Journal of *Marine Science and Engineering*

MDPI

Article

Auxiliary Equipment Detection in Marine Engine Rooms Based on Deep Learning Model

Jiahao Qi, Jundong Zhang * and Qingyan Meng

College of Marine Engineering, Dalian Maritime University, Dalian 116026, China; halo754@dlmu.edu.cn (J.Q.); qingyanmeng@dlmu.edu.cn (Q.M.)
* Correspondence: zhjundong@dlmu.edu.cn

Abstract: In the intelligent perception of the marine engine room, visual identification of auxiliary equipment is the prerequisite for defect recognition and anomaly detection. To improve the detection accuracy, this study presents an auxiliary equipment detector in the cabin based on a deep learning model. Owing to the compact layout of pipeline networks and the large disparity in the equipment scales, we initially adopted RetinaNet as the basic framework, and introduced the single channel plain architecture RepVGG as the feature extraction network to simplify the complexity and improve realtime detection. Secondly, the Neighbor Erasing and Transferring Mechanism (NETM) was applied in the feature pyramid to deal with more complicated scale variations. Then, the complete IoU (CIoU) regression loss function was used instead of smooth L1, and the DIoU Soft-NMS mechanism was proposed to alleviate the misdetection in congested cabins. Further, comparison experiments and ablation experiments were performed on the auxiliary equipment in a marine engine room (AEMER) dataset to validate the efficacy of these strategies on the model performance boost. Specifically, our model can correctly detect 93.44% of coolers, 100.00% of diesel engines, 60.26% of meters, 95.30% of pumps, 55.01% of reservoirs, 97.68% of oil separators, and 74.37% of valves in a practical cabin.

Keywords: AEMER; CIoU loss; DIoU Soft-NMS; NETM; RepVGG; RetinaNet

Citation: Qi, J.; Zhang, J.; Meng, Q. Auxiliary Equipment Detection in Marine Engine Rooms Based on Deep Learning Model. *J. Mar. Sci. Eng.* **2021**, *9*, 1006. https://doi.org/10.3390/jmse9091006

Academic Editor: Marco Cococcioni

Received: 15 August 2021
Accepted: 12 September 2021
Published: 14 September 2021

1. Introduction

The intelligent monitoring and alarm system in a marine engine room can perform realtime monitoring of the various running statuses of the power system to guarantee the safety of the ship's operation. Whenever a failure happens, the system will send various signals such as sound-light to alarm and simultaneously backup the relevant data of the operating and system status, so that the engineers can find the cause of the failure and repair it promptly. Even if there is no one on duty, the engineer can respond to the signals by the extended alarm system when the intelligent monitoring in cabin transmits them to all corners of the ship. However, these alarm systems seem to lose focus on the defect recognition and anomaly detection of the appearance of the equipment because the sensors cannot obtain the information above. For example, some screws may be loose or the pipelines could leak when there is no engineer watching in the engine room, the consequences could be deadly.

If the liquid level of the bilge well is abnormally high, the bilge would overflow due to an unsupervised alarm; seawater pipelines could corrode and penetrate; the pipelines connected with submarine gates or other discharge overboard valves could break up; the bulkheads of a measuring tube could be missing. If any of the faults is not troubleshot immediately, it will cause electrical equipment to trip and will endanger the safety of the ship. If visual sensors are applied to identify the appearance information of the equipment in the engine room automatically, and the monitoring information is integrated into the centralized monitoring and alarm system, this can predict some faults or defects early to help engineers deal with the hidden dangers and minimize the odds of failure in advance.

However, the current visual perception technologies for ship's intelligent engine room are lesser-known. With the exploitation of computer vision, our main intention of this paper is to propose a detection model for auxiliary equipment in a marine engine room, which will replace the engineer's eyes and recognize the equipment autonomously when unattended. At the same time, it might provide a potential guide for subsequent visual inspection to find the appearance defects in the cabin equipment. Despite the emergence of convolutional neural networks that have greatly achieved considerable progress both in detection robustness and accuracy, tasks of auxiliary equipment detection in practical cabin still face difficulties and challenges for depth frameworks, which can be summarized as the following:

- There are unavailable datasets for the marine engine room. The number of valves and meters accounts for a large proportion of equipment in an engine room, while other equipment accounts for only a small proportion;
- The auxiliary equipment is multiscale in size ranging from tiny valves to giant diesel engines;
- The engine room is congested and formless, the pipelines with corresponding equipment are densely distributed in cabin, which means there is a large amount of occlusion or obscuring equipment.

Considering the challenges mentioned above, this study proposes a realtime detection model for auxiliary equipment in a marine engine room. To recapitulate briefly, the contributions of this paper can be shown below:

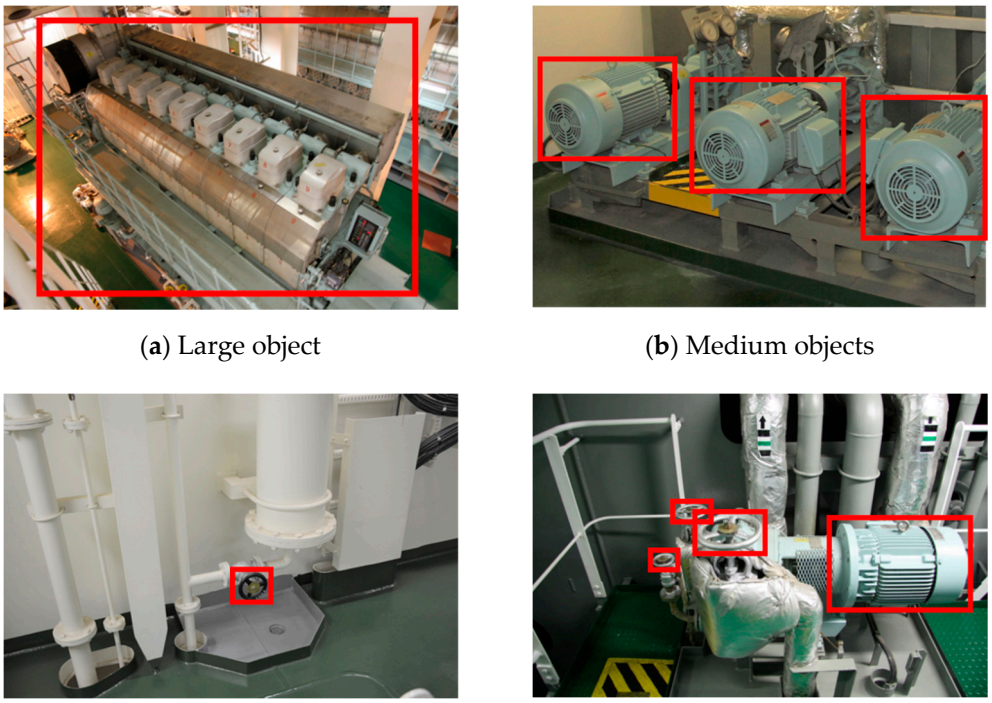
- In consideration of the currently unavailable public datasets, we filtered the original image resources and expanded the samples of the equipment in small proportion relying on our 3D virtual engine room team. Moreover, we built the auxiliary equipment in a marine engine room (AEMER) dataset, whose equipment classes included diesel engine, oil separator, cooler, reservoir, pump, valve, and meter;
- To facilitate the deployment of the detector in the cabin monitoring and alarm, we replaced the backbone in RetinaNet with RepVGG, which combined the plain architecture of VGG and the residual branch of ResNet. Furthermore, to ameliorate the situation of small-scale equipment misdetection in the cabin, we adopted the Neighbor Erasing and Transferring Mechanism (NETM) with FPN to filter out the redundant features of large-scale objects in the shallow feature pyramid layers and transfer them to the deeper layers;
- Because of the characteristics of the cabin layout, we applied the DIoU Soft-NMS to undermine the destructive impact on undetected errors, which can ensure the precision and recall in cabin. At the same time, we replaced the regression loss of smooth L1 with CIou loss, which not only ensures prediction boxes fit the targets better but also accelerates the speed of convergence and regression accuracy of training.

