DOI: 10.19884/j.1672-5220.202406015

DrownACB-YOLO: an Improved YOLO for Drowning Detection in Swimming Pools

ZENG Xiaoya[1], XU Wujun[1, 2*], ZHANG Xiunian[3]
1. College of Information Science and Technology, Donghua University, Shanghai 201620, China
2. Engineering Research Center of Digitalized Textile and Fashion Technology, Donghua University, Shanghai 201620, China
3. Lian Ren Digital Health Company Limited, Shanghai 201200, China

Abstract: With the rise in drowning accidents in swimming pools, the demand for the precision and speed in artificial intelligence (AI) drowning detection methods has become increasingly crucial. Here, an improved YOLO-based method, named DrownACB-YOLO, for drowning detection in swimming pools is proposed. Since existing methods focus on the drowned state, a transition label is added to the original dataset to provide timely alerts. Following this expanded dataset, two improvements are implemented in the original YOLOv5. Firstly, the spatial pyramid pooling (SPP) module and the default upsampling operator are replaced by the atrous spatial pyramid pooling (ASPP) module and the content-aware reassembly of feature (CARAFE) module, respectively. Secondly, the cross stage partial bottleneck with three convolutions (C3) module at the end of the backbone is replaced with the bottleneck transformer (BotNet) module. The results of comparison experiments demonstrate that DrownACB-YOLO performs better than other models.

Keywords: drowning detection; YOLO; atrous spatial pyramid pooling (ASPP); content-aware reassembly of feature (CARAFE)
CLC number: TP391 **Document code**: A
Article ID: 1672-5220(2025)04-0417-08

Open Science Identity (OSID)

0 Introduction

Swimming is a popular sport. However, drowning in swimming pools has become a major cause of accidental death worldwide in the report of world health organization (WHO)[1-2]. To overcome this risk, some swimming venues are equipped with underwater cameras to assist lifeguards in manually monitoring blind spots, but this heavy-intensity approach is susceptible to negligence. Therefore, there is an inevitable trend towards automating the process of underwater drowning detection[3].

Compared to traditional automatic algorithms, artificial intelligence (AI) is widely applied in drowning detection due to its precision and learning ability[4].

Venkata et al.[5] applied the faster region-based convolutional neural network (Faster R-CNN) and the single shot multibox detector (SSD) for drowning detection. The experimental results reveal that SSD exhibits a higher detection speed, while Faster R-CNN achieves a higher precision. Hayat et al.[6] integrated the feature pyramid network (FPN) into the mask region-based convolutional neural network (Mask R-CNN), achieving a precision of 94.1% but with a speed of only 6 frame/s (FPS). Handalage et al.[7] merged the deep simple online real-time tracking (DeepSORT) with YOLOv3 to devise a three-stage drowning detection method, achieving a high precision. Although the precision of the latter two methods is high, there is still considerable space for improvement in the detection speed.

Here, an improved YOLO-based method, named DrownACB-YOLO, is proposed to take the detection precision and speed into account. Three improvements of the proposed DrownACB-YOLO are depicted:

1) The transition label is added to Zhang's[8] dataset, a binary classification dataset containing the normal and drowning labels. This change expands the dataset to tri-classification, compensating for the lack of an intermediate state and providing timely alerts.

2) The atrous spatial pyramid pooling (ASPP) module and the content-aware reassembly of feature (CARAFE) module are integrated into YOLOv5 to improve precision.

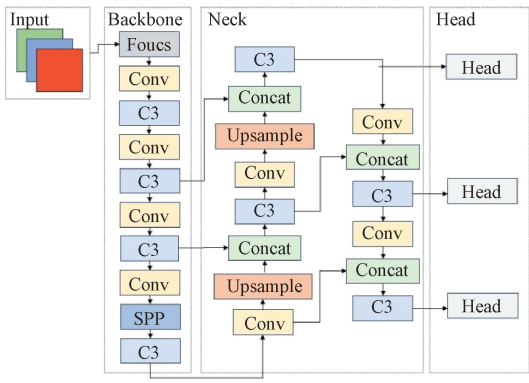
3) The cross stage partial bottleneck with three convolutions (C3) module at the end of the backbone in YOLOv5 is replaced with the bottleneck transformer (BotNet) module to further improve the detection performance.

