electronics

MDPI

Article

A Foam Line Position Detection Algorithm for A/O Pool Based on YOLOv5

Yubin Xu *, Yihao Wu and Yinzhang Guo

College of Computer Science and Technology, Taiyuan University of Science and Technology, Taiyuan 030000, China; hqw212hqw@gmail.com (Y.W.); guoyinzhang@tyust.edu.cn (Y.G.)
* Correspondence: xyub@163.com

Abstract: During the biochemical pretreatment process of leachate in urban landfill sites, if the foam in the A/O pool is not promptly addressed, it can lead to overflow, posing hazards to the surrounding environment and personnel. Therefore, a real-time foam line detection algorithm based on YOLOv5x was proposed, which enhances feature information and improves anchor box regression prediction to accurately detect the position of foam lines. Firstly, in the preprocessing stage, employing a rectangular box to simultaneously label the foam line and the edge of the A/O pool within the same region, enhances the feature information of the foam line. Then, the C3NAM module was proposed, which applies weight sparse penalties to attention modules in the feature extraction section, to enhance the capability of extracting foam line features. Subsequently, a B-SPPCSPC module was proposed to enhance the fusion of shallow and deep feature information, addressing the issue of susceptibility to background interference during foam line detection. Next, the Focal_EIOU was introduced to ameliorate the issue of class imbalance in detection, providing more accurate bounding box predictions. Lastly, optimizing the detection layer scale improves the detection performance for smaller targets. The experimental results demonstrate that the accuracy of this algorithm reaches 98.9%, and the recall reaches 88.1%, with a detection frame rate of 26.2 frames per second, which can meet the actual detection requirements of real-world application scenarios.

Keywords: foam line detection; C3NAM; B-SPPCSPC; YOLOv5x; Focal_EIOU

Citation: Xu, Y.; Wu, Y.; Guo, Y. A Foam Line Position Detection Algorithm for A/O Pool Based on YOLOv5. *Electronics* **2024**, *13*, 1834. https://doi.org/10.3390/electronics13101834

Academic Editor: Daniel Riccio

Received: 10 March 2024
Revised: 16 April 2024
Accepted: 21 April 2024
Published: 9 May 2024

1. Introduction

During the treatment process of leachate from urban landfill sites, a significant amount of foam is continuously generated within the Anaerobic/Oxic (A/O) pool [1]. At the pool edge, due to the adhesive nature and movement characteristics of foam, an accumulation of foam that is too high may lead to overflow if not addressed promptly. This necessitates continuous observation by observers to make timely and accurate judgments, heavily relying on the subjective assessments and experience of the observer. Particularly in cases where the A/O pool operates for extended periods, is of considerable scale, or requires high monitoring frequency, manual visual inspection is severely limited in terms of accuracy and efficiency. It is not only time-consuming and labor-intensive, but also prone to inaccuracies, resulting in failure to meet the demands of current production requirements.

Articles and reports specifically addressing foam line detection are currently lacking. However, due to the overall shape of foam lines being approximately similar to water level lines, one could refer to methods used for water level detection as a basis for developing foam line detection techniques. Currently, water level detection methods can be broadly categorized into two types. One is the traditional method, which includes installing a water gauge, visually reading its measurements, or using sensors to measure the water level. Among these methods, installing water gauges is inefficient and lacks timeliness, while the method of using sensors to automatically collect analog signals related to water levels, and then converting them into water level quantities, is costly and difficult to maintain.

Both of these methods are highly susceptible to the influence of harsh environmental conditions. Another category is water level measurement based on image processing. Iwahashi et al. [2] improved detection accuracy by adding the frame images of moving objects and applying frame averaging to reduce interference in water level line detection. However, this algorithm requires affine transformation for slanted water level lines, making it unsuitable for irregular water level lines. Additionally, the use of the LDA algorithm results in excessively long processing times for large-sized images. Lin et al. [3] employed the Maximum Interclass Variance method to segment the water gauge, then conducted template matching on the segmented water gauge characters, and lastly calculated the water level in conjunction with the water level line. However, this method does not eliminate the segmentation residue of the Maximum Interclass Variance method, leading to larger-than-actual measurement results. Chen et al. [4] employed morphological methods to enhance grayscale images, improving binarization in water gauge segmentation, and subsequently recognizing the water level. However, this method relies on grayscale images and is susceptible to lighting conditions. Zhou et al. [5] employed image differencing to extract areas of water level change, calibrated the camera using water gauge scales, and calculated the actual water level height based on the camera calibration results. However, the position of the water gauge and the camera can have a great impact on the measurement results. Bao et al. [6] determined the water level position by calculating the maximum value generated when sliding the Haar feature module between the centroid positions of boundary and non-boundary points. However, this algorithm provides only a rough calibration of the water level line, with lower accuracy in areas with many folds. Additionally, it is prone to false detections in regions with strong illumination and reflections.

