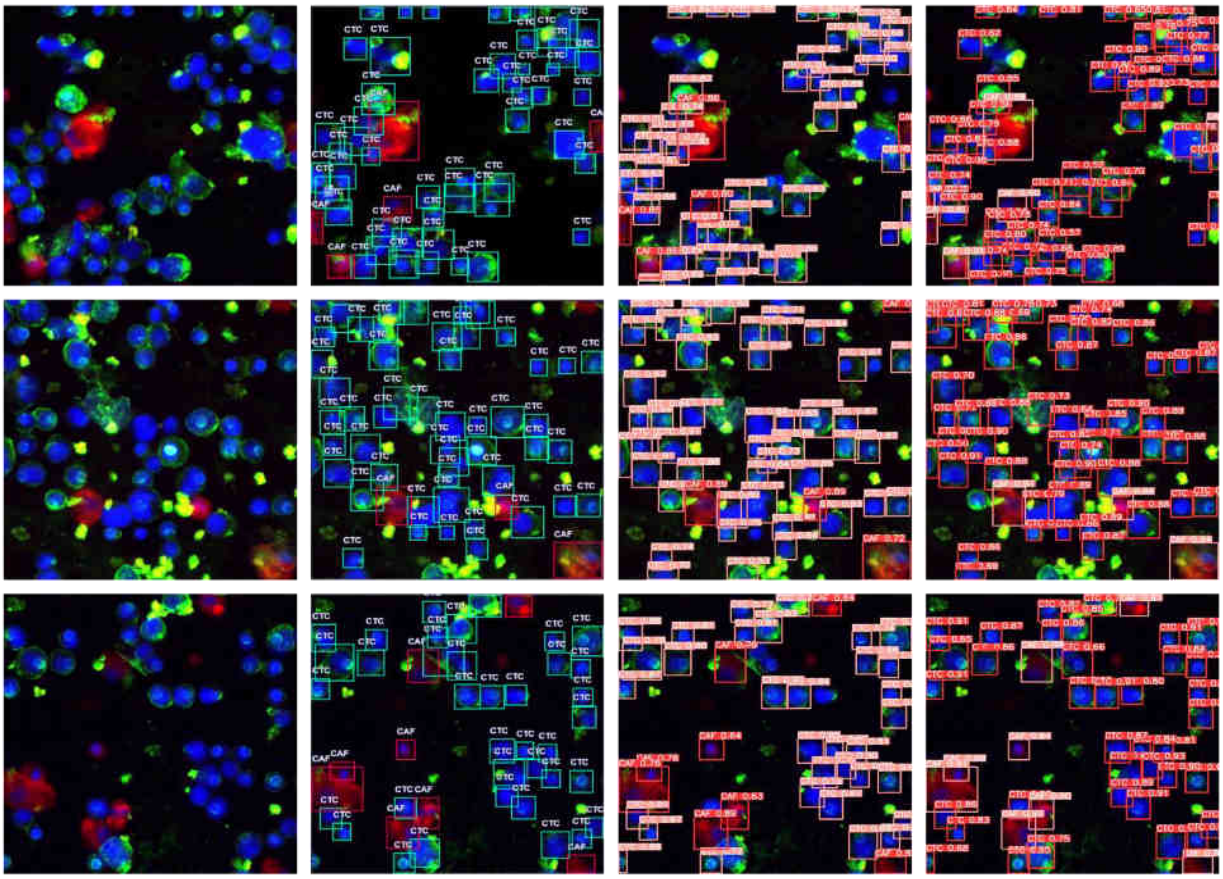
To highlight the performance of the improved algorithm, this experiment compares the performance of CYOLOv8 with other mainstream target detection algorithms in the same experimental setting using the CAC dataset. As shown in Table 1, compared with the two-stage algorithm Faster R-CNN, CYOLOv8 improves 5.5 percentage points and 9.9 percentage points in mAP50 and mAP50-95, respectively, with a significant gap in the detection effect, and CYOLOv8 has a huge advantage in terms of the number of parameters and computation. Compared with the single-stage algorithm RetinaNet, CYOLOv8 improves by 3.5 percentage points and 9.6 percentage points in mAP50 and mAP50-95, respectively, and still has a huge advantage in the number of parameters and the amount of computation. and RTMDet[25] compared to CYOLOv8, CYOLOv8 improves by 4.1 percentage points and 9.7 percentage points in mAP50 and mAP50-95, respectively, while the number of parameters and computation are reduced by 42.8% and 1.3%, respectively. Compared with the classical YOLO algorithms, CYOLOv8 outperforms YOLOv3-tiny by 2.2 percentage points and 7.5 percentage points on mAP50 and mAP50-95 and outperforms YOLOv5n by 1.9 percentage points and 4.5 percentage points. In terms of the number of parameters and the amount of computation, CYOLOv8 still has a big advantage over YOLOv3, but it has more parameters and computation than YOLOv5. It can be seen that the YOLO family of algorithms has a great advantage in the cell detection task.

