remote sensing

MDPI

Article

Improved Neural Network with Spatial Pyramid Pooling and Online Datasets Preprocessing for Underwater Target Detection Based on Side Scan Sonar Imagery

Jinrui Li [1,2,†], Libin Chen [1,†], Jian Shen [3,†], Xiongwu Xiao [2,*], Xiaosong Liu [4], Xin Sun [4], Xiao Wang [5] and Deren Li [2]

[1] College of Resources and Environment, Yangtze University, Wuhan 430100, China
[2] State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University, Wuhan 430072, China
[3] School of Computer Science, Wuhan University, Wuhan 430072, China
[4] Coal Geological Geophysical Exploration Surveying & Mapping Institute of Shanxi Province, Jinzhong 030621, China
[5] School of Marine Technology and Geomatics, Jiangsu Ocean University, Lianyungang 222000, China
* Correspondence: xwxiao@whu.edu.cn; Tel.: +86-186-0276-2010
† These authors contributed equally to this work.

Abstract: Fast and high-accuracy detection of underwater targets based on side scan sonar images has great potential for marine fisheries, underwater security, marine mapping, underwater engineering and other applications. The following problems, however, must be addressed when using low-resolution side scan sonar images for underwater target detection: (1) the detection performance is limited due to the restriction on the input of multi-scale images; (2) the widely used deep learning algorithms have a low detection effect due to their complex convolution layer structures; (3) the detection performance is limited due to insufficient model complexity in training process; and (4) the number of samples is not enough because of the bad dataset preprocessing methods. To solve these problems, an improved neural network for underwater target detection—which is based on side scan sonar images and fully utilizes spatial pyramid pooling and online dataset preprocessing based on the You Look Only Once version three (YOLO V3) algorithm—is proposed. The methodology of the proposed approach is as follows: (1) the AlexNet, GoogleNet, VGGNet and the ResNet networks and an adopted YOLO V3 algorithm were the backbone networks. The structure of the YOLO V3 model is more mature and compact and has higher target detection accuracy and better detection efficiency than the other models; (2) spatial pyramid pooling was added at the end of the convolution layer to improve detection performance. Spatial pyramid pooling breaks the scale restrictions when inputting images to improve feature extraction because spatial pyramid pooling enables the backbone network to learn faster at high accuracy; and (3) online dataset preprocessing based on YOLO V3 with spatial pyramid pooling increases the number of samples and improves the complexity of the model to further improve detection process performance. Three-side scan imagery datasets were used for training and were tested in experiments. The quantitative evaluation using Accuracy, Recall, Precision, mAP and F1-Score metrics indicates that: for the AlexNet, GoogleNet, VGGNet and ResNet algorithms, when spatial pyramid pooling is added to their backbone networks, the average detection accuracy of the three sets of data was improved by 2%, 4%, 2% and 2%, respectively, as compared to their original formulations. Compared with the original YOLO V3 model, the proposed ODP+YOLO V3+SPP underwater target detection algorithm model has improved detection performance through the mAP qualitative evaluation index has increased by 6%, the Precision qualitative evaluation index has increased by 13%, and the detection efficiency has increased by 9.34%. These demonstrate that adding spatial pyramid pooling and online dataset preprocessing can improve the target detection accuracy of these commonly used algorithms. The proposed, improved neural network with spatial pyramid pooling and online dataset preprocessing based on the YOLO V3 method achieves the highest scores for underwater target detection results for sunken ships, fish flocks and seafloor topography, with mAP scores of 98%, 91% and 96% for the above three kinds of datasets, respectively.

check for updates

Citation: Li, J.; Chen, L.; Shen, J.; Xiao, X.; Liu, X.; Sun, X.; Wang, X.; Li, D. Improved Neural Network with Spatial Pyramid Pooling and Online Datasets Preprocessing for Underwater Target Detection Based on Side Scan Sonar Imagery. *Remote Sens.* **2023**, *15*, 440. https://doi.org/10.3390/rs15020440

Academic Editor: Andrzej Stateczny

Received: 1 December 2022
Revised: 29 December 2022
Accepted: 7 January 2023
Published: 11 January 2023

Keywords: underwater target detection; spatial pyramid pooling; online datasets preprocessing; YOLO V3; side scan sonar imagery

1. Introduction

In the field of underwater target detection, there are many cases of effective object detection and recognition. For example, balancing learning (BL) for high-quality Synthetic Aperture Radar (SAR) ship detection [1], two-stage faster region-based convolutional neural network [2], novel quad feature pyramid network [3] and hybrid task cascade plus (HTC+) for SAR ship instance segmentation [4] have been applied in the synthetic aperture radar learning approach for target detection and recognition from remote sensing images. Taken together, these experimental results demonstrate improvements in the recognition and the detection of underwater targets when utilizing a high-speed focusing algorithm for circular synthetic aperture radar [5], high-speed and high-accurate detector (H2Det) with SAR images [6], HyperLi-Net for high-accuracy and high-speed SAR ship detection [7]. Exploiting a balance scene learning mechanism for offshore and inshore target detection [8] improves detection accuracy and efficiency.

In the underwater target detection field, side scan sonar has been widely used in marine research [9], seabed terrain classification [10] and oceanographic surveys [11] because it can acquire a large number of images within the sonar measurement range [12], which is also a classification of detection capability of side-scan sonar and the need for the coverage of a specified sweeping area [13]. But with the processing of the side-scan sonar datasets, the traditional method is cumbersome and inefficient; it does not work very well. The traditional machine learning method is cumbersome and inefficient, cannot completely realize high-accuracy end-to-end target detection and is limited by the structure of the models influencing the detection time [14]. At present, the application of deep learning methods in many fields has achieved excellent results and, in recent years, has also been successful [15].

At present, the methods of target detection in side scan sonar images can be classified as contour extraction and image segmentation methods. Wang et al. [16] used a particle swarm optimization algorithm to improve the threshold segmentation method to segment the vector and improve the accuracy and efficiency of underwater target segmentation. Gong et al. [17] combined the fast fuzzy c-means clustering algorithm (SCFFCM) with MRF to effectively improve the ability of noise suppression in sonar image segmentation. Dzieciuch et al. [18] reported on the preliminary methods and results of applying a non-linear classification method, convolutional neural networks (CNN) to mine detection in noisy sonar imagery. Rhinelander et al. [19] adopted extra feature extraction, and data engineering can result in better classification performance compared to parameter optimization alone for target classification with side scan sonar. Song et al. [20] proposed using deep CNN to segment images of side scan sonar into three parts: highlight areas with objects, regions of shadow and sea-bottom reverberation areas. Zhu et al. [21] proposed and tested an approach on a set of sonar images obtained by a UUV equipped with side scan sonar. Automatic target detection was achieved through the use of matched filters, while target classification is achieved with the trained SVM classifier based on features extracted by the CNN. Einsidler et al. [22] demonstrated the application of deep learning for ATR in side scan sonar imagery; in particular, supervised-learning Region-based CNN (R-CNN), for detecting objects in sonar images. Kim et al. [23] proposed a convolutional neural network algorithm based on Faster R-CNN (Region based Convolutional Neural Networks) learning based on the region of interest in which the details of the neural network are self-organized to fit the data. Wu et al. [24] proposed a novel Convolutional neural network with an Efficient Convolutional Network (ECNet) architecture to implement semantic segmentation of side scan sonar images. Wang et al. [25] proposed a real-time semantic segmentation network RT-Seg for side scan sonar images. Yu et al. [26] employed the R2CNN module to obtain accurate sonar image features, which reduces errors and improves accuracy. Additionally, a self-guidance module was introduced to ensure the network's stability and to optimize the segmentation results. Wang et al. [27] proposed an underwater object detection method based on the YOLO V3 network, demonstrating that the YOLO V3 network is an effective way to improve the accuracy of underwater object detection. Li et al. [28] studied the problems of low accuracy, low efficiency and low missed detection rate in sonar image target detection. The You Only Look Once version 3 (YOLO V3) model is used to perform one-time neural network processing on side scan sonar images applied to sonar image detection in this paper. Jin et al. [29] proposed inputting sonar images directly into the improved convolutional neural network (CNN) for seabed sediment classification without extracting image features.

As demonstrated by this research, deep learning has been widely used in side scan sonar image detection. However, deep learning networks cannot be applied to different resolutions and different scales, while the sample input scales and scales of the network processing create inconsistency issues because of the limited inputting scales of the images. The usual approach is to compress, expand or crop the images [30], but this will ultimately limit the accuracy of object detection because of the loss of image information [31] and add distortion to the entire image [32]. Some methods of side scan sonar image target detection and some conventional neural networks, such as Alexnet, GoogleNet, ResNet, etc. have low accuracy [33] and add complexity to network structures in the process of side scan sonar image processing [34].

Because of limiting the scales of the input images and classical target detection methods, some strategies and solutions that we can adopt in typical target detection based on side scan sonar images are as follows: (1) For the problem that the existing network is difficult to apply to different resolutions and different scales, and the inconsistency between the sample input scales [35] and the network processing scales. We propose to add an algorithm to the model to break the limiting of the input scales on the detection process. Spatial pyramid pooling can use pooling to divide the CNN features into several fixed sizes [36], which can solve the problems in the process of image detection in a certain extent. (2) To solve the problem of low accuracy of side scan sonar image detection using some classical convolutional neural network methods due to the structure of the model, we can use the YOLO network for further network optimization and improvement. Besides, we also adopted online dataset preprocessing (ODP) for improving the experimental results and achieving high detection accuracy.

In this paper, we propose an improved deep learning network based on spatial pyramid pooling and online dataset preprocessing for side scan sonar image object detection to resolve the multi-scale inputting problem [37]. Our proposed approach uses online dataset processing, with random flips to increasing the number of samples and enhancing the complexity of the model [38], so as to improve detection accuracy. We use the data expansion method to preprocess the data [39], thus meeting the pre-data preparation requirements for image classification. Our approach uses spatial pyramid pooling in which multiple pools of different sizes with the size of the output feature vector [40] are fixed to realize the input operations with images of different sizes. The spatial pyramid pooling network is used for target detection as it can adapt to multi-dimensional inputting [41], automatically extracting missing input vector features. Spatial pyramid pooling has successfully achieved accuracy improvements in image target detection as it is added between the feature extraction module and the fully connected layer [42]. To further address the limited multi-scale inputting problem, we also adopted the YOLO V3 as this is an output network that uses multi-scale features for object detection and adjusts the basic network structure to realize efficient high-accuracy detection performance.

The improvement of the lightweight network algorithm based on YOLO V3 has had a lot of fruitful research. Yue Li et al. [43] proposed an improved YOLO-SC (YOLO-Submarine Cable) detection method, based on the YOLO V3 algorithm, to build a testing environment for submarine cables and to create a submarine cable image dataset. Moran

Ju et al. [44] proposed a multi-scale target detection algorithm involving the improvement of YOLO V3 by a mathematical derivation method based on Intersection over Union (IOU). Xu et al. [45] adopted DenseNet (Densely Connected Network) based on the YOLO V3 in detecting remote sensing targets to enhance feature extraction capability. What this research has in common with the above proof applications is the improved YOLO V3 algorithm. What distinguishes this research from similar applications is that we propose an online data preprocessing method to increase the amount of data from the data input; thus, the complexity of the model can be improved, and better experimental results can be obtained.

A large number of systematic experiments indicate that, by adding the spatial pyramid pooling, the evaluation of side scan image target detection with the commonly used networks can be significantly improved. Three kinds of datasets were used for quantitative evaluation in the experiments, including the Sunken ship, Fish flock and Seafloor topography. The evaluation indicators, [46] including Accuracy, Recall, Precision, and F1 Score, measured the overall accuracy of the model prediction. These evaluation experiments assessed the efficiency of the network model as modified in the experiment compared with the unimproved networks. Based on the spatial pyramid pooling, the YOLO V3 used in this paper for three kinds of targets shows greater efficiency than AlexNet, GoogleNet, VGGNet and ResNet neural networks. This result shows that spatial pyramid pooling effectively deals with the problem of missing data, improves the efficiency of the model and can have higher detection accuracy.

The results show that the proposed approach effectively realizes higher detection accuracy. The proposed online dataset preprocessing approach incorporating random flip, based on the YOLO V3 model with spatial pyramid pooling, increased the number of samples and enhanced the complexity of the model, thus improving detection accuracy. This study can provide theoretical and practical value for improved detection when using side scan sonar images. The rest of this paper is organized as follows. Section 2 presents the principle methodology for the paper. Experiment details, including datasets description and preprocessing, training model process, results and discussion, etc., are shown in Section 3. Conclusions are drawn in Section 4.

2. Methodology

2.1. Improved Neural Networks with Spatial Pyramid Pooling and Online Datasets Preprocessing for Underwater Target Detection

As shown in Figure 1, we start with dataset preprocessing. In the experiment, five kinds of neural network frameworks for target detection were adopted, and then spatial pyramid pooling was added into the five neural network structures, respectively. Online dataset preprocessing (ODP) is input to the network that is our proposed target detection method based on the YOLO V3 algorithm with spatial pyramid pooling and online dataset preprocessing for side scan sonar underwater images.

