Research on Innovative Methods and Implementation Paths of Ship Target Detection in Deep Learning

Kaizhi Li *
University of Glasgow, Glasgow, UK
**Corresponding Author.*

Abstract:
In order to improve the recognition accuracy of ship targets in SAR images, this paper adopts a data-driven deep learning ship detection scheme, and attempts to use YOLOv5, YOLOv7, and YOLOv7-X to recognize ship targets in the HRSID dataset and a self-constructed HRSID dataset enhanced for small targets, and finally presents them in a visual form. While recognizing images, it is also possible to recognize ship targets in videos and real-time cameras. This article mainly explores in depth from four aspects: selection and reinforcement of datasets, selection of deep learning algorithms, Python scripts and visual interfaces for assisting model training. This article aims to simplify the repetitive operations in the experimental process and further enhance the convenience of detection. Relevant Python scripts and visual interfaces have been designed and written to achieve auxiliary functions such as batch and random addition of small targets in images, stacking of real boxes and recognition boxes of test samples, and improving the efficiency of ship detection. The experimental results in the article show that using YOLOv7-X and the enhanced HRSID dataset can achieve a recognition accuracy of 92.53%, and also have good recognition effects on small targets and ship targets in complex backgrounds such as ports. Compared with traditional ship detection methods, there is a significant improvement in recognition accuracy.

Keywords: ship detection: SAR image, HRSID dataset, YOLOv7, visualization

INTRODUCTION

The People's Republic of China has a coastline of 18000 kilometers and a claimed jurisdictional over sea areas of 3 million square kilometers. It is a land sea composite power that borders the Eurasian continent to the west and the Pacific Ocean to the east. It has many natural deep-water ports and extremely abundant marine resources. In the late 19th and early 20th centuries, with the development of the Industrial Revolution, advances in transportation and communication technology, and reforms in various trade policies, trade between countries around the world began to grow rapidly, gradually forming a wave of trade globalization. As a result, the status of the ocean continues to rise, and the volume of goods trade between countries through the ocean has repeatedly reached new highs. According to statistics from the World Trade Organization (WTO), over 80% of international trade in goods is achieved through ocean transportation. China has been the world's largest country in goods trade for six consecutive years, and attaching importance to maritime traffic management and protecting China's maritime rights and interests has become a necessary condition for comprehensively building a socialist modernized strong country [1]. At the same time, under the influence of various economic interests, geopolitical factors, and other factors, the security of China's territorial seas faces a complex and intricate pattern, full of various potential dangers and challenges. The ship target detection system can monitor ship targets at sea and ensuring the safety of ports, providing rapid emergency response, and effectively protecting marine ecology and resources. Therefore, continuously strengthening ship detection technology plays a crucial role in safeguarding China's maritime rights and improving port governance efficiency, and requires researchers to explore it in depth.

The initial ship detection method was based on visual inspection of optical images, which required operators to manually identify ships by observing aerial photographs or satellite images. This resulted in very low detection efficiency and accuracy, which was subjectively influenced by operators, without objective and accurate standards. As shown in Figure 1 the optical imaging method cannot achieve large-scale and real-time monitoring and is also limited by factors such as weather, lighting, and image resolution, making it difficult to accurately detect ships or small vessels in complex scenes.

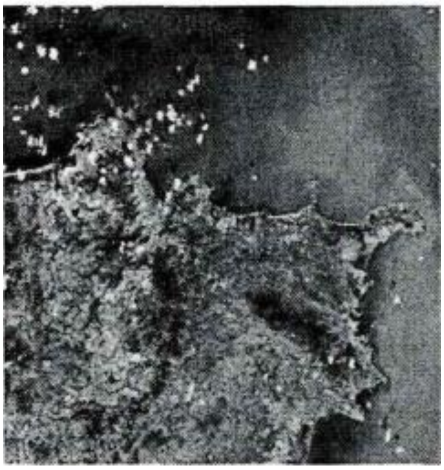


Figure 1. Schematic diagram of optical image in Weihai area. Shandong Province

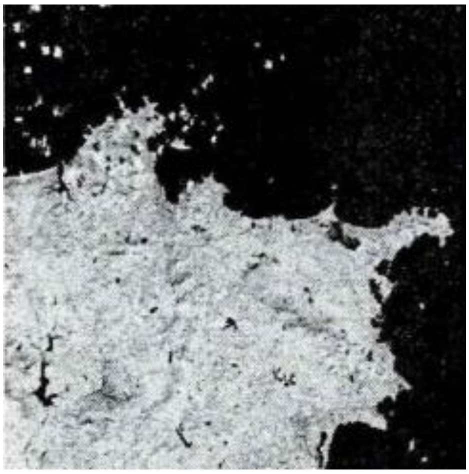


Figure 2. Schematic diagram of near-infrared image in Weihai area. Shandong Province

Researchers have started using infrared to detect ship targets in response to nighttime and low visibility conditions. The near-infrared image is shown in Figure 2 and its strong anti-interference ability further expands the detection range of the observer. Observers determine the outline and position of a ship by detecting it's heal distribution. However, when there is a temperature change within the detection range, infrared radiation will be affected by the temperature difference, producing a large amount of noise, which will affect the clarity and quality of the image. Meanwhile, the color, material, geometric shape, and other characteristics of the ship's surface may affect the absorption, reflection, and scattering of infrared waves. Therefore, infrared radiation cannot accurately detect ship targets in real-time around the clock.

In response to the above limitations, researchers tend to use radar to detect ship targets, which can effectively solve problems such as climate, temperature, long-distance detection, and real-time performance. The predecessor of the radar was the concept of a planar angle measuring radar proposed by British scientist Marconi, which could prevent ships from colliding with each other. In 1936. The British built the first radar station (Radio Detection and Ranging). The detection principle of early radar was to obtain the relative distance between the measured object and the radar station by measuring and calculating the echo delay. But it was still limited by the low resolution of the radar at that time and the susceptibility to interference from terrain, waves, and other interferences. Therefore, radar is not good at detecting targets in complex situations. Since the 1980s. Detecting ship targets through Synthetic Aperture Radar (SAR) images has become the preferred method for surveyors. Firstly. SAR inherits the advantages of radar not being affected by weather conditions, temperature and climate, and light intensity, and has high resolution and long-distance observation, making it the best choice for ship detection. It cannot be ignored that SAR has high operating and maintenance costs, and SAR data processing is complex, requiring personnel with professional knowledge to perform appropriate operations. At the same time, due to the limited resolution of SAR images at that time, even the clearest SAR images could not accurately recognize some complex scenes or dense small ships.

With the advent of the intelligent era, researchers have begun to improve and expand the methods of ship target detection based on intelligent, high-precision scientific technology and high-performance computers. By improving traditional methods and utilizing various algorithms, researching deep learning models and parameters, continuously improving the accuracy and efficiency of object detection, and striving towards automated ship target precise detection.

Traditional statistical feature detection methods mainly rely on statistical analysis of target features in images, mapping from the original image to a high data space, which includes: detecting specific colors of ships and identifying them through color features; Detecting specific textures of ships and identifying them through texture features; Detecting specific shapes of ships and identifying them through shape features; Ships usually have clear boundaries, and edge detection algorithms are used to detect the contours of ships in images. For example. Chen etc al [2]. (2021) proposed a method for identifying targets based on the specific color of the detected ship, reflecting candidate regions in the HSV color space, and inferring the direction of target movement based on color features. In 2008. Grabner etc al [3] chose to use texture feature 4. Such as Local Binary Pattern (LBP) which converts the pixel values in each small area into binary data based on the grayscale relationship between the central pixel and adjacent pixels. The binary data is then arranged in a clockwise order of 5 or counterclockwise as a string of binary numbers, and converted to decimal as the LBP feature value of the pixel. In 1996, Cameron [4] further developed a classification method based on coherent target decomposition theory, extracting the maximum symmetric scattering component in every 5 resolution units and dividing them into six basic types of symmetric scatterers (such as dihedral angle, trihedral angle, dipole, etc.). In the same year. Cloude summarized the theory of polarization target decomposition, which included a polarization entropy calculation method based on polarization coherence matrix decomposition. Therefore, for scattering targets that satisfy reciprocity characteristics, the polarization measurement vector is expressed in Pauli basis as shown in formula 1:

$$K = \frac{1}{\sqrt{2}} [S_{hh} + S_w S_w - S_{hh} 2 S_{hv}]^T \tag{1}$$

The polarization coherence matrix and eigen decomposition are shown in formula 2:

$$T = K * K^{*T} = \sum_{i=1}^{3} \lambda_i \, (\bar{e_i} - \bar{e}_i^{*T}) \tag{2}$$

Note: It Ais the eigenvalue of the coherence matrix; Ah is a feature vector.

