ISSN:3007-9640 DOI: https://doi.org/10.70114/acmsr.2025.2.1.P377

Sea-Surface Ship Detection and Embedded Deployment in SAR Images Based on Improved YOLOv5

Lei Fang[1, a]*, Gao Sun[1, b], Tianqi Wang[1, c], Xiao Wang[1, d], Hongwen Dong[1, e], Biwu Chen[1, f]

[1]Department of Shanghai Radio Equipment Research Institute, Shanghai, China.

[a] 18296971027@163.com, [b] gao_sun@163.com, [c] tianqiwang205@163.com,

[d] xiaowanghit@163.com, [e] donghw602@163.com, [f] cbw_think@whu.edu.cn

Abstract. The detection of sea-surface ships represents a crucial means by which countries can compete for marine resources in the context of the current international strategic environment. The ongoing advancement of deep learning has led to a notable enhancement in the accuracy of target detection tasks involving surface ships, particularly when these are conducted using intelligent algorithms. Concurrently, the depth and width of target detection network models are expanding, accompanied by an increase in complexity. This has resulted in a challenge for deploying models that are limited by their own computational complexity in real surface ship target detection tasks. The field programmable gate array (FPGA) offers a solution to the issues of complex target detection network structures, high computational resource consumption and high storage space requirements. The FPGA's high parallelism and customizability make it an ideal choice for addressing these challenges. In light of the aforementioned background and problems, this paper puts forth an enhanced YOLOv5 algorithm for the detection and identification of SAR targets associated with sea-surface ships. To this end, the network model has been converted from floating-point to 8-bit integer parameter distribution through the optimization and quantification of the target detection network model. This enables the embedded deployment of the target detection and identification algorithm in real scenarios and tasks based on FPGA. The implementation of target detection and recognition algorithms in real scenarios and tasks.

Keywords: Sea-surface Ship; Target Detection; FPGA; Embedded Deployment.

1. Introduction

In the field of maritime security monitoring and military reconnaissance, the detection of targets on surface vessels has consistently been a pivotal technology. In recent years, the advancement of artificial intelligence has been marked by the continuous development of disciplines, particularly the ongoing improvement and breakthrough of deep learning algorithms. Among these, the YOLOv5 algorithm, a single-stage intelligent algorithm in the field of target detection and recognition, has emerged as a research focus in the domain of sea surface ship detection and recognition. This is due to its high computational efficiency and rapid recognition speed, achieved at the cost of reduced computational precision.

In the field of target detection and recognition, the single-stage YOLO series network detection algorithm model offers a more efficient approach to target localization and recognition than the two-stage intelligent algorithm, which has demonstrated enhanced detection speed but slightly reduced accuracy. Consequently, in order to enhance the precision of target detection, the network model structure of the YOLO series is evolving towards greater irregularity and complexity, accompanied by an increase in the depth and width of the network. YOLOv1 is capable of predicting multiple bounding boxes and class probabilities simultaneously through a single convolutional neural network. This network contains several convolutional layers with different convolutional kernel sizes, as well as pooling and fully connected layers [1]; The YOLOv2 model incorporates several enhancements to the YOLOv1 architecture, including batch normalization, high-resolution classifiers, multi-scale training, anchor frame mechanisms and fine-grained features. These modifications have led to a notable increase in irregularities within the network model structure, as evidenced by the findings presented in reference [2]; The YOLOv3 network employs

ISSN:3007-9640 DOI: https://doi.org/10.70114/acmsr.2025.2.1.P377
Darknet-53 as its fundamental structure, incorporates multi-scale prediction and up-sampling feature fusion techniques, and enhances the bounding box prediction and category prediction methodologies. However, the network model depth surpasses 100 layers, resulting in a progressively intricate structure [3]; YOLOv4 identifies the optimal model structure and training method through a multitude of experiments based on YOLOv3. It employs CSPDarknet53 as the backbone network and PANet as the neck. Despite achieving a balance between detection speed and accuracy, the complex network structure of YOLOv4 results in significant computational and memory resource consumption during the inference process; In contrast, YOLOv5 offers a range of model versions to accommodate varying performance requirements. It employs a more streamlined and flexible PyTorch framework, facilitating the deployment and utilization of FPGA-based embedded systems in sea surface ship target detection and recognition tasks [5].

