Fuzzy Systems and Data Mining IX
A.J. Tallón-Ballesteros and R. Beltrán-Barba (Eds.)

doi:10.3233/FAIA231052

E-Yolov7: A Life Jacket Detection Algorithm Based on Yolov7

Jiajia XU[a,1] and Changming ZHU[a]
[a] *School of Information Engineering, Shanghai Maritime University, China*

Abstract. As the core equipment of water safety, life jackets play an indispensable role in the safety field and rescue applications. Although existing deep learning techniques and object detection methods have achieved remarkable success in various fields, there are still many challenges in applying them to life jacket detection. Complex environmental changes, occlusion problems caused by water surface fluctuations, etc. will affect the detection effect. In addition, there are relatively few datasets suitable for life jacket detection, which makes many existing detection models lack stability and generalization ability in practical application scenarios. In order to address these challenges, this paper aims to propose an efficient and accurate detection method for life jacket object detection. Specifically, we first review the relevant work of sea rescue, then collect and label a life jacket-related dataset, and then introduce EVC block based on the Yolov7 network to aggregate the local corner area information of the image, and propose an improved life jacket target detection method E-Yolov7. In the experimental and evaluation section, the proposed method was tested, comparing it with Yolov5s, Yolov7-Tiny, Yolov7x, and Yolov7. Experiments show that the improved model E-Yolov7 has better performance in the life jacket detection task, and compared with Yolov7, the precision is increased by 3.7 %, the mAP is increased by 2.5% and the F1 score is increased by 2 %.

Keywords. Yolov7, life jacket detection, deep learning, EVC

1. Introduction

Detecting life jackets is an important application of object detection technology in the field of water safety. To ensure the safety of water activities and improve the efficiency of emergency rescue operations, researchers have conducted extensive work in the detection of life jackets.

In the early stages of life jacket detection research, traditional computer vision methods such as feature-based detection and classifiers were predominantly employed. Researchers used techniques like edge detection and color segmentation to identify the features of life jackets. However, these approaches exhibited limited robustness in complex scenes and under varying lighting conditions, making them inadequate to meet practical requirements.

With the emergence of deep learning, especially the development of Convolutional Neural Networks (CNNs), researchers began to explore the application of deep learning

[1] Corresponding Autho: Jiajia Xu, School of Information Engineering, Shanghai Maritime University, Shanghai, China. E-mail: 1353171026@qq.com.

to life jacket detection. Some studies utilized classic CNN architectures like AlexNet and VGG to train detection models for life jackets.

Ye et al. [1] proposed an automated detection model based on semantic segmentation. To achieve better results, improvements and optimizations were made to the classic U-net network in semantic segmentation. Special convolutional layers were used instead of max-pooling layers, an additional dropout layer was added to prevent overfitting, and reflective padding was applied to enable the extraction of more features by convolutional kernels. Liu et al. [2] introduced a method to simultaneously identify multiple ship targets based on ship color, size, and kinematic features. They established a transformation calculation model between image coordinates and real-world environment coordinates. Zhang et al. [3] applied Adaboost and Haar-like feature algorithms to image recognition. However, most of these approaches were mainly targeted at single-background image libraries. The actual images captured often presented complex backgrounds and significant shape variations due to shooting angles, which made traditional methods less effective in identifying water-based images. Fan et al. [4] used AlexNet as a basic network and employed a deep belief network (DBN) to classify images based on spectral and texture features. Yang et al. [5] designed an algorithm to detect life jackets in complex ferry terminal environments. This algorithm utilized statistical HSV color range information and boundary tracking methods to locate life jacket regions in images. Liu et al. [6] proposed a single-camera vision-based approach for unmanned boats involved in water search and rescue operations. An algorithm based on RGB color features was used for target localization, segmenting the target regions in the images. The VGG-F model was then used to extract and identify the water surface images.

In recent years, the Yolo (You Only Look Once) series has also found application in life jacket detection. Weng et al. [7] utilized Yolov5 to propose a detection algorithm capable of identifying life jackets in backlit environments on rivers, aiming to enhance law enforcement efficiency in water-based units.