Introducing polarization entropy to calculate the logarithmic arithmetic sum of eigenvalues to represent the ratio between eigenvalues M, as shown in formulas 3 and 4:

$$H = -\sum_{i=1}^{3} P_i \, log_3 \, P_i \tag{3}$$

$$P_i = \frac{\lambda_i}{\sum_{j=1}^{3} \lambda_j} \tag{4}$$

Intelligent detection based on deep learning refers to the use of deep learning algorithms to detect ship targets. After decades of development, currently, deep learning based object detection algorithms can be divided into two types: one-stage object detection and two-stage object detection. Representative two-stage detectors include RCNN. FastRCNN [5] and FasterRCNN. These algorithms first use deep convolutional neural networks to extract features, then generate a series of region suggestions that may contain the target, and further accurately classify and locate the target by line-tuning the bounding box coordinates of these regions. At present, the development focus of ship recognition direction is on improving target detection accuracy, reducing missed detection rate, eliminating the influence of complex scenes, enhancing real-time performance, reducing model size, and strengthening for small targets. This technology combines machine vision with deep learning techniques, continuously improving the accuracy and efficiency of object detection through the exploration of various algorithms. It relies on neural networks to automatically learn effective features from massive data, integrates feature extraction with classifier design, and achieves end-to-end detection. This method is beneficial for detecting large-scale and high-dimensional data. In the face of increasingly complex scenarios, continuous optimization of remote sensing target detection algorithms based on deep learning has become an inevitable trend in development. Domestic and foreign researchers arc jointly promoting the continuous innovation and progress of deep learning based object detection methods in terms of accuracy and real-time performance.

Having a mature ship target detection model, the article monitor the surrounding area in real-time and accurately will help improve the management efficiency and security of the port. It adopts a data-driven artificial intelligence ship detection method. The current ship target detection system has low accuracy in recognizing small targets and complex background ship targets, and there is no visual interface to intuitively display the recognition effect. This article attempts to use YOLOv5. YOLOv7. And YOLOv7-X to recognize ship targets in a self-built HRSID dataset that enhances small targets and complex backgrounds, and ultimately presents it in a visual form. The innovation of this article lies in dataset selection and reinforcement, deep learning algorithm models, and program visualization.

There are innovation points as the following: Firstly, In terms of dataset selection and reinforcement, this article compares multiple SAR datasets containing ship targets and selects the IIRSID dataset, and classifies the HRSID dataset into sea surface class, background port and shore class, and large ship class. In addition. 18 small targets near the shore and port were selected, randomly rotated and repositioned, and added to the initial dataset to enhance its specificity. Take the initial HRSID dataset as Dataset I and the enhanced HRSID dataset as Dataset 2. After comparative experiments, it has been proven that the enhanced dataset is beneficial for detecting ship targets. Secondly, in terms of deep learning algorithm models, this article selects and deeply understands the operating logic of YOLOv5. YOLOv7, and YOLOv7-X algorithms after comparing various deep learning algorithms. The focus is on the innovation of YOLOv7 compared to previous YOLO versions, including modules such as ELAN. SPPCSPC. RepConv. etc. After combining with episode, it was successfully proven through six rounds of experiments that YOLOv7-X is more suitable for ship target detection, further improving the final recognition accuracy of the model for small targets and complex port situations. Thirdly, in terms of program visualization, this article designs a login management interface, a SAR image visualization detection interface, a SAR image video visualization detection interface, and real-time camera detection of ship target SAR images to more intuitively demonstrate the accuracy of the model in identifying ships. By visualizing the interface of ship target recognition programs based on deep learning, ship targets can be accurately, quickly, and conveniently detected.

EXPERIMENTAL METHODS OF SAR IMAGE TARGET DETECTION METHOD BASED ON YOLOV7

YOLOv7

YOLOv7 is a product that innovatively upgrades key parts such as the model backbone and network architecture based on previous versions of YOLO. It accuracy and running speed of YOLOv7 exceed those of previous YOLO versions, as shown in Figure 3 which compares the average accuracy and prediction efficiency of YOLOv7 with the previous two YOLO models and Table 1. which contains information on the parameter count, computational complexity, image size, and accuracy of each YOLO version l2^. YOLOv7 is the detection method with the fastest prediction efficiency and highest average accuracy among previous YOLO models.

Improved Parts of YOLOv7 Compared to Previous YOLO Versions

YOLOv7 adopts a multi-branch stacked structure for feature extraction in both the backbone and enhanced feature extraction parts of the algorithm. Compared with the previous YOLO model, this structure is more dense and adopts the latest down sampling style structure at that time, using maximum pooling and 2*2 step size feature parallelism for feature extraction and compression.

In addition. YOLOv7 also introduces a special SPP structure, namely SPPCSPC. which can reduce the computational load by half and accelerate the running speed. Among them, the SPP structure with CSP structure is used to expand the receptive field, increase the network's understanding of a wider range of input image regions, and use large residual edges to assist in optimization and feature extraction.

YOLOv7 adopts an adaptive multi positive sample matching strategy. Unlike previous versions, during the training process, each real box no longer corresponds to only one positive sample, but can be predicted by multiple prior boxes. This improvement aims to enhance the training efficiency of the model. For each real box. the algorithm will determine the loss function based on the category and IOU calculation between the adjusted predicted box and the real box in order to find the most suitable prior box for that real box.

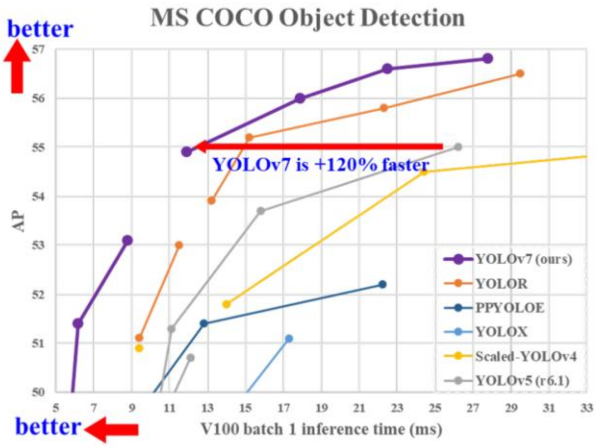


Figure 3. Comparison of average accuracy and prediction efficiency between YOLOv7 and previous excellent YOLO models

Table 1. Relevant information of YOLO algorithm in various versions

Model	#Param.	FLOPs	Size	APval	APval50	APval75	APvalS	APvalM	ApvalL
YOLOv4	64.4M	142.8G	640	49.70%	68.20%	54.30%	32.90%	54.80%	63.70%
YOLOR-u5(r6.1)	46.5M	109.1G	640	50.20%	68.70%	54.60%	33.20%	55.50%	63.70%
YOLOv4-CSP	52.9M	120.4G	640	50.30%	68.60%	54.90%	34.20%	55.60%	65.10%
YOLOR-CSP	52.9M	120.4G	640	50.80%	69.50%	55.30%	33.70%	56.00%	65.40%
YOLOv7	36.9M	104.7G	640	51.20%	69.70%	55.50%	35.20%	56.00%	66.70%
improvement	-43%	-15%	-	+0.4	+0.2	+0.2	+1.5	=	+1.3
YOLOR-CSP-X	96.6M	226.8G	640	52.7%	71.3%	57.4%	36.3%	57.5%	68.3%
YOLOv7-X	71.3M	189.9G	640	52.9%	71.1%	57.5%	36.9%	57.7%	68.6%
improvement	-36%	-19%	-	+0.2	-0.2	+0.1	+0.6	+0.2	+0.3
YOLOv4-tiny	6.1	6.9	416	24.9%	42.1%	25.7%	8.7%	28.4%	39.2%
YOLOv7-tiny	6.2	5.8	416	35.2%	52.8%	37.3%	15.7%	38.0%	53.4%
improvement	+2%	-19%	-	+10.3	+10.7	+11.6	+7.0	+9.6	+14.2
YOLOv4-tiny-3l	8.7	5.2	320	30.8%	47.3%	32.2%	10.9%	31.9%	51.5%
YOLOv7-tiny	6.2	3.5	320	30.8%	47.3%	32.2%	10.0%	31.9%	52.2%
improvement	-39%	-49%	-	=	=	=	-0.9	=	+0.7
YOLOR-E6	115.8M	683.2G	1280	55.7%	73.2%	60.7%	40.1%	60.4%	69.2%
YOLOv7-E6	97.2M	515.2G	1280	55.9%	73.5%	61.1%	40.6%	60.3%	70.0%
improvement	-19%	-33%	-	+0.2	+0.3	+0.4	+0.5	-0.1	+0.8
YOLOR-D6	151.7M	935.6G	1280	56.1%	73.9%	61.2%	42.4%	60.5%	69.9%
YOLOv7-D6	154.7M	806.8G	1280	56.3%	73.8%	61.4%	41.3%	60.6%	70.1%
YOLOv7-E6E	151.7M	843.2G	1280	56.8%	74.4%	62.1%	40.8%	62.1%	70.6%
improvement	=	-11%	-	+0.7	+0.5	+0.9	-1.6	+1.6	+0.7

YOLOv7 also incorporates the structural design of RepVGG. Integrating RepConv and modules into specific parts of the network to reduce the network's parameter count and computational cost. RepVGG is a lightweight convolutional neural network architecture.

As part of the YOLO series. YOLOv7 has undergone improvements in its overall architecture, but has not undergone significant innovation. Its overall framework can still be divided into three main parts: the backbone feature extraction network, the enhanced feature extraction network, and YOLO Head.

Feasibility Analysis

Because the YOLO series algorithm is currently one of the most popular object detection algorithms, it not only has extremely fast detection speed, but also has high detection quality. The YOLO series algorithm has now developed to YOLOv9 version, and YOLOv7 technology' is relatively mature with a stable model, making it suitable for this ship target recognition project. In addition, the YOLO series code is all open-source algorithms, so it has a wide audience, strong stability, and fast update frequency, making it suitable for this ship target recognition project.