The experimental results show that, compared with other mainstream target detection algorithms, CYOLOv8 achieves the best detection results in a smaller model volume for the detection tasks of CTCs and CAFs, with sufficient accuracy for the detection of cells in the dense intersection region.

Table 1. Comparison results of different algorithms

Algorithms	mAP50 (%)	mAP50-95 (%)	Params(M)	FLOPs (G)
Faster R-CNN	91.8	69.9	41.4	134.0
RetinaNet	93.8	70.2	36.3	128.0
RTMDet-tiny	93.2	70.1	4.9	8.0
YOLOv3-tiny	95.1	72.3	12.1	19.0
YOLOv5n	95.4	75.3	2.5	7.2
YOLOv8n	96.2	75.6	3.0	8.2

CYOLOv8	97.3	79.8	2.8	7.9



(a) Original (b) Ground Truth (c) YOLOv8 (d) CYOLOv8

Fig. 8. Comparison between the improved algorithm and YOLOv8 detection effect

IV. Conclusion

In this paper, a real-time detection method of CTCs and CAFs with improved YOLOv8 is proposed to solve the time-consuming and laborious problem of traditional cell detection. The addition of BoT module and EMA module in YOLOv8 can increase the feature weights of CTC and CAF in the image, so that the algorithm pays more attention to the features of the detection target, thus ignoring other information interference, and also effectively solves the problem of leakage of detection due to the large number of overlapping CTCs and the high density of CTCs in the image. The introduction of the C2f-SCConv module in the neck of YOLOv8 enables the improved algorithm to reduce the number of parameters substantially while effectively improving the detection accuracy. Experiments show that the mAP50 of the CYOLOv8 model in the CAC dataset is as high as 97.3% and the mAP50-95 is as high as 79.8%, which are both better than the existing models, indicating that the model has a good ability of classification and recognition. This CYOLOv8 model provides an effective method for CTC and CAF detection in medical images and is expected to be applied to CTC and CAF recognition in medical images.

Acknowledgment

This research was not supported by any funding project.