1 Methods

The YOLO algorithm belongs to one-stage algorithms. Compared to other versions, the lighter

Received date: 2024-06-29
* Correspondence should be addressed to XU Wujun, email: wujun.hsu@qq.com
Citation: ZENG X Y, XU W J, ZHANG X N. DrownACB-YOLO: an improved YOLO for drowning detection in swimming pools[J]. *Journal of Donghua University (English Edition)*, 2025, 42(4): 417-424.

design of YOLOv5 enables faster and easier deployment. The series of YOLOv5 includes YOLOv5s, YOLOv5m, YOLOv5l and YOLOv5x. Among them, YOLOv5s has the smallest weight file[9]. Therefore, YOLOv5s is chosen as the baseline for improvement. YOLOv5s consists of input, backbone, neck and head. The input utilizes mosaic augmentation to improve the imbalance of input data. The backbone and neck perform feature extraction and fusion at different granularities, respectively[10]. The head performs the final detection and classification. The overall architecture of YOLOv5s is shown in Fig. 1.



SPP—spatial pyramid pooling; Conv—convolution; Concat—concatenation.

Fig. 1 Overall architecture of YOLOv5s

1.1 Improving YOLOv5 model

Here, the optimization focuses on enhancing the feature extraction and fusion abilities of the backbone and neck. Therefore, some advanced modules are introduced in the original YOLOv5 to improve the detection precision and performance.

1.1.1 ASPP

Convolutional neural networks (CNNs) have outstanding performance in computer vision (CV) owing to their rapid feature extraction and robust generalization ability. The components include convolutional layers, pooling layers, fully connected layers and an output layer. In particular, the pooling layers dramatically reduce the resolution of feature maps, leading to disadvantages in image processing. Thus, the atrous convolution (AtrConv) module could be introduced to optimize the feature extraction process[11].

Unlike normal convolution, a dilation rate is introduced to AtrConv to define the interval of the kernel. In CNNs, the dilation rate makes convolution kernels discontinuous, thus forming AtrConv. This structure enables capturing diverse receptive fields without increasing the number of parameters and compromising the feature map resolution. However, AtrConv regards adjacent pixels as separate entities, leading to the lack of correlation and the loss of local features. To solve this problem, ASPP is introduced and its structure is shown in Fig. 2.

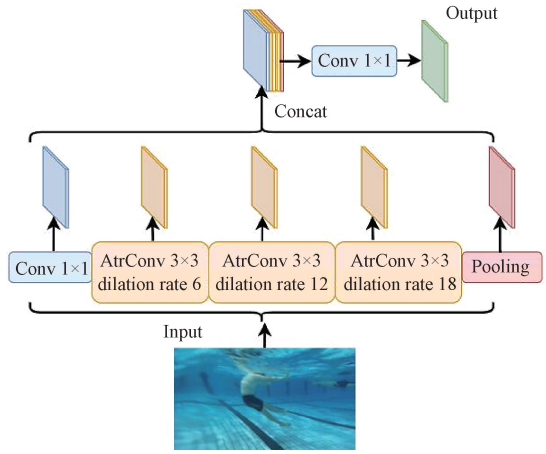


Fig. 2 Structure of ASPP

ASPP enhances the ability to capture contextual information at multiple scales by connecting AtrConv with different dilation rates in parallel, thereby realizing feature aggregation and reducing the loss of image information. Consequently, integrating YOLOv5 with ASPP is more efficient in the image analysis and object detection.

1.1.2 CARAFE

In the CV field, the upsampling operator is essential for capturing the details of images and restoring feature maps. CARAFE stands out for the ability to achieve broader receptive fields than traditional methods, such as nearest neighbor and linear interpolation, due to its unique structure.

The CARAFE operator comprises the kernel prediction and the content-aware reassembly modules. For a given feature map $\boldsymbol{\chi}$, the $k \times k$ subregion centered on the position l is denoted as $N(\boldsymbol{\chi}_l, k_{\text{encoder}})$, where k_{encoder} is the prediction kernel size. The kernel prediction module ψ estimates the direction kernel $\boldsymbol{U}_{l'}$ for the corresponding position l' of l in the new feature map $\boldsymbol{\chi}'$ by $N(\boldsymbol{\chi}_l, k_{\text{encoder}})$[12]. Then, the content-aware reassembly module ϕ utilizes $\boldsymbol{U}_{l'}$ to perform feature recombination within the predefined region. This recombination is centered around a weighted combination that generates $\boldsymbol{\chi}'$, where weights are produced in a content-aware manner. For the modules in CARAFE,

$$\begin{cases} \boldsymbol{U}_{l'} = \psi(N(\boldsymbol{\chi}_l, k_{\text{encoder}})), \\ \boldsymbol{\chi}'_{l'} = \phi(N(\boldsymbol{\chi}_l, k_{\text{up}}), \boldsymbol{U}_{l'}), \end{cases} \tag{1}$$

where k_{up} is the reassembly kernel size. The method could be summarized as initially predicting the recombination kernel at each specified position, using the predicted recombination kernel and then extracting the recombination features.