The remainder of our paper is organized as follows. Section 2 discusses the related works on computer vision. Section 3 introduces the proposed novel auxiliary equipment detection model in a marine engine room based on RetinaNet. Section 4 analyzes the ablation study and comparison experiments based on the AEMER dataset. Finally, we summarize the full text and identify the future work in Section 5.

2. Related Work

Visual identification is a prerequisite for the inspection task in a marine engine room. Whether the information can be detected comprehensively and accurately will greatly affect the reliability of subsequent equipment prediction and evaluation. With the development of deep learning, the convolutional neural network made major breakthroughs in accuracy and speed compared with traditional object detection methods [1–3]. In general, there are two main schools among the detection models: the two-stage algorithm represented by the R-CNN [4–6] and the one-stage algorithm represented by YOLO [7–9] and SSD [10]. Specifically, the two-stage algorithm firstly generates candidate regions on the image, then

classifies and regresses them individually. Conversely, the one-stage algorithm directly locates and classifies all targets on the entire image, which bypasses the step of generating candidate regions. Both the two-stage and one-stage have their own advantages, generally speaking, the former is more accurate and the latter is faster. For the current object detection task, no matter which genre of algorithm is adopted, one must face the challenge of multiscale, that is, the size of the target to be detected differs greatly from the proportion of the entire images and between different images, even within the same image. In Figure 1a–c, we can see the scale of the diesel engine, pump, and valve are from large to small while Figure 1d contains multiscale targets. The challenges caused by the scale variations severely limit the overall performance of the existing detectors. Therefore, how to better achieve multiscale object detection has always been a central issue in scholarship.



(**a**) Large object (**b**) Medium objects

(**c**) Small object (**d**) Multiscale objects

Figure 1. Examples of the multiscale objects in a marine engine room. In (**a**), the engine occupies almost the whole image. In (**b**), the pixels of every pump account for about 10% of the image. In (**c**), the pixels of the tiny valve are less than 1% of the whole image. Here, in (**d**), are some small valves and a middle size valve.

Object detection includes the two subtasks of regression and classification. The root of the scale problem is that, as the convolutional neural network deepens, its ability to express abstract features strengthens. However, shallow spatial and semantic information is gradually lost during downsampling, which results in the inability of deep features to provide fine-grained spatial information, so it cannot accurately locate the target. Therefore, a generic strategy to solve the scale variations is to construct multiscale feature expression. At present, the commonly used methods for constructing multiscale features include: (1) The use of the feature pyramid network (FPN) [11] to sequentially perform object detection on different resolutions [12,13]. (2) In the neural network, the connection of feature maps of different depths to reconstruct a feature pyramid for object detection [14,15]. (3) The design of parallel branches in the internal neural network to build a spatial pyramid for object detection [16,17]. In addition to constructing multiscale feature expressions, some scholars have studied strategies to reduce the accuracy gap of different scales from a more detailed level in the algorithm process, such as bounding box regression loss function [18–21] and anchor mechanism [22,23].

Among many strategies, the typical single-shot detector is SSD [10], which combines both the main point of YOLO [7] and the anchor mechanism of Faster R-CNN [6] to ensure that the feature maps of different receptive fields can adapt to different scale targets. However, the representative ability of shallow features is much weaker than the deep, which leads to poor performance in detecting small objects. DSSD [24] proposed a complex feature pyramid network on the basis of SSD, promoting feature fusion between different levels and achieving better accuracy at the price of calculating efficiency. FSSD [25] inserted a fusion mechanism into the original SSD, making full use of local detailed features and global semantic features. ASSD [26] added an attention module [27] to each feature layer and achieved the accuracy of RetinaNet [28]. Considering the single-shot detectors are prone to scale-confusion during feature fusion, Li et al. [29] proposed the Neighbor Erasing and Transferring Mechanism (NETM) that erases salient large-scale features in the shallow feature maps by the Neighbor Erasing Module (NEM), and transfers them to the deep by the Neighbor Transferring Module (NTM) to ensure that small-scale features can be better perceived by the network. Their experimental results were significantly better than other single shot detectors. To this end, the RepVGG-RetinaNet equipped with NETM was used to detect auxiliary equipment of various scales in marine engine room.

3. Methodologies

In this section, the overview of our proposed framework for auxiliary equipment detection in marine engine room is introduced firstly. Next, we present the simple but powerful plain architecture of the convolutional neural network RepVGG used for the feature extraction network in RetinaNet, and describe how to convert a trained block into the inference layer. Then, we discuss the Neighbor Erasing and Transferring Mechanism (NETM), which is adopted in the feature pyramid to deal with more complicated scale variations. Finally, the DIoU Soft-NMS postprocessing mechanism and CIou loss function are introduced in detail.

3.1. Overview of the Proposed RetinaNet

The basic framework in this paper is RetinaNet, which is mainly comprised of a backbone, feature pyramid network (FPN), and subnet. The backbone is designed to pick up low-level general features, such as shape and texture. FPN is a U-shaped network structure, the pyramid generated by feature fusion can effectively combine the semantic representations of different depths and dimensions, which can individually explore multiscale features in different layers. The subnet module includes classification and regression branches.

In the RetinaNet as shown in Figure 2, the low-level features are first extracted through the RepVGG [30], and feature maps of different resolutions are output through five stages, which are labeled as C1, C2, C3, C4, and C5 according to the output sequence. Then the maps are fused by the FPN, and generate the fused pyramid layers P3, P4, and P5 with the same resolution as C3, C4, and C5. At the same time, the pyramid layer P6 is obtained through 3×3 convolution with stride 2 on P5, and then the P7 is obtained through 3×3 convolution with stride 2 on P6. In FPN, pyramid features are fused in a top-down strategy, which may introduce large-scale object features to shallow maps and exert passive influence on detecting tiny objects. Therefore, the feature maps need to be aggregated by NETM [29] except for P7, and the salient large object features in P3 and P4 erased by NEM are transferred to P5 and P6 via NTM. Finally, the five feature maps are sent to the subnet to classify and regress. Both the classification and the regression subnet adopt the FCN [31] structure. After a series of straight convolution operations, the former can obtain the confidence score that each anchor contains the ground truth. Similarly, the latter can obtain a set of location offsets that each anchor regresses to the ground truth. Specifically, both the object class and precise coordinate position are obtained in the subnet.

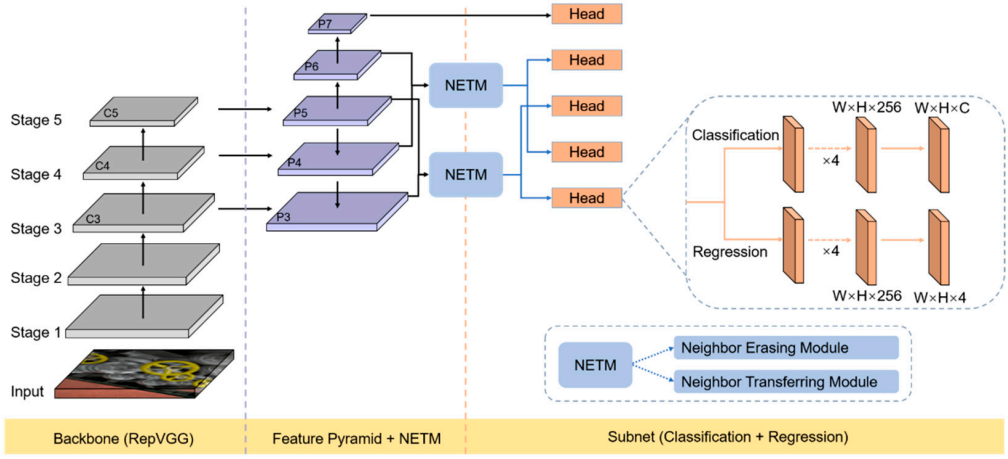


Figure 2. Illustration of the proposed framework.