FPGA are low-cost, low-power, highly parallel and programmable, customizable chips. Their real-time pipelined computing and high real-time performance characteristics make them suitable for on-chip development and deployment of convolutional neural network intelligent algorithms. The pipeline structure provided by FPGA can be well matched to convolutional neural network algorithms, particularly under conditions of limited logic computing resources, the design of intelligent algorithms based on FPGA is a viable approach. By accelerating the operator and operation architecture, the FPGA-based intelligent algorithm can enhance efficiency. Furthermore, through appropriate lightening and simplification of the intelligent algorithm, it can be adapted to a range of hardware environments, including embedded deployment tasks that require real-time and accuracy. The optimization of multi-channel data transmission and parameter reordering, as employed in literature [6], effectively improves the utilization of the FPGA device bus bandwidth, thereby enabling the computational performance of the intelligent algorithm to be enhanced at the hardware level. In order to reduce the off-chip access memory and the algorithm's on-chip storage, while simultaneously increasing the length of the data transfer, the literature [7,8] employed a process of retraining and quantization of the YOLO parameters, utilizing binary weights and low-bit activation operations. This resulted in the rearrangement of the weight parameters of the intelligent algorithms.

This paper presents a study of the intelligent algorithm YOLOv5 in the context of target detection and identification of surface ships. It demonstrates the on-chip deployment of this algorithm based on embedded FPGA. In order to address the specific network architecture of the YOLOv5 target identification and detection model, we have devised a basic module operator comprising a standard convolution module, a depth-separable convolution module and a CSP bottleneck structure module. This operator can be utilized within the computational processing unit of an FPGA. Subsequently, the 32-bit floating-point parameter distribution weights are converted to create a frozen model, thereby ensuring the adaptability of the algorithmic models under different training architectures. This is followed by the conversion of the 32-bit floating-point parameter distribution weights to the aforementioned frozen model. Subsequently, the floating-point weight model is quantized, thereby transforming the network model structure into an 8-bit fixed-point model. This results in a reduction in the complexity of the model, a reduction in the amount of logic, memory and bandwidth resources required by the edge device, and an increase in energy efficiency.

2. Embedded deployment methodology

2.1 Overall idea of the proposed methodology

Fig. 1 illustrates the comprehensive process of implementing the YOLOv5 algorithm with intelligent algorithms, from the training phase to the deployment phase, as described in this paper. Firstly, the design, training and optimization of the algorithm model is carried out in order to obtain the accuracy of the network training model, ensuring that it meets the actual task requirements. Subsequently, the quantization of the model is achieved, with the FPGA processor quantizing the 32-bit float model into an 8-bit int model. At the same time, a simulation test is carried out in order

ISSN:3007-9640 DOI: https://doi.org/10.70114/acmsr.2025.2.1.P377

to meet the performance index. Finally, if the Subsequently, if the quantized model is found to have passed the simulation test and met the requisite task requirements, the corresponding deployment quantization design coding must then be carried out in order to complete the engineering quantization process and generate the deployment file. Finally, the actual test of the deployment performance is carried out, and the deployment accuracy and operational performance of the model are returned according to the AI calculation results.