Finally, five kinds of neural networks were compared between those with spatial pyramid pooling added and those without it added to verify the effects of spatial pyramid pooling; based on spatial pyramid pooling with YOLO V3, the performance was compared between the online datasets preprocessing being adopted and without it being adopted to verify the effectiveness of the online datasets preprocessing.

2.2. Datasets Preprocessing

Seafloor conditions are complex and changeable [47], affected by irregular water flow and light [48], and the imaging quality is spotty. In addition, the characteristics of sonar images, such as blurred edges and low contrast [49], will eventually cause serious problems, such as image blurring and color distortion. Therefore, preprocessing should improve the quality of side scan sonar images, which can make the target contour more prominent. Preprocessing permits effective identification of observation targets. The preprocessing scheme of datasets is as follows.

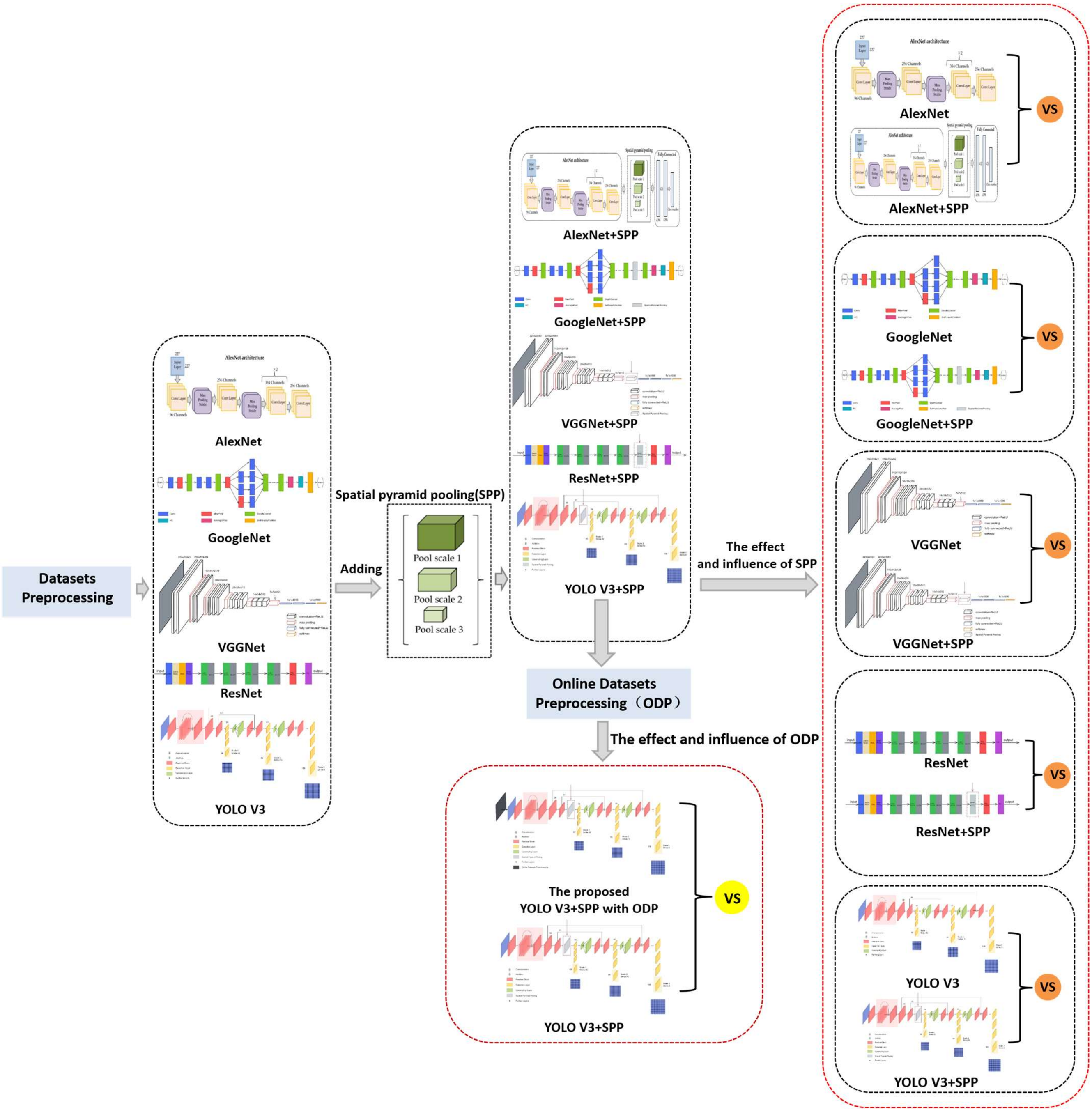

Figure 1. The overall experiment flow chart and our proposed ODP+YOLO V3+SPP method.

(1) The expanded datasets

The original dataset consisted of three categories (sunken ship, fish flock and seafloor topography), each with 50 images and a total of 150 images and with 250 images and a total of 750 images (Figure 2).

(2) Data Augmentation

The original sample data dataset used in this paper is small, and thus subject to overfitting [50] in the absence of more training samples. The number of different types of images in the training set is often not balanced enough [51], leading to bias phenomena [52] in the trained model representation to effectively train the network. The original dataset is augmented to improve the detection accuracy and generalization ability of the model.

By introducing a correlate specifying the directory where the original data is located, the augmentation method is set to geometrically rotate (to the left or the right) with a probability of 0.7 to within 25 degrees of geometric rotation. Sampling 250 images, i.e., augmentation to 250 images per category, for a total of 750 images (Table 1).

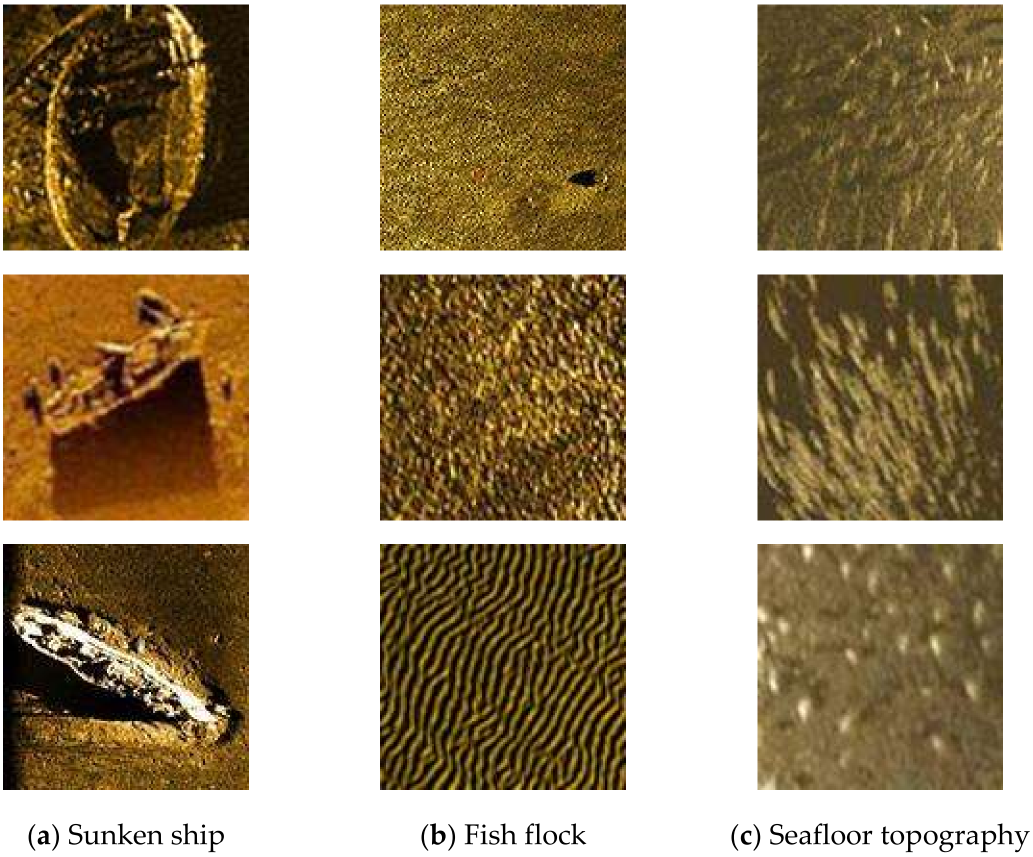

Figure 2. The demonstration of the original side scan sonar images from three kinds of datasets including Sunken Ship, Fish Flock, Seafloor Topography.

Table 1. Detailed description of the datasets.

Dataset Class	Sunken Ship	Fish Flock	Seafloor Topography
Image Quantity	250	250	250

(3) Partition the datasets

The newly expanded post-dataset was divided into the training set and the validation set in a 7:3 ratio, that is, 175 for training and 75 for testing in each category.

(4) YOLO dataset annotation

Image datasets are annotated and stored in a text file with the same name as the image. Example: 00.50.51.01.0. The first integer represents the object class, with three categories (0,1,2); the second and third values represent the scale of the object's central location to the entire image; the fourth and fifth values represent the scale of the object's width and height to the entire image. Since there is only one object in each image, the following four numbers are the same, except for the first number, which is the category of the object.

2.3. Using Online Dataset Preprocessing to Improve Target Detection Performance

2.3.1. DataLoader Platform

The DataLoader is a tool in Pytorch for processing model input data [53]. It combines datasets and samplers by providing single or multi-threading of the object [54] on the datasets. The goal is to do an initialization of the data [55] and divide the training data into groups. This function is used in the training model to break up the training data into groups [56]. During the training process, the DataLoader encapsulates the custom Dataset into batch-sized tensors according to batch size, shuffle, and so on operation process.

2.3.2. Random Flip

As a kind of data enhancement method, the random flip is adopted, which includes horizontal and vertical flips according to the actual target [57], and is usually realized by modifying the configuration file [58]. Random flips will randomly produce numbers between 0 and 1.

Random flips randomly produce numbers ranking between 0 and 1 (Algorithm 1). The main effect of horizontal flipping on labels is the x coordinate, that is, x = 1 − x0, labels [: , 1] = 1 − labels [: , 1]. The main effect of vertical flipping on labels is the y coordinate, which is y = 1 − y0, labels [: , 2] = 1 − labels [: , 2].

Algorithm 1: Random Flip Algorithm

Input: All initialized Datasets
Based platform: Dataloader
Output: Updated Random Flip Datasets
Listing 1: Random Flip Process

```
    dataset = create_dataloader
    mlc = int(np.concatenate (dataset.labels, 0)[:,0].max()) #max label class
    nb = len (train_loader) #number of batches
        assert mlc < nc Possible class labels are 0-{nc—1}
        if Rank in [–1,0]:
            val_loader =create_dataloader
        if not resume:
            labels = np.concatenate (datasets.labels, 0)
                if plots:
                    plot_labels (labels, names, save_dir)
End procedure
```

2.4. Networks Adopted in the Experiments

The convolution backbone network acts as a feature extractor and is related to accuracy and speed performance, playing an important role in object detection. Convolution Neural Networks (CNNs) are a deep learning model widely used in sonar image classification. CNNs can realize advanced solutions for object detection and classification in the field of computer vision [59]. The core feature of CNN is the capture of local features and the automatic acquisition of semantic information at different levels of abstraction [60]. Because of its relatively fast training speed, it has also achieved very good results in image based target detection and has been widely used. For example, HOG-ShipLSNet [61] for ship classification, Polarization fusion with geometric feature embedding for SAR ship classification [62], and there have been many mature experiments. CNN can pool and map the initial image to extract higher-level semantic information [63] layer by layer. The specific convolution process is below:

$$z_j^m = f\left(\sum_{i\in T_j} k_{i.j}^m * z_i^{m-1} + b_j^m\right) \tag{1}$$

where f is the activation function; $\Updownarrow$ is the number of layers; $\mathrm{k}_{i,j}$ is filtering; b_j is the bias; $*$ is the convolution operation; and $\mathrm{T_j}$ represents a collection to store input matrices.

Through years of development, CNNs have gradually achieved remarkable performance and accuracy by increasing the number of layers of the neural network; thus, they have also increased the number of samples for learning and training. In its process, a back-propagation algorithm feeds back the error into the model layer by layer [64], gradually training it by forwarding feedback, finally causing the neural network model to converge.

2.4.1. AlexNet

AlexNet is used to input images constrained to a fixed size, randomly truncated into 224 × 224 blocks [65]. Different feature maps are passed to subsequent convolutional layers [66] after a series of operations. This network uses five convolutional layers as the base layer and adds another three layers as a fully connected network structure [67], which acts as a classifier. Through layers of the network, more computing resources are calculated; the two convolutional layers will communicate with each other; the feature dimension [68] will be added at the same time; and the calculation amount will be superimposed in convolutional layers with almost no size limit on images.

2.4.2. GoogleNet

The Google team came up with a web architecture called GoogLeNet [69] for image detection and classification, based on the Hebbian model, which consists of nine Inception modules stacked on top of each other [70]. This architecture increases the depth and width of the network, improves the utilization of the internal computing resources [71] and constructs the mathematical model of the GoogLeNet network. The success of the GoogLeNet mainly depends on the Inception module, which can deal with the problems of over-fitting, large amounts of computation and low accuracy.