3.2. Backbone Feature Extraction Network

For the current computer vision tasks, ResNet [32] and MobileNet [33–35] appear frequently. A large collection of experimental analyses argue that the ResNet can extract robust feature representations, and the customers can flexibly choose ResNet-50 or 101 according to their requirements; MobileNet is suitable for some embedded devices with low computing power, which can significantly balance the detection speed and accuracy. RepVGG is improved on the basis of classical VGG [36], whose main idea is to add the essence of ResNet to the VGG Block, namely, the identity branch and the residual branch. The sketch of the RepVGG architecture is shown in Figure 3a represents the classical ResNet that contains the residual structure of identity and 1×1 convolution, which commendably solve the vanishing gradient problem in the deep layers and make the model easier to converge Figure 3b represents the RepVGG training enlightened by ResNet but in a different way that the identity and 1×1 branches can be removed by structural reparameterization, which not only allows the deep network to obtain robust feature performance, but also solves the vanishing gradient problem quite nicely Figure 3c represents the RepVGG inference, we performed the transformation in identity and 1×1 branches to accelerate the network deployment, which can be converted into a stack of 3×3 convolutional structure with simple algebra.

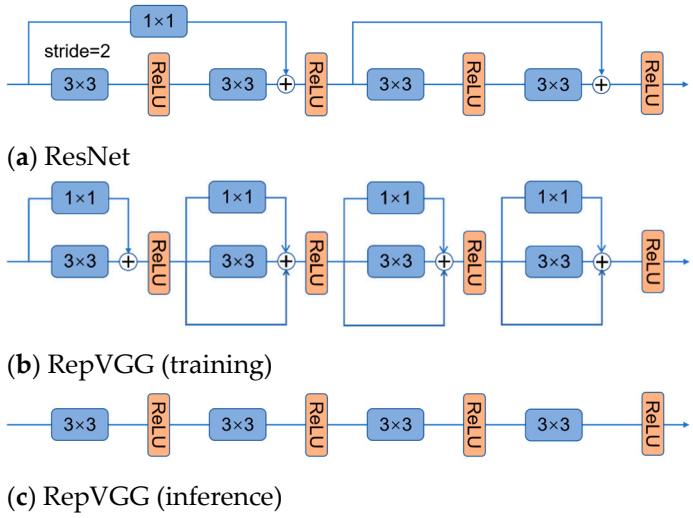
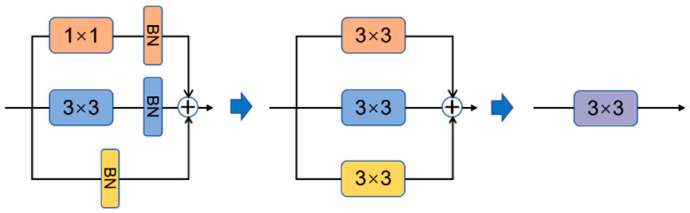
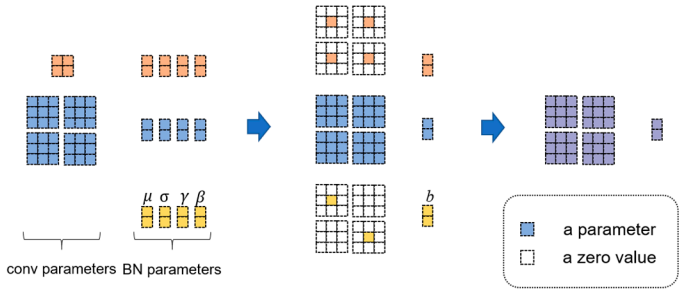


Figure 3. Sketch of RepVGG architecture. (**a**) The body of ResNet. (**b**) The body of the training time RepVGG. (**c**) The body of the inference time RepVGG. Only stage 2 of RepVGG-B1g4 is shown, which has 4 layers and conducts downsampling via stride-2 convolution at the beginning.

Figure 4 describes how to subtly convert a trained RepVGG block into a single plain 3×3 conv layer for RepVGG inference. Firstly, the convolutional layer in the residual block is fused, then the fused convolution layer is transformed into 3×3 convolution, and finally the 3×3 convolution in the residual branches is merged, that is, the weights and offsets of all branches are added together.



(**a**) Perspective of structure



(**b**) Perspective of parameter

Figure 4. Structural reparameterization of a RepVGG block. (**a**) Structural reparameterization of a RepVGG block from the perspective of structure. (**b**) Structural reparameterization from the perspective of parameter.

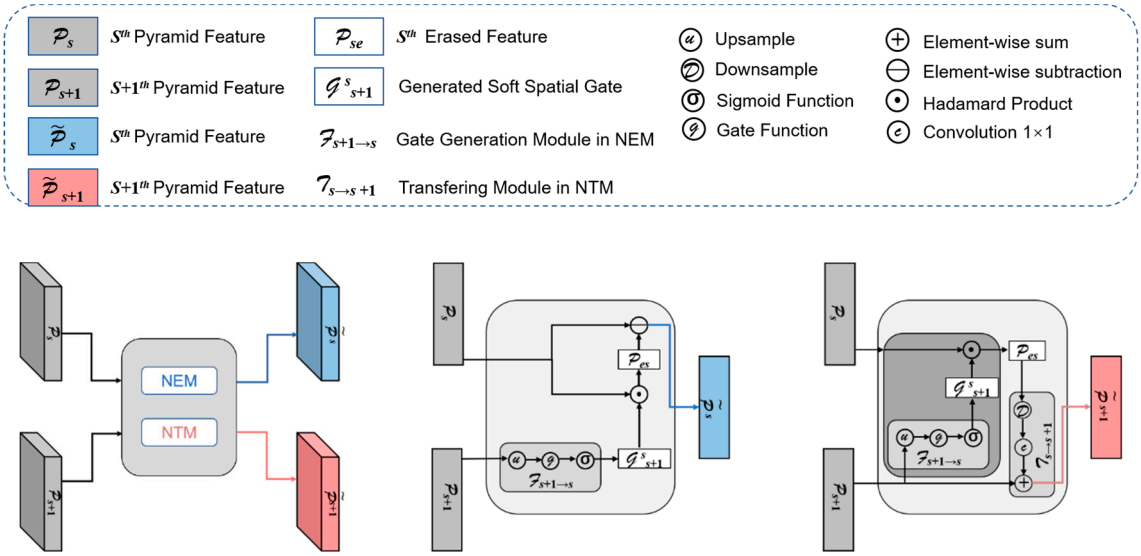
We used $W^3 \in \mathbb{R}^{C_2 \times C_1 \times 3 \times 3}$, $W^1 \in \mathbb{R}^{C_2 \times C_1}$ to denote the kernel of 3×3 and 1×1 convolutional layers respectively, and used μ, σ, γ, β as the accumulated mean, standard deviation, learned scaling factor, and bias of the identity branch or the BN layer, respectively. Let $M^1 \in \mathbb{R}^{N \times C_1 \times H_1 \times W_1}$, $M^2 \in \mathbb{R}^{N \times C_2 \times H_2 \times W_2}$ be the input and output respectively, and $*$ be the operator of convolution. If $C_1 = C_2$, $W_1 = W_2$, $H_1 = H_2$ in M^1, M^2, we have the Equation (1):

$$M^2 = \mathrm{BN}(M^1, \mu^0, \sigma^0, \gamma^0, \beta^0) + \mathrm{BN}(M^1 * W^1, \mu^1, \sigma^1, \gamma^1, \beta^1) + \mathrm{BN}(M^1 * W^3, \mu^3, \sigma^3, \gamma^3, \beta^3) \quad (1)$$

3.3. FPN with NETM

As shown in Figure 5a, the NETM contains the neighbor erasing module (NEM) and the neighbor transferring module (NTM). The former was proposed to erase the redundant salient features of large objects and highlight the small objects in shallow feature maps. The latter was designed to receive these erased features from NEM, and transfer them to enhance the deep features.