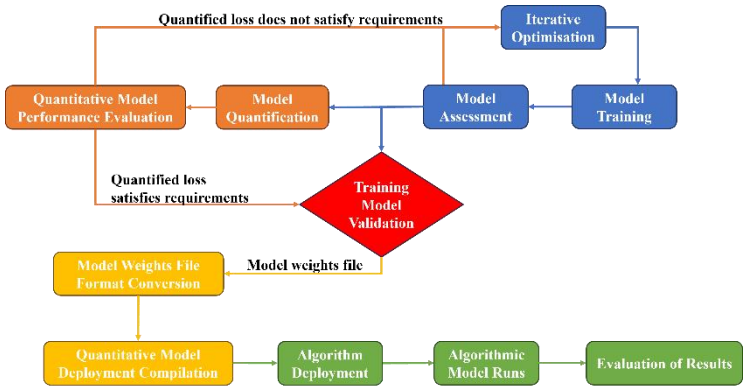

Fig. 1 Embedded Deployment Implementation Overall Process

2.2 Detailed process for proposing the methodology

Firstly, this paper employs the YOLOv5 target detection and recognition algorithm for the training of the model on surface ship SAR images. The network structure model is trained and tested using the aforementioned algorithm. Figure 2 illustrates the overall structural framework of this paper's improved YOLOv5 model, which comprises the backbone network module, the neck module, and the detection head module. In this network structure, two avenues for improvement have been identified. Firstly, modifying the hyper-parameter convolutional layer parameters and increasing the small target detection layer can enhance the model accuracy without significantly depleting on-chip resources. Secondly, optimizing the SAR image dataset of sea-surface ships can also elevate target recognition accuracy.

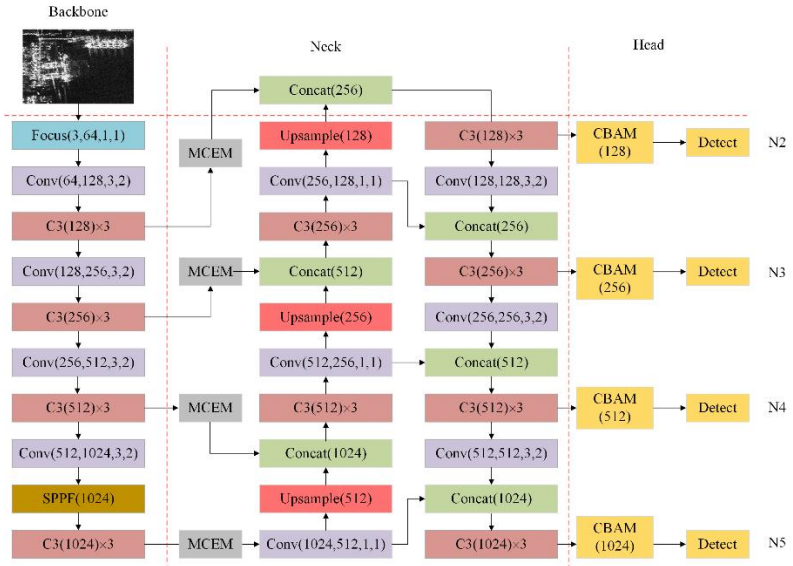

Fig. 2 Improved overall framework of the YOLOv5 algorithm

And then, the model is quantized, and the quantized network structure is constructed. The 32-bit floating-point type model weights can be converted to an 8-bit integer weight model. This entails transforming the trained YOLOv5 neural network weights, activation values, and other elements from high precision to low precision. However, it is essential to ensure that the accuracy of the

ISSN:3007-9640 DOI: https://doi.org/10.70114/acmsr.2025.2.1.P377
model remains comparable to that of the original model before the transformation. In the quantization process, the weight parameters are static values that remain unchanged once the network model is solidified. The distribution range of the values can be obtained through weight analysis, which allows for the calculation of quantization coefficients. In contrast, the activation values are dynamic, with the range of values varying with different inputs. Therefore, it is necessary to determine the distribution range of the activation values by using representative calibration data, which in turn allows for the calculation of quantization coefficients. Model quantization establishes an effective data mapping relationship between fixed-point and floating-point formats, facilitating more precise gains while minimizing accuracy losses. The conversion formula from floating point to fixed point is as follows:

$$Q = \frac{R}{S} + Z \tag{1}$$

Where R denotes the input floating-point data, Q denotes the fixed-point data after quantization, Z denotes the value of the zero point, and S denotes the quantization coefficient, which determines the mapping relationship between the floating-point and the fixed-point based on the two parameters, S and Z. In this paper, the quantization coefficient S and the zero offset value Z are solved using the maximum minimum value method, i.e.:

$$S = \frac{R_{max} - R_{min}}{Q_{max} - Q_{min}} \tag{2}$$

$$Z = Q_{max} - \frac{R_{max}}{S} \tag{3}$$

where R_{max} and R_{min} denote the maximum and minimum values of the floating-point data, and Q_{max} and Q_{min} denote the maximum and minimum values of the fixed-point data, which are 127 and -128, respectively, when 8-bit fixed-point quantization is used.In general, the numerical distribution of the model weights is similar to that of a normal distribution, with a centroid point close to 0, and the majority of the values concentrated in a smaller range on either side of the centroid point . In order to make the network structure model less computational process on FPGA, this paper assumes that the numerical distribution of the weights is the symmetric distribution centered at the 0 point, then the formula for the quantization coefficients is simplified as:

$$Q = \frac{R}{S} \tag{4}$$

$$S = \frac{\widehat{R}_{max}}{\widehat{Q}_{max}} \tag{5}$$

where $\widehat{R}_{max}$ is the maximum of the two numbers $|R_{max}|$ and $|R_{min}|$; $\widehat{Q}_{max}$ is the maximum of the two numbers $|Q_{max}|$ and $|Q_{min}|$, and for 8-bit fixed-point quantization, this value is 128.

Subsequently, an AI compiler is employed to translate the quantized neural network model into a specific file comprising binary instruction sequences, which can be recognized by the AI parallel computing unit. The nodes in the network model and the corresponding operations are analogous to the hardware logic units in the AI parallel computing unit. Through the AI compiler, the neural network model and the AI parallel computing unit can be associated. The AI compiler initially parses the optimized and quantized neural network topology into a corresponding computational graph, designated as the Intermediate Representation (IR). Subsequently, it performs the corresponding optimization to map the nodes in the computational graph into the corresponding arithmetic operations, including convolution, fully connected, activation, and others. Additionally, it utilizes the AI parallel computing unit in FPGA to the fullest extent possible. The parallelism of the AI parallel computing unit hardware in the FPGA is leveraged to the greatest extent possible, and ultimately, code that can be recognized and executed by the AI parallel computing unit is generated.

The last is the transformation of the model, mainly based on the heterogeneous AI parallel computing platform implemented on the FPGA, under the control scheduling of the CPU, the AI parallel computing unit implemented by the logic end of the example to complete the accelerated computation of the heterogeneous acceleration platform, for the sake of the system's rapid iterative upgrading, it is generally only necessary to update the model and generate the corresponding xmodel file, and then compile the completed xmodel file as the global data compiled into the CPU

ISSN:3007-9640 DOI: https://doi.org/10.70114/acmsr.2025.2.1.P377
executable program, which is submitted by the CPU to the AI parallel computing unit can complete the edge inference. The transformation of the model is to convert the compiled xmodel file into a *.o format file for the C/C++ compiler linker on the embedded deployment side to generate the final executable program.

2.3 Results

In this paper, the improved YOLOv5s model is trained using the surface ship SAR dataset, a total of 1160 SAR images of offshore and harbor surface ship targets, with the image size of 512×512, which are used to train the 32-bit floating-point model weight distribution parameters under the PyTorch architecture, and obtain a certain accuracy rate. The 32-bit floating-point weight file *.pt is then used to quantize, compile and transform the parameter distribution to obtain an 8-bit integer weight distribution file that can be deployed on-chip, and the deployable file is also used to test with the same test set to obtain the results shown in Fig. 3. A comparison of the parameter set accuracy between the 32-bit floating-point network model and the 8-bit integer deployable network model is shown in Table 1.