In conclusion, CARAFE enables upsampling based on the input content, aggregates contextual semantic information and generates an adaptive kernel to improve content-aware processing capability.

1.1.3 BotNet

Attention mechanisms could dynamically allocate computational resources based on the importance of processed information, thereby enhancing model performance and expanding their applications in the CV field.

According to the functional scope, attention mechanisms could be categorized into soft attention, hard attention and self-attention. Self-attention enables parallel computing on lengthy inputs, allowing for the simultaneous processing of disparate image regions. Additionally, self-attention focuses on input without requiring additional information, capturing the correlations and dependencies among different parts of the input image[13].

The structure of self-attention is illustrated in Fig. 3. Firstly, each element x_i of the input $\boldsymbol{X} = [x_1, x_2, \cdots, x_i, \cdots, x_n]$ is encoded by the embedding layer to obtain y_i, where $i \in \{1, 2, \cdots, n\}$ indexes the sequence positions. Secondly, y_i passes through three fully connected layers to generate the query, key and value, namely, q_i, k_i and v_i, respectively. The formulas of q_i, k_i and v_i are

$$\begin{cases} q_i = w_q \cdot y_i, \\ k_i = w_k \cdot y_i, \\ v_i = w_v \cdot y_i, \end{cases} \tag{2}$$

where w_q, w_k and w_v are trained by the network. The matrix forms of the formulas are

$$\begin{cases} \boldsymbol{Q} = \boldsymbol{W}_q \cdot \boldsymbol{Y}, \\ \boldsymbol{K} = \boldsymbol{W}_k \cdot \boldsymbol{Y}, \\ \boldsymbol{V} = \boldsymbol{W}_v \cdot \boldsymbol{Y}. \end{cases} \tag{3}$$

Then, each element $a_{i,j}$ of the correlation matrix $\boldsymbol{A}$ is obtained by calculating the dot product between the query q_i derived from x_i and the key k_j derived from x_j. Finally, $\boldsymbol{A}$ is normalized by Softmax to obtain the attention matrix $\boldsymbol{A}'$, where each element is denoted as $a'_{i,j}$. $\boldsymbol{A}'$ is

$$\boldsymbol{A}' = \mathrm{softmax}(\boldsymbol{K}^{\mathrm{T}}\boldsymbol{Q}). \tag{4}$$

The output of self-attention O_s is

$$O_s = \boldsymbol{V} \odot \boldsymbol{A}'. \tag{5}$$

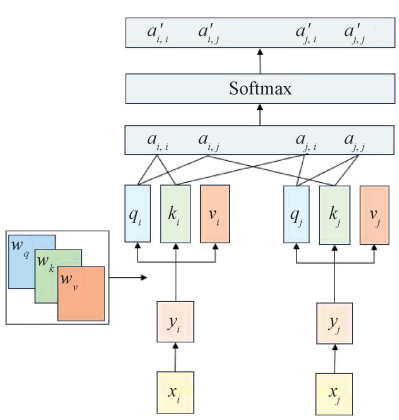
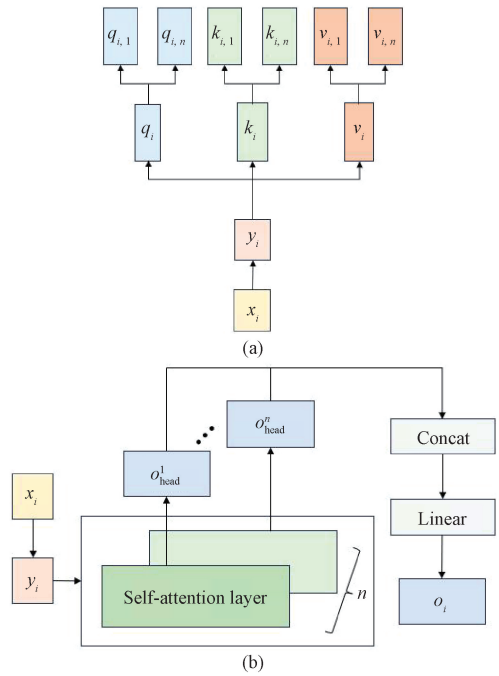


Fig. 3 Structure of self-attention

Self-attention only focuses on the single feature relationship, whereas the multi-head self-attention (MHSA) module, as depicted in Fig. 4, is a superposition of self-attention, enhancing the ability of models to represent complex semantic structures and relationships. The essence of MHSA lies in dividing q_i, k_i and v_i into multiple heads through linear transformations followed by concatenating all the outputs o_i together to form the nth head self-attention mechanism[13].



o^n_{head}— output of the nth head self-attention.