To ease the feature scale-confusion, the NEM was designed to take out the superfluous features. As shown in Figure 5b, s^{th} and $(s+1)^{th}$ are two adjacent pyramid layers, $p_s \in \mathbb{R}^{c_s \times h_s \times w_s}$ has more semantic information about object x_s than $p_{s+1} \in \mathbb{R}^{c_{s+1} \times h_{s+1} \times w_{s+1}}$. Based on the distribution of features, $\widetilde{p}_s$ for object s from the original p_s can be generated by the filtering features p_{es} of objects in $[s+1, S]$ as Equation (2), and the feature p_{es} from p_s is extracted by Equation (3):



(**a**) NETM (**b**) NEM (**c**) NTM

Figure 5. Sketch of the NETM. (**a**) Neighbor Erasing and Transferring Mechanism. (**b**) Neighbor Erasing Module. (**c**) Neighbor Erasing Module. The salient large object features erased by NEM are transferred to the deeper layer via NTM.

$$\widetilde{p}_s = f_s(x_s) = p_s \ominus p_{es} = f_s(x_s, x_{s+1}, \ldots, x_S) \ominus f_s(x_{s+1}, \ldots, x_S) \tag{2}$$

$$p_{es} = p_s \odot \mathrm{g}^s_{s+1} = p_s \odot \mathrm{F}_{s+1\to s}(p_{s+1}) = p_s \odot \frac{1}{1 + e^{-\mathrm{g}(\mathrm{u}(p_{s+1}); W^s_{s+1})}} \tag{3}$$

As formulated in Equation (2), p_{es} helps extract the refined information of large objects. In Figure 5c, we transferred it and obtained a specific pyramid layer $\widetilde{p}_{s+1}$ as Equation (4):

$$\widetilde{p}_{s+1} = \mathrm{T}_{s\to s+1}(p_{es}, p_{s+1}) = \mathrm{c}_{1\times 1}(\mathrm{D}(p_{es}); W^{s+1}_s) \oplus p_{s+1} \tag{4}$$

where $\mathrm{D}(p_{es})$ represents a downsampling operation in order for p_{es} to match the feature resolution with p_{s+1}. In addition, $\mathrm{c}_{1\times 1}$ represents a 1×1 convolutional layer, which can discern the corresponding channel number with learnable $W^{s+1}_s \in \mathbb{R}^{1\times 1\times c_s \times c_{s+1}}$.

3.4. DIoU Soft-NMS

How to better achieve object detection in dense scenarios has always been a research hotspot. With our observation in the AEMER dataset, the equipment obstruction in marine engine room can be roughly divided into the following conditions: the same class of equipment obstruction, different classes of equipment obstruction, and nonequipment obstruction. If we take the first case as an example in traditional NMS, all of the bounding boxes must first be sorted by the confidence score in descending order, and then the highest score box is selected. In addition, the rest of boxes might be suppressed if there is an obvious overlap with the selected box. However, what if the suppressed boxes have better location information than the selected box? As shown in Figure 6, when the separator (S2) overlaps with separator (S1), the detectors are readily confused as a result of the similar physical features of the separators. Therefore, the bounding box2 (B2) that should regress to S2 may be misguided to S1 or suppressed by the bounding box1 (B1) that should regress to S1, resulting in inaccurate positioning.

Considering that the model used in this paper might generate multiple prediction boxes, we adopted the Soft-NMS [37] postprocessing mechanism to ensure that each target was detected, and replaced the IoU metric with DIoU [20]. In other words, we should not abandon the prediction boxes that were mistakenly deleted due to the excessive overlap such NMS, but retain them by lowering their confidence scores. The pseudo code of DIoU Soft-NMS is shown in Figure 7.

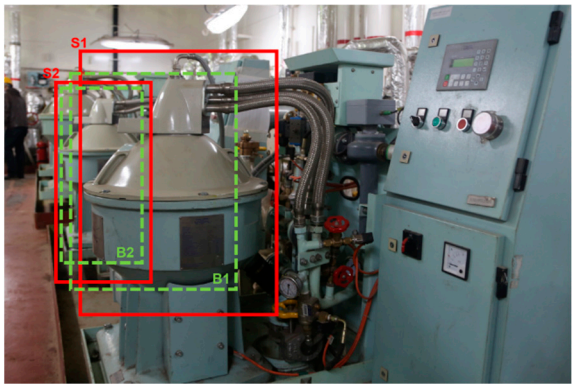


Figure 6. An example of auxiliary equipment detection in a congested marine engine room. B1 and B2 are the prediction boxes for the S1 and S2 target separators. In traditional NMS, B2 may be misguided to S1 or suppressed by B1.

Input:
$\mathcal{B} = \{b_1, .., b_N\}$, $\mathcal{S} = \{s_1, .., s_N\}$, N_t
$\mathcal{B}$ is the list of initial detection boxes
$\mathcal{S}$ contains corresponding detection scores
N_t is the NMS threshold

begin
  $\mathcal{D} \leftarrow \{\}$
  while $\mathcal{B} \neq$ *empty* **do**
    $m \leftarrow$ argmax $\mathcal{S}$
    $\mathcal{M} \leftarrow b_m$
    $\mathcal{D} \leftarrow \mathcal{D} \cup \mathcal{M}$; $\mathcal{B} \leftarrow \mathcal{B}$ - $\mathcal{M}$
    for b_i *in* $\mathcal{B}$ **do**
      if $DIoU(\mathcal{M}, b_i) \geq N_t$ **then**
        $s_i \leftarrow s_i f(DIoU(\mathcal{M}, b_i))$
      end
    end
  end
  return $\mathcal{D}, \mathcal{S}$
end

Figure 7. The pseudo code of DIoU Soft-NMS.

To improve the recall and eliminate the redundancy in marine engine room, the postprocessing mechanism takes into account the distance and the overlap relationship between multiple boxes. Here, the IoU metric in NMS was replaced by Equation (5):

$$DIoU = IoU - \left(\frac{\rho^2(b, b^{cand})}{c^2}\right)^{\beta} \tag{5}$$

where ρ is the Euclidean distance between the midpoints of two boxes. c is the diagonal length of the outer rectangular bounds covering the two boxes. b^{cand} and b represent the candidate boxes and highest score box. In addition, the penalty term, β, is assigned 1 generally, when it approaches zero, nearly all prediction boxes whose center points do not overlap with the center points of the highest score box are preserved; when it approaches infinity, the DIoU will degenerate to IoU, that is to say, the effectiveness of DIoU Soft-NMS can assimilate with greedy-NMS [38].