Cost analysis

Given the above introduction to YOLOv7 and YOLO technology, as it is an open source project, the main deployment costs arc model reinforcement costs, dataset collection and reinforcement costs, visualization costs, storage costs, and server deployment costs.

Model reinforcement cost: Duc to YOLO model being one of the most popular deep learning models at present, it has a high technical barrier and requires high technical requirements for algorithm modification. Therefore, there is a lack of high-quality technical personnel, and the training period is long, resulting in high training costs. The model is still constantly innovating and improving, which requires technical personnel to keep up with the times and requires a large amount of resources to ensure that the model remains at the forefront of the industry.

The cost of dataset collection and reinforcement: Due to the fact that the detection results of deep learning models are generally proportional to the quantity and quality of the dataset.

Therefore, if you want to achieve high recognition accuracy, you need to provide a high-quality amount of data. SAR images require professional radar equipment and corresponding ground stations, which require high costs, complex technology, and high-quality technical personnel.

Therefore, seeking professional companies for business cooperation or collecting and utilizing publicly available datasets is a better choice.

Training cost: A deep learning based object recognition model requires multiple sets of comparative experiments, with hundreds or thousands of training sessions each time, in order to obtain data parameters that are more suitable for detecting the target. Therefore, strong computing power is needed as the foundation.

Development prospect analysis and prediction

The use of Yolo model for deep learning based ship target recognition has broad development prospects and potential in the future. The Yolo model is an open-source model, so with the efforts of the entire community, the model version is updated frequently. With the continuous advancement of deep learning technology and the continuous optimization of Yolo models. Yolo models will be able to more accurately identify ship targets, including small targets and targets in complex scenes. By introducing more training data and improving the network structure, the performance of the model will continue to improve. Future research will address end-to-end ship target recognition solutions, including the integration of data collection, model training, inference, and application deployment, in order to further improve the overall performance and efficiency of the system. Ultimately, deep learning based ship target recognition will gradually introduce techniques such as adaptive learning and reinforcement learning, enabling models to achieve autonomous learning and optimization in constantly changing environments, thereby improving their adaptability and robustness in complex scenarios. In summary, the ship target detection method using the Yolo model has great potential for development, and future research will continue to promote technological progress and application innovation in this field, providing more reliable and efficient solutions for marine safety, cargo transportation, and marine resources.

EXPERIMENTS AND RESULTS ANALYSIS

Dataset Preparation

Dataset selection

With the explosive progress of technology, large-scale transportation of ships, and the complex international situation, higher requirements have been put forward for the accuracy and timeliness of ship inspection. The resolution of SAR images is no longer sufficient to provide higher accuracy ship detection and recognition results. Therefore, a high-resolution SAR image dataset IIRSID (High Resolution SAR Image Dataset) can further improves the recognition accuracy of deep learning based ship recognition in complex backgrounds or small target ships. In high-resolution SAR images, ships are no longer just a highlight as before, but have detailed and accurate ship features. Compared with low resolution SAR images, instance segmentation in high-resolution SAR images can realistically and effectively depict the shape of ships pixel by pixel. Therefore, high-resolution SAR images arc beneficial for precise tasks such as maritime transportation safety and fisheries law enforcement.

The dataset includes 136 panoramic high-resolution SAR images with a resolution range of In 1 to 5m, cropped into 5604 SAR images with a resolution of 8(H)* 800 pixels. These SAR images have various polarization, imaging modes, imaging conditions, and other features, and there are a total of 16951 ships in the entire HRSID. In order to reduce errors and omissions in ship labelling [6]. The optical remote sensing images similar to SAR images were selected to eliminate potential interference environments from ships. The model trained through IIRSID can effectively detect ships in large-sized SAR images.

When constructing HRSID. efforts were made to avoid the deficiencies in the existing SAR ship dataset. However. as the background becomes more complex, the difficulty of detection also increases accordingly. Therefore, in order to train a more powerful ship recognition model, the proportion of offshore scenes in this dataset is 18.4%. While the proportion of offshore scenes is 81.6%. HRSID also supports instance segmentation technology, which is beneficial for accurate ship detection and recognition Example: In a SAR image, if only target detection is performed, only the approximate position and bounding box of each ship can be obtained, and it is impossible to distinguish different ships in the image. However, instance segmentation techniques can separate each vessel from its surrounding background and assign a unique identifier to each vessel. This means that each vessel will be individually identified and will not be confused with other vessels or backgrounds. Therefore, segmentation technology provides more detailed and accurate ship identification, enabling researchers to better understand each ship in the image, thereby improving the accuracy and performance of ship detection and recognition.

Dataset processing

After selecting the initial dataset HRSID. browse all images, perform data cleaning, confining the integrity, correlation, and consistency of all SAR images, and classify the dataset into three categories: complex terrain background map of ports, ocean background map. and large vessel map.

By writing the COCO2VOC. py script as shown in Figure 4. this script can be implemented to

modify the Microsoft Common Objects in Context (MS COCO) format of the original dataset to VOC (Visual Object Classes) format and store it as an*ML annotation file.

By writing the copy file. py script as shown in Figure 5. this script can be implemented to automatically extract the corresponding*ML annotation files from the*ML folder corresponding to the original image and save them to a new folder after selecting the original image for data augmentation. This improves the efficiency of the dataset augmentation process and reduces the time for operators to search for the*ML files corresponding to the photos that need to be enhanced.

By writing the advanced image 2. py script as shown in Figure 6. this script can be implemented to: select a small target in a complex image, determine its specific coordinates (XML annotation file), and then fill in the image name and path in the script, as well as the coordinates of the small target that need to be replicated; Next, enter the name and path of the dataset where this target needs to be added; After running the program, the selected small targets will undergo random rotation angles, random movement positions, and other operations, and will be added to each image in the corresponding dataset. The coordinates of the small targets will also be automatically added to the*ML annotation file corresponding to the image. After performing the above operations on all the small targets that need to be enhanced, an HRS1D dataset enhanced for the small target data is generated.

```
createChildNode(doc, 'xmin', str(int(attrs[0])),
                bndbox_node)
createChildNode(doc, 'ymin', str(int(attrs[1])),
                bndbox_node)
createChildNode(doc, 'xmax', str(int(attrs[2]) + int(attrs[0])),
                bndbox_node)
createChildNode(doc, 'ymax', str(int(attrs[3]) + int(attrs[1])),
                bndbox_node)
object_node.appendChild(bndbox_node)

return object_node
```

Figure 4. Key code of COCO2VOC. py script

```
for image_file in image_files:
    image_name = os.path.splitext(image_file)[0]
    source_annotation_file = os.path.join(source_annotation_dir, image_name + ".xml")
    if os.path.exists(source_annotation_file):
        target_annotation_file = os.path.join(target_annotation_dir, image_name + ".xml")
        shutil.copyfile(source_annotation_file, target_annotation_file)
```

Figure 5. Key code of copy_file.py

```
x_offset = random.randint(0, target_image.shape[1] - target_object.shape[1])
y_offset = random.randint(0, target_image.shape[0] - target_object.shape[0])

rotation_angle = random.uniform(0, 360)

rotation_matrix = cv2.getRotationMatrix2D((target_object.shape[1] / 2, target_object.shape[0] / 2),
                                          rotation_angle, 1)

rotated_target_object = cv2.warpAffine(target_object, rotation_matrix,
                                       (target_object.shape[1], target_object.shape[0]))

target_image_with_object = target_image.copy()
target_image_with_object[y_offset:y_offset + rotated_target_object.shape[0],
x_offset:x_offset + rotated_target_object.shape[1]] = rotated_target_object

cv2.imwrite(target_image_path, target_image_with_object)

update_voc_xml(target_image_name, x_offset, y_offset, rotated_target_object.shape[1],
               rotated_target_object.shape[0])
```

Figure 6. Key code of advanced imagc2. py

By writing the VOC annotation.py script as shown in Figure 7. this script can randomly partition the entire dataset: The ratio of training set » validation set to test set is 9:1; The ratio of training set Io validation set in (training sct+validation set) is 9: 1.

```
trainval_percent = 0.9
train_percent = 0.9
```

Figure 7. voc_annotation. py

Dataset enhancement

In this chapter, both the 11RSID dataset and the IIRSID dataset enhanced for small targets and complex backgrounds were used.

Prcproccss all images and use a Python script (COCO2VOC. py) to modify the original COCO (Microsoft Common Objects in Context) format of the IIRSID dataset to the VOC (Visual Object Classes) formal consistent with this YOLOv7 model, and store it as an*ML annotation file.

Randomly partition the entire dataset using a Python script (voc annotation, py):

The ratio of (training sct+validation set) to the test set, (training set^validation set): Test sct=9: l

The ratio of training set to validation set in (training sct+validation set) is 9: 1

Dataset 1: Initial HRS1D dataset, consisting of 5604 SAR images with a resolution of 800* 8(M) pixels and a total of 16951 ships

Dataset 2: Enhanced dataset for small targets. After selecting 18 small targets that are difficult to identify near the shore and port, their random angles and positions are added to the initial dataset for targeted reinforcement of the entire dataset. Among them, there are 5604 SAR images with a resolution of 800* 800 pixels and a total of 117823 ships. Add small targets to an original SAR image as shown in Figures 8. 9 and 10, and modify the corresponding*ML annotation file of the image as shown in Figure 9.