Due to the high requirements of traditional image processing methods on lighting, environment, and shooting positions, the accuracy and timeliness of water level measurements are often compromised in complex situations. However, with the rapid combination of neural networks and image processing, the accuracy of water level detection methods based on image processing has been effectively enhanced. Xu et al. [7] utilized the preliminary positioning of the water gauge in addition to dynamic mapping or Convolutional Neural Networks (CNN) character recognition on segmented characters to determine the water level position. However, the water gauge is susceptible to the surrounding environment, leading to the instability of segmented characters. Cheng et al. [8] utilized a U-shaped Convolutional Neural Network (U-Net) to determine the water level line. After segmenting the water surface, the points with the maximum pixel value difference formed the water level line. However, in regions with complex situations and multiple folds, adjacent pixel values are too close, leading to segmented areas closely approximating the straight line. Xiao et al. [9] employed SSD to determine the position of the water gauge, relying on the "E" scale in the water to establish the water level line position. However, the tilt of the water gauge is highly susceptible to changes in water surface fluctuations, subsequently affecting the area ratio of non-zero pixels within the boundary range and impacting the determination of the water level line position. Huang et al. [10] proposed the deep learning-based universal paradigm, DeepWL, for diagnosing captured images and determining whether they represent the water level line based on scalar values. However, this method requires establishing a recognition field within the observation area, making the computation complex and unsuitable for small-scale scenarios. Wu et al. [11] utilized a denoising model combining CNN and residual networks to obtain the water level line through grayscale stretching, indirectly determining the actual water level value. This method similarly relies on the water gauge as a medium, requires the surrounding water environment statutes to remain nearly consistent, and is highly susceptible to environmental influences. Liao et al. [12] designed a new loss function based on slope and intercept for CNN training, predicted the water level line based on the predicted coordinates of the intersection point between the water level line and the left boundary of the image, and the predicted angle value between the water level line and the horizontal direction. However, this method needs to assume that the water level line is approximately straight and is

not suitable for scenarios where many curves cannot be approximated as straight lines. Additionally, it is prone to a significant number of error samples.

The difference between foam lines and water level lines lies in the fact that, in a segment of the water flow where the velocity is relatively steady, the water level does not undergo sharp changes, and exhibits relatively stable characteristics. Consequently, the above-mentioned methods for detecting water levels are mostly based on water gauges. However, in real-world application scenarios, the water gauge can only play a role in a relatively calm water flow due to the width itself limits, and this method is highly susceptible to the complex surrounding environment. Additionally, whether it is image edge or image color, the water gauge positioning will be greatly affected by them. Unlike the relatively stable characteristics of the water level line. The characteristic of foam lines is as follows: due to the continuous motion within the A/O pool, there are almost never any steady, straight-line shape foam lines present at the same moment, nor are there identical foam lines at adjacent moments. If we divide the foam line into small segments, the changing situation in each segment of the foam line is different; no single segment of the foam line can represent the motion situation of the overall foam line. At the position of the A/O pool edge, foam accumulates together, and its shape is irregular and constantly changing. At any given moment, the water gauge readings in any area are inadequate for representing the changing situation in the overall foam line. Hence, methods relying on water gauge measurements, such as HSV color space, morphological transformations, and multi-frame water gauge image processing, have limitations in detecting foam lines.

2. Related

In recent years, with the continuous innovative development of machine learning in the field of image processing, deep learning-based image processing algorithms have begun to play an increasingly important role in industrial applications. Deep learning-based detection algorithms utilize neural networks to comprehend hierarchical features within data. It employs multiple predictive layers to automatically learn the hidden internal structures within training data [13–15]. It involves designing specific network architectures based on the characteristics of input data and continuously optimizing algorithms to make the output results as close to the input data as possible, and has gradually become an indispensable and important technology in industrial production [16,17], particularly in the field of water level surveying [18].