2.4.3. VGGNet

VGGNet network structure is simple, regular and efficient. Compared with VGGNet, AlexNet and GoogleNet replace a 5 × 5 convolutional kernel with two 3 × 3 small convolutional cores [72] and reduce the parameters. The outstanding contribution of VGGNet is to demonstrate that very small convolutions can be effectively improved by increasing network depth. The VGGNet network structure has a total of 16 layers including 13 convolution layers [73], 5 pooling layers and three full connection layers.

2.4.4. ResNet

When the VGGNet propagates backwards, there will be a loss in the transmission process, and even a gradient explosion and a gradient disappearance will occur when the transmission reaches a deeper layer. In 2015, Kaiming He proposed deep residual networks (ResNet) for target detection and classification, target localization [74] and segmentation [75]. In contrast to VGGNet, ResNet has skip connections between different layers [76], adding shallow information directly to the convolution layer and jumping some gradients directly to the shallow layer to mitigate the loss problem.

The deeper the network layer is, the more abundant the features that can be extracted [77] and the more semantic the image can reflect, but the simple stack will lead to the disappearance of network gradients [78]. ResNet introduces residual learning to deal with the problem that the deep learning network is difficult to be optimized. Its essence is to use the optimal mapping and use the stack nonlinear layer to fit another mapping.

2.4.5. YOLO V3

The classification network is the backbone network, and target detection and image match the sub-direction of the classification task. From AlexNet to ResNet, the backbone network is getting better. We add the regression head of the predicted position and the estimation of the NMS maximum a series of content and operations [79] to form the model of YOLO V3.

YOLO is an abbreviation for You Only Look Once. The YOLO versions are in the process of upgrading iterations. Fast speed is the most outstanding core characteristic of the YOLO algorithm in its small size [80]. YOLO shows great ability in generalization due to its generalized features [81] and good performance transferring to others. YOLO has many lower sampling layers [82], improving the effectiveness of the network because the target features are not exhaustive. The original YOLO uses global information and reduces detection errors when using the background as an object. YOLO V2 includes two-stage training, which adds Anchor with K-means. In YOLO V3, the network adopts FPN [83] for multi-scale detection.

Inspired by ResNet, YOLO V3 uses Darknet-53 as the backbone feature extraction [84] network, with better robustness and deeper network layers to increase the number of convolutional layers; the residual network is used in the convolution layer many times to make the network more easily converge and to enhance the adaptability of the model to the complex scene.

2.5. Multi-Scale Inputting Based on the Spatial Pyramid Pooling

In recent years, the study of multi-dimension input based on CNN has been one of the hot fields in deep learning, and two main classes of multi-dimension object detection methods have been developed [85]. To detect the multi-feature graph extracted from different layers of networks independently [86], the graph is fused [87]. To improve the precision of multi-scale detection, the multi-feature map is fused with the feature map and receptive fields of different sizes.

Based on the CNN model, the Spatial Pyramid Pooling Network method was proposed in 2014, which enables the non-fixed input image size to map arbitrary-sized features into fixed-sized feature vectors [88]; it shows the powerful performance of detecting multi-scale object fusion.

The process of the Spatial Pyramid Pooling Network is to extract features of different sizes to effectively utilize global information. If an image is an input, it will be divided into different sizes, and each block will extract a feature [89]. In max pooling, the value of the largest block in the image is calculated, and then the corresponding neuron output is obtained in turn.

The Spatial Pyramid Pooling Network has many pools of different sizes to ensure that the outputting feature vector size [90] is fixed to achieve the pooling goal. The CNN and its characteristics greatly limit the input of image size. If the images are processed, it will lead to image quality shrinkage and information loss [91]. To solve this problem, the Spatial Pyramid Pooling Network method is added to the CNN.

2.6. Networks with Spatial Pyramid Pooling Adopted Experiments in the Models

Based on the ability of the spatial pyramid pool to break through the restriction of input size, we add it to the traditional classical algorithms AlexNet, GoogleNet, VGGNet and ResNet network models, respectively. The effect, however, is restricted by the structure of the above network models; the detection accuracy has some limitations [92]. Therefore, we add a spatial pyramid module in YOLO V3 to realize the function of stitching multi-scale region features. Multi-scale aggregation is used to extract features and to converge the network in the spatial pyramid pool.

Therefore, we compare YOLO V3 network with other networks vertically, using the same networks with and without the pyramid pool model, using the qualitative evaluation by evaluation indicators, to highlight the high performance of the YOLO V3 Model based on spatial pyramid pooling.

2.6.1. AlexNet + SPP

Spatial pyramid pooling is a method of combining input images through three pool modules and convolving different input images to extract the features of each module, solving the problem of training multiple images after a selective search. Adding Spatial Pyramid Pooling to AlexNet (Figure 3) eliminates the crop/warp image normalization process and solves the information loss and storage problems caused by image distortion. We used Spatial Pyramid Pooling to fix the convolution layer in the process of fine-tuning the AlexNet network structure (Algorithm 2), replacing the last Pooling layer before the full connection layer, thus the CNN network can train the role of different sizes of images. Finally, the original image can be directly sent to the network for training, accelerating the network to achieve the end-to-end process of object detection.

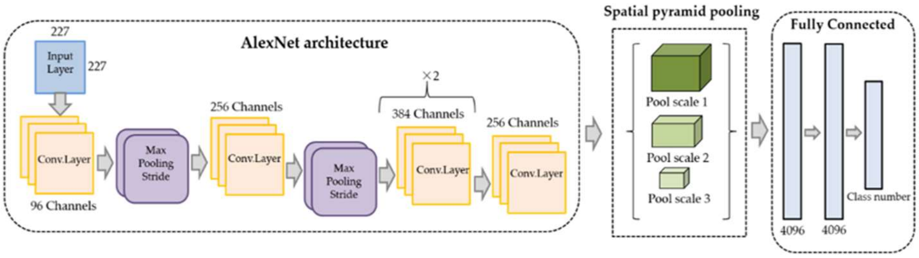

Figure 3. The model of AlexNet + SPP.

Algorithm 2: AlexNet + SPP Algorithm

Input: SPP algorithm input
Based network: AlexNet algorithm
Output: AlexNet + SPP algorithm output
Listing 2: SPP Process

```
→ class SPPLayer (nn.Module):
→ def __init__(self, num_levels, pool_type = 'max_pool'):
→ → super (SPPLayer, self).__init__()
→ → self.num_levels = num_levels
→ → → self.pool_type = pool_type
```

End procedure

2.6.2. GoogleNet + SPP

The success of the GoogleNet is largely due to the Inception module, which can be seen as a stack of Inception modules throughout the GoogLeNet mainframe. The spatial pyramid algorithm structure in GoogleNet is added after the Inception module (Figure 4). The Inception Module mainly adds a convolution layer to CNN (Algorithm 3), using ReLU as the activation function, and its main function is to reduce the dimension of network features and to reduce a lot of computation without sacrificing the performance of the network model. The Spatial pyramid pooling structure in GoogleNet is added after the Inception template.

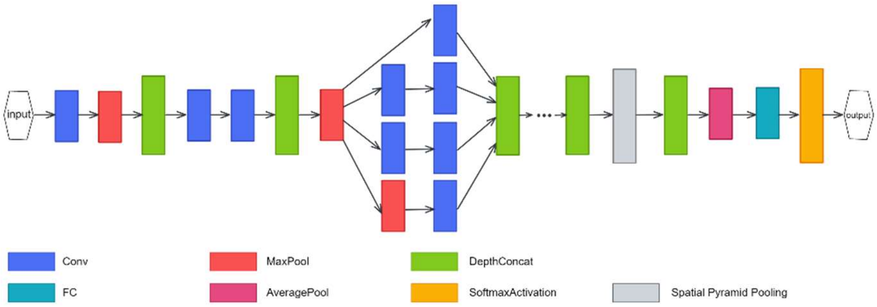

Figure 4. The model of GoogleNet + SPP.

Algorithm 3: GoogleNet + SPP Algorithm

```
Input: SPP algorithm input
Based network: GoogleNet algorithm
Output: GoogleNet + SPP algorithm output
Listing 3: SPP Process
→ class SPPLayer (nn.Module):
→ def __init__(self, num_levels, pool_type = 'max_pool'):
→ → super (SPPLayer, self).__init__()
→ → self.num_levels = num_levels
→ → → self.pool_type = pool_type
End procedure
```

2.6.3. VGGNet + SPP

VGGNet is compact, uses maximum pooling connections and uses ReLU as the activation function between the hidden layers. Therefore, the VGGNet is often used to extract image features, increase the deep of the convolutional neural network and use small convolutional cores, which play a great role in the final detection of the network (Figure 5). We add the spatial pyramid pooling to the last layer (Algorithm 4) of the VGGNet's convolution layer.

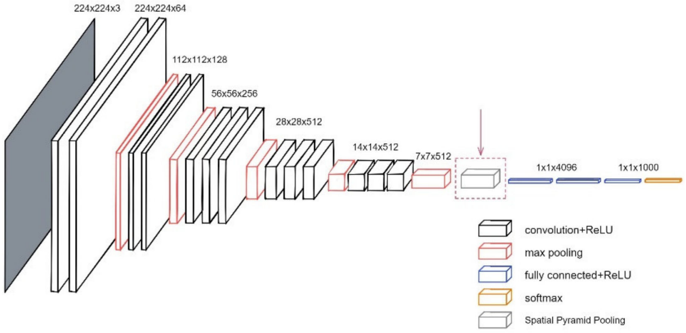

Figure 5. The model of VGGNet + SPP.

Algorithm 4: VGGNet + SPP Algorithm

```
Input: SPP algorithm input
Based network: VGGNet algorithm
Output: VGGNet + SPP algorithm output
Listing 4: SPP Process
→ class SPPLayer (nn.Module):
→ def __init__(self, num_levels, pool_type = 'max_pool'):
→ → super (SPPLayer, self).__init__()
→ → self.num_levels = num_levels
→ → → self.pool_type = pool_type
End procedure
```

2.6.4. ResNet + SPP

The residual module is very important in the deep neural network, which is often used in the process of modeling. The success of ResNet is that it solves the problem of network degradation with the help of a residual module, which improves the depth of the network, gets the feature of stronger expression ability and has higher accuracy (Figure 6).

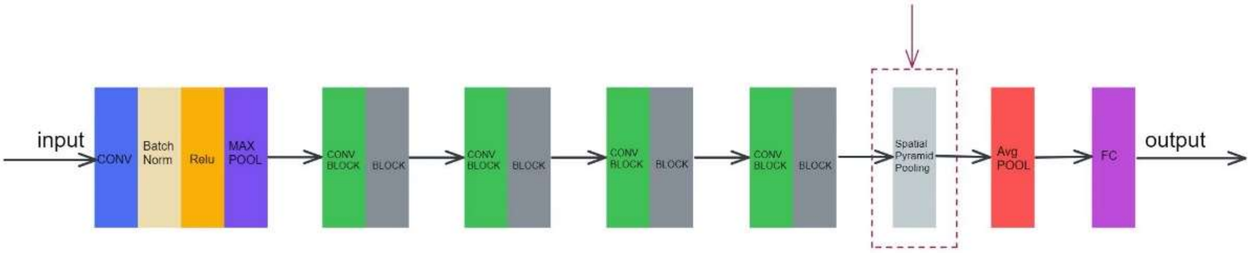

Figure 6. The model of ResNet + SPP.

The spatial pyramid pooling structure in ResNet is added between ResNet and the full connection layer (Algorithm 5), and the deep residual network without activating the normalized layer is designed. The main goal is to achieve different input sizes while ensuring the fixed-size network, which is now well-optimized and fully utilized. It is helpful to increase the depth of the network while avoiding gradient disappearance.

Algorithm 5: ResNet + SPP Algorithm

```
Input: SPP algorithm input
Based network: ResNet algorithm
Output: ResNet + SPP algorithm output
Listing 5: SPP Process
→ class SPPLayer (nn.Module):
→ def __init__(self, num_levels, pool_type = ‘max_pool’):
→ → super (SPPLayer, self).__init__()
→ → self.num_levels = num_levels
→ → → self.pool_type = pool_type
return solution
```

2.6.5. YOLO V3 + SPP

YOLO V3 outputs three feature maps of different sizes, from top to bottom, corresponding to the features of the deep layer, middle layer and shallow layer, respectively. In contrast to the small size and large receptive field of the deep feature map, the shallow feature map is more convenient for detecting small-scale objects, which is similar to the FPN structure.

With the addition of a spatial pyramid pooling structure, YOLO V3 can increase performance with little time consumption (Figure 7). The fusion of local features and global features is realized by the SPP module, in which the SPP part is added in the first Set block (Algorithm 6). By maximizing the pool of different receptive fields and finally stitching the dimensions, the fusion information of different scales can be obtained, thus improving the performance of the model.