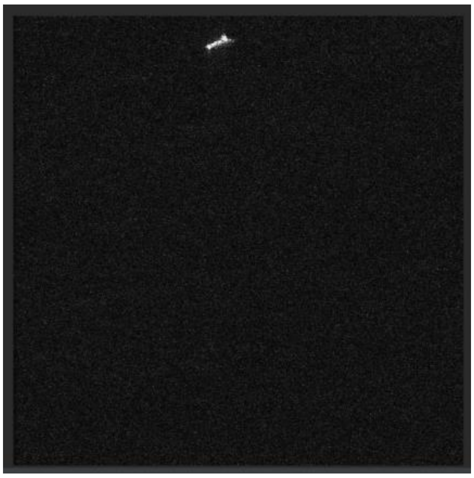


Figure 8. Photo I of the dataset before small target enhancement.

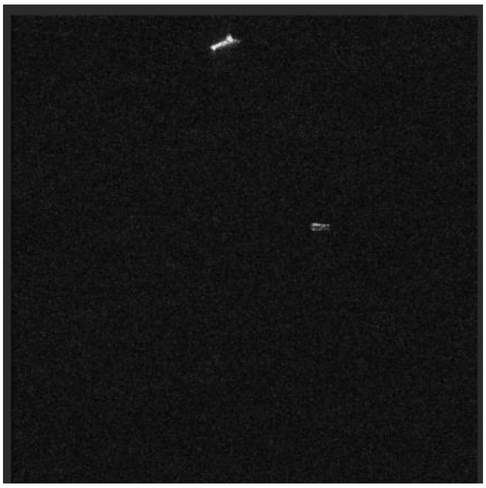


Figure 9. Photo 2 of the dataset after small target enhancement

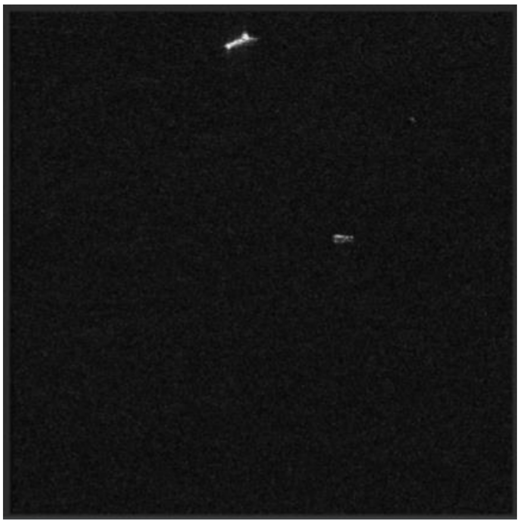


Figure 10. Photo 3 of the enhanced dataset for small targets

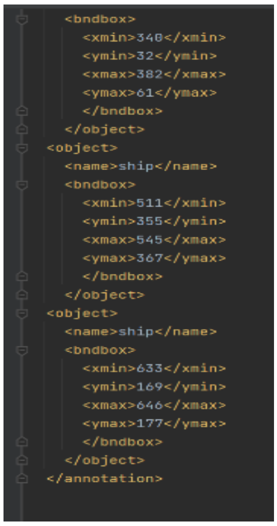


Figure 11. Enhanced*ML file 3

Experimental Preparation

Experimental hardware and software environment

In terms of experimental hardware, two NVIDIA Gcforcc GTX 1080 graphics cards arc used;

The CPU used is Intel Core i9 WOOK. In terms of experimental software, the operating system used for training is Ubuntu 22.04: The operating system used during testing is Windows 1 1; Using Python version 3.11.7; Important libraries and versions added to Pycharm arc matplotlib==3.1.2. opencv_python==4.1.2.30. tqdm 4.60.0. Pillow==8.2.0. h5py=2.1O.O.

Hypcrparamctcr settings: confidcncc=0.45. non maximum suppression overlap threshold (nms iou1=0.3. input size (input sthap) = [640. 640]. anchor box= [12. 16. 19. 36.40. 28. 36. 75. 76.55

72. 146. 142. 110. 192. 243. 459. 401],

Select three SAR images with distinct features that are difficult to recognize: one containing a large number of small targets and a high density of ships, as shown in Figure 12. and two SAR images with interference such as ports and docks in the background, as shown in Figures 13 and 14, as test samples for the experiment. The correct answers in the examples are illustrated in Figures 15, 16, and 17.

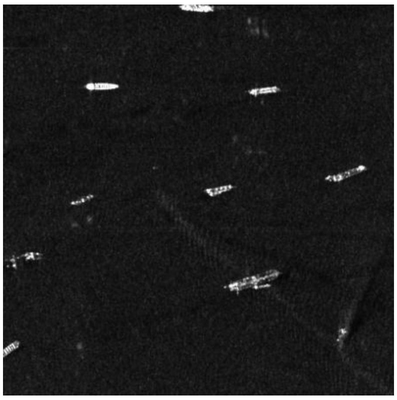


Figure 12. Test Figure 1

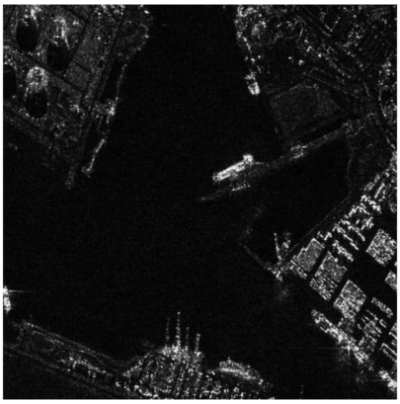
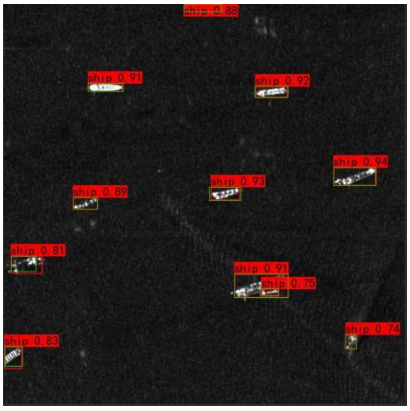


Figure 13. Test Figure 2

Figure 14. Test Figure 3

The green marked box indicates the correct ship target in these three test images



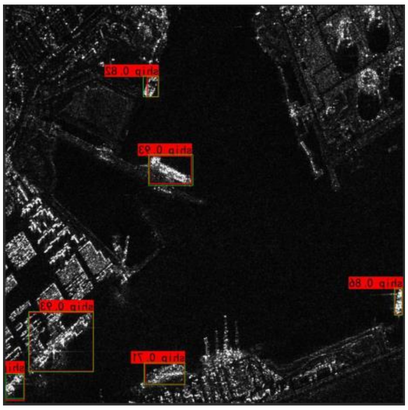


Figure 15. Test Figure 1: A total of 1 1 ships

Figure 16. Test Figure 2: A total of f> ships

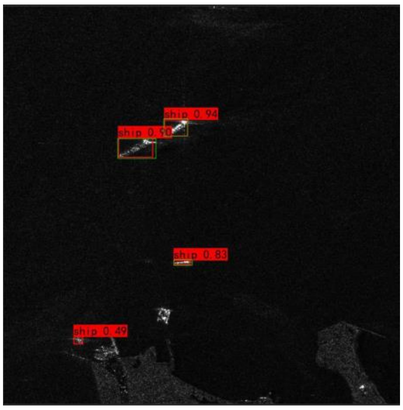


Figure 17. Test Figure 3: A total of 3 ships

Evaluation indicators

Evaluation indicators refer to quantitative indicators used to measure the performance of a model. The values of evaluation indicators can help experimenters understand the performance of the model in a certain experimental environment. The most commonly used evaluation indicator is Map (Mean Average Precision). Map is a commonly used comprehensive evaluation index in object detection tasks, which integrates the accuracy and recall of all categories. For each experiment under different confidence thresholds, calculate the Precision Recall curve and calculate the area under the curve (integral under the precision recall curve). Finally, take the average of the areas under all conditions to obtain the value of the Map. The use of Map0.5 in this article refers to the Map value when the loll threshold is 0.5. and the expected value of Map0.5 in the model is 90%.

If Map0.5 is greater than W, it is considered that the model recognition is successful and meets the basic recognition requirements.

Design experiment

In order to further improve the recognition accuracy of the model for small targets and ship targets under complex background conditions, experiments will be conducted in two main directions: replacing the model subject and enhancing the dataset with specificity. To test the impact of different model subjects on detection accuracy, it is necessary to design comparative experiments using YOLOv 5. YOLOv7. and YOLOv7-X as model subjects while keeping the dataset unchanged.

On test the impact of different datasets on detection accuracy, it is necessary to design a comparison experiment using the original HRSID dataset and the targeted enhanced HRSID dataset while keeping the model subject unchanged. To increase the usability of ship target detection, experiments need to be designed to recognize SAR images in videos containing SAR images and real-time camera footage using different model subjects. Therefore, based on the above design ideas, the following experiments are designed for testing:

Experiment 1: Train for 200 rounds using YOLOv5 and the original HRSID dataset.

Experiment 2: Train for 150 rounds using YOLOv7 and the original HRSID dataset.

Experiment 3: Use YOLOv7 and the original HRSID dataset to continue training for 50 rounds on the basis of Experiment 2.

Experiment 4: Train 200 rounds using YOLOv 7 and HRSID dataset reinforced for small target data.