If the DIoU of candidate boxes with the highest score prediction box is greater than or equal to θ, the confidence score will be punished by Gauss rather than harshly setting as zero. As a result, the final confidence score function is shown in Equation (6):

$$s_i = \begin{cases} s_i e^{-\frac{DIoU(M,B_i)^2}{\delta}} & DIoU(M, B_i) \geq \theta \\ s_i & DIoU(M, B_i) < \theta \end{cases} \tag{6}$$

3.5. Loss Function

In the self-made dataset, we found that valves and meters made up a large majority of the auxiliary equipment in a marine engine room, compared with diesel engines or oil separators. To solve the imbalanced sample class, we first expanded the small percentage of equipment to minimize the negative impact of the dataset. Since the focal loss can well solve the passive influence of the traditional cross entropy loss on class imbalance and difficult to classify samples, the original classification function focal loss in RetinaNet was retained and defined by Equation (7):

$$FL(p, y) = \alpha^t * (1 - p_t)^\gamma * CE(p, y) \tag{7}$$

where γ represents the focusing parameter. α_t is an indicator variable. In our experiments, we set them as suggested in Ref. [28].

Regarding the regression loss function of the bounding boxes, Girshick et al. [5] proposed the smooth L1 loss, which combines the characteristics of L1 and L2. Considering that these functions cannot directly reflect the similarity of boxes, Yu et al. [18] proposed the IoU loss function, which treats the rectangular box as a whole. However, if there is no overlap between boxes, the IoU will always be zero and the model cannot learn. Therefore, Rezatofighi et al. [19] proposed a Generalized IoU (GIoU) loss function, which added a penalty term on the basis of the IoU. Zhang et al. [20] argued that the GIoU loss will degenerate into an IoU loss if the prediction box is wholly surrounded by the ground truth box, which might cause the model to fail to distinguish the relative position relationship. Accordingly, they proposed Distance IoU (DIoU) and Complete IoU (CIoU) loss, the former adds a penalty term for the center distance between boxes on the basis of IoU loss, and the latter adds a penalty term for the similarity of the aspect ratio on the basis of DIoU. By comprehensive comparison, we used the CIoU loss defined by Equation (8) as the regression loss.

$$L_{CIoU} = 1 - DIoU + \alpha v \tag{8}$$

where α and v, respectively, denote a positive tradeoff parameter and the consistency of aspect ratio, which is defined as Equation (9).

$$\begin{cases} v = \frac{4}{\pi^2}\left(\arctan\frac{w^{gt}}{h^{gt}} - \arctan\frac{w}{h}\right)^2 \\ \alpha = \frac{v}{(1-IoU)+v} \end{cases} \tag{9}$$

4. Experiments

The process of auxiliary equipment detection in marine engine room is presented in Figure 8. Raw images of auxiliary equipment were collected by the cabin acquisition device. Some of the images were processed by data augmentation, and then the auxiliary equipment in marine engine room (AEMER) dataset was built completely. We trained the RepVGG-RetinaNet detector on the AEMER, and the equipment was detected through the trained model eventually.

4.1. AEMER Dataset

With the resources of our 3D virtual engine room team, we quickly collected various images in the Very Large Container Ship (VLCS), Very Large Ore Carrier (VLOC), and Very Large Crude Carrier (VLCC) cabin scenes. Most of them were taken by our team through Canon digital cameras, and others were photographed by cabin monitoring. Due to fact that the angular variation of the image acquisition devices might cause inconsistent light intensity, we preprocessed the original images. Furthermore, to ease the passive influence of imbalanced class, the data augmentations we used included adding Gaussian noise, mirroring, rotating, shifting, color translation, and cutout. Specifically, we combined several augmentations to enhance the auxiliary equipment images that accounted for a small proportion. Then, we built the AEMER dataset with 7375 images, which contained

Cooler, Engine, Meter, Pump, Reservoir, Separator, and Valve. Figure 9 displays some raw image samples in the AEMER. In this paper, we randomly selected 70% of the data in AEMER as the training set, 20% as the validation set, and 10% as the test set.

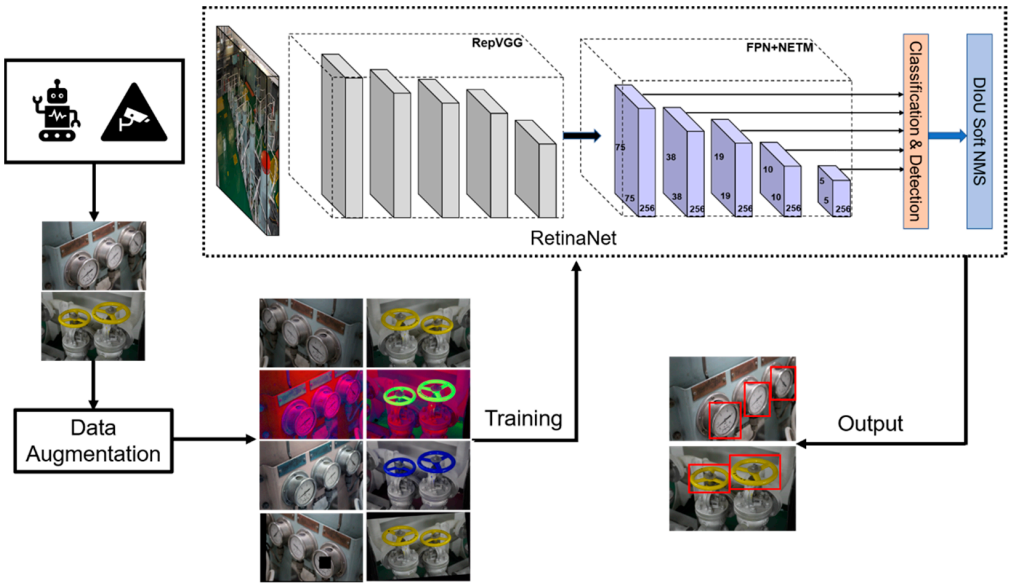


Figure 8. The process of auxiliary equipment detection in a marine engine room.

Figure 9. The original image samples in the AEMER.

4.2. Implementation

Our experiments were implemented according to the configuration in Table 1. For training details, we selected the RepVGG-B1g4 as the backbone for our proposed RetinaNet and trained it on the PASCAL VOC 07++12 trainval dataset (see the following Section 4.4.1). Next, we joined this to the trained weights of Section 4.4.1 to conduct contrast experiments on AEMER dataset. During the AEMER training period, we set the number of whole training iterations and the initial learning rate to 100 epochs and 1×10^{-4}, respectively. If the total loss did not reduce noticeably in four straight epochs, the learning rate would drop to 50% of the previous stage. Then, Adam was used to update the weights to accelerate model convergence. Furthermore, we treated the bounding box as a positive sample if the IoU was greater than 0.5, and as a negative sample if the IoU was less than 0.4. The hyperparameters of weighting factors and focusing parameter in focal loss were set to 2.5 and 0.25 respectively.

Table 1. The implementation of our experiments.