Target detection algorithms based on deep learning can be broadly classified into two categories. The first category comprises two-stage target detection algorithms [19], which include Region-based CNN (R-CNN) [20], Spatial Pyramid Pooling Networks (SPP-Net) [21], and Faster R-CNN [22]. While these algorithms boast high accuracy, they suffer from a slower detection speed. The second category consists of single-stage target detection algorithms. In this approach, each region of interest is directly classified as either an object or background, and then the object is directly detected and located through a stage, including the You Only Look Once (YOLO) [23–27] series, You Only Look One-level eXtreme (YOLOX) [28], and Single Shot MultiBox Detector (SSD) [29]. These algorithms are advantageous for their fast detection speed, but they come at the expense of lower precision. Compared to YOLO algorithms, SSD utilizes multi-scale features for detection. It replaces fully connected layers with fully convolutional networks [30], assigns multiple prior bounding boxes with different aspect ratios for each network, and employs data augmentation techniques. However, YOLOv5 features a more lightweight architecture with a smaller model size. It achieves superior performance in terms of inference speed due to optimization strategies such as model pruning, model quantization, and model trimming. Additionally, YOLOv5 demonstrates superior performance in detection accuracy, attributed to the introduction of adaptive convolution [31] and Path Aggregation Network (PANet) [32]. In various scenarios, such as image analysis, autonomous driving, and industrial, many scholars have proposed different improvement methods for deep learning algorithms. Chan et al. [33] redesigned the classifiers by collecting the features produced

between the network sets, and presenting the constituent layers and the activation function for the classifiers, to calculate the classification score of each classifier. Compared to the original model, it exhibits better performance. Lin et al. [34] adopted a novel framework for identifying captured individuals in the presence of multiple cameras while providing some methods for fast person detection and tracking in multi-view cameras. Collins et al. [35] employed the FCNN-based ML model to classify beach and water returns in lidar line scan time series. After training is completed, the classified data are used to estimate swash depths and define the rising position. Gao et al. [36] enhanced the backbone network of YOLOv5 by introducing two attention mechanisms. This modification allows the model to focus on crucial information in the images, resulting in more accurate detection of fabric defects. Li et al. [37] introduced the Channel-Global Attention Mechanism (CGAM) into the backbone network to enhance the feature extraction capabilities for targets of different scales and suppress interference from redundant information. Additionally, they incorporated the Dense Upsampling Convolution (DUC) module to expand low-resolution convolutional feature maps, effectively enhancing the fusion of various convolutional feature maps. Zhao et al. [38] employed an improved attention module and a BatchNorm layer in a multi-regularized adaptive network pruning algorithm. This algorithm achieves the optimal pruning of the network structure through a collaborative process of pruning and training. Moreover, they applied a combined training and INT8 quantization algorithm, implementing both saturated and unsaturated mapping, to meet the accuracy and real-time requirements for autonomous target recognition in complex environments. Chen et al. [39] proposed an improved feature extraction network that integrates Deformable Convolutional Networks Version 2 (DCNv2) and Coordinate Attention (CA). This modification broadens the receptive field for small target defects. Additionally, they introduced an improved PANet called CA-PANet, facilitating the reuse of shallow features. This method strengthens feature fusion for defects of different scales, enhances the feature representation capability of defects, and improves the accuracy of defect detection boxes. Li et al. [40] employed depth-wise separable convolution modules to reduce model parameters, optimized weight allocation based on the CA mechanism and an adaptive spatial feature fusion strategy, and improved the loss function to enhance the reliability of the detected target bounding boxes.

Considering the strong learning capabilities and fast real-time processing of the YOLOv5 algorithm, along with its ability to simultaneously balance model detection accuracy, this paper opts for the higher-accuracy You Only Look Once version 5 extra-large (YOLOv5x) to implement the detection of foam lines. As the motion state of foam within the A/O pool is continually changing, YOLOv5x has some difficulties in extracting foam line features, particularly in areas with intense motion, making it difficult to detect foam lines in smaller regions. Therefore, this paper proposes the BYOLOv5x (Balance-YOLOv5x) algorithm, which primarily focuses on researching and improving the feature extraction part, spatial pyramid part, loss function part, and detection layers part of YOLOv5x. The main innovations of the study are as follows:

(1) In the image preprocessing stage, the entire foam line is annotated in segments. For each region formed by the highest and lowest points, the points on the foam line within that region, along with their vertically corresponding points on the A/O pool edge, are annotated using rectangular bounding boxes, and annotations are applied segment by segment. This process enhances the feature information of the foam line, addressing the issue of missed detections in smaller regions, and ultimately improving the accuracy of foam line detection.
(2) In the feature extraction stage, the C3NAM module is proposed, which considers contribution factors to enhance the ability of the model to extract foam line features.
(3) In the spatial pyramid part, the B-SPPCSPC module is proposed to enhance the perception degree of feature information in two branches. Simultaneously, it strengthens the fusion capability of deep and shallow feature information.