Experiment 5: Train 1 50 rounds using YOLOv7-X and the original HRSID dataset.

Experiment 6: Train 200 rounds using YOLOv 7-X and HRSID dataset reinforced for small target data.

Experiment 7: Use YOLOv7 to detect SAR videos containing ship targets.

Experiment*: Use YOLOv7-X to detect SAR videos containing ship targets.

Experiment 9: Use YOLOv7 to detect ship targets in SAR images through a camera.

Experiment 10: Use YOLOv7-X to detect ship targets in SAR images through a camera.

Comparative Experiment

Using YOLO series models for image detection

According to the experimental protocol in design experiment, different YOLO senes models were used to train on different datasets. In Figure 18. the numerical variation curves of Map0.5 and Loss in epochs were recorded. The performance results of testing the same image under multiple experimental conditions are shown in Figure 19. and the final values of Map0.5 and Loss under various experimental conditions are summarized in Table 2.

Table 2. Final experimental results under multiple experimental conditions

Experiment Number	Model	Dataset	Training Epochs	Map 0.5	Loss
1	Yolov5	HRSID	200	0.9090	0.0248
2	Yolov7	HRSID	150	0.9039	0.0210
3	Yolov7	HRSID	200	0.9060	0.0172
4	Yolov7	Enhanced HRSID	200	0.9080	0.0248
5	Yolov7-X	HRSID	150	0.9083	0.0204
6	Yolov7-X	Enhanced HRSID	200	0.9253	0.0225

Tables 3 and 4 respectively show the number of missed and false detections of test.

Images 1. 2. and 3 under multiple experimental conditions, and provide comparisons.

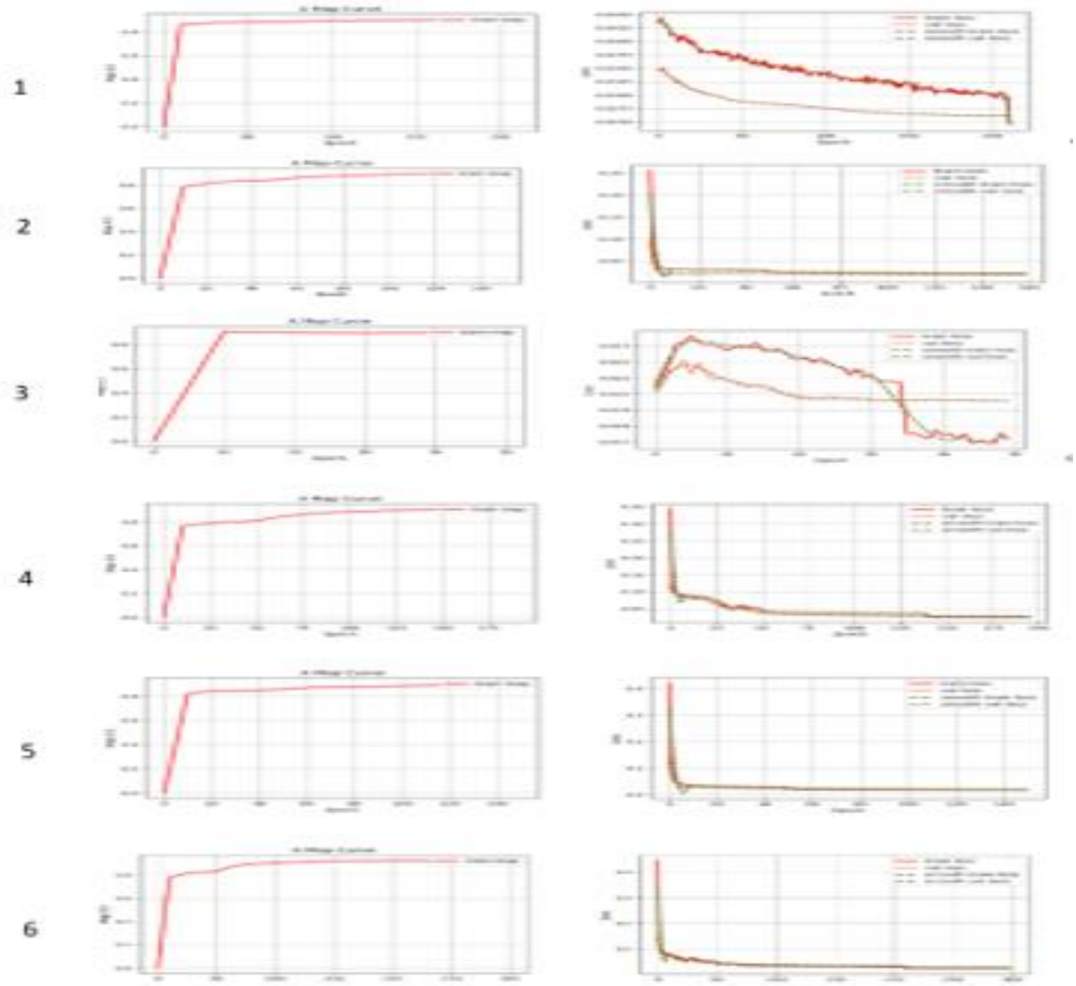


Figure 18. Numerical curves of Map0.5 and loss for each epoch under multiple experimental conditions

Table 3. Summan of identification of test figure 1 under multiple experimental conditions

Experiment Number	Missed Detection in Test Image 1	False Detection in Test Image 1
1	1	0
2	0	0
3	0	0
4	1	0
5	0	0
6	0	0

After analyzing three tables (Table 3,4,5), it was found that Experiment 4 using YOLOv7 and the enhanced HRSID dataset only missed one ship in Test Figure 1. while Experiment 5 using YOLOv7-X as the model subject only identified one ship incorrectly in Test Figure 3. Experiment 6 using YOLOv7-X and the enhanced HRSID dataset successfully identified all ship targets without omission or recognition errors. This indicates that using more suitable algorithms and enhancing the dataset with specificity can help improve the accuracy of ship target recognition.

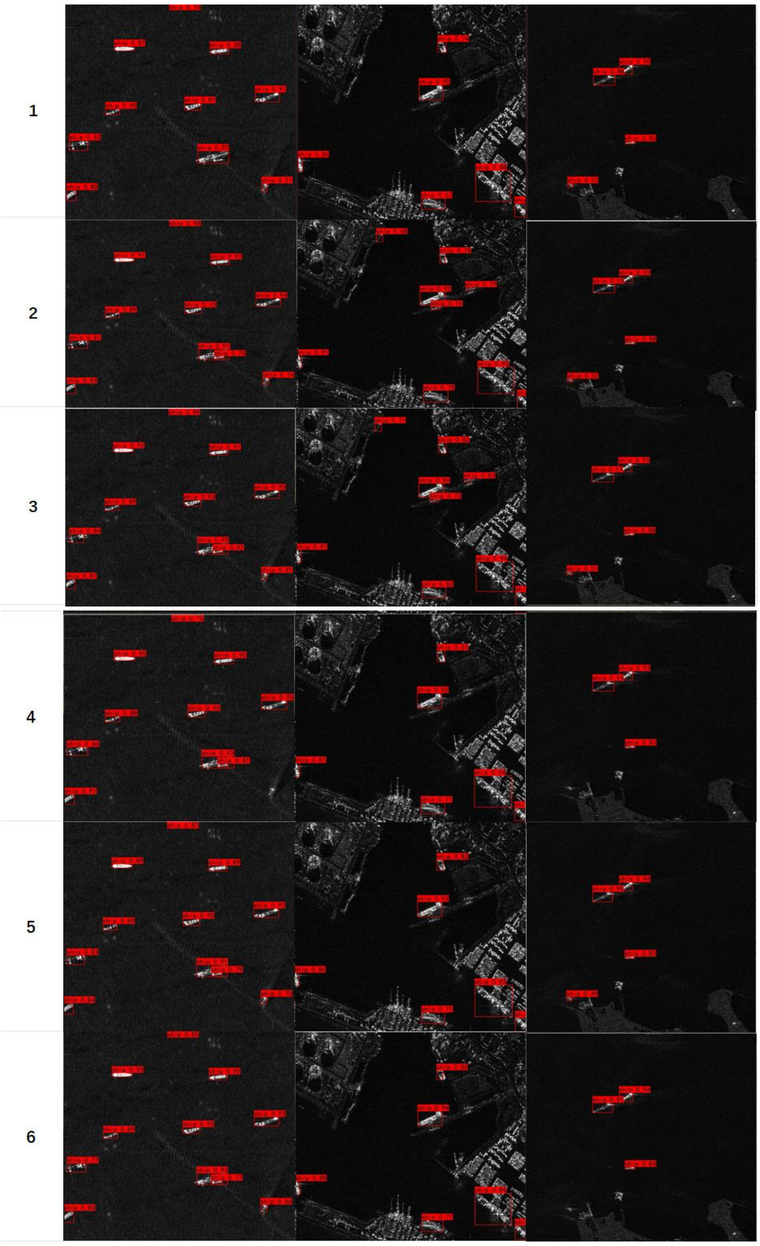


Figure 19. Comparison of test results for the same image under multiple experimental conditions

Table 4. Summan of identification of test figure 2 under multiple experimental conditions

Experiment Number	Missed Detection in Test Image 2	False Detection in Test Image 2
1	0	0
2	0	3
3	0	3
4	0	0
5	0	0
6	0	0

Table 5. Summary of identification of test figure 3 under multiple expenmental conditions