Despite the significant successes achieved by existing deep learning techniques and object detection methods in various fields, challenges still exist in applying them to life jacket detection. Challenges include complex maritime environments, varying sea conditions, water turbulence, and occlusions, all of which can impact the accuracy and real-time performance of detection algorithms. To this end, many scholars have also made efforts in this regard [9][11].

Additionally, there is a relative scarcity of specialized datasets suitable for life jacket detection, which limits the stability and generalization capabilities of many existing detection models in real-world application scenarios. In order to address these challenges, this paper aims to propose an efficient and accurate detection method for life jacket object detection. The main contributions of this paper are as follows:

- A detection framework based on Yolov7 to determine whether people are wearing life jackets is proposed.
- The dataset of object detection for whether people are wearing life jackets is collected and annotated, and the situation under different lighting and fogging at sea is simulated by using image enhancement to improve the robustness of the model.
- Through a series of experimental evaluations and comparisons, it is proved that the method proposed in this paper is effective and meaningful.

2. Materials and Methods

2.1. Dataset

Since there are relatively few specialized datasets suitable for life jacket testing, in this research paper, we obtained relevant images on the web through keyword searches, and manually removed images that did not meet our requirements (such as the case of too much obscuration and the person being too small for the human eyes to distinguish whether the person is wearing the life jacket), and then processed the images, unified the size, and enriched the dataset with image enhancement and data enhancement methods. Use color dithering to adjust the saturation, brightness, contrast, etc. of the picture, which is used to simulate the situation under different lighting and climate. The labeling work uses the lableimg, which is used for the annotation. Considering that life jackets are the main detection object, so we pay more attention to life jackets when labeling.

Our proposed dataset contains a total of 4138 images, of which 3100 are in the training set, 534 in the validation set and 504 in the test set. The height and width of the images are 640 pixels. Some of the images in the dataset are shown in Figure 1. The quantitative distribution of labels is shown in Table 1. It contains a total of 7101 'nowear' and 7657 'wear' labels, of which there are 5390 'nowear' labels and 5648 'wear' labels in the training set, 895 'nowear' 993 'wear' labels in the validation set and 816 'nowear' labels and 1016 ' wear' tags in the validation set, 816 'nowear' tags and 1016 ' wear' tags in the test set.

Figure 1. Some examples of dataset.

Table 1. The labels distribution of the dataset

	Train	Val	Test	Total
Nowear	5390	895	816	7101
Wear	5648	993	1016	7657
Total	11038	1888	1832	14758

2.2. Method

So far, Yolo has appeared in several versions [12][20], each with certain improvements over the previous version. The method proposed in this paper is based on Yolov7 [20]. In this article, we used Yolov7 [20] as infrastructure and improved it. The improved network structure in this paper is shown in Figure 2, which consists of a backbone and a head network.

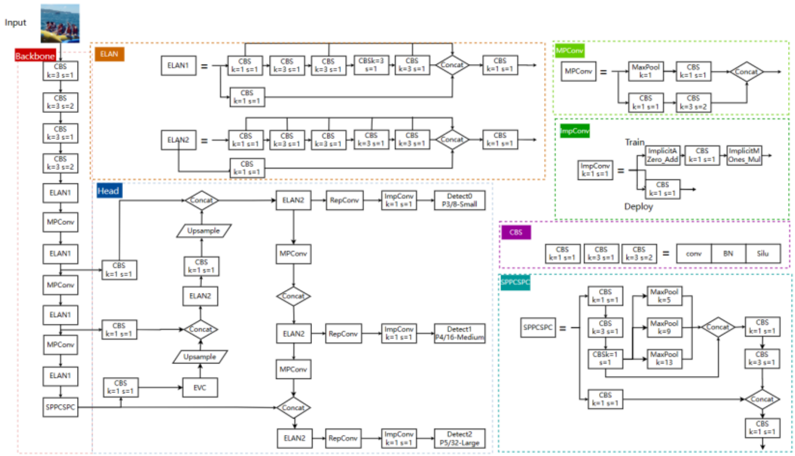

Figure 2. The network structure of E-Yolov7.