(4) By introducing the Focal and Efficient-Intersection over Union Loss (Focal_EIOU Loss), the model could focus on the high-quality anchor boxes during the bounding box regression process, enhancing the precision of bounding box regression predictions. Finally, optimizing the detection layer scales allows the model to pay more attention to smaller target areas, thereby improving the model's detection capability for foam lines.

3. Proposed Algorithm

3.1. Overall Structure

For enhancing the extraction capability of multi-scale feature information, promoting the fusion of feature information of shallow and deep layers, improving the feature expression capability of foam lines, and addressing issues related to susceptibility to background interference in the foam line detection process, this paper designed the B-SPPCSPC module, reconstructed the spatial pyramid pooling part of YOLOv5x, and thus proposed the BYOLOv5x algorithm based on the YOLOv5x framework. At the same time, a C3NAM attention module is proposed in the backbone to enhance the feature extraction capability for foam lines. In the output layer, the Focal_EIOU Loss is introduced as the bounding box loss function to provide more precise bounding box regression predictions. Finally, the detection layer scale is optimized to enhance the model's overall detection capability. The specific network framework is illustrated in Figure 1.

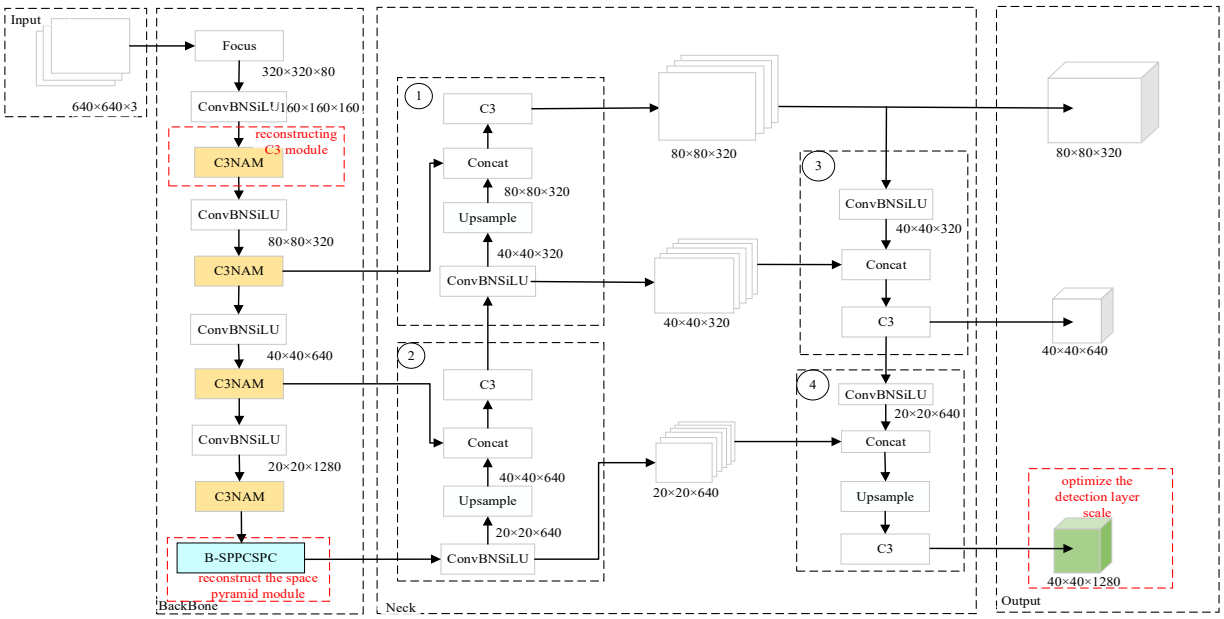


Figure 1. The overall structure of the BYOLOv5x algorithm.

3.2. C3NAM

The attention mechanism plays a significant role in enhancing the performance of deep convolutional neural networks, especially its plug-and-play manifestation form, demonstrating noticeable effectiveness in deep-layered networks within the domain of deep learning at profound levels. However, in the process of extracting target feature information, if the attention mechanism lacks consideration for the contribution factors of weights, it may result in fewer remarkable features being obtained. Therefore, this paper proposed a C3NAM attention module, which leverages the contribution factors of weights to enhance the effect of attention. The NAM module adopts integrated modules of the Hybrid Channel Attention Mechanism (CBAM), and redesigns the channel attention and spatial attention submodules, embedding them at the end of each network block. By utilizing the scaling factor from Batch Normalization (BN) to reflect the magnitude of changes and degree of importance in each channel, where the scaling factor represents the variance in BN, the output formula is as shown in Formula (1). The larger the variance,

the more intense the changes in that channel, indicating richer information and higher importance in the channel. Conversely, smaller variance implies less diverse information and lower importance.