Fig. 4 Structure of MHSA: (a) local; (b) global

Considering the advantages of MHSA, Srinivas et al.[14] integrated a four-head self-attention module into a residual network (ResNet) and replaced the bottleneck in the fourth block to form a new module. It is observed that this new module closely resembles the transformer in both structural design and operational efficiency. As a result, it is BotNet. This module utilizes both the feature extraction ability of CNNs and the self-attention mechanism of the transformer, yielding improved performance over CNNs.

1.2 Framework of DrownACB-YOLO

The first improvement of DrownACB-YOLO to YOLOv5 involves replacing SPP with ASPP, which results in DrownA-YOLO. This improvement expands the receptive field of the convolution kernel, mitigating the resolution loss linked to the three pooling structures in SPP. The second improvement involves the integration of CARAFE with YOLOv5. This improvement, combined with the previously introduced ASPP module, constitutes DrownAC-YOLO. The default upsampling operator in YOLOv5, which is the zero-order interpolation operator, suffers from a limited receptive field and inadequate semantic understanding. CARAFE is introduced to

compensate for this drawback. The feature extraction network of YOLOv5 adopts the C3 structure, which brings a high model complexity[15]. In order to further improve the detection performance and reduce the minimal delay overhead, BotNet is finally introduced. The architecture of DrownACB-YOLO is depicted in Fig. 5.

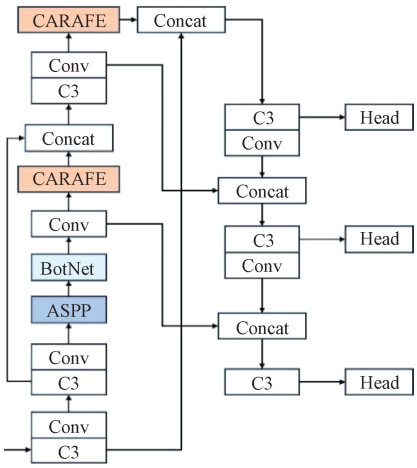


Fig. 5 Architecture of DrownACB-YOLO

1.3 Dataset

There are few open datasets on swimming in pools. Zhang[8] used two 12-megapixel rear cameras, equipped with a five-element lens and a six-element lens, in an Apple iPhone 11, to capture swimming videos. Representative frames were then extracted from the recorded videos for the frame analysis. The dataset contains 1 140 images with 96 dot/in (DPI), and swimmers in images are labeled as the normal or drowning in LabelImg software. Since there are usually multiple swimmers in the image, the dataset ultimately contains 724 normal labels and 808 drowning labels.

2 Experiments

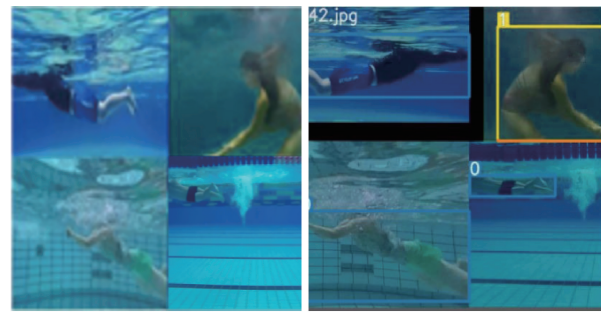
In order to design an effective model, the dataset must be preprocessed before model training. The optimal model is then selected.

2.1 Dataset preprocessing

Due to the limited size of Zhang's dataset, 2 088 images collected from the network were added to the original dataset, resulting in an expanded dataset with a total of 3 228 images.

Currently, most existing datasets are binary classification, leading to delayed alerts and injuries when the drowning persons are detected. To address this, a transition label is added to the expanded dataset. Behaviors in the pool are categorized as normal, drowning and transition. The normal includes standing and regular swimming postures. The drowning is characterized by the chaotic behavior with significant body folding and irregular limb movements. The transition is identified by a large body tilt angle and partial confusion in physical actions. This new dataset contains 2 894 normal labels, 981 drowning labels and 2 069 transition labels. This process enables lifeguards to initiate timely rescues upon detecting the transition state.

To meet the training requirements, the dataset is divided into a training set, a validation set and a test set at a ratio of 8∶1∶1. Additionally, due to the limited size of the expanded dataset, data augmentation methods such as mosaic augmentation, rotation and scaling are applied to preprocess the dataset and prevent overfitting during training. The effects of some samples in the expanded dataset are shown in Fig. 6. These samples all exhibit decreased brightness and other transformations are also applied. Specifically, the first and the second samples are enlarged and undergo mirrored rotation; the third and the fourth samples are enlarged.