Algorithm 6: YOLO V3 + SPP Algorithm

```
Input: SPP algorithm input
Based network: YOLO V3 algorithm
Output: YOLO V3 Net + SPP algorithm output
Listing 6: SPP Process
→ class SPP (nn.Module):
→ def __init__(self, c1, c2, k = (5, 9, 13)):
→ → super ()._init_()
→ → self.cv1 = Conv (c1, c_, 1, 1)
→ → self.cv2 = Conv (c_ *(len(k) + 1), c2, 1, 1)
→ → → self.m = nn.ModuleList ([nn.MaxPool])
End procedure
```

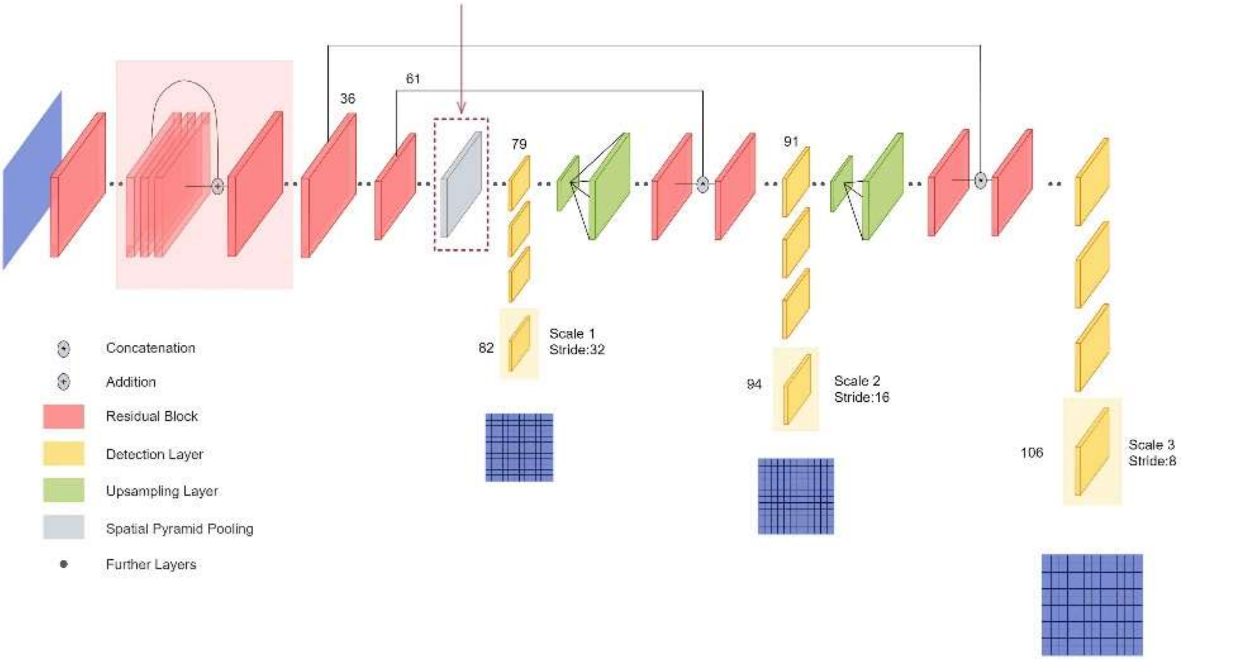

Figure 7. The model of YOLO V3 + SPP.

2.6.6. Online Dataset Preprocessing with YOLO V3 + SPP

We adopt online dataset preprocessing (ODP) to improve detection performance based on the DataLoader platform in Section 2.3. The input part of the improved neural network with spatial pyramid pooling is based on the YOLO V3 method (YOLO V3 + SPP model with Online Dataset Preprocessing) to form an experimental group (Figure 8). It is used to compare with the YOLO V3 model with spatial pyramid pooling in Section 2.6.5 to test the effect of random data flipping before the input part (Algorithm 7) to the impact on the evaluation of identification.

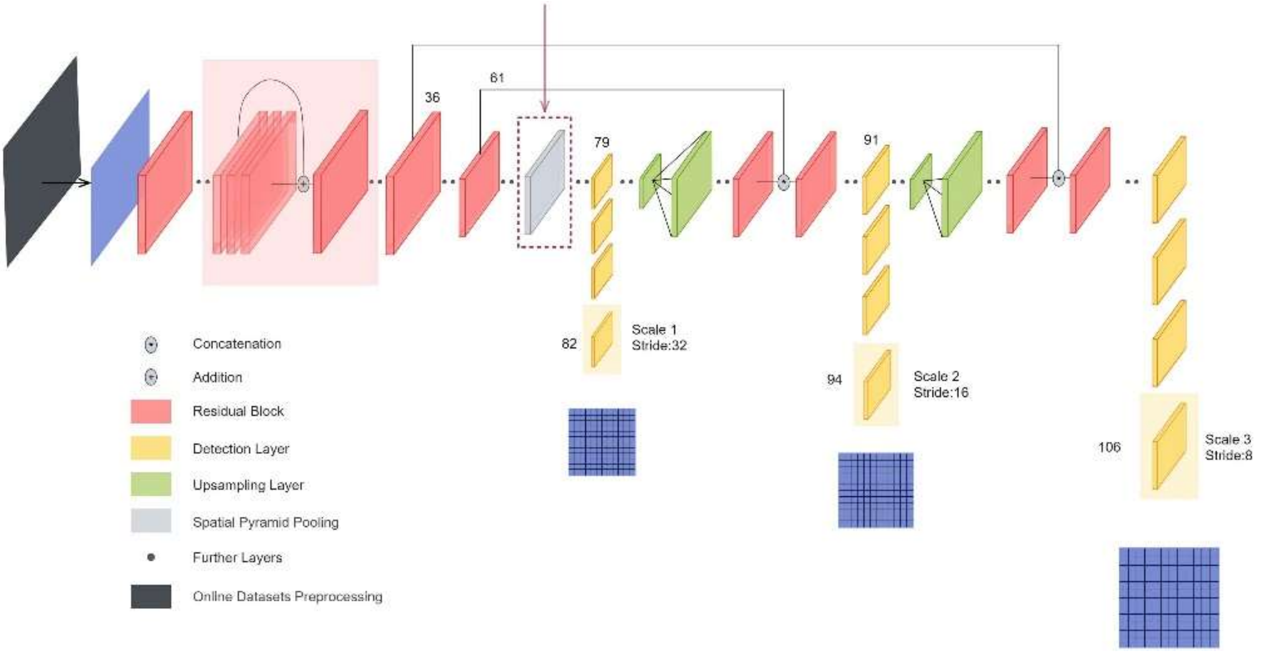

Figure 8. The model of Online Datasets preprocessing with YOLO V3 + SPP.

Algorithm 7: Online Datasets Preprocessing YOLO V3 + SPP Algorithm

```
Input: SPP algorithm input
Based network: YOLO V3 algorithm
Output: YOLO V3 Net + SPP algorithm output
Listing 7: SPP Process
→ class SPP (nn.Module):
→ def __init__(self, c1, c2, k = (5, 9, 13)):
→ → super ()._init_()
→ → self.cv1 = Conv (c1, c_, 1, 1)
→ → self.cv2 = Conv (c_ *(len(k) + 1), c2, 1, 1)
→ → → self.m = nn.ModuleList ([nn.MaxPool])
End procedure
```

2.6.7. Performance Comparisons

Through the above experiments, we made four groups of comparison, testing the influence and effect of adding spatial pyramid pooling on four kinds of traditional classical deep learning network methods and then, through a group of experimental comparison, testing the effect of spatial pyramid pooling on YOLO V3 deep learning network method. A set of experiments was carried out to compare the effects of spatial pyramid pooling on YOLO V3 deep learning network method, to test the effect of on-line preprocessing adopting random flip on YOLO V3 with high accuracy and high speed, which has been added to spatial pyramid pooling.

3. Experiments

3.1. Datasets

3.1.1. Datasets Description

To verify the validity of sonar image based target detection by using pyramid pooling improved convolution neural network, the original raw sample of the dataset used in this paper comes from 150 sonar images with 50 images in each category collected by individuals divided into three categories, including sunken ships, fish flocks and seafloor topography (Figure 2), to form a dataset for the research of the underwater target detection algorithm. The description of the Dataset is shown in Table 1.

The amount of data is far from sufficient; it is difficult to train the network, and the network will be overfitting, affecting the accuracy. To efficiently train the network, the original dataset is processed as follows.

3.1.2. Datasets Preprocessing

Data expansion makes the training data as close as possible to the test data, thus improving the prediction accuracy. At the same time, the network is forced to learn more robust features so that the model has stronger generalization ability. After data expansion, the sample of the training set can be increased, which can effectively alleviate the over-fitting of the model and also can bring stronger generalization ability to the model. According to the preprocessing method of Section 2.2, we extend the original dataset and introduce the correlation package, specify the directory in which the original data resides and geometrically rotate the original data (to the left or the right) with a probability of 0.7 to within 25 degrees. The original label of the image should be kept unchanged when the data expansion operation is carried out, and the data expansion should be carried out without changing the label; 250 images were added to each category, for a total of 750 images. The newly expanded post-dataset is divided into a training set and verification set in a 7:3 ratio, that is, 175 training sets and 75 test sets.

3.2. Experimental Preparation

3.2.1. Experimental Condition

In this experiment, the training and testing environments were conducted on a Ubuntu 18.04 PC with an Intel® Xeon® CPU, 31.2 GB memory and an NVIDIA-SMI 470.82.00 GPU with 250.9 GB memory. The procedure was operated on the PyCharm 11.0.13 platform using python language. The deep learning framework written by PyCharm for the underlying language has high flexibility and strong code execution ability. Based on this, the following experiments are designed.

3.2.2. Experiment Model Training

The deep learning framework written by torch for the underlying language has high flexibility and strong code execution ability. Based on this, the following four sets of experiments are designed (Figure 9); the learning rate is set to 0.0001; and a total of 60 steps (epoch) are trained. Firstly, AlexNet, GoogleNet, VGGNet and ResNet neural networks were used as initialization model group to preprocess three kinds of datasets: sunken ships, fish flocks and seafloor topography.

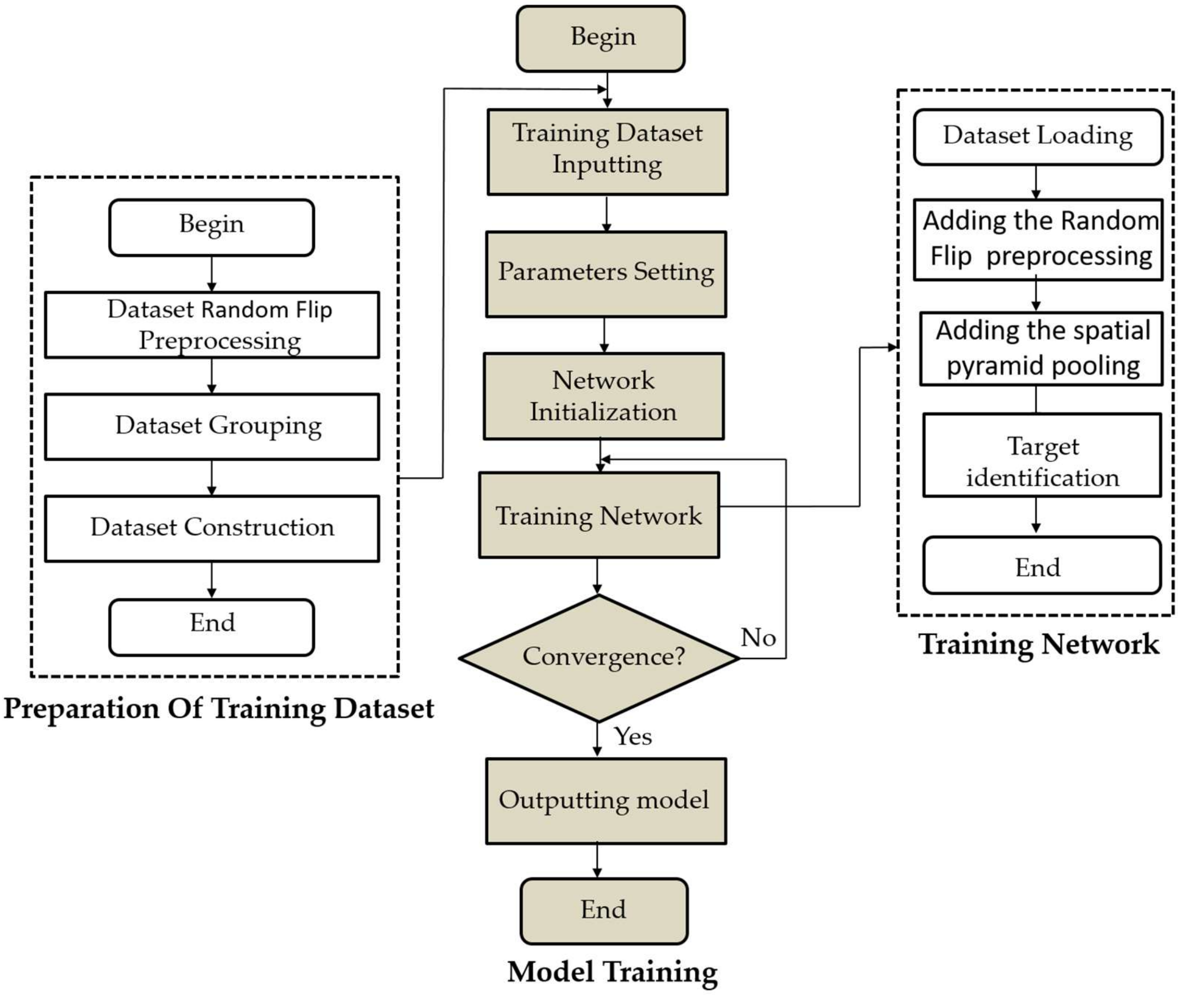

Figure 9. Flow chart of the target detection processes.