Experiment Number	Missed Detection in Test Image 3	False Detection in Test Image 3
1	0	1
2	0	1
3	0	1
4	0	0
5	0	1
6	0	0

Detection of VIdeos containing SAR ship images

As shown in Figure 20, both Experiment 7 (using YOLOv7 to detect SAR videos containing ship targets) and Experiment 8 (using YOLOv7-X to detect SAR videos containing ship targets) can recognize ship targets

7

8

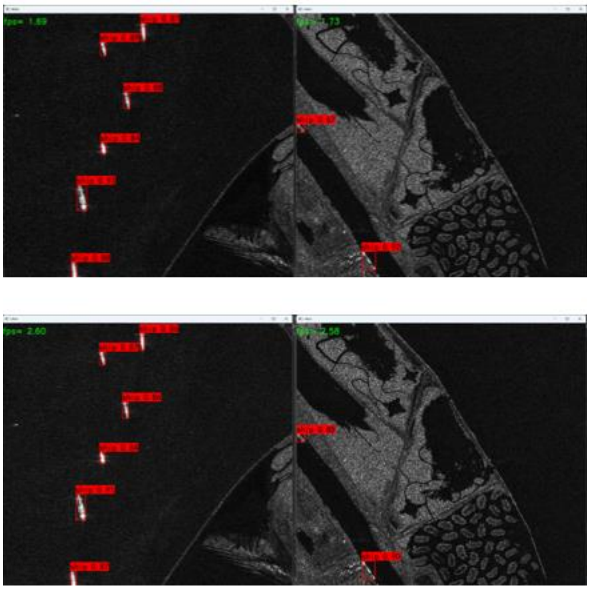


Figure 20. Detection results of videos containing SAR ship images using YOLOv7 and YOLOv7-X

Detection of SAR ship images through cameras

As shown in Figure 21. both Experiment 9 (using YOLOv7 to detect ship targets in SAR images through a camera) and Experiment 10 (using YOLOv7-X to detect ship targets in SAR images through a camera) can recognize ship targets.

Result Analysis

As shown in Figure 22, the Map0.5 values of the first six experiments were all greater than 90%, meeting the requirements of 23 and the last four experiments successfully achieved detection of ship targets. Based on further analysis of the above data, it can be concluded that in terms of ship target detection, as shown in the comparison between Experiment 2 and Experiment 3, increasing the number of training rounds after a certain number of rounds has less help in improving recognition accuracy; As shown in the comparison between Experiment 2 and Experiment 5.

9

10

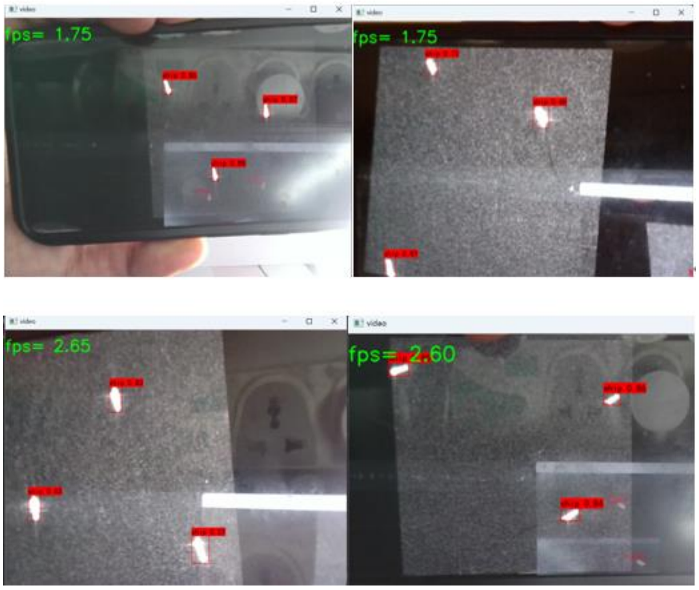


Figure 21. Detection results using YOLOv7 and YOLOv7-X via camera

electing a model with a larger number of parameters will consume longer training time, but it can help improve recognition accuracy; As shown in the comparison between Experiment 3 and Experiment 4. as well as Experiments 5 and 6. training with datasets reinforced for small targets and complex backgrounds also helps to improve recognition accuracy. Therefore, in this article.

YOLOv7-X was ultimately used to achieve the best recognition performance with HRSID enhanced for small targets and complex backgrounds, accurately identifying all ships in the three test images

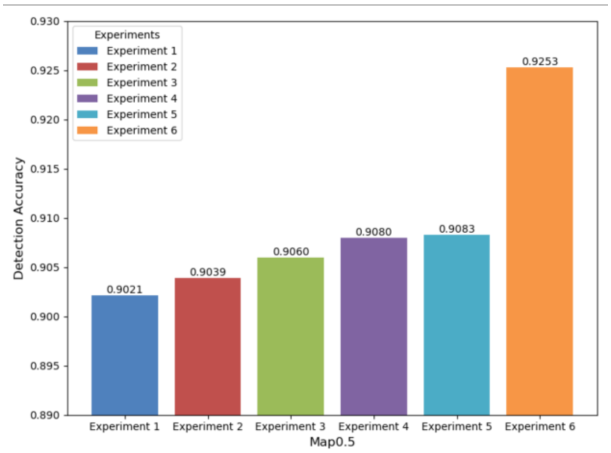


Figure 22. Comparison of Recognition Accuracy among Six Experiments

VISUAL INTERFACE

QT Designer was used in conjunction with PySidc6 to visualize the entire ship recognition model. QT Designer is a graphical user interface design tool that is part of the QT toolkit and can be used to edit or create Qt interfaces. PySideb is a module in Python that supports the development and creation of graphical user interfaces (GUI) based on the Qt framework.

Login Interface

In order to increase the security and practicality of the entire model, a login interface was designed. As shown in Figures 23 and 24 after entering the correct account name and password, the ship detection main page can be successfully opened. When entering the wrong account name and password, the terminal window will prompt "username or password incorrect" to remind the user in a timely manner.

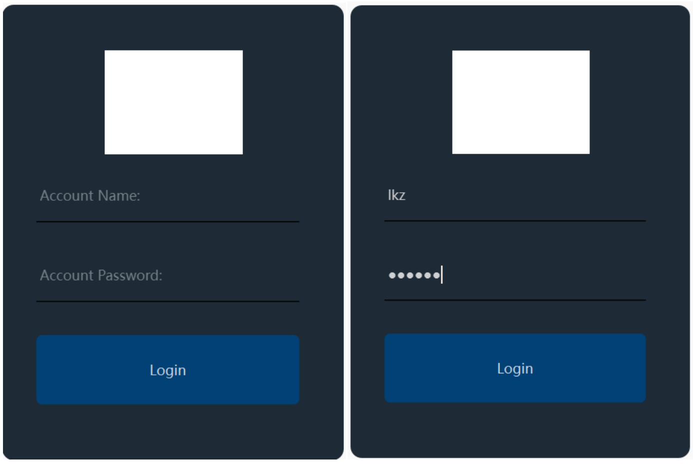


Figure 23. Login interface homepage Figure 5.2 Enter name and password

Main Interface

After successful login, you will enter the main page of ship target detection as shown in Figure 24 On this page, you can choose to detect ship targets through images, videos, or using cameras

according to your needs.

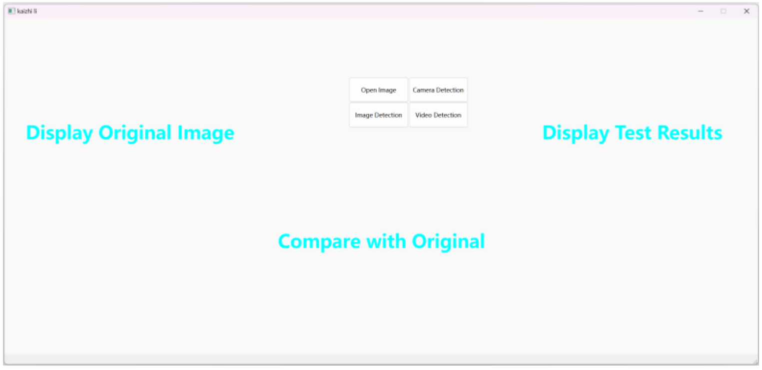


Figure 24. Main page of ship target detection

After clicking the 'Open Image' button, you can select the image you want to detect, as shown in Figure 25. After successful selection, the original image file path will be displayed in the terminal window, and the initial graphics will be shown in the "Display Original linage" section as shown in Figure 26.