Configuration	Detail
CPU	Inter i7-9700 (3.00 GHz) 8-core
GPU	A single NVIDIA GeForce GTX 1660Ti
RAM	16 GB
Operating System	Windows 10
IDE	PyCharm 2020.1.4
Framework	GPU-based PyTorch-1.4.0
Toolkit	CUDA 11.3

4.3. Evaluation Criteria

In the object detection task, the image information generally consisted of background and foreground (targets). When the current foreground was correctly detected by detectors, we denoted the prediction boxes as true positive (TP). When the current foreground was misdetected as background or other foregrounds by detectors, we denoted the prediction boxes as false positive (FP). When the background was misdetected as foreground by detectors, we denoted the prediction boxes as false negative (FN). Otherwise, we denoted the prediction boxes as true negative (TN). On the basis of the four situations, precision defined by Equation (10) and recall defined by Equation (11) were introduced to evaluate the detection accuracy. Every class can generate a P–R curve according to precision and recall, and the enclosed area of the curve and the coordinate axis in the range of (0,1) was the average precision (AP). The mean average precision (mAP) has been widely used in target detection and evaluation. In addition to detection accuracy, another important evaluation metric for detection is speed. Only high speed can achieve realtime detection. Generally, FPS is used to evaluate the speed of object detection, that is, the number of images that can be processed per second. In this paper, the evaluation criteria we used contained: precision, recall, AP, mAP, and FPS.

$$P_{recision} = \frac{TP}{TP + FP} \tag{10}$$

$$R_{ecall} = \frac{TP}{TP + FN} \tag{11}$$

4.4. Results Analysis

4.4.1. Results on PASCAL VOC

In this subsection, The PASCAL VOC2007 trainval and 2012 trainval were used as the training datasets. The backbone of RepVGG-B1g4 and NETM were employed in RetinaNet. For training details, we trained the RepVGG-B1g4-RetinaNet for 120 epochs and set the initial learning rate to 0.001. The learning rate would drop to 10% of the previous stage if the iterations reached the 50th epoch and 80th epoch.

We demonstrate our method on PASCAL VOC 2007 test set with the major purpose of comparing the FPS and mAP with other typical methods. The comparison results are listed in Table 2. With a comparable detection speed, our proposed RepVGG-RetinaNet carried out an appreciable improvement in detection compared to others. Specifically, the mAP of our detector on PASCAL VOC 2007 test set was 3.3%, 0.2%, 6.0%, 2.5%, 2.0%, 1.1%, 1.2%, 0.9%, 0.2%, 0.2%, and 0.4% better than Faster R-CNN [6], R-FCN [39], YOLOv2 [8], SSD [10], DSOD [40], DSSD [23], RSSD [41], FSSD [25], ASSD [26], RefineDet [42], and RetinaNet, respectively.

Table 2. Comparison of the test detection results on PASCAL VOC2007. The detection speed is measured by FPS with a single 1660Ti GPU using a batch size of 1.

Method	Pretrain	Backbone	Input Size	GPU	FPS	mAP
Faster R-CNN	√	VGG	~600 × 1000	Titan X	7	73.2
	√	ResNet-101		K40	2.4	76.4
R-FCN	√	ResNet-50	~600 × 1000	-	-	77.0
	√	ResNet-101		K40	5.8	79.5
YOLOv2	√	DarkNet-19	352 × 352	Titan X	81	73.7
SSD	√	VGG	300 × 300	Titan X	46	77.2
DSOD	×	DS/64-192-48-1	300 × 300	Titan X	17.4	77.7
DSSD	√	ResNet-101	321 × 321	Titan X	9.5	78.6
RSSD	√	VGG	300 × 300	Titan X	35	78.5
FSSD	√	VGG	300 × 300	1080Ti	65.8	78.8
ASSD	√	ResNet-101	321 × 321	K40	11.4	79.5
RefineDet	√	VGG	320 × 320	K80	12.9	79.5
RetinaNet	√	ResNet-50	600 × 600	1660Ti	17.4	79.3
Ours	√	RepVGG-B1g4	600 × 600	1660Ti	21.8	79.7

4.4.2. Ablation Study on AEMER

To validate the contribution of our strategies, we performed an ablation study on the AEMER dataset to explore the effects of backbone, NETM, DIoU Soft-NMS, and CIoU loss on detection accuracy and speed. In this experiment, we reconstructed the feature pyramid network with NETM and replaced the bounding box regression loss function of the original RetinaNet. At the same time, we applied the DIoU Soft-NMS postprocessing mechanism during training. The comparison results are shown in Table 3, where the largest difference of M1-4 and M5-8 lay in the backbone. M5 was 0.47% and 16% better than M1 in terms of mAP and Time, M6 was 1.12%/34% better than M2, M7 was 0.91%/16% better than M3, M8 was 2.07%/34% better than M4. With the support of NETM, DIoU Soft-NMS, and CIoU, the result of each strategy enjoyed significant mAP improvement and a faster inference time.

Table 3. Ablation study on AEMER dataset. Here we investigated three models, the original RetinaNet (M1), ResNet-RetinaNet (M2-4), and RepVGG-B1g4-RetinaNet (M5-8). The NETM represented the neighbor erasing and transferring mechanism. The BBRL was the bounding box regression loss. The detection speed was measured by Time with a single 1660Ti GPU using a batch size of 1.

Backbone	M	NETM	CIoU BBRL	DIoU Soft-NMS	Time (ms)	mAP
ResNet-50	1	-	-	-	58	79.34
	2	√	-	-	61	80.03
	3	-	√	√	58	79.46
	4	√	√	√	61	80.21
RepVGG-B1g4	5	-	-	-	49	79.81
	6	√	-	-	40	81.15
	7	-	√	√	49	80.37
	8	√	√	√	40	82.29

4.4.3. Comparison with Others

As can be seen from Table 4, we evaluated our RepVGG-RetinaNet on the AEMER dataset with other methods and presented the contrastive AP results of each equipment. The mAP of our detector (82.29%) was 6.16% higher than Faster R-CNN (76.13%), 5.44% higher than R-FCN (76.85%), 3.47% higher than YOLOv3 (78.82%), 6.08% higher than SSD (76.21%), 4.33% higher than FSSD (77.96%), 3.23% higher than ASSD (79.06%), 5.34% higher than RefineDet (76.95%), and 2.95% higher than original RetinaNet (79.34%). In summary, Table 4 shows the favorable accuracy-speed tradeoff of RepVGG-RetinaNet and verifies the significance of our proposed detector in a cabin.

Table 4. Comparison of the test detection results with the other classical methods on the AEMER dataset.

Method	Backbone	Cooler	Engine	Meter	Pump	Reservoir	Separator	Valve	FPS	mAP
Faster R-CNN	ResNet-50	90.96	93.77	43.81	82.11	86.95	84.83	50.49	8.53	76.13
R-FCN	ResNet-101	88.47	94.12	55.02	86.65	71.65	87.55	54.38	21.72	76.85
YOLOv3	DarkNet-53	87.95	100.00	57.16	88.34	73.93	87.87	56.49	29.86	78.82
SSD	VGG	83.53	100.00	46.22	89.46	71.05	91.71	51.48	27.99	76.21
FSSD	VGG	84.30	100.00	47.94	89.79	76.49	93.60	53.62	24.26	77.96
ASSD	ResNet-101	85.85	100.00	49.53	90.39	78.57	93.90	55.18	17.94	79.06
RefineDet	VGG	93.91	100.00	45.95	89.92	67.20	90.51	51.16	23.67	76.95
RetinaNet	ResNet-50	88.80	100.00	57.21	94.03	48.81	95.69	70.86	17.24	79.34
Ours	RepVGG-B1g4	93.44	100.00	60.26	95.30	55.01	97.68	74.37	24.98	82.29

4.4.4. Visualization

We randomly selected several images from the AEMER dataset to visually compare the detection results between the original RetinaNet (upper) and the RepVGG-RetinaNet (lower) in Figure 10. We set the IOU threshold and confidence threshold to 0.5 and 0.25 respectively. There were nine whole valves in (a), both RetinaNet and RepVGG-RetinaNet detected all of them, but the latter had higher confidence scores than the former. In (b), the RetinaNet mistakenly identified one meter as a valve, and completely missed the four small valves in the lower left corner. By comparison, RepVGG-RetinaNet detected one more valve than RetinaNet. There were some multi\scale objects in (c), including two meters, one reservoir, and one valve. Both RetinaNet and RepVGG-RetinaNet detected the large reservoir, but the latter correctly detected one tiny valve more than the former. Furthermore, we present the detection results in the congested scenarios (d–f), the RepVGG-RetinaNet had a higher confidence probability and smaller position deviation in the practical cabin. From the perspective of the detection error, our detector alleviated the problem of missed detection and false detection, but the accuracy on valves and meters was not perfect. In summary, the overall performance of our RepVGG-RetinaNet was superior to the original RetinaNet, and had the robust ability of adaptive filtering feature information.