(a) (b)

Fig. 6 Comparison of samples in expanded dataset: (a) before preprocessing; (b) after preprocessing

2.2 Experimental environment and parameter settings

The parameters of the experimental environment and network training are shown in Tables 1 and 2.

Table 1 Parameters of software and hardware

Item	Value
CPU	Intel Xeon Platinum 8255C with 2.50 GHz
GPU	NVIDIA GeForce RTX 3080 with 10 GB
Operating system	Ubuntu 18.04
Python	3.8
Deep learning framework	PyTorch

Table 2 Parameters of network training

Item	Value
Batch	16
Learning rate	0.001
Epoch	100
Image size/(pixel × pixel)	640 × 640

2.3 Evaluation metrics

The precision of the drowning detection method is evaluated by metrics such as recall rate R_r, precision rate R_p, and mean average precision $\overline{P_A}$. These metrics are closely linked to the intersection and union (IoU) of the

predicted box and the actual box. The IoU is considered as true positive T_P when it exceeds the predefined threshold. Conversely, if the IoU is below the threshold, it is marked as false positive F_P. When there is no predicted box on the target, it is classified as false negative F_N [10]. The formulas of R_r and R_p are

$$\begin{cases} R_r = \dfrac{T_P}{T_P + F_N}, \\ R_p = \dfrac{T_P}{T_P + F_P}. \end{cases} \tag{6}$$

For each label, the precision-recall (R_p-R_r) curve is plotted on the coordinate axes. The average precision P_A is quantified by the area under the R_p-R_r curve and bounded by the abscissas[10]. $\overline{P_A}$ is the average of all P_A values, serving as a comprehensive metric to evaluate model performance across different categories. The formulas of P_A and $\overline{P_A}$ are

$$\begin{cases} P_A = \int_0^1 R_p(R_r)\,\mathrm{d}R_r, \\ \overline{P_A} = \dfrac{1}{n}\sum_{i=1}^{n} P_A, \end{cases} \tag{7}$$

where n is the number of classes. Here, $n=3$.

The assessment of the model complexity includes considerations of both the number of parameters and computational cost, where the computational cost is in terms of floating-point operations (FLOPs). Meanwhile, the detection speed for drowning incidents is in terms of FPS. These evaluation metrics allow for a balanced understanding of the model performance.

2.4 Model training and experimental results

The convergence curves of training losses are depicted in Fig. 7. It is observed that the proposed model eventually tends to converge.

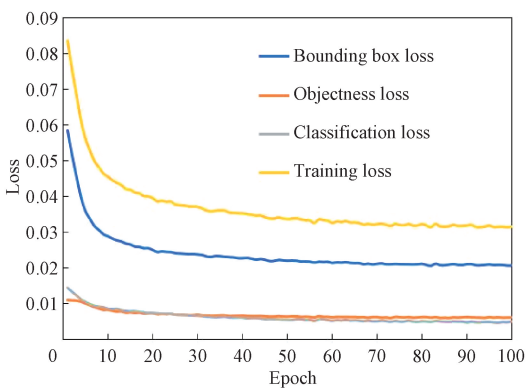


Fig. 7 Convergence curves of training losses

2.4.1 Model complexity

The complexity of different models is illustrated in Table 3. RetinaNet and YOLOv7 have a higher number of parameters and computational cost than other models. Consequently, the two models possess a higher model complexity, implying more demanding hardware requirements for practical applications. The remaining models decrease in complexity from YOLOv10 to YOLOv5 to EfficientDet.

Table 3 Number of parameters and computational cost in different models

Model	Number of parameters	Computational cost/GFLOPs
EfficientDet	6.56×10^6	11.52
RetinaNet	3.637×10^7	82.10
YOLOv10	8.04×10^6	24.50
YOLOv7	3.721×10^7	105.14
YOLOv5	7.06×10^6	16.30
DrownA-YOLO	8.96×10^6	17.82
DrownAC-YOLO	9.10×10^6	18.13
DrownACB-YOLO	8.44×10^6	17.65

YOLOv10 has fewer parameters than DrownACB-YOLO but more computational cost, which demonstrates that the number of parameters and computational cost are not necessarily proportional. The number of parameters and computational cost are metrics of model complexity, but they focus on different aspects. The computational cost measures the computational resources needed during detection inference, while the number of parameters reflects the training complexity. Despite requiring more memory and storage space during training, DrownACB-YOLO demonstrates lower computational demands during detection inference, leading to reduced hardware requirements.

2.4.2 Detection speed

The experimental results regarding the detection speed of different models are shown in Fig. 8. The order of the detection speed from the fastest to the slowest is YOLOv10, YOLOv5, DrownACB-YOLO, DrownA-YOLO, DrownAC-YOLO, YOLOv7, RetinaNet and EfficientDet.