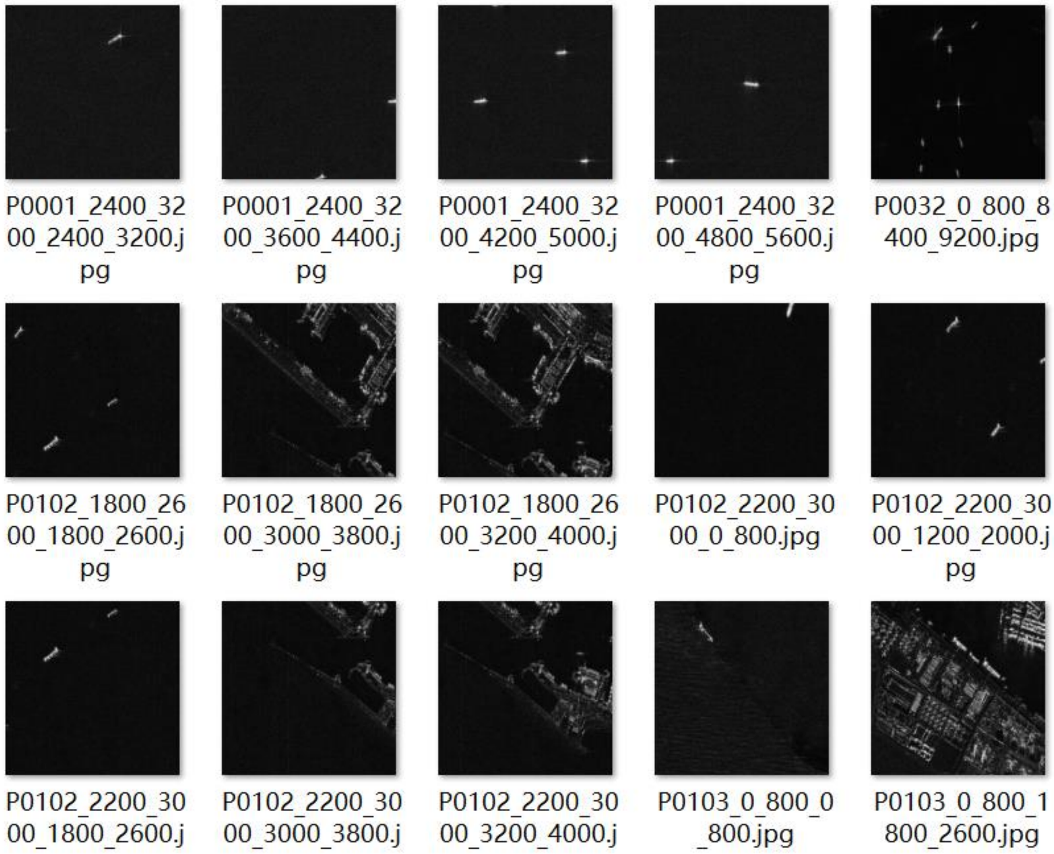


Figure 25. Select the image to be detected

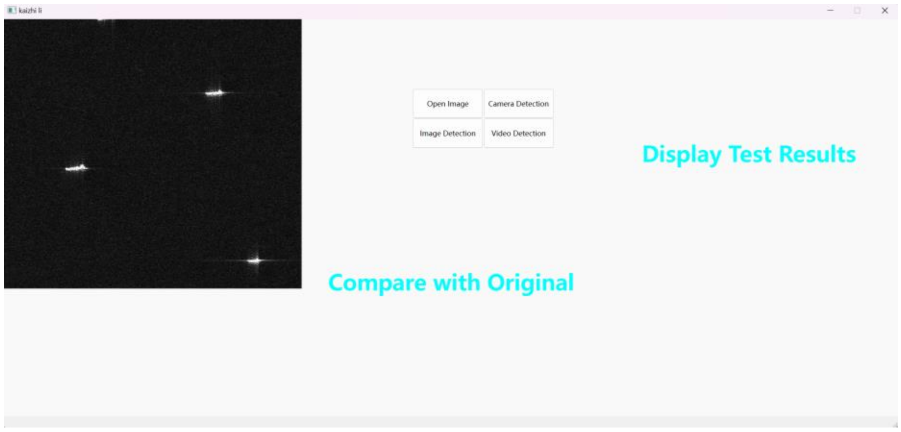


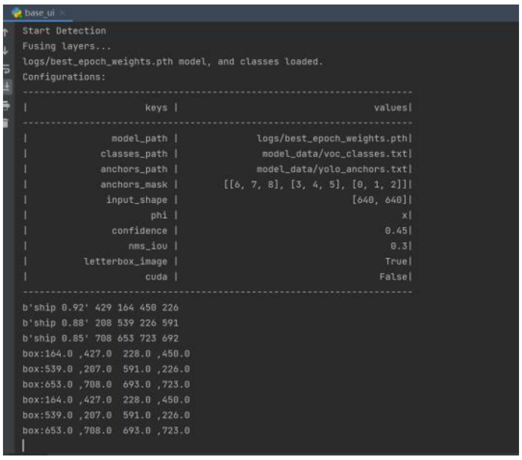
Figure 26. shows the original image

After clicking the image detection button, the program will automatically start detecting the ship targets in the selected image. After prompting "start detection" in the terminal window, the key parameter information used by the model in this test will be displayed as shown in Figure 27.

Providing the confidence and coordinate information of each ship in the detected image. The detection results and comparison w ith the original image w ill be displayed on the visualization page as shown in Figure 28. Click the open image button again to select the image and repeat the above steps to perform continuous detection on the image.

As shown in Figure 28 by writing the dataloader.py script, the method of converting real.

values into real boxes (green boxes) accompanying with the detection boxes (red boxes) in the same image is achieved. As shown in Figure 28. the detection rate is presented more clearly and intuitively, which improves the efficiency of judging the accuracy of model prediction.



```
base_ui
Start Detection
Fusing layers...
logs/best_epoch_weights.pth model, and classes loaded.
Configurations:
----------------------------------------------------------------------
|                  keys |                                       values|
----------------------------------------------------------------------
|            model_path |                  logs/best_epoch_weights.pth|
|          classes_path |                  model_data/voc_classes.txt|
|          anchors_path |                  model_data/yolo_anchors.txt|
|          anchors_mask |           [[6, 7, 8], [3, 4, 5], [0, 1, 2]]|
|           input_shape |                                   [640, 640]|
|                   phi |                                            x|
|            confidence |                                         0.45|
|               nms_iou |                                          0.3|
|       letterbox_image |                                         True|
|                  cuda |                                        False|
----------------------------------------------------------------------
b'ship 0.92' 429 164 450 226
b'ship 0.88' 208 539 226 591
b'ship 0.85' 708 653 723 692
box:164.0 ,427.0  228.0 ,450.0
box:539.0 ,207.0  591.0 ,226.0
box:653.0 ,708.0  693.0 ,723.0
box:164.0 ,427.0  228.0 ,450.0
box:539.0 ,207.0  591.0 ,226.0
box:653.0 ,708.0  693.0 ,723.0
```

Figure 27. Display of Image Deicclion Parameters

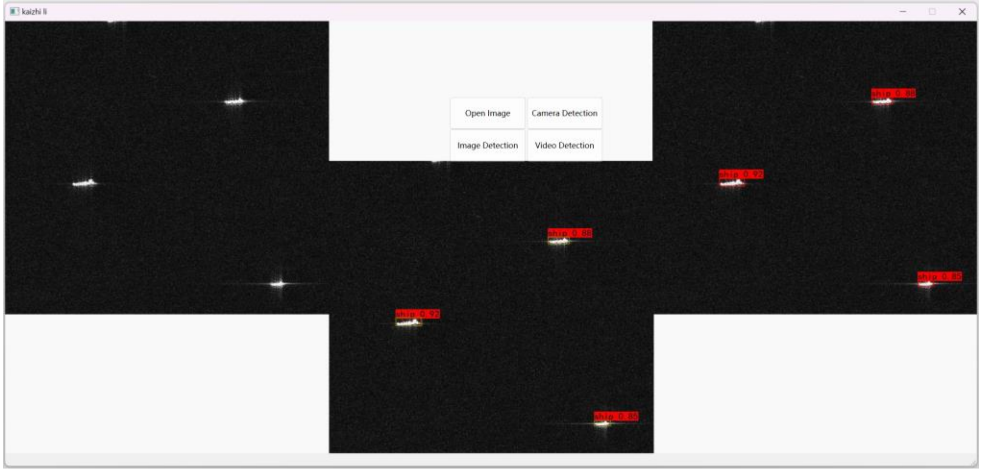


Figure 28. Visualization effect after detection completion

```
box[:, [0, 2]] = box[:, [0, 2]] / self.input_shape[1]
box[:, [1, 3]] = box[:, [1, 3]] / self.input_shape[0]

box[:, 2:4] = box[:, 2:4] - box[:, 0:2]
box[:, 0:2] = box[:, 0:2] + box[:, 2:4] / 2
```

Figure 29. dataloader.py The key code of the script

Video Detection Page

After clicking the video detection button, the page will jump to the video detection page as shown in Figure 31. After clicking the "Select Video" button, you can choose the desired video to be detected. Once the selection is successful, the terminal window will display the original video file path.

After clicking the video detection button, the program will automatically start detecting the ship targets in the selected video. After the prompt "Start detecting video" appears in the terminal window, the key parameter information used by the model in this test will be displayed, and the folder containing the output video will be searched to delete the video with the same name; Then provide the confidence and coordinate information of each vessel in the detected video: After the entire video detection is completed, as shown in Figure 33. the original video will be automatically synchronized with the detected video on the visualization page for playback; After playing, click the Select Video button again, select the video and repeat the above steps to perform continuous detection on the video.