The backbone network is used to extract image features. This part follows the backbone network of Yolov7, which is mainly composed of four modules: CBS (Conv-BN-SiLU), ELAN1, MPConv, and SPPCSPC. Besides, There are some changes in the head network. We added ECV block to the top-level feature map.

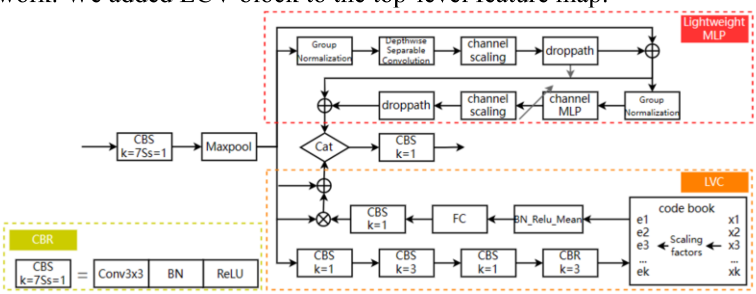

Figure 3. The structure of EVC block.

The new EVC block originated from the CFP network [11], and is mainly composed of lightweight MLP module and LVC (learnable Visual Center) module. The structure of it is shown in Figure 3. Firstly, the obtained feature map is smoothed by a CBS module, and then the obtained feature X_{in} is used as input to the lightweight MLP module and the LVC module through the maxpool. lightweight MLP mainly consists of two residual blocks, one is a residual block based on DWConv and the other is a residual block based on Channel MLP. Specifically, X_{in} is solved by Group Normalization for group normalization, and then the channel number is changed after Depthwise Separable Convolution, and then DropPath randomly drops some branches to improve the generalization ability and robustness of the network.

The obtained result is concated with X_{in} on the channel to get the output $LMLP\_DW_{out}$ of the first residual block, and it is used as the input of the second residual block. In the second residual block, $LMLP\_DW_{out}$ undergoes Group Normalization, then fuses the channel information through the Channel MLP module, then changes the number of channels and performs DropPath to randomly discard some of the branches, and then concat the result with $LMLP\_DW_{out}$ on the channel to obtain the output $LMLP_{out}$ of the lightweight MLP. The output of lightweight MLP is $LMLP_{out}$. The LVC module is an encoder with a fixed codebook. It consists of a fixed codebook and a set of learnable Scaling Factors. Specifically, X_{in} is first encoded by a convolutional group containing 1x1, 3x3, 1x1, and then grouped by the CBR module, the result obtained. After that, the result is fed into the codebook and a set of scaling factors is used to obtain the encoding result, the k-th codeword can be calculated by Eq. (1).

$$e_k = \sum_{i=1}^{N} \frac{e^{-s_k \|x_i - b_k\|^2}}{\sum_{j=1}^{K} e^{-s_k \|x_i - b_k\|^2}} (x_i - b_k) \tag{1}$$

$$e = \sum_{k=1}^{K} \phi(e_k) \tag{2}$$

Where N is the total spatial number of input features, s_k is the k-th scaling factor, x_i is the k-th pixel point, and b_k is the k-th learnable visual codeword, then x_i-b_k is the relative position information of both, and K is the total number of visual centers. Then use Eq. (2) to get the whole image about k codewords all the information e, where contains BN layer with ReLU and mean layer, and then put the e input to the fully connected layer, after connecting a 1x1 convolution, and then multiply the obtained result with X_{in} on the channel, and then add to get the output LVC_{out} , and finally splice $LMLP_{out}$ and LVC_{out} on the channel and then pass through a 1x1 CBS to get the output EVC_{out} of the EVC module.

3. Experiment

3.1. Experimental Environment and Evaluation Indicators

The training task is implemented on Python 3.8 and Pytorch 2.0.1. The device information for training includes NVIDIA RTX A4000, and 15.73GB of memory.