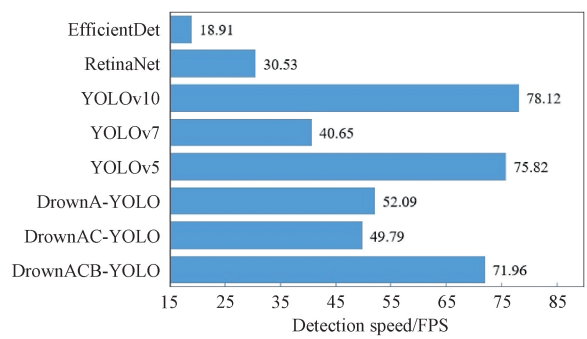


Fig. 8 Detection speed of different models

By combining Table 3 and Fig. 8, it is observed that the model complexity increases, correspondingly decreasing the detection speed. Thus, DrownACB-YOLO achieves a detection speed of 71.96 FPS and inevitably

lags behind the original YOLOv5 in the detection speed.

Notably, EfficientDet has the smallest number of parameters and computational cost among these algorithms, but it only achieves a detection speed of 18.91 FPS. This is because EfficientDet relies heavily on the depthwise convolution. Despite achieving a computational cost of 11.52 GFLOPs, it incurs substantial data read and write operations. As a result, a significant amount of time is spent on memory access, preventing the computing power of GPU from being fully utilized.

2.4.3 Detection precision

The detection precision is depicted in Fig. 9. Combined with the analyses of Table 3 and Fig. 9, the results reveal that the gradual integration of ASPP and CARAFE into YOLOv5 leads to a sustained increase in $\overline{P_A}$. Furthermore, BotNet not only reduces the model complexity but simultaneously elevates the model precision. $\overline{P_A}$ of DrownACB-YOLO reaches 0.917, marking an absolute increase by nearly six percentage points compared to YOLOv5 and outperforming EfficientDet, RetinaNet and YOLOv10. YOLOv7 achieves the highest $\overline{P_A}$ of 0.939. However, its model complexity is the highest, which sacrifices the detection speed and is unfavorable for deployment and practical applications.

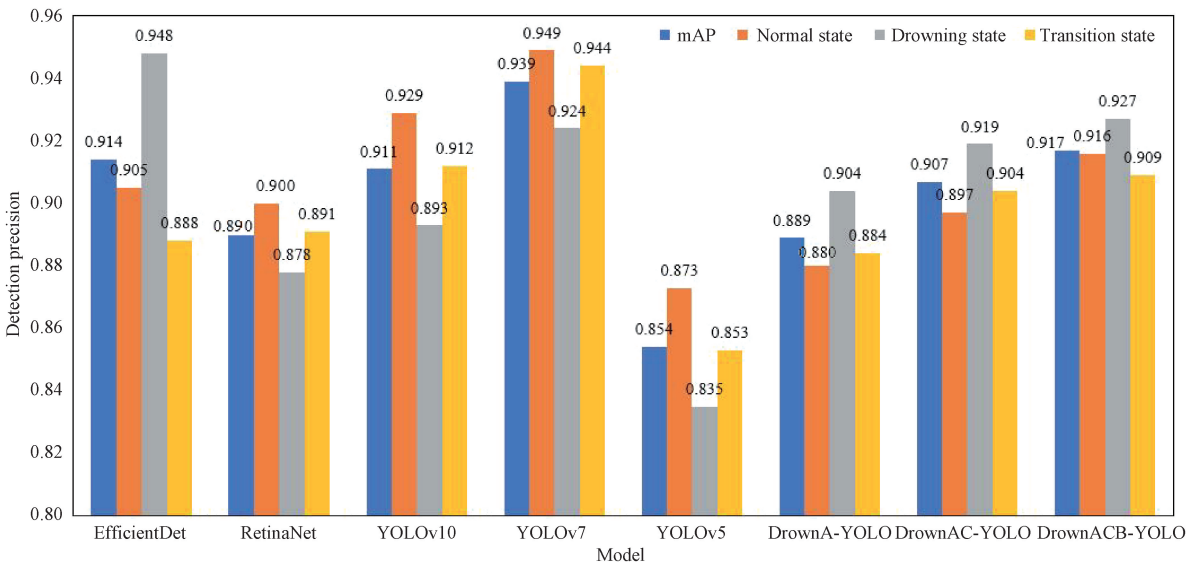
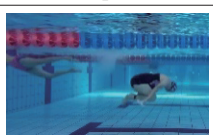
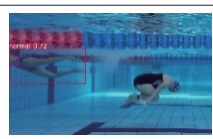
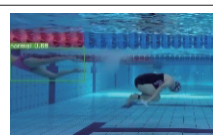
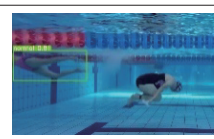
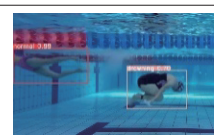
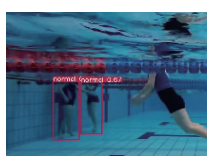
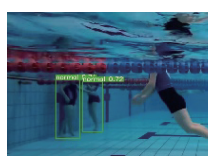
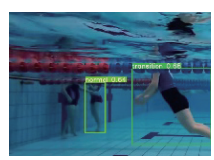
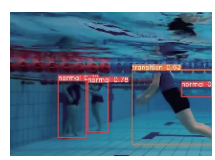
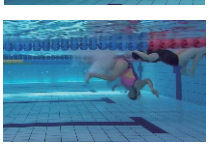
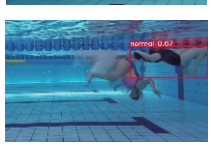
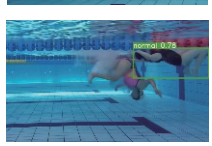
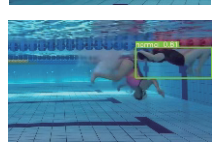
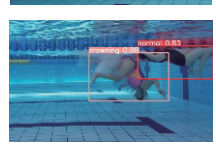