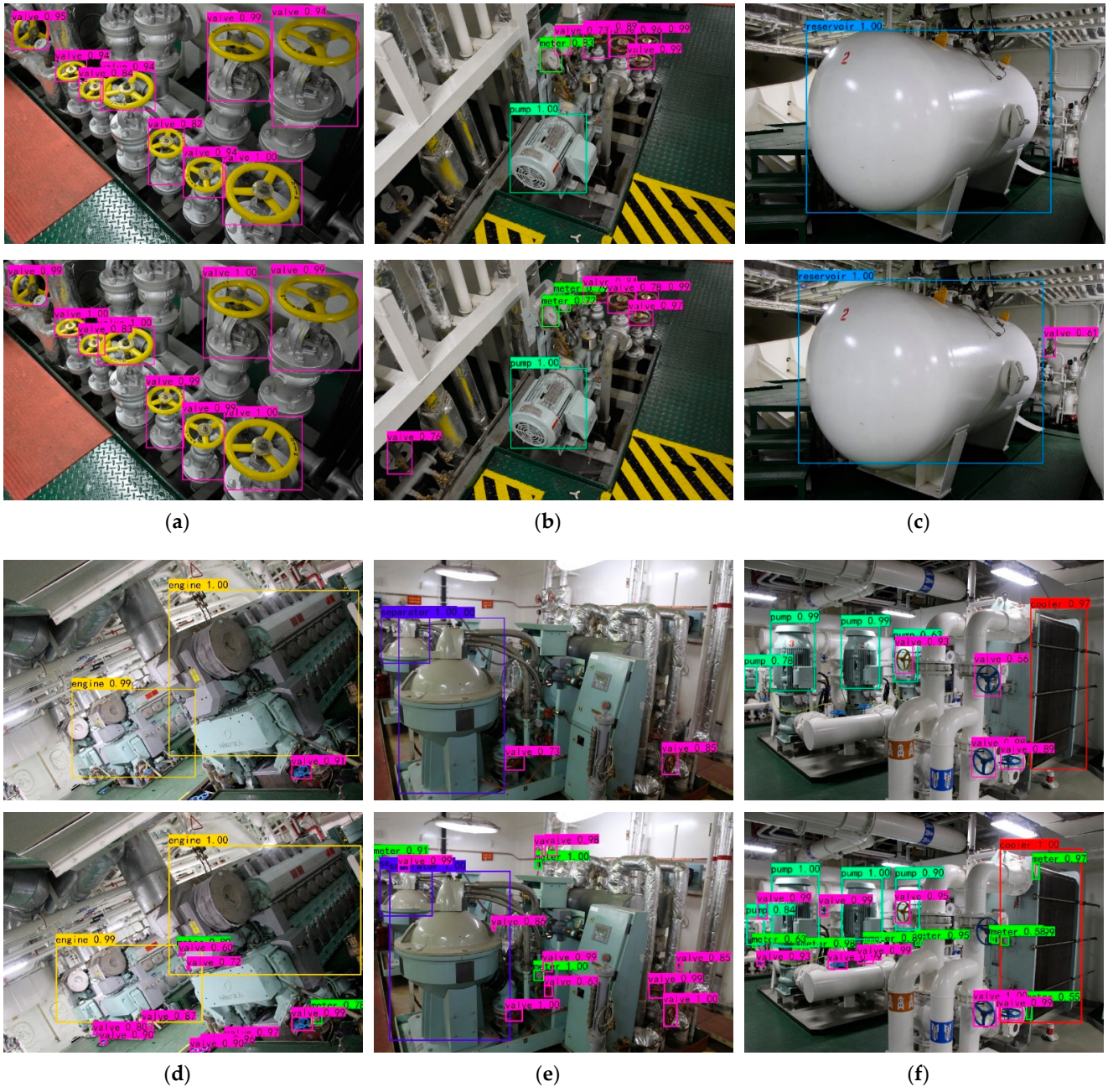


Figure 10. Detection results visualization on AEMER dataset. In column (**a**) we present the auxiliary equipment of valves. In other columns, we present the equipment of pumps (**b**), meters (**b**), reservoirs (**c**), auxiliary engines (**d**), coolers (**f**), and oil separators (**e**). In six of the comparisons, the upper images are the output of the original RetinaNet and the lower images are the output of the proposed RepVGG-RetinaNet.

5. Conclusions and Discussion

Considering the key technology of intelligent perception in a marine engine room, we built the AEMER dataset and proposed a RepVGG-RetinaNet detector for auxiliary equipment in a congested cabin. According to the analysis of experimental data, the following main conclusions can be reached: (1) Compared with ResNet, the backbone of RepVGG in RetinaNet has better detection performance in practical cabin scenes. (2) The FPN with the feature scale unmixing NETM is capable of helping the detector to have an adaptive filtering function, which enhances the expression ability of small-scale features and effectively solves the misdetection and false positive problems. (3) Through the improvement of the RetinaNet based on the CIoU regression loss as well as DIoU Soft-NMS

postprocessing mechanism, we have further advanced the detection accuracy of auxiliary equipment in the cabin. (4) The proposed RepVGG-RetinaNet has comparable detection speed and accuracy, which meets the elementary demands of the inspection tasks in marine engine room, and effectively provides technical support for the defect recognition and anomaly detection of the appearance of equipment in the centralized monitoring and alarm system.

As for the low detection accuracy of Reservoir and Meter, we tried to modify the framework and parameters, but the results were unsatisfactory. Therefore, we strategically shifted focus on improving the mAP and temporarily gave up the AP. In future work, we will fully consider the AP of single class and try to further improve the detection accuracy of small-scale targets in the cabin. Moreover, the AEMER constructed for the inspection task has not been ideal, it will be supplemented and expanded in the future. Meanwhile, the new semantic information will be added to the dataset, and unknown object detection will be carried out by modifying the model and combining zero-sample classifiers, so that the detector might have the ability of self-learning and self-updating, which will further realize the full-smart visual perception.

Author Contributions: Conceptualization, J.Q. and Q.M.; methodology, J.Q.; software (PyCharm Version: 2020.1.4. Access date: 10 June 2021), Q.M.; validation, J.Q. and Q.M; formal analysis, J.Z.; investigation, J.Q.; resources, J.Z.; data curation, J.Q.; writing—original draft preparation, J.Q.; writing—review and editing, J.Z.; visualization, Q.M.; supervision, J.Z.; project administration, J.Q.; funding acquisition, J.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by National Science Foundation under Grant U1905212.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The processed data cannot be shared at this time as the data also forms part of an ongoing study.