Afterwards, then the spatial pyramid pooling algorithm structure is added to the above network deep learning training in contrast to the group without the addition of the spatial pyramid pooling, for proving the effectiveness and reliability of the spatial pyramid pooling algorithm. The second step is to define some variables required for training and placeholders for inputting images and to import the prepared dataset into networks. The third step is to collect and summarize the data obtained from the experiment, reading all the pictures in all subfolders under the path, and storing them in a list. Finally, when loading the data, the data is reordered; the training set and the verification set are loaded in turn; and the data results are displayed through the samples.

In the experiment of the YOLO model, the datasets are enhanced, and the training set is labeled and input into the network. Then the model configuration file is constructed according to the network structure; the configuration file is analyzed through the script; and the YOLO V3 network is constructed by adding the pyramid pool layer by layer. YOLO V3 is used to train three kinds of experimental data, and then, based on YOLO V3, a spatial pyramid pool algorithm is added to construct the network; the spatial pyramid pool network layer is divided into four branches and sampled by the different maximum pool, and three parallel pool layers are added to the network layer 75 to 77, the spatial pyramid structure can achieve the fusion of different scale features, especially for small target detection, which has a good performance and enhances the ability of feature extraction. We input the YOLO V3 model with randomly flipped dataset preprocessing online and compared them to verify the dataset preprocessing effect.

3.3. Evaluation Indicators

In the process of data training, there will be overfitting. To avoid it as much as possible without adding training samples to make the dataset more complex, performing various transformations on the original image [81] can greatly improve the numerical amount of learning and training in the experiment. According to the progress of the experiment, the model parameters were analyzed and adjusted to improve the detection accuracy of the model [82].

To compare, analyze and evaluate accurately, the different network models used in the experiment, their respective detection accuracy, selecting accuracy, precision, recall, mAP and Time were used as indicators. There are four situations that describe the classification results. These situations are:

The true positive (TP): positive samples predicted by the model as positive classes;

The true negative (TN): negative samples predicted as negative classes by the model;

The false positive (FP): positive samples predicted as negative classes by the model;

The false negative (FN): negative samples predicted by the model as positive classes.

The accuracy rate is the percentage of the number of samples that predicts and judges correct data in the model to the total number of data samples. Precision and Recall are mutually constrained. Their calculations follow:

$$OA = \frac{(TP + TN)}{(TP + TN + FP + FN)} \tag{2}$$

$$precision = \frac{TP}{TP + FP} \tag{3}$$

$$recall = \frac{TP}{TP + FN} \tag{4}$$

$$mAP = \frac{\sum_n precision}{n} \tag{5}$$

3.4. Experimental Results

This section presents the analysis of the data results, to verify the improvement effect of spatial pyramid pooling and to verify the effectiveness of adopting online dataset preprocessing (ODP) random flip in the field of sonar image based target detection. To improve the reliability and stability of the experiment, the amount of data in the training set is much larger than that in the test set, and the size of the training set is more than twice that of the test set. After 60 iterations and repeated experiments, a comparison of experimental results is shown in Table 2 for the AlexNet Group, the AlexNet + SPP Group, the GoogleNet Group, the GoogleNet +SPP Group, the ResNet Group and ResNet + SPP Group. This table reports the performance results of eight groups from the number of iterations from three scenes. In order to display the experimental results efficiently, we adopted the median accuracy of the verification set of the multi-round model after training for a more straightforward presentation of the results.

Table 2. The comparison of evaluation indicators on classic network performance.

Networks	Testing Time Used (Mins)	Accuracy	Precision	Recall	F1-Score
AlexNet	5.10	0.76	0.79	0.74	0.71
AlexNet + SPP	4.27	0.78	0.80	0.73	0.74
GoogleNet	5.81	0.71	0.71	0.82	0.76
GoogleNet + SPP	3.64	0.75	0.75	0.73	0.78
VGGNet	14.34	0.84	0.81	0.97	0.88
VGGNet + SPP	8.25	0.86	0.89	0.86	0.89
ResNet	4.39	0.87	0.89	0.87	0.87
ResNet + SPP	4.18	0.89	0.91	0.86	0.88

In the first experiment comparing (+SPP), our proposed adding SPP method yielded higher detection accuracy and faster detection efficient performance (Table 2). For example, In the AlexNet groups, the AlexNet + SPP method shows time used 0.83 min faster than the AlexNet method; the AlexNet + SPP method shows 2% accuracy higher than the AlexNet method; the AlexNet + SPP method shows 1% precision higher than the AlexNet method; the AlexNet + SPP method shows 1% recall lower than the AlexNet method; the AlexNet + SPP method shows 3% F1-score higher than the AlexNet method, indicating that adding SPP has a positive influence on improving detection accuracy and detection efficient performance.

In the second experiment comparison (Table 3), our proposed ODP+YOLO V3+SPP method yielded the highest detection accuracy and the fastest processing efficiency performance. From a comparison with Table 3, the proposed ODP+YOLO V3+SPP outperformed the other tested networks in the YOLO group comparisons. In the mAP evaluation, the Online Datasets Preprocessing YOLO V3 + SPP group shows 4% higher than the YOLO V3 + SPP group; the Online Datasets Preprocessing YOLO V3 + SPP group shows 6% higher than the YOLO V3 group. In Precision evaluation respect, the Online Datasets Preprocessing YOLO V3 + SPP group shows 2% higher than the YOLO V3 + SPP group; the Online Datasets Preprocessing YOLO V3 + SPP group shows 13% higher than the YOLO V3 group. In Recall evaluation respect, the Online Datasets Preprocessing YOLO V3 + SPP group shows 5% lower than the YOLO V3 + SPP group; the Online Datasets Preprocessing YOLO V3 + SPP group shows 3% higher than the YOLO V3 group. In Time used evaluation respect, the Online Datasets Preprocessing YOLO V3 + SPP group shows 0.02 h faster than the YOLO V3 + SPP group; the YOLO V3 + SPP group shows 0.08 h faster than the YOLO V3 group; the Online Datasets Preprocessing YOLO V3 + SPP group shows 0.1 h faster than the YOLO V3 group, that is to say, the detection efficiency has increased by 9.34%. The above experimental results show that compared with the original YOLO V3 model, the proposed ODP+YOLO V3+SPP underwater target detection algorithm model has improved detection performance through the mAP qualitative evaluation index has increased by 6%, the Precision qualitative evaluation index has increased by 13%, and the detection efficiency has increased by 9.34%. The results also indicate our proposed ODP+YOLO V3+SPP method has a positive improving effect on detection accuracy and detection efficiency.

Table 3. Comparison of evaluation indicators on YOLO Network performance.

Networks	Testing Time Used (Hours)	mAP	Precision	Recall
YOLO V3	1.17	0.89	0.78	0.8
YOLO V3 + SPP	1.09	0.91	0.89	0.88
The proposed ODP + YOLO V3 + SPP	1.07	0.95	0.91	0.83

In Figure 10, we can see that the AlexNet network added to the spatial pyramid pooling has higher detection accuracy and better detection performance than without the spatial pyramid pooling. It improves the spatial pyramid pooling showing a positive effect on the image detection of side scan sonar imagery.

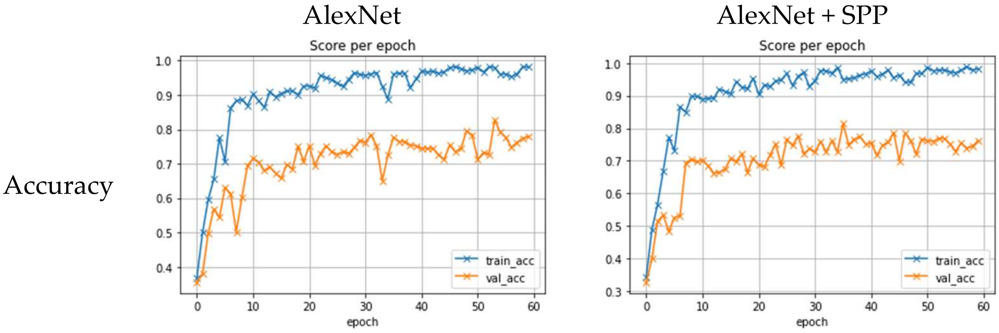

Figure 10. Performance comparison analysis between AlexNet and AlexNet + SPP.

In the GoogleNet groups (Table 2), the GoogleNet + SPP method shows time used 2.17 mins faster than the GoogleNet method; the GoogleNet + SPP method shows 4% accuracy higher than the GoogleNet method; the GoogleNet + SPP method shows 4% precision higher than the GoogleNet method; the GoogleNet + SPP method shows 9% recall lower than the GoogleNet method; the GoogleNet + SPP method shows 2% F1-score higher than the AlexNet method.

In Figure 11, we can see that the GoogleNet network added to the spatial pyramid pooling has higher detection accuracy and better detection performance than without adding spatial pyramid pooling. It improves the spatial pyramid pooling showing a positive effect on the image detection of side scan sonar imagery.

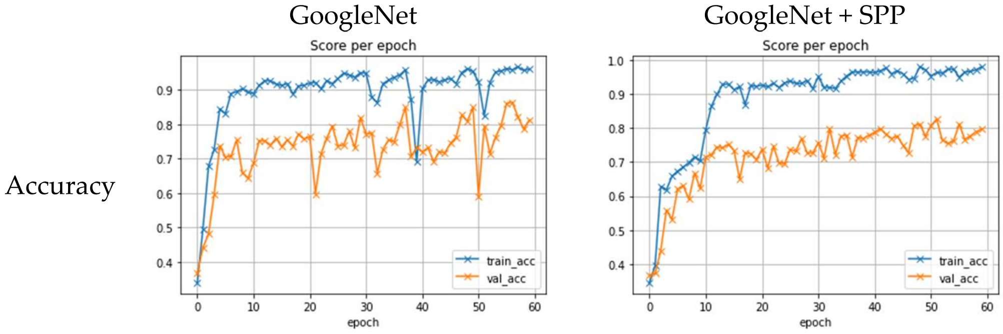

Figure 11. Performance comparison analysis between GoogleNet and GoogleNet + SPP.

In the VGGNet groups (Table 2), the VGGNet + SPP method shows time used 6.09 mins faster than the VGGNet method; the VGGNet + SPP method shows 2% accuracy higher than the VGGNet method; the VGGNet + SPP method shows 8% precision higher than the VGGNet method; the VGGNet + SPP method shows 11% recall lower than the VGGNet method; the VGGNet + SPP method shows 1% F1-score higher than the VGGNet method.

In Figure 12, we can see that the VGGNet network added to the spatial pyramid pooling has higher detection accuracy and better detection performance than without the spatial pyramid pooling. It improves the spatial pyramid pooling, showing a positive effect on the image detection of side scan sonar imagery.

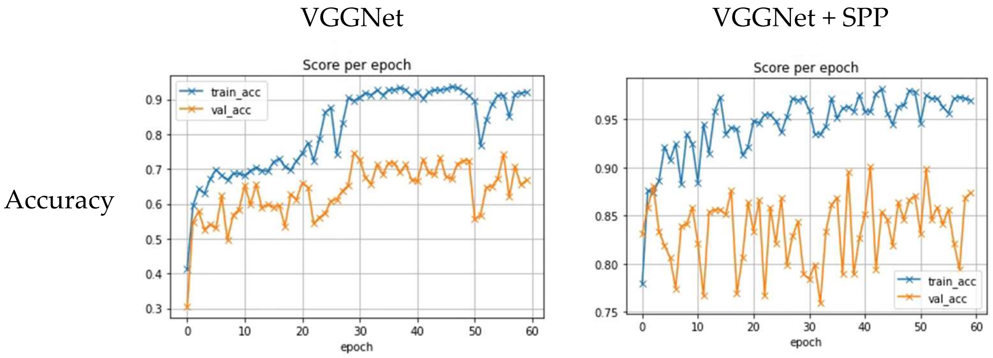

Figure 12. Performance comparison analysis between VGGNet and VGGNet + SPP.

In the ResNet groups (Table 2), the ResNet + SPP method shows time used 0.21 mins faster than the ResNet method; the ResNet + SPP method shows 2% accuracy higher than the ResNet method; the ResNet + SPP method shows 2% precision higher than the ResNet method; the ResNet + SPP method shows 1% recall lower than the ResNet method; the ResNet + SPP method shows 1% F1-score higher than the VGGNet method.