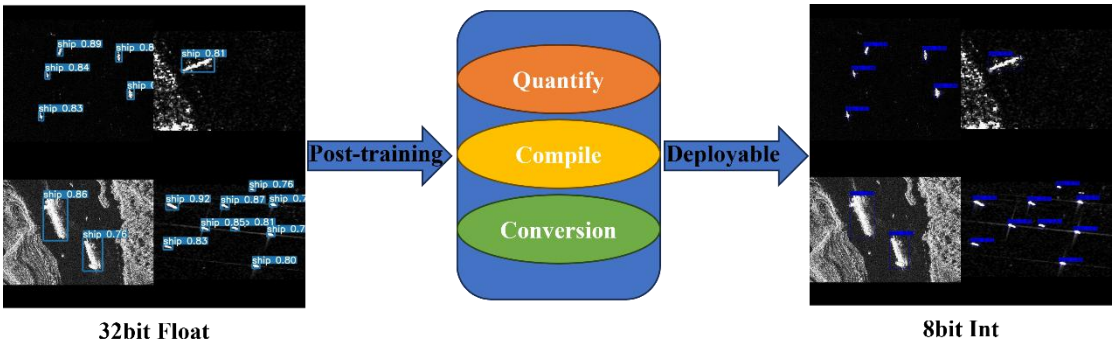

Fig. 3 Comparison of results before and after embedded deployment

The results show that the weight format file after quantization, compilation and transformation also has excellent performance in the SAR target detection and recognition task of surface ships, and the average accuracy error is less than 3%, which indicates that after the 32-bit floating-point network model parameter distribution trained in the host computer is migrated to the embedded edge-on-chip device, under the conditions of limited resources, memory and bandwidth, the improved YOLOv5 algorithm is able to obtain a large gain in target detection and recognition effectiveness at the cost of a small loss in accuracy.

Table 1. Comparison of parameters and accuracy before and after embedded deployment

Model Type	Layer Numbers	Operation(GFLOPs)	Inference(FPS)	Precision	Recall	mAP
32bit Float	134	11.816596	10.5	0.991	0.849	0.973
8bit Int	134	11.816596	3.1	0.995	0.733	0.966

3. Summary

This paper proposes an embedded deployment method for SAR image target detection and recognition of surface ships based on the improved YOLOv5 algorithm. 32-bit floating-point weight file parameters are obtained by optimizing and training the improved YOLOv5s network structure model, and 8-bit integer weight distribution of the weight file is obtained by the FPGA quantization, compilation and transformation method based on the parallel computation unit proposed herein, and it is tested on the embedded deployment model of surface ships. The network model weight parameter distribution, which will be tested for embedded deployment model on surface ships, the results show that under the limited on-chip resource conditions, the method can obtain better target detection and identification gains at the expense of smaller accuracy, with an average error of 3%. The embedded deployment method based on intelligent target detection and

ISSN:3007-9640 DOI: https://doi.org/10.70114/acmsr.2025.2.1.P377
recognition algorithm proposed in this paper has a greater research value in edge device integration of sea surface ship target detection and recognition tasks or other target detection and recognition tasks due to its good accuracy results and consumption of small on-chip resources.

References

[1] M. Hussain, YOLOv1 to v8: Unveiling Each Variant–A Comprehensive Review of YOLO, in IEEE Access, 2024, 12: 42816-42833.

[2] R. Li and J. Yang, Improved YOLOv2 Object Detection Model, 2018 6th International Conference on Multimedia Computing and Systems (ICMCS), Rabat, Morocco, 2018, 1-6.

[3] K. Wu, C. Bai, D. Wang, Z. Liu, T. Huang and H. Zheng, Improved Object Detection Algorithm of YOLOv3 Remote Sensing Image, in *IEEE Access*, 2021, 9: 113889-113900.

[4] C. Kumar B., R. Punitha and Mohana, YOLOv3 and YOLOv4: Multiple Object Detection for Surveillance Applications, *2020 Third International Conference on Smart Systems and Inventive Technology (ICSSIT)*, 2020, 1316-1321.

[5] M. Alameri and Q. A. Memon, YOLOv5 Integrated With Recurrent Network for Object Tracking: Experimental Results From a Hardware Platform, in *IEEE Access*, 2024, 12: 119733-119742.

[6] R. A. Amin, M. Hasan, V. Wiese and R. Obermaisser, FPGA-Based Real-Time Object Detection and Classification System Using YOLO for Edge Computing, in *IEEE Access*, 2024, 12:73268-73278.

[7] S. Zhang, J. Cao, Q. Zhang, Q. Zhang, Y. Zhang and Y. Wang, An FPGA-Based Reconfigurable CNN Accelerator for YOLO, *2020 IEEE 3rd International Conference on Electronics Technology (ICET)*, Chengdu, China, 2020, 74-78.

[8] L. Zhang and Z. Pan, "Design Implementation of FPGA-Based Neural Network Acceleration," *2024 4th International Conference on Consumer Electronics and Computer Engineering (ICCECE)*, Guangzhou, China, 2024, 163-166.