Acknowledgments: This original images of AEMER dataset were taken by our 3D virtual engine room team. Thanks for their support in this work.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Sasaki, Y.; Emaru, T.; Ravankar, A.A. SVM based Pedestrian Detection System for Sidewalk Snow Removing Machines. In Proceedings of the IEEE/SICE International Symposium on System Integration, Iwaki, Japan, 11–14 January 2021; pp. 700–701.
2. Ranjan, R.; Patel, V.M.; Chellappa, R.A. Deep Pyramid Deformable Part Model for Face Detection. In Proceedings of the IEEE 7th International Conference on Biometrics Theory, Applications and Systems, Arlington, VA, USA, 8–11 September 2015.
3. Zhao, Z.Q.; Zheng, R.; Xu, S.T.; Wu, X.D. Object detection with deep learning: A review. *IEEE Trans. Neural Netw. Learn. Syst.* **2019**, *30*, 3212–3232. [CrossRef] [PubMed]
4. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 580–587.
5. Girshick, R. Fast R-CNN. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Santiago, Chile, 7–13 December 2015; pp. 1440–1448.
6. Ren, S.Q.; He, K.M.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *Adv. Neural Inf. Process. Syst.* **2015**, *39*, 1137–1149. [CrossRef] [PubMed]
7. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
8. Redmon, J.; Farhadi, A. YOLO9000: Better, Faster, Stronger. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 6517–6525.
9. Redmon, J.; Farhadi, A. Yolov3: An incremental improvement. *arXiv* **2018**, arXiv:1804.02767.
10. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. SSD: Single shot multibox detector. In Proceedings of the 14th European Conference on Computer Vision, Amsterdam, Netherlands, 8–16 October 2016; Volume 9905, pp. 21–37.
11. Lin, T.Y.; Dollar, P.; Girshick, R.; He, K.M.; Hariharan, B.; Belongie, S. Feature Pyramid Networks for Object Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 936–944.

12. Singh, B.; Davis, L.S. An analysis of scale invariance in object detection snip. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 3578–3587.
13. Singh, B.; Najibi, M.; Davis, L.S. SNIPER: Efficient multi-scale training. *Adv. Neural Inf. Process. Syst.* **2018**, *31*, 9310–9320.
14. Cai, Z.W.; Fan, Q.F.; Feris, R.S.; Vasconcelos, N. A unified multi-scale deep convolutional neural network for fast object detection. In Proceedings of the 14th European Conference on Computer Vision, Amsterdam, Netherlands, 8–16 October 2016; Volume 9908, pp. 354–370.
15. Kong, T.; Sun, F.C.; Huang, W.B.; Liu, H.P. Deep feature pyramid reconfiguration for object detection. In Proceedings of the 15th European Conference on Computer Vision, Munich, Germany, 8–14 September 2018; Volume 11209, pp. 172–188.
16. He, K.M.; Zhang, X.Y.; Ren, S.Q.; Sun, J. Spatial pyramid pooling in deep convolutional networks for visual recognition. In Proceedings of the 13th European Conference on Computer Vision, Zurich, Switzerland, 6–12 September 2014; Volume 8691, pp. 346–361.
17. Liu, S.T.; Huang, D.; Wang, Y.H. Receptive field block net for accurate and fast object detection. In Proceedings of the 15th European Conference on Computer Vision, Munich, Germany, 8–14 September 2018; Volume 11215, pp. 404–419.
18. Wang, X.L.; Xiao, T.T.; Jiang, Y.N.; Shao, S.; Sun, J. Repulsion Loss: Detecting Pedestrians in a Crowd. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 7774–7783.
19. Yu, J.H.; Jiang, Y.N.; Wang, Z.Y.; Cao, Z.M.; Huang, T. UnitBox: An advanced object detection network. In Proceedings of the 2016 ACM Multimedia Conference, Amsterdam, Netherlands, 15–19 October 2016; pp. 516–520.
20. Rezatofighi, H.; Tsoi, N.; Gwak, J.; Sadeghian, A.; Reid, I.; Savarese, S. Generalized intersection over union: A metric and a loss for bounding box regression. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 658–666.
21. Zheng, Z.H.; Wang, P.; Liu, W.; Li, J.Z.; Ye, R.G.; Ren, D.W. Distance-IoU loss: Faster and better learning for bounding box regression. In Proceedings of the 34th AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; pp. 12993–13000.
22. Zhu, C.C.; He, Y.H.; Savvides, M. Feature Selective Anchor-Free Module for Single-Shot Object Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 840–849.
23. Zhu, C.C.; Chen, F.Y.; She, Z.Q.; Savvides, M. Soft Anchor-Point Object Detection. In Proceedings of the 16th European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020; Volume 12354, pp. 91–107.
24. Fu, C.; Liu, W.; Ranga, A.; Tyagi, A.; Berg, A. DSSD: Deconvolutional single shot detector. *arXiv* **2017**, arXiv:1701.06659.
25. Li, Z.; Zhou, F. FSSD: Feature Fusion Single Shot Multibox Detector. *arXiv* **2017**, arXiv:1712.00960.
26. Yi, J.R.; Wu, P.X.; Metaxas, D.N. ASSD: Attentive single shot multibox detector. *Comput. Vis. Image Underst.* **2019**, *189*, 102827. [CrossRef]
27. Gu, Y.; Luxburg, U.V.; Bengio, S.; Wallach, H.; Fergus, R.; Vishwanathan, S.; Garnett, R. Attention Is All You Need. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 5998–6008.
28. Lin, T.S.; Goyal, P.; Girshick, R.; He, K.M.; Dollar, P. Focal Loss for Dense Object Detection. In Proceedings of the 16th IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2999–3007.
29. Li, Y.Z.; Pang, Y.W.; Cao, J.L.; Shen, J.B.; Shao, L. Improving Single Shot Object Detection with Feature Scale Unmixing. *IEEE Trans. Image Process.* **2021**, *30*, 2708–2721. [CrossRef] [PubMed]
30. Ding, X.; Zhang, X.; Ma, N.; Han, J.; Ding, G.; Sun, J. RepVGG: Making VGG-style ConvNets Great Again. *arXiv* **2021**, arXiv:2101.03697.
31. Shelharmer, E.; Long, J.; Darrell, T. Fully Convolutional Networks for Semantic Segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 640–651. [CrossRef] [PubMed]
32. He, K.M.; Zhang, X.Y.; Ren, S.Q.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
33. Howard, A.G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Adam, H. MobileNets: Efficient convolutional neural networks for mobile vision applications. *arXiv* **2017**, arXiv:1704.04861.
34. Sandler, M.; Howard, A.G.; Zhu, M.; Zhmoginov, A.; Chen, L.C. MobileNetV2: Inverted Residuals and Linear Bottlenecks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 4510–4520.
35. Howard, A.G.; Sandler, M.; Chen, B.; Wang, W.J.; Chen, L.C. Searching for MobileNetV3. In Proceedings of the 17th IEEE International Conference on Computer Vision, Seoul, Korea, 27 October–2 November 2019; pp. 1314–1324.
36. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. In Proceedings of the 3rd International Conference on Learning Representations 2015, San Diego, CA, USA, 7–9 May 2015.
37. Bodla, N.; Singh, B.; Chellappa, R.; Davis, L.S. Soft-NMS—Improving Object Detection with One Line of Code. In Proceedings of the 16th IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 5562–5570.
38. Neubeck, A.; Van Gool, L. Efficient non-maximum suppression. In Proceedings of the 18th International Conference on Pattern Recognition, Hong Kong, China, 20–24 August 2006; Volume 3.
39. Dai, J.F.; Li, Y.; He, K.M.; Sun, J. R-FCN: Object Detection via Region-based Fully Convolutional Networks. *Adv. Neural Inf. Process. Syst.* **2016**, *29*, 379–387.
40. Shen, Z.Q.; Liu, Z.; Li, J.G.; Jiang, Y.G.; Chen, Y.R.; Xue, X.Y. DSOD: Learning Deeply Supervised Object Detectors from Scratch. In Proceedings of the 16th IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 1937–1945.

41. Jeong, J.; Park, H.; Kawak, N. Enhancement of SSD by concatenating feature maps for object detection. In Proceedings of the 28th British Machine Vision Conference, London, UK, 4–7 September 2017.
42. Zhang, S.F.; Wen, L.Y.; Bian, X.; Lei, Z.; Li, S.Z. Single-Shot Refinement Neural Network for Object Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 4203–4212.