In Figure 13, we can see that the ResNet network added to the spatial pyramid pooling has higher detection accuracy and better detection performance than without the spatial pyramid pooling. It improves the spatial pyramid pooling, showing a positive effect on the image detection of side scan sonar imagery.

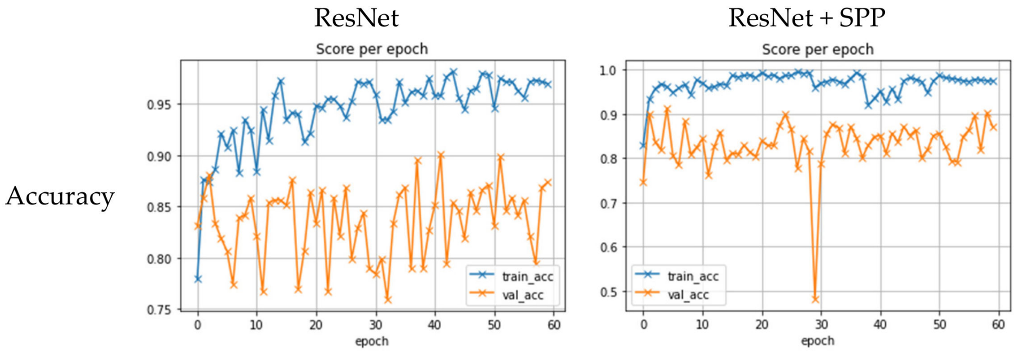

Figure 13. Performance comparison analysis between ResNet and ResNet + SPP.

Above all, the experimental results show that: for AlexNet, GoogleNet, VGGNet, ResNet algorithms, after spatial pyramid pooling is added to their backbone networks in this paper, the average detection accuracy of the three sets of data is increased by 2%, 4%, 2% and 2%, respectively, comparing with the original algorithm. This proves that adding spatial pyramid pooling can improve the target detection accuracy of these commonly used algorithms.

In Figure 14, we can see that the YOLO V3 network added to the spatial pyramid pooling (YOLO V3 + SPP) has higher detection accuracy and better detection performance than without the spatial pyramid pooling. Besides, our proposed improved neural networks with spatial pyramid pooling and online datasets preprocessing (ODP + YOLO V3 + SPP) show the best detection performance in the figure. It improves the spatial pyramid pooling and online datasets preprocessing, showing a positive influence on the image detection of side scan sonar imagery. The detection accuracy has been improved through our proposed method.

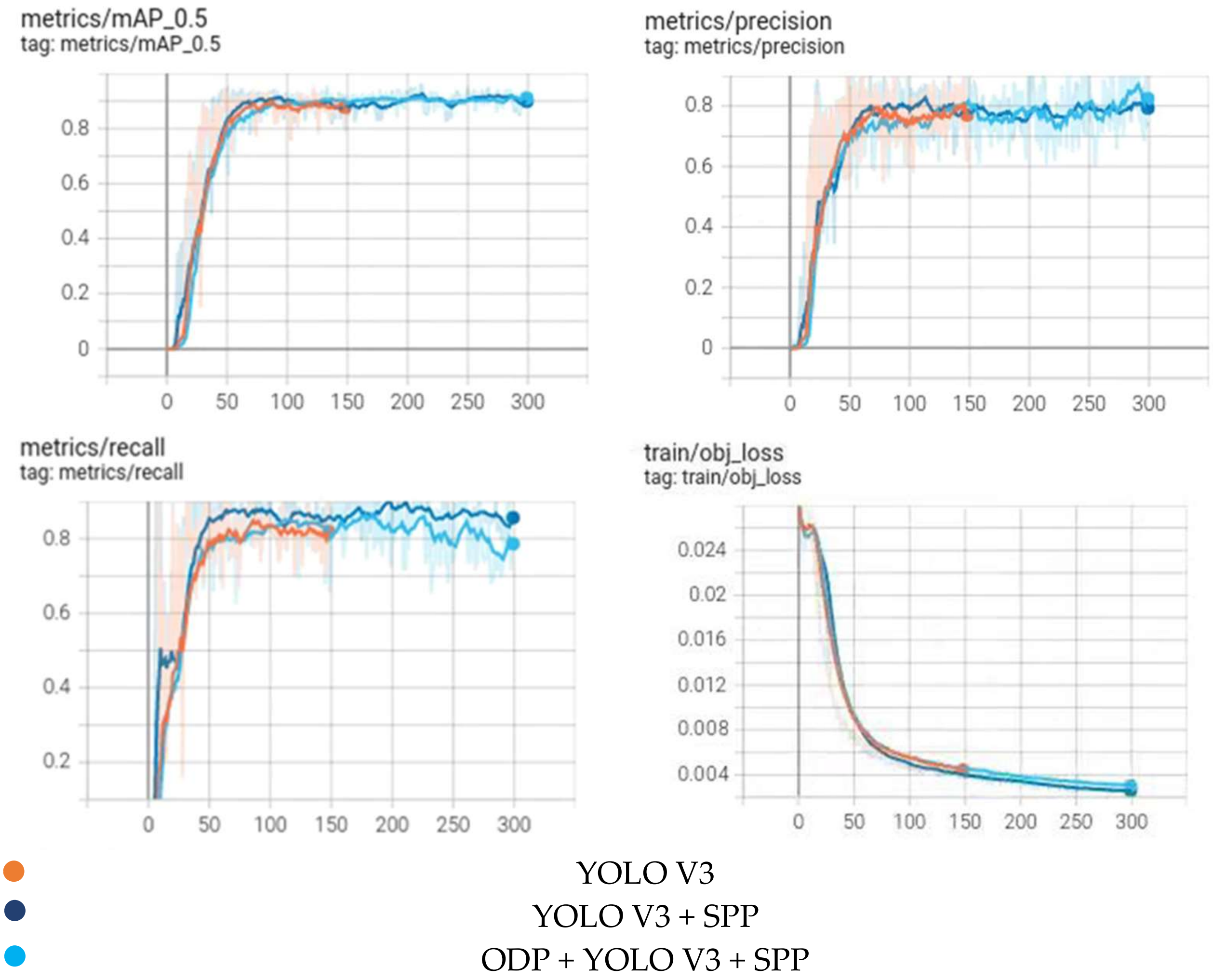

Figure 14. The performance of YOLO groups comparison analysis.

In Table 4 the evaluation indicators for three kinds of side scan sonar datasets based on the proposed YOLO V3 + SPP with online datasets preprocessing method shows that the mAP is 98% for the sunken ship, 91% for the fish flock, 96% for seafloor topography; the Precision is 89% for the sunken ship, 98% for the fish flock, 82% for seafloor topography; the Recall is 88% for the sunken ship, 77% for the fish flock, 84% for seafloor topography. In Figure 15, it shows some example target detection results of the proposed YOLO V3 + SPP with online datasets preprocessing (ODP) method for the three sets of side scan sonar datasets including Sunken Ship, Fish Flock and Seafloor Topography.

Table 4. The evaluation indicators for three kinds of side scan sonar datasets based on the YOLO V3 + SPP with online datasets preprocessing method.

Target Classes	mAP	Precision	Recall
All	0.95	0.9	0.83
Sunken ship	0.98	0.89	0.88
Fish flock	0.91	0.98	0.77
Seafloor topography	0.96	0.82	0.84

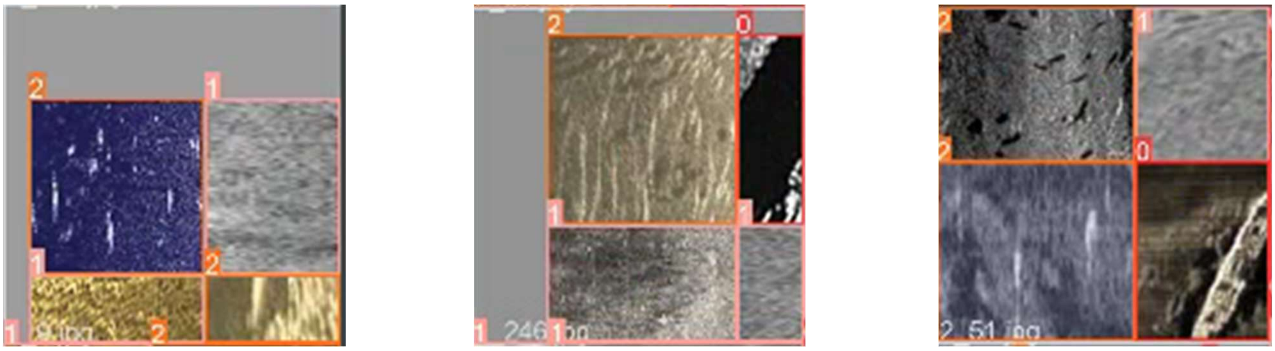

Figure 15. Some example target detection results of the proposed ODP+YOLO V3 + SPP method for the three sets of side scan sonar datasets, including Sunken Ship, Fish Flock, Seafloor Topography: label 0 refers to Sunken Ship; label 1 refers to Fish Flock; and label 2 refers to Seafloor Topography.

To test the efficiency of the ODP + YOLO V3 + SPP method proposed in this paper with the above four traditional classical neural networks and four improved algorithms (a total of eight algorithms) after adding SPP respectively. Considering the length of the paper and other issues, this paper will compare the efficiency of the ODP + YOLO V3 + SPP method proposed in this paper with the fastest method among the 8 traditional classical neural networks (Google + SPP). The overall detection time of all images and the average detection time of each image in the validation set were used to evaluate the efficiency of the method. The overall detection time for all images is the median of the overall detection time for all images in the validation set. The average detection time of each image is obtained by dividing the total detection time of all images in the validation set by the total number of images to obtain the median of the results.

In Table 5, we compare the total detection time of all images and the average detection time of each image in the traditional classical algorithm, the Google + SPP method with the shortest time and the highest efficiency, the Proposed ODP + YOLO V3 + SPP is used for comparison with The Proposed ODP + YOLO V3 + SPP. As mentioned in the 3.1 Datasets section, there are three types of validation sets for the entire dataset, namely sunken ship, fish flock and seafloor topography. The number of images in each type of validation set is 75, and the total number of images in the three types of validation set is 225. We then carry on the experimental result analysis according to this.

Table 5. Comparing the average detection time between the original classic Google + SPP and the proposed ODP + YOLO V3 + SPP in the seconds for each image.

Methods	The Overall Detection Time for All Images	The Average Detection Time for Each Image
Google + SPP	218.25	0.97
The Proposed ODP + YOLO V3 + SPP	128.25	0.57

According to Table 5, on the NVIDIA-SMI 470.82.00, the average detection time for each image with the Google + SPP method was 0.97 s and the overall detection time for all images was 218.25 s. The average detection time of The Proposed ODP + YOLO V3 + SPP for each image was 0.57 s and the overall detection time for all images was 128.25 s. Our method proposed ODP + YOLO V3 + SPP method is faster than the Google + SPP method. The efficiency for all datasets by The Proposed ODP + YOLO V3 + SPP method is improved by 41%. Therefore, The Proposed ODP + YOLO V3 + SPP method in this paper shows the best detection efficiency.

From the above experimental results, our method an improved neural network with the spatial pyramid pooling and online dataset preprocessing based on the YOLO V3 shows good performance in improving detection accuracy and improving detection speed. Compared with Google+SPP group, which is the best detection efficient model in the several classic networks (Table 2), the proposed ODP+YOLO V3+SPP model's detection efficiency has increased by 41% (Table 5). Compared with the original YOLO V3 model, the proposed ODP+YOLO V3+SPP underwater target detection algorithm model has improved detection performance through the mAP qualitative evaluation index has increased by 6%, the Precision qualitative evaluation index has increased by 13%, and the detection efficiency has increased by 9.34% (Table 3). Our approach shows the best detection performance, and it is conducive to improvement in the detection of typical side scan sonar datasets.

3.5. Discussion

The method proposed in this paper combines the spatial pyramid pooling algorithm and the online datasets preprocessing based on the dataLoader platform to improve the detection performance (e.g., detection accuracy and efficiency) of the side scan sonar images. After conducting a comparison of the proposed approach in the experiments, the results show our approach yielded higher detection performance without adopting the Spatial Pyramid Pooling Method, without adopting the Online Dataset Preprocessing Method.

The results prove in the process of target detection, the detection evaluation indicators of the same network with the spatial pyramid pooling compared with that of the original network show better performance. It has a certain stability, which verifies that the pools of different sizes of the spatial pyramid pooling algorithm can make the fixed size of the output feature vector helps to improve the detection accuracy of sonar images. For example, compared with the AlexNet Group, the AlexNet + SPP Group has less time used and higher accuracy, etc. Spatial pyramid pooling better preserves more feature information of images, which is beneficial to improve detection accuracy and efficiency.

This result shows the performances of YOLO group comparisons on detection. We can find that YOLO V3 with adding spatial pyramid pooling has better performance than YOLO V3 without adding spatial pyramid pooling. The detection accuracy of the YOLO V3 network can be improved by using spatial pyramid pooling. Besides, online datasets preprocessing YOLO V3 with adding spatial pyramid pooling has the best performance in the three groups, which demonstrates the good performance of YOLO models in the design of experiments, especially which includes spatial pyramid pooling and online datasets preprocessing, showing better performance than other network methods in the paper.