The experiments are conducted using end-to-end training, for the fairness of the experiments, all the experiments in this case do not use pre-trained weights. The experiments are conducted using data augmentation such as mosaic, mixup, and so on. To assess the performance of the improved model, we use four evaluation metrics, including precision, recall, mAP and F1 score.

3.2. Comparison with Other Object Detection Models

In order to prove that the improved model in our work is effective, we compare the method proposed in this paper with the current mainstream Yolo series of object detection algorithms, such as Yolov5s, Yolov7-tiny, Yolov7x, Yolov7. The experimental results are shown in the Table 2, from which we can see that the precision of the improved model in this paper has 3.7% improvement compared with the results of the original model, 0.3% improvement in recall, 2.5% improvement in mAP, and 2% improvement in F1 score. And compared with other models in the experiment, the improved model in this paper is higher than other models except that Recall is lower than Yolov7x. Besides, The pr-curves of the proposed method and other target detection models are shown in Figure 4, from which we can see that E-Yolov7 is superior to several others.

It can be seen that the introduction of EVC module enables the model to better learn corner area information and long-range dependencies, which can improve the model's ability to extract features, and also proves that the text improvement is effective.

Table 2. The results of proposed method compared with other object detection models

method	p	R	mAP	F1
Yolov5s[16]	0.806	0.775	0.832	0.790
Yolov7-tiny[20]	0.790	0.776	0.818	0.780
Yolov7x[20]	0.832	**0.836**	0.873	0.830
Yolov7[20]	0.856	0.831	0.877	0.840
E-Yolov7	**0.893**	0.834	**0.902**	**0.860**

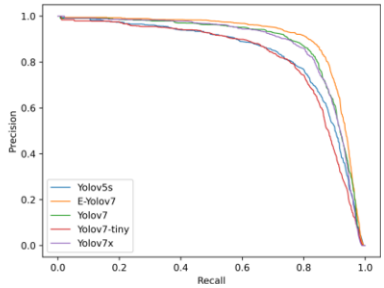

Figure 4. The pr-curves of the proposed method and other target detection models

3.3. Prediction Results of Yolov7 and E-Yolov7

(a)
(b)


Figure 5. The results of using Yolov7 and E-Yolov7 to predict the dataset presented in this paper.

On the dataset made in this study, Yolov7 and E-Yolov7 were used for prediction, the results are shown in the figure 5. Where (a) is the result of Yolov7 prediction, (b) is the result of E-Yolov7 prediction, through comparison, the improved method in this paper can reduce the false detection and missed detection, which proves that this study is effective for the detection of life jackets.

4. Conclusion

This study proposes a detection framework based on deep learning algorithm to determine whether people are wearing life jackets. In view of the problem of insufficient sea rescue datasets, we collected and annotated life jacket datasets. Using the current mature Yolov7 as the benchmark network, and the feature extraction capability of the model is improved by adding an EVC block at the top level of the feature pyramid. The improved model in this paper can obtain information on whether people are wearing life jackets and people's location information, through which it is convenient for administrators to manage marine or water projects, facilitate supervision of whether people wear life jackets as required, and timely determine their location information and remind tourists who are not wearing life jackets. The life jacket detection model proposed in this paper proves that the method proposed in this paper is effective and meaningful through a series of experimental evaluations and comparisons.

In addition, life jacket inspection faces many challenges, firstly, the time requirements for detection, which require our model to be lightweight enough to achieve real-time results. Secondly, models need to be universal to adapt to different environments and data sets. The performance of the model presented in this article may degrade when applied to scenarios that are significantly different from the training dataset. Despite these challenges, our findings suggest that the improved model proposed in this paper has great potential to solve real-world problems. In the future, we plan to improve the detection accuracy and speed of life jacket inspection.