References

[1] Lambert, Arthur W., Diwakar R. Pattabiraman, and Robert A. Weinberg, “Emerging biological principles of metastasis,” Cell, vol. 168, 2017, pp. 670-691.
[2] Taftaf, R., et al., “ICAM1 initiates CTC cluster formation and trans-endothelial migration in lung metastasis of breast cancer,” Nature Communications, vol. 12, 2021, pp. 4867.
[3] Ao, Z., et al., “Identification of Cancer-Associated Fibroblasts in Circulating Blood from Patients with Metastatic Breast Cancer,” Cancer Res., vol. 75, 2015, pp. 4681-4687.
[4] Sahai, E., et al., “A framework for advancing our understanding of cancer-associated fibroblasts,” Nature Reviews Cancer, vol. 20, 2020, pp. 174-186.
[5] Allard, W. Jeffrey, and Leon WMM Terstappen, “CCR 20th Anniversary Commentary: Paving the way for circulating tumor cells,” Clinical Cancer Research, vol. 21, 2015, pp. 2883-2885.
[6] Guo, Z., et al., “Circulating Tumor Cell Identification Based on Deep Learning,” Frontiers in Oncology, vol. 12, 2022, pp. 843879-843879.
[7] Shen, Wang., et al. “Label-free detection of rare circulating tumor cells by image analysis and machine learning,” Scientific Reports, vol. 1, 2020, pp. 12226.
[8] Leonie, L., Zeune., et al., “Deep learning of circulating tumor cells,” Nature Machine Intelligence, vol. 2, 2022, pp. 124-133.
[9] Shen, Cheng, et al., “Automatic detection of circulating tumor cells and cancer associated fibroblasts using deep learning,” Scientific Reports, vol. 13, 2023 pp.5708.
[10] Lin T Y, Goyal P, Girshick R, et al., “Focal loss for dense object detection,” Proceedings of the IEEE International Conference on Computer Vision(ICCV), 2017, pp. 2980-2988.
[11] S. Ren, K. He, R. Girshick, and J. Sun, “Faster r-cnn: Towards real-time object detection with region proposal networks,” Advances in neural information processing systems, vol. 28, 2015.

[12] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition(CVPR), 2016 pp. 779-788.
[13] He K, Gkioxari G, Dollár P, et al., “Mask r-cnn,” Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition(CVPR), 2017, pp. 2961-2969.
[14] Jocher, G., Chaurasia, A., and Qiu, J, “YOLO by Ultralytics (Version 8.0.0),” 2023, https://github.com/ultralytics/ultralytics.
[15] Srinivas A, Lin T Y, Parmar N, et al., “Bottleneck transformers for visual recognition,” Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition(CVPR), 2021, pp. 16519-16529.
[16] D. Ouyang et al., “Efficient Multi-Scale Attention Module with Cross-Spatial Learning,” ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2023, pp. 1-5.
[17] J. Li, Y. Wen and L. He, “SCConv: Spatial and Channel Reconstruction Convolution for Feature Redundancy,” in 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2023, pp. 6153-6162.
[18] J. Redmon, A. Farhadi, “Yolo9000: better, faster, stronger,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition(CVPR), 2017, pp. 7263-7271.
[19] J. Redmon, A. Farhadi, “Yolov3: An incremental improvement,” arXiv preprint arXiv:1804.02767, 2018.
[20] Bochkovskiy A, Wang C Y, Liao H Y M, “Yolov4: Optimal speed and accuracy of object detection,” arXiv preprint arXiv:2004.10934, 2020.
[21] G. Jocher, “YOLOv5 by Ultralytics,” https://github.com/ultralytics/yolov5.
[22] Wang C Y, Bochkovskiy A, Liao H Y M. “YOLOv7:Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors,” Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition(CVPR), 2023, pp. 7464-7475.
[23] Dosovitskiy A, Beyer L, Kolesnikov A, et al. “An image is worth 16x16 words: transformers for image recognition at scale,” arXiv preprint arXiv:2010.11929, 2020.
[24] He K, Zhang X, Ren S, et al. “Deep residual learning for image recognition,” Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition(CVPR), 2016, pp. 770-778.
[25] Lyu C, Zhang W, Huang H, et al. “Rtmdet: An empirical study of designing real-time object detectors,” arXiv preprint arXiv:2212.07784, 2022.

Wang Xiandong graduate student, research direction: deep learning, object detection.

Ma Xin graduate tutor, Ph.D. in Engineering, research direction: (1) Microfluidic biochip detection technology (2) Embedded system medical device design (3) Artificial intelligence drug design/screening, cell image analysis (4) Organoid chip bionics Model. He hosted and participated in 7 national, provincial, and ministerial level fund projects, and published more than ten academic papers as the first author.