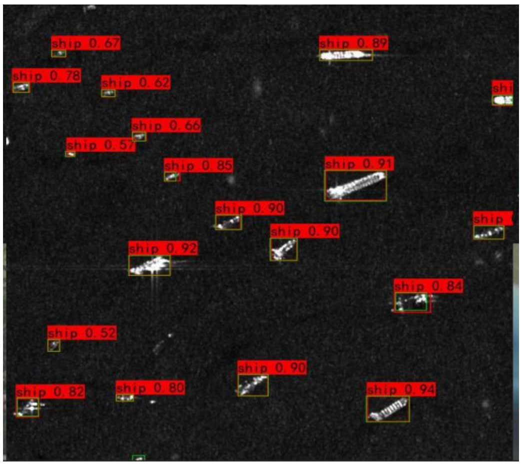


Figure 30. Overlapping the detection box (red) with the real box (green)

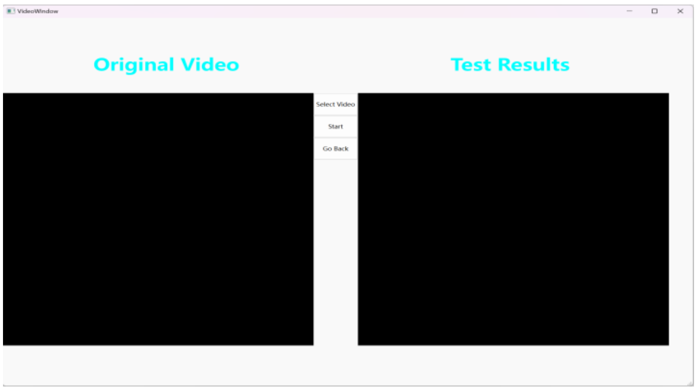


Figure 31 Video Detection Visualization Page

Click the 'Return to Previous' button to return to the main page of ship target detection.

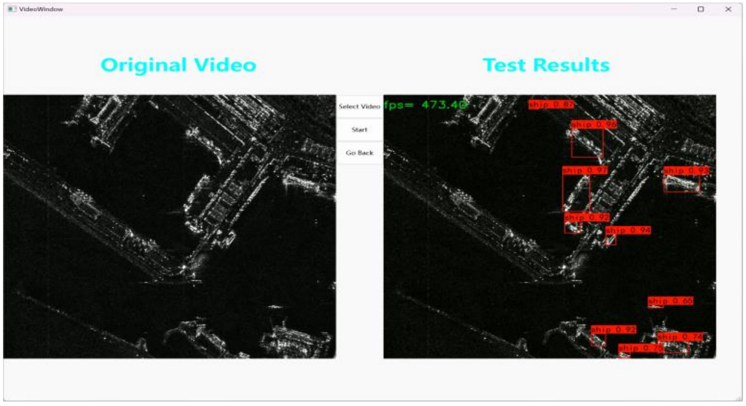


Figure 32. Shows the detection results and plays them synchronously with the onginal video

As shown in Figure 33 by writing the video maker. py script, this script can be implemented to create corresponding videos for all photos in any folder, and play each photo for 1 second at 30 frames per second.

```
images = [img for img in os.listdir(image_folder) if img.endswith(".jpg")]

frame = cv2.imread(os.path.join(image_folder, images[0]))
height, width, layers = frame.shape

video = cv2.VideoWriter(video_name, cv2.VideoWriter_fourcc(*'mp4v'), 30, (width, height))

for image in images:
    for _ in range(30):
        video.write(cv2.imread(os.path.join(image_folder, image)))
```

Figure 33. Key code of video_maker.exe script

Camera Detection Page

After clicking the camera detection button, the page will jump to the camera detection page as the Figure 35. clicking on 'Start Detection' will automatically open the camera and detect the ship targets appearing in the camera. After the prompt "Start detecting camera" appears in the terminal window, the key parameter information used by the model in this test will he displayed, followed by the confidence and coordinate information of each detected vessel. At the same time, (the visual interface will detect and mark the ship targets appearing in the camera.

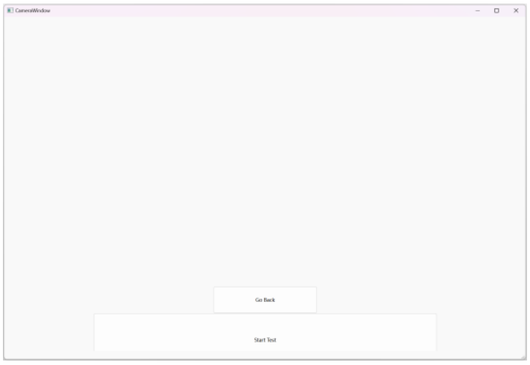


Figure 34. Camera fktection Visualization Page

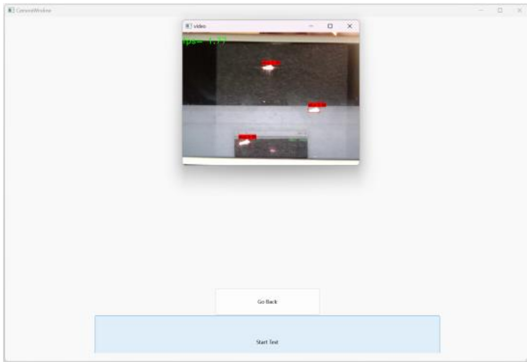


Figure 35. Visualization effect of camera detection

CONCLUSION

This article analyzes the current research status and trends of ship target recognition based on deep learning, starting from the practical needs of port management and marine safety. Early methods based on manually extracting the characteristics of ship targets for recognition required a significant amount of manpower and had low accuracy, and could not provide timely and effective detection results. At present, deep learning based object detection algorithms have effectively improved detection efficiency and accuracy, but they are still difficult to accurately recognize small targets and complex background ship targets, and there is no visual interface to intuitively display the recognition effect after recognition is completed. The research summary of this article regarding such issues is as follows:

(1) Under the condition of running for a certain number of rounds, it is difficult to further improve the recognition accuracy by increasing the number of training rounds. As the number of training rounds increases, the accuracy of identifying ship targets will gradually increase but not significantly, and it will take a longer time. Comparing with the YOLOv7 and IIRSID datasets, increasing the training epochs in an attempt to improve recognition accuracy resulted in a smaller improvement in accuracy.

(2) By changing the prediction model subject while ensuring that the dataset remains unchanged, the recognition accuracy of ship targets can be further improved, while also enhancing the recognition accuracy of small targets and complex background ship targets. In this article.

OLOv7-X is more suitable for ship target detection than YOLOv7 and YOLOv5, effectively improving recognition accuracy. Compared to the same HRSID dataset, replacing the YOLOv7-X algorithm with a larger number of parameters to attempt to improve recognition accuracy. After practical application, it was found that the model's recognition accuracy for normal targets, small targets, and ship targets with complex backgrounds was further improved after replacing the algorithm subject, which was the highest among the four experimental groups. However, due to the large number of parameters in YOLOv7-X, the training time per round is longer than in YOLOv5 and YOLOv7.

(3) By replacing the targeted enhanced dataset while ensuring that the main body of the prediction model remains unchanged, the recognition accuracy of small targets and complex background ship targets can be further improved. In this article, an IIRSID dataset was constructed for small targets and complex background ship targets, and relevant Python scripts were written to improve the efficiency of creating the dataset; Comparing the use of the enhanced HRSID dataset to improve recognition accuracy in both YOLOv7 and YOLOv7-X; The results showed that after dataset reinforcement, the model's recognition accuracy for small targets and ship targets with complex backgrounds was improved.

(4) In terms of related functions, YOLOv7 model and YOLOv7-X model were successfully used to recognize videos of ship targets containing SAR images. At the same time, the YOLOv7 model and YOLOv7-X model were successfully used to recognize ship targets in SAR images appearing in real-time cameras.

(5) In terms of project visualization, a visual display interface was designed to compare the real values with the detection boxes (red boxes) in the form of real boxes (green boxes), in order to more intuitively display whether the ship target recognition is correct or not, and to present the det effect more clearly and intuitively for subsequent statistics. At the same time, this article also designed a visual interface for detecting ship target videos and camera functions, which can synchronize the playback of the detected video and the detected video for easy viewing of the detection effect.

This article successfully achieved a recognition success rate of over 92% for ship targets in SAR images through a data-driven deep learning ship detection model, and presented it in a visual interface. However, further improvement is still needed in the subsequent content: after manually finding the smallest ship (4 pixels* 9 pixels) and the largest ship (133 pixels* 199 pixels) in the entire dataset, it was not successful to further enhance the accuracy of model detection by strengthening the fit between anchors and target sizes. This requires repeated attempts in the future and coordination w ith other parameters in order to summarize the regularity of better solutions.

REFRENCES

[1] Zhang Hanvcn National maritime security and its legal response under the unprecedented major changes in a century Theoretical Exploration. 2022 (01): 5-11.

[2] Wang Xuesong. Wang Jian. Wang Tao, et al. Instantaneous measurement method for polarization scattering matrix of radar targets Electronic Journal. 2006 (06): 1020-1025.

[3] Chen Z. Long C. Zhang L. et al. Can: A context aware network for shadow removal Proceedings of the IEEE/CVF International Conference on Computer Vision 2021 4743-4752.

[4] Grabner H. Nguyen T T. Gruber B. et al. On-line boosting-based car detection from aenal images. ISPRS Journal of Photogrammetry and Remote Sensing. 2008. 63(3): 382-396 [4].

[5] Hinton G E. Salakhutdinov R R Reducing the dimensionality of data with neural networks, science. 2006. 313(5786) 504-507.

[6] Wei. Shunjun et al. HRSID A High-Resolution SAR Images Dataset for Ship Detection and Instance Segmentation. IEEE Access 8 (2020): 120234-120254.