Fig. 9 Detection precision of different models

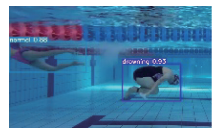
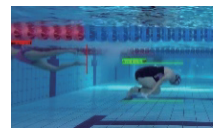
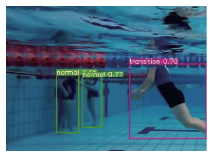
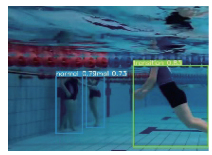
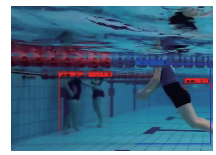
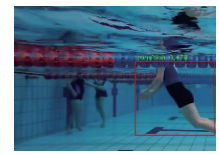
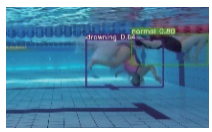
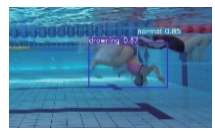
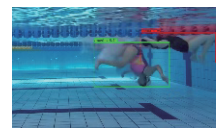
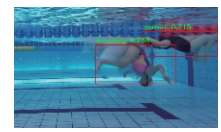
2.4.4 Detection performance

As shown in Table 4, after gradual improvements, DrownACB-YOLO successfully detects all targets in the image. The performance of original YOLOv5 is limited due to its SPP structure and zero-order upsampling operator. These components reduce image resolution and limit the receptive field, leading to insufficient feature extraction. Consequently, it struggles to differentiate between drowning and transition states, as well as between normal and transition states. DrownACB-YOLO addresses these shortcomings. Additionally, DrownACB-YOLO maintains a higher detection precision than other models, except YOLOv7, and successfully detects small-scale targets that YOLOv7 and YOLOv10 fail to detect.

Table 4 Detection results of different models

Input	YOLOv5	DrownA-YOLO	DrownAC-YOLO	DrownACB-YOLO
				
				
				

(Table 4 continued)

Input	YOLOv10	YOLOv7	EfficientDet	RetinaNet
				
				
				

3 Conclusions

The main work of this paper was focused on enriching the number of images and labels in the original dataset to create a new dataset, and proposing DrownACB-YOLO with a high detection speed and a high detection precision for drowning detection in swimming pools. The experimental results indicate that DrownACB-YOLO reaches a detection speed of 71.96 FPS and a mean average precision of 0.917. Compared to the original YOLOv5, DrownACB-YOLO shows an absolute increase of nearly six percentage points, maintains a lower model complexity, keeps a higher detection speed and demonstrates a superior performance across various scales and in obscured scenarios. The proposed model more closely meets the essential requirements for drowning detection in swimming pools than other models.

Compared to the normal and drowning states, the transition state is our main focus. Although DrownACB-YOLO has significantly improved the detection precision of the transition state, it remains slightly inferior to YOLOv7 and YOLOv10, presenting a future challenge. Additionally, labeling datasets relies on subjective judgments, which affects the feature extraction capabilities of the model. To address this, we plan to incorporate human pose estimation before labeling. By using human skeleton points obtained from pose estimation, we could achieve a more scientific classification of swimmer behaviors, thereby enhancing the detection precision. This paper primarily focuses on algorithmic improvements. Future work will shift toward hardware deployment to further assess the practical feasibility of the algorithm.