The method we proposed can better realize the balance of improving accuracy and efficiency than other comparison methods in the paper, which is an improved neural network with spatial pyramid pooling and online dataset preprocessing based on the YOLO V3 method for typical underwater target detection.

4. Conclusions

In this paper, we propose an improved neural network with spatial pyramid pooling and online dataset preprocessing based on the YOLO V3 method for underwater target detection, realizing the balance of improving accuracy and efficiency. The algorithm proposed in this paper delivers high detection accuracy, but more importantly, it can achieve an optimal balance between improved detection accuracy and improved detection speed when tackling underwater targets. Compared with the original YOLO V3 model, the proposed ODP+YOLO V3+SPP underwater target detection algorithm model has improved detection performance through the mAP qualitative evaluation index has increased by 6%, the Precision qualitative evaluation index has increased by 13%, and the detection efficiency has increased by 9.34%.

The spatial pyramid pooling algorithm was added to improve sonar image detection to break the limitations of the input image size. The limited size of input data is a common problem; therefore, the overall quality of input data may be lacking. The traditional methods to solve this problem often result in missing information, and furthermore, simple data processing methods often cause information uncertainty. In this research, we adopted methods that improved the integrity of the information and thus increased the accuracy of the classification model. YOLO V3 with spatial pyramid pooling. As compared with other tested models, this approach uses a more streamlined structure that enhances feature extraction and reduces vanishing gradients by incorporating a spatial pyramid structure. Results show that the application of spatial pyramid pooling can improve the accuracy of side scan sonar image based target detection. Using online dataset preprocessing with random flips improved detection performance. Lots of experiments are needed to find the optimal convolution kernel parameters to reduce the dependence on neural network parameters and data volume.

For the typical target detection based on side scan sonar images, the detection accuracy of common neural network algorithms was improved over the original algorithms after adding spatial pyramid pooling. Therefore, (1) the detection accuracy of typical target detection based on side scan sonar image can be notably improved by using spatial pyramid pooling as proposed in this paper; (2) in this paper, online dataset preprocessing flip helps improve underwater target detection accuracy based on side scan sonar images.

In the future, our research will further focus on the deep learning based feature extraction methods, and more effective methods will be integrated. The proposed model can be used in the future to improve the accuracy of side scan image classification tasks. At the same time, our framework will be enhanced with more advanced deep learning architectures to make further efforts to improve model efficiency.

Author Contributions: J.L., X.X., L.C., J.S. and D.L. proposed the idea and wrote the manuscript; J.L., X.X., J.S., X.L., X.S. and X.W. designed and performed experiments; X.X., L.C. and D.L. helped to revise the manuscript. All authors have read and agreed to the published version of the manuscript.

Funding: The work was supported by the National Natural Science Foundation of China (Grant No. 42101449), the Natural Science Foundation of Hubei Province, China (Grant Nos. 2022CFB773, 2020CFA001), the Key Research and Development Project of Jinzhong City, China (Grant No. Y211006), the Science and Technology Program of Southwest China Research Institute of Electronic Equipment (Grant No. JS20200500114), the Chutian Scholar Program of Hubei Province, the Yellow Crane Talent Scheme and LIESMARS Special Research Funding. We sincerely thank the editors and anonymous reviewers for their outstanding comments.

Data Availability Statement: The data are not publicly available due to the restrictions of containing information that could compromise the privacy of research participants. So participants of this study did not agree for their data to be shared publicly, so supporting data is not available.

Acknowledgments: The authors wish to thank the editor and reviewers for their suggestions, and thanks to Xiao Wang at the Jiangsu Ocean University for providing us with several sets of experimental data. We are thankful to Stephen C. McClure for providing us English editing of the manuscript freely.

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

References

1. Zhang, T.; Zhang, X.; Liu, C.; Shi, J.; Wei, S.; Ahmad, I.; Zhou, X.; Pan, D.; Li, J.; Su, H. Balance learning for ship detection from synthetic aperture radar remote sensing imagery. *ISPRS J. Photogramm. Remote Sens.* **2021**, *182*, 190–207. [CrossRef]
2. Siradjuddin, I.A.; Muntasa, A. Faster Region-based Convolutional Neural Network for Mask Face Detection. In Proceedings of the 2021 5th International Conference on Informatics and Computational Sciences (ICICoS), Semarang, Indonesia, 24–25 November 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 282–286.
3. Zhang, T.; Zhang, X.; Ke, X. Quad-FPN: A novel quad feature pyramid network for SAR ship detection. *Remote Sens.* **2021**, *13*, 2771. [CrossRef]
4. Zhang, T.; Zhang, X. HTC+ for SAR Ship Instance Segmentation. *Remote Sens.* **2022**, *14*, 2395. [CrossRef]
5. Bara, M.; Sagues, L.; Paniagua, F.; Broquetas, A.; Fàbregas, X. High-speed focusing algorithm for circular synthetic aperture radar (C-SAR). *Electron. Lett.* **2000**, *36*, 1. [CrossRef]
6. Zhu, M.; Hu, G.; Zhou, H.; Wang, S. H2Det: A high-speed and high-accurate ship detector in SAR images. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2021**, *14*, 12455–12466. [CrossRef]
7. Zhang, T.; Zhang, X.; Shi, J.; Wei, S. HyperLi-Net: A hyper-light deep learning network for high-accurate and high-speed ship detection from synthetic aperture radar imagery. *ISPRS J. Photogramm. Remote Sens.* **2020**, *167*, 123–153. [CrossRef]
8. Zhang, T.; Zhang, X.; Shi, J.; Wei, S.; Wang, J.; Li, J.; Su, H.; Zhou, Y. Balance scene learning mechanism for offshore and inshore ship detection in SAR images. *IEEE Geosci. Remote Sens. Lett.* **2020**, *19*, 1–5. [CrossRef]

9. Shang, X.; Zhao, J.; Zhang, H. Automatic overlapping area determination and segmentation for multiple side scan sonar images mosaic. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2021**, *14*, 2886–2900. [CrossRef]
10. Qin, X.; Luo, X.; Wu, Z.; Shang, J. Optimizing the sediment classification of small side-scan sonar images based on deep learning. *IEEE Access* **2021**, *9*, 29416–29428. [CrossRef]
11. Zhu, M.; Song, Y.; Guo, J.; Feng, C.; Li, G.; Yan, T.; He, B. PCA and kernel-based extreme learning machine for side-scan sonar image classification. In Proceedings of the 2017 IEEE Underwater Technology (UT), Busan, Republic of Korea, 21–24 February 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 1–4.
12. Fallon, M.F.; Kaess, M.; Johannsson, H.; Leonard, J.J. Efficient AUV navigation fusing acoustic ranging and side-scan sonar. In Proceedings of the 2011 IEEE International Conference on Robotics and Automation, Shanghai, China, 9–13 May 2011; IEEE: Piscataway, NJ, USA, 2011; pp. 2398–2405.
13. Wu, Z.; Yang, F.; Tang, Y. Side-scan sonar and sub-bottom profiler surveying. In *High-Resolution Seafloor Survey and Applications*; Springer: Singapore, 2021; pp. 95–122.
14. Al-Qatf, M.; Lasheng, Y.; Al-Habib, M.; Yu, L. Deep Learning Approach Combining Sparse Autoencoder With SVM for Network Intrusion Detection. *IEEE Access* **2018**, *6*, 52843–52856. [CrossRef]
15. Bertasius, G.; Shi, J.; Torresani, L. Deepedge: A multi-scale bifurcated deep network for top-down contour detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 4380–4389.
16. Wang, Y.; Sun, Y.; Lv, P.; Wang, H. Detection of line weld defects based on multiple thresholds and support vector machine. *NDT E Int.* **2008**, *41*, 517–524. [CrossRef]
17. Gong, M.; Su, L.; Jia, M.; Chen, W. Fuzzy clustering with a modified MRF energy function for change detection in synthetic aperture radar images. *IEEE Trans. Fuzzy Syst.* **2013**, *22*, 98–109. [CrossRef]
18. Dzieciuch, I.; Gebhardt, D.; Barngrover, C.; Parikh, K. Non-Linear Convolutional Neural Network for Automatic Detection of Mine-Like Objects in Sonar Imagery. In Proceedings of the International Conference on Applications in Nonlinear Dynamics, Rome, Italy, 21–25 May 2017; Springer: Cham, Switzerland, 2017; pp. 309–314.
19. Rhinelander, J. Feature extraction and target classification of side-scan sonar images. In Proceedings of the 2016 IEEE Symposium Series on Computational Intelligence (SSCI), Athens, Greece, 6–9 December 2016.
20. Song, Y.; Zhu, Y.; Li, G.; Feng, C.; He, B.; Yan, T. Side scan sonar segmentation using deep convolutional neural network. In Proceedings of the OCEANS 2017-Anchorage, Anchorage, AK, USA, 18–21 September 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 1–4.
21. Zhu, P.; Isaacs, J.; Fu, B.; Ferrari, S. Deep learning feature extraction for target recognition and classification in underwater sonar images. In Proceedings of the 2017 IEEE 56th Annual Conference on Decision and Control (CDC), Melbourne, Australia, 12–15 December 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 2724–2731.
22. Einsidler, D.; Dhanak, M.; Beaujean, P.P. A deep learning approach to target recognition in side scan sonar imagery. In Proceedings of the OCEANS 2018 MTS/IEEE Charleston, Charleston, SC, USA, 22–25 October 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 1–4.
23. Kim, J.; Choi, J.W.; Kwon, H.; Oh, R.; Son, S. The application of convolutional neural networks for automatic detection of underwater object in side scan sonar images. *J. Acoust. Soc. Korea* **2018**, *37*, 118–128.
24. Wu, M.; Wang, Q.; Rigall, E.; Li, K.; Zhu, W.; He, B. ECNet: Efficient convolutional networks for side scan sonar image segmentation. *Sensors* **2019**, *19*, 2009. [CrossRef] [PubMed]
25. Wang, Q.; Wu, M.; Yu, F.; Feng, C.; Li, K.; Zhu, Y.; Rigall, E.; He, B. Rt-seg: A real-time semantic segmentation network for side scan sonar images. *Sensors* **2019**, *19*, 1985. [CrossRef] [PubMed]
26. Yu, F.; He, B.; Li, K.; Yan, T.; Shen, Y.; Wang, Q.; Wu, M. Side scan sonar images segmentation for AUV with recurrent residual convolutional neural network module and self-guidance module. *Appl. Ocean Res.* **2021**, *113*, 102608. [CrossRef]
27. Wang, Y.; Liu, J.; Yu, S.; Wang, K.; Han, Z.; Tang, Y. Underwater Object Detection based on YOLO-v3 network. In Proceedings of the 2021 IEEE International Conference on Unmanned Systems (ICUS), Beijing, China, 15–17 October 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 571–575.
28. Li, J.W.; Cao, X. Target Recognition and Detection in Side scan Sonar Images based on YOLO v3 Model. In Proceedings of the 2022 41st Chinese Control Conference (CCC), Hefei, China, 25–27 July 2022; IEEE: Piscataway, NJ, USA, 2022; pp. 7186–7190.
29. Jin, S.; Zhang, N.; Bian, G.; Cui, Y. A seabed sediment classification model based on PSO-AlexNet. In Proceedings of the 2nd International Conference on Signal Image Processing and Communication (ICSIPC 2022), Qingdao, China, 20–22 May 2022; SPIE: Bellingham, WA, USA, 2022; Volume 12246, pp. 349–361.
30. Jia, X.; Wei, X.; Cao, X.; Foroosh, H. Comdefend: An efficient image compression model to defend adversarial examples. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 6084–6092.
31. Yang, X.; Yan, J. Arbitrary-oriented object detection with circular smooth label. In Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020; pp. 677–694.
32. Sahin, F.E.; Tanguay, A.R. Distortion optimization for wide-angle computational cameras. *Optics Express* **2018**, *26*, 5478–5487. [CrossRef]
33. Zhang, X.; Karaman, S.; Chang, S.F. Detecting and simulating artifacts in gan fake images. In Proceedings of the 2019 IEEE International Workshop on Information Forensics and Security (WIFS), Delft, The Netherlands, 9–12 December 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 1–6.