References

[1] Cheng Ye, Ji Li. Life Jacket Wearing Recognition Based on Semantic Segmentation. In: 2022 9th International Conference on Digital Home (ICDH). 2022; p. 141–146.
[2] Chenguang Liu, Xiumin Chu, Shuo Xie, et al. Multi-target Locating Method of Surface Ship Based on Monocular Vision. Journal of Traffic and Transportation Engineering. 2015;15(05):91-100.
[3] Junjie Zhang, Shuyan Ding, Lunbo Li. NAO Robot Player Recognition Based on Haar-like Features and Color Features. Computer And Modernization. 2017;(2):30-35.
[4] Hongwei Fan, Deshun Hu, Shuoqi Gao, et al. Intelligent Recognition Method and Experimental Verification of Ferrography Image Based on Deep Belief Network. Lubrication Engineering. 2021; 46(7):15-22.
[5] Xuesong Yang, Biye Cai, Jianming Zhang, et al. Life Jacket Detection Algorithm Based on HSV Color Feature and Contour Area. Computer Engineering and Applications. 2016;52(03):184-188+210.
[6] Mengjia Liu, Hui Feng, Haixiang Xu, et al. CNN-based Multi-object Detection Experiment for Search and Rescue Unmanned Boat on Water Surface. Journal of Wuhan University of Technology (Transportation Science & Engineering).2019;43(05):910-913+919.
[7] Qinglong Weng, Xiaoxiao Du, Kun Zhang. Deep Learning-based Law Enforcement Algorithm For Crew Life Jacket Detection with Color Features and Contour Areas. Chana Water Transport. 2023;(04 vo 23):42-43+46.
[8] Guang Han, Wang Zhou, Ning Sun, Jixin Liu and Xiaofei Li. Feature Fusion and Adversary Occlusion Networks for Object Detection. IEEE Access. 2019;7:124854-124865.
[9] Chunluan Zhou, Junsong Yuan. Occlusion Pattern Discovery for Object Detection and Occlusion Reasoning. IEEE Trans Circuits Syst Video Technol. 2020;2067-2080.
[10] Xuexue Li, Wenhui Diao, Yongqiang Mao, Peng Gao, et al. OGMN: Occlusion-guided Multi-task Network for Object Detection in UAV Images. ISPRS Journal of Photogrammetry and Remote Sensing. 2023;199:242–257.
[11] Yu Quan, Dong Zhang, Liyan Zhang, Jinhui Tang. Centralized Feature Pyramid for Object Detection. IEEE Trans on Image Process. 2023;32:4341–4354.
[12] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You Only Look Once: Unified, Real-time Object Detection. Computer vision and pattern recognition. 2016;779-788.
[13] Joseph Redmon, and Ali Farhadi. YOLO9000: Better, Faster, Stronger. arXiv; arXiv preprint arXiv:1612. 08242,2016.
[14] Joseph Redmon and Ali Farhadi. Yolov3: An Incremental Improvement. arXiv preprint arXiv:1804. 02767, 2018. 1, 2, 3.
[15] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. Yolov4: Optimal Speed and Accuracy of Object Detection. arXiv preprint arXiv:2004. 10934, 2020. 1, 2, 3, 6.
[16] Glenn Jocher, et al. Yolov5. https://github. com/ ultralytics/yolov5, 2021. 1, 2, 3, 5, 6.
[17] Zheng Ge, Songtao Liu, Feng Wang, Zeming Li, and Jian Sun. YOLOX: Exceeding YOLO series in 2021. arXiv preprint arXiv:2107. 08430, 2021. 1, 2, 7, 10.
[18] Xin Huang, Xinxin Wang, Wenyu Lv, Xiaying Bai, et al. Pp-yolov2: A Practical Object Detector. arXiv preprint arXiv:2104. 10419, 2021. 3, 6.
[19] Chuyi Li, Lulu Li, Hongliang Jiang, Kaiheng Weng, et al. YOLOv6: A Single-Stage Object Detection Framework for Industrial Applications. arXiv preprint arXiv: 2209. 02976,2022.
[20] Chien-Yao Wang, Alexey Bochkovskiy, and Hong-Yuan Mark Liao. YOLOv7: Trainable Bag-of-freebies Sets New State-of-the-art For Real-time Object Detectors. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, CB, Canada, 18–22 June 2023; pp. 7464–7475.