References

[1] World Health Organization. Global report on drowning: preventing a leading killer[R]. World Health Organization, 2014.

[2] HE X Y, YUAN F, LIU T Z, et al. A video system based on convolutional autoencoder for drowning detection[J]. *Neural Computing and Applications*, 2023, 35(21): 15791-15803.

[3] URRUCHI C, CERVANTES-CHAUCA D, HUAMANCHAHUA D. Proposal of a swimming pool drowning detection system using cameras and raspberry Pi based on machine learning [C]//2022 2nd International Conference on Robotics, Automation and Artificial Intelligence (RAAI). New York: IEEE, 2022: 178-181.

[4] HE Q N, ZHANG H S, MEI Z Q, et al. High accuracy intelligent real-time framework for detecting infant drowning based on deep learning [J]. *Expert Systems with Applications*, 2023, 228: 120204.

[5] VENKATA M, NISHANT S. Detecting and tracking of humans in an underwater environment using deep learning algorithms[D]. Karlskrona: Blekinge Institute of Technology, 2019.

[6] HAYAT M A, YANG G T, IQBAL A. Mask R-CNN based real time near drowning person detection system in swimming pools[C]//2022 Mohammad Ali Jinnah University International Conference on Computing (MAJICC). New York: IEEE, 2022: 1-6.

[7] HANDALAGE U, NIKAPOTHA N, SUBASINGHE C, et al. Computer vision enabled drowning detection system [C]//2021 3rd International Conference on Advancements in Computing (ICAC). New York: IEEE, 2021: 240-245.

[8] ZHANG X N. Research on swimming pools drowning warning based on pose estimation and edge computing [D]. Shanghai: Donghua University, 2023. (in Chinese)

[9] HUANG C, ZHU Y, WANG J Y, et al. Water surface target detection algorithm for unmanned cleaning ship based on improved YOLO V5[C]// 2022 International Conference on Cyber-Physical

Social Intelligence (ICCSI). New York: IEEE, 2022: 386-391.
[10] XIE R L, ZHU Y J, LUO J, et al. Detection algorithm for bearing roller end surface defects based on improved YOLOv5n and image fusion [J]. *Measurement Science and Technology*, 2023, 34(4): 045402.
[11] HUANG Y, WANG Q Q, JIA W J, et al. See more than once: kernel-sharing atrous convolution for semantic segmentation [J]. *Neurocomputing*, 2021, 443: 26-34.
[12] WANG J H, GAO X H, LIU Z, et al. GSC-YOLOv5: an algorithm based on improved attention mechanism for road creak detection [C]//2023 IEEE 12th Data Driven Control and Learning Systems Conference (DDCLS). New York: IEEE, 2023: 1664-1671.
[13] HUANGFU X Y, QIAN H M, HUANG M. A review of deep neural networks combined with attention mechanism [J]. *Computer and Modernization*, 2023, 2: 40.
[14] SRINIVAS A, LIN T Y, PARMAR N, et al. Bottleneck transformers for visual recognition [C]//2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). New York: IEEE, 2021: 16519-16529.
[15] LI Y, LI R, XU Y. Design of online vision detection system for stator winding coil [J]. *Journal of Donghua University* (*English Edition*), 2023, 40(6): 639-648.

DrownACB-YOLO：一种用于游泳池溺水检测的改进 YOLO 算法

曾小雅[1]，许武军[1,2*]，张修念[3]

1. 东华大学 信息科学与技术学院，上海 201620
2. 东华大学 数字化纺织服装技术教育部工程研究中心，上海 201620
3. 联仁健康医疗大数据科技股份有限公司，上海 201200

摘　要：随着游泳池溺水事故的增多，人们对人工智能溺水检测方法的精度和速度要求也越来越高。该文提出一种基于改进的 YOLO 游泳池溺水检测方法，命名为 DrownACB-YOLO。现有方法都只关注溺水状态，该文在原始数据集中添加了一个过渡标签，以提供及时警报。在此基础上，对 YOLOv5 算法进行了两点改进。首先，将 YOLOv5 的空间金字塔池化模块和默认的上采样算子分别替换为空洞空间金字塔池化模块和内容感知特征重组模块。其次，将 YOLOv5 的主干部分末尾的 C3 模块替换为 bottleneck transformer 模块。对比实验的结果表明，DrownACB-YOLO 性能优于其他模型。

关键词：溺水检测；YOLO 算法；空洞空间金字塔池化；内容感知特征重组