34. Sheikh, H.R.; Bovik, A.C. Image information and visual quality. *IEEE Trans. Image Process.* **2006**, *15*, 430–444. [CrossRef]
35. Zhang, W.; Ma, K.; Yan, J.; Deng, D.; Wang, Z. Blind image quality assessment using a deep bilinear convolutional neural network. *IEEE Trans. Circuits Syst. Video Technol.* **2018**, *30*, 36–47. [CrossRef]
36. Canziani, A.; Paszke, A.; Culurciello, E. An analysis of deep neural network models for practical applications. *arXiv* **2016**, arXiv:1605.07678.
37. Zhou, Z.; Cui, Z.; Zang, Z.; Meng, X.; Cao, Z.; Yang, J. UltraHi-PrNet: An Ultra-High Precision Deep Learning Network for Dense Multi-Scale Target Detection in SAR Images. *Remote Sens.* **2022**, *14*, 5596. [CrossRef]
38. Han, B.; Hu, Z.; Su, Z.; Bai, X.; Yin, S.; Luo, J.; Zhao, Y. Mask_LaC R-CNN for measuring morphological features of fish. *Measurement* **2022**, *203*, 111859. [CrossRef]
39. Wang, H.; Shi, Y.; Yue, Y.; Zhao, H. Study on freshwater fish image recognition integrating SPP and DenseNet network. In Proceedings of the 2020 IEEE International Conference on Mechatronics and Automation (ICMA), Osaka, Japan, 24–26 April 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 564–569.
40. Guo, H.; Gao, H.; Guo, C.; Lu, J.; Lin, Y. Dock detection method in remote sensing images based on improved YOLOv4. In Proceedings of the Fourteenth International Conference on Digital Image Processing (ICDIP 2022), Wuhan, China, 20–23 May 2022; SPIE: Bellingham, WA, USA, 2022; Volume 12342, pp. 105–112.
41. Han, Q.; Yin, Q.; Zheng, X.; Chen, Z. Remote sensing image building detection method based on Mask R-CNN. *Complex Intell. Syst.* **2022**, *8*, 1847–1855. [CrossRef]
42. Le, Y.; Yang, X. Tiny imagenet visual recognition challenge. *CS 231n* **2015**, *7*, 3.
43. Li, Y.; Zhang, X.; Shen, Z. YOLO-Submarine Cable: An Improved YOLO-V3 Network for Object Detection on Submarine Cable Images. *J. Mar. Sci. Eng.* **2022**, *10*, 1143. [CrossRef]
44. Ju, M.; Luo, H.; Wang, Z.; Hui, B.; Chang, Z. The application of improved YOLO V3 in multi-scale target detection. *Appl. Sci.* **2019**, *9*, 3775. [CrossRef]
45. Xu, D.; Wu, Y. Improved YOLO-V3 with DenseNet for multi-scale remote sensing target detection. *Sensors* **2020**, *20*, 4276. [CrossRef] [PubMed]
46. Karathanassi, V.; Kolokousis, P.; Ioannidou, S. A comparison study on fusion methods using evaluation indicators. *Int. J. Remote Sens.* **2007**, *28*, 2309–2341. [CrossRef]
47. Shang, R.; Wang, J.; Jiao, L.; Rustam, S.; Hou, B.; Li, Y. SAR targets classification based on deep memory convolution neural networks and transfer parameters. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2018**, *11*, 2834–2846. [CrossRef]
48. Pang, Y.; Zhao, X.; Zhang, L.; Lu, H. Multi-scale interactive network for salient object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 9413–9422.
49. Sriram, S.; Vinayakumar, R.; Sowmya, V.; Aamoun, A.; Soman, K. Multi-scale learning based malware variant detection using spatial pyramid pooling network. In Proceedings of the IEEE INFOCOM 2020-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), Toronto, ON, Canada, 6–9 July 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 740–745.
50. Li, S.; Yuan, S.; Liu, S.; Wen, J.; Huang, Q.; Zhang, Z. Characteristics of Low-Frequency Acoustic Wave Propagation in Ice-Covered Shallow Water Environment. *Appl. Sci.* **2021**, *11*, 7815. [CrossRef]
51. Sebens, K.P.; Witting, J.; Helmuth, B. Effects of water flow and branch spacing on particle capture by the reef coral Madracis mirabilis (Duchassaing and Michelotti). *J. Exp. Mar. Biol. Ecol.* **1997**, *211*, 1–28. [CrossRef]
52. Wang, X.; Wang, L.; Li, G.; Xie, X. A Robust and Fast Method for Sidescan Sonar Image Segmentation Based on Region Growing. *Sensors* **2021**, *21*, 6960. [CrossRef] [PubMed]
53. Ying, X. An overview of overfitting and its solutions. *J. Phys. Conf. Ser.* **2019**, *1168*, 022022. [CrossRef]
54. Pecorelli, F.; Di Nucci, D.; De Roover, C.; De Lucia, A. A large empirical assessment of the role of data balancing in machine-learning-based code smell detection. *J. Syst. Softw.* **2020**, *169*, 110693. [CrossRef]
55. Osetsky, Y.; Barashev, A.V.; Zhang, Y. Sluggish, chemical bias and percolation phenomena in atomic transport by vacancy and interstitial diffusion in NiFe alloys. *Curr. Opin. Solid State Mater. Sci.* **2021**, *25*, 100961. [CrossRef]
56. Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Chintala, S. Pytorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems*; MIT Press: Cambridge, MA, USA, 2019; Volume 32.
57. Moura, P.; Crocker, P.; Nunes, P. High-level multi-threading programming in logtalk. In Proceedings of the International Symposium on Practical Aspects of Declarative Languages, San Francisco, CA, USA, 7–8 January 2008; Springer: Berlin/Heidelberg, Germany, 2008; pp. 265–281.
58. He, K.; Girshick, R.; Dollár, P. Rethinking imagenet pre-training. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 4918–4927.
59. Jaikrishnan, S.V.J.; Chantarakasemchit, O.; Meesad, P. A breakup machine learning approach for breast cancer prediction. In Proceedings of the 2019 11th International Conference on Information Technology and Electrical Engineering (ICITEE), Pattaya, Thailand, 10–11 October 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 1–6.
60. Chu, H.; Xiong, X.; Gao, Y.J.; Luo, J.; Jing, H. Diffuse reflection and reciprocity-protected transmission via a random-flip metasurface. *Sci. Adv.* **2021**, *7*, eabj0935. [CrossRef]
61. Li, Z.; Peng, C.; Yu, G.; Zhang, X.; Deng, Y.; Sun, J. Detnet: A backbone network for object detection. *arXiv* **2018**, arXiv:1804.06215.

62. Albawi, S.; Mohammed, T.A.; Al-Zawi, S. Understanding of a convolutional neural network. In Proceedings of the 2017 International Conference on Engineering and Technology (ICET), Antalya, Turkey, 21–23 August 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 1–6.
63. Wang, Z.J.; Turko, R.; Shaikh, O.; Park, H.; Das, N. CNN explainer: Learning convolutional neural networks with interactive visualization. *IEEE Trans. Vis. Comput. Graph.* **2020**, *27*, 1396–1406. [CrossRef]
64. O'Shea, K.; Nash, R. An introduction to convolutional neural networks. *arXiv* **2015**, arXiv:1511.08458.
65. Zhang, T.; Zhang, X.; Ke, X.; Liu, C.; Xu, X. HOG-ShipCLSNet: A novel deep learning network with hog feature fusion for SAR ship classification. *IEEE Trans. Geosci. Remote Sens.* **2021**, *60*, 1–22. [CrossRef]
66. Wang, Y.; Wang, C.; Zhang, H. Ship classification in high-resolution SAR images using deep learning of small datasets. *Sensors* **2018**, *18*, 2929. [CrossRef] [PubMed]
67. Nguyen, D.P.T.; Matsuo, Y.; Ishizuka, M. Exploiting syntactic and semantic information for relation extraction from wikipedia. In Proceedings of the IJCAI Workshop on Text-Mining & Link-Analysis (TextLink 2007), Hyderabad, India, 6–12 January 2007.
68. Sacramento, J.; Ponte Costa, R.; Bengio, Y.; Senn, W. Dendritic cortical microcircuits approximate the backpropagation algorithm. In *Advances in Neural Information Processing Systems*; MIT Press: Cambridge, MA, USA, 2018; Volume 31.
69. Alom, M.Z.; Taha, T.M.; Yakopcic, C.; Westberg, S.; Asari, V.K. The history began from alexnet: A comprehensive survey on deep learning approaches. *arXiv* **2018**, arXiv:1803.01164.
70. Alippi, C.; Disabato, S.; Roveri, M. Moving convolutional neural networks to embedded systems: The alexnet and VGG-16 case. In Proceedings of the 2018 17th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN), Porto, Portugal, 11–13 April 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 212–223.
71. Yuan, Z.W.; Zhang, J. Feature extraction and image retrieval based on AlexNet. In Proceedings of the Eighth International Conference on Digital Image Processing (ICDIP 2016), Chengu, China, 20–22 May 2016; SPIE: Bellingham, WA, USA, 2016; Volume 10033, pp. 65–69.
72. Ballester, P.; Araujo, R.M. On the performance of GoogLeNet and AlexNet applied to sketches. In Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, Phoenix, AZ, USA, 12–17 February 2016.
73. Anand, R.; Shanthi, T.; Nithish, M.S.; Lakshman, S. Face recognition and classification using GoogleNET architecture. In *Soft Computing for Problem Solving*; Springer: Singapore, 2020; pp. 261–269.
74. Salavati, P.; Mohammadi, H.M. Obstacle detection using GoogleNet. In Proceedings of the 2018 8th International Conference on Computer and Knowledge Engineering (ICCKE), Mashhad, Iran, 25–26 October 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 326–332.
75. Muhammad, U.; Wang, W.; Chattha, S.P.; Ali, S. Pre-trained VGGNet architecture for remote-sensing image scene classification. In Proceedings of the 2018 24th International Conference on Pattern Recognition (ICPR), Beijing, China, 20–24 August 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 1622–1627.
76. Sathish, K.; Ramasubbareddy, S.; Govinda, K. Detection and localization of multiple objects using VGGNet and single shot detection. In *Emerging Research in Data Engineering Systems and Computer Communications*; Springer: Singapore, 2020; pp. 427–439.
77. Wu, Z.; Shen, C.; Van Den Hengel, A. Wider or deeper: Revisiting the ResNet model for visual recognition. *Pattern Recognit.* **2019**, *90*, 119–133. [CrossRef]
78. Li, S.; Jiao, J.; Han, Y.; Weissman, T. Demystifying ResNet. *arXiv* **2016**, arXiv:1611.01186.
79. Chen, Z.; Xie, Z.; Zhang, W.; Xu, X. ResNet and Model Fusion for Automatic Spoofing Detection. In Proceedings of the Interspeech, Stockholm, Sweden, 20–24 August 2017; pp. 102–106.
80. Khan, R.U.; Zhang, X.; Kumar, R.; Aboagye, E.O.; Kumar, R. Evaluating the performance of ResNet model based on image recognition. In Proceedings of the 2018 International Conference on Computing and Artificial Intelligence, Chengdu, China, 12–14 March 2018; pp. 86–90.
81. He, F.; Liu, T.; Tao, D. Why ResNet works? Residuals generalize. *IEEE Trans. Neural Netw. Learn. Syst.* **2020**, *31*, 5349–5362. [CrossRef]
82. Redmon, J.; Farhadi, A. Yolov3: An incremental improvement. *arXiv* **2018**, arXiv:1804.02767.
83. Zhao, L.; Li, S. Object detection algorithm based on improved YOLOV3. *Electronics* **2020**, *9*, 537. [CrossRef]
84. Won, J.H.; Lee, D.H.; Lee, K.M.; Lin, C.H. An improved YOLOv3-based neural network for de-identification technology. In Proceedings of the 2019 34th International Technical Conference on Circuits/Systems, Computers and Communications (ITC-CSCC), Jeju, Republic of Korea, 23–26 June 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 1–2.
85. Lang, P.; Fu, X.; Martorella, M.; Dong, J.; Xie, M. A comprehensive survey of machine learning applied to radar signal processing. *arXiv* **2020**, arXiv:2009.13702.
86. Lee, Y.H.; Kim, Y. Comparison of CNN and YOLO for Object Detection. *J. Semicond. Disp. Technol.* **2020**, *19*, 85–92.
87. He, K.; Zhang, X.; Ren, S.; Sun, J. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *37*, 1904–1916. [CrossRef]
88. Yue, J.; Mao, S.; Li, M. A deep learning framework for hyperspectral image classification using spatial pyramid pooling. *Remote Sens. Lett.* **2016**, *7*, 875–884. [CrossRef]
89. Huang, Z.; Wang, J.; Fu, X.; Yu, T.; Wang, R. DC-SPP-YOLO: Dense connection and spatial pyramid pooling based YOLO for object detection. *Inf. Sci.* **2020**, *522*, 241–258. [CrossRef]

90. Liu, Y.; Gross, L.; Li, Z.; Li, X.; Qi, W. Automatic building extraction on high-resolution remote sensing imagery using deep convolutional encoder-decoder with spatial pyramid pooling. *IEEE Access* **2019**, *7*, 128774–128786. [CrossRef]
91. Zhang, X.; Wang, W.; Zhao, Y.; Xie, H. An improved YOLOv3 model based on skipping connections and spatial pyramid pooling. *Syst. Sci. Control Eng.* **2021**, *9* (Suppl. 1), 142–149. [CrossRef]
92. Xu, F.; Wang, H.; Sun, X.; Fu, X. Refined marine object detector with attention-based spatial pyramid pooling networks and bidirectional feature fusion strategy. *Neural Comput. Appl.* **2022**, *34*, 14881–14894. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.