$$B_{out} = BN(x_i) = \gamma \hat{x}_i + \beta = \gamma \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \varepsilon}} + \beta, \tag{1}$$

The implementation process of Formula (1) in the neural network is as follows: Firstly, compute the mean μ_B of the features for a mini-batch B, as shown in Formula (2). Then, compute the variance σ_B^2 of the features for the same small batch B, as shown in Formula (3). Next, normalize the features using the mean and variance, as shown in Formula (4). Finally, the batch normalization layer uses the γ and β to linearly transform the values of normalized.

$$\mu_B = \frac{1}{m} \sum_{i=1}^{m} x_i \tag{2}$$

$$\sigma_B^2 = \frac{1}{m} \sum_{i=1}^{m} (x_i - \mu_B)^2 \tag{3}$$

$$\hat{x}_i = \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \varepsilon}} \tag{4}$$

where γ and β are trainable affine transformation parameters, m is the number of samples in the batch, and ε is a very small number to prevent division by zero.

x_i represents the feature values of a single sample in the input data. In batch normalization, for each feature dimension, the mean and variance of the entire mini-batch are all computed, and these statistics are used to normalize each feature $x_{i,j}$ of each sample. This is expressed as shown in Formula (5), in which j represents the index of the feature.

$$\hat{x}_{i,j} = \frac{x_{i,j} - \mu_{B,j}}{\sqrt{\sigma_{B,j}^2 + \varepsilon}} \tag{5}$$

Therefore, x_i as the input to the BN layer, is normalized, re-scaled, and shifted, serving as the input to the next layer. This helps the network converge faster and reduces overfitting during the training process.

For the channel attention submodule, as illustrated in Figure 2, initially, the scaling factor is employed to measure the pixel magnitude of the channel dimension in the spatial dimension of the input feature map. The output results undergo the influence of channel weights and activation function successively, ultimately yielding the output feature formula, as shown in Equation (6), where M represents the output feature, and γ represents the scaling factor for each channel. When the object is the channel dimension, the value of n is taken as γ, and W_i represents the channel dimension weights.

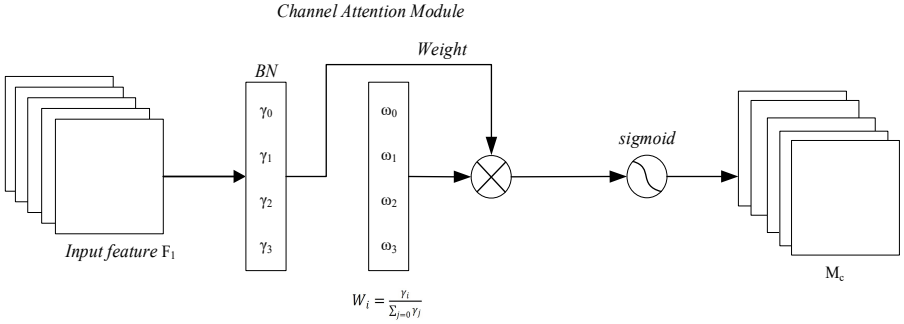


Figure 2. Channel attention submodule.

The spatial attention submodule is illustrated in Figure 3. If the same normalization method is applied to each pixel in the spatial dimension to measure pixel importance, it is referred to as pixel normalization. The formula for the output feature is as shown in Equation (6). λ represents the scaling factor for each channel, and when the object is the spatial dimension, the value of n is taken as λ. W_λ represents the channel dimension weight.

$$M = sigmoid(W_n(BN(F))), \tag{6}$$

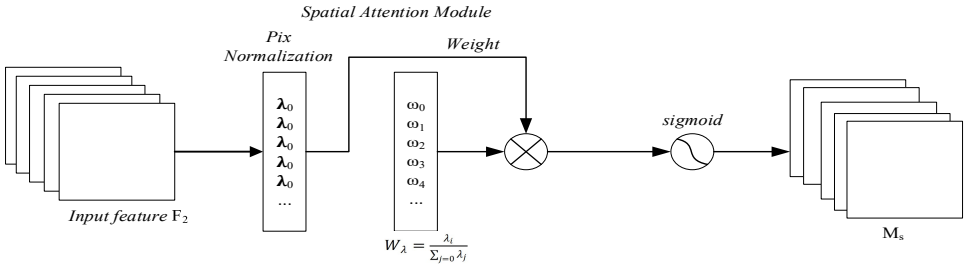


Figure 3. Spatial attention submodule.

Simultaneously, to suppress less important features, a regularization term is introduced in the loss function, as shown in Equation (7), where x represents the input, y represents the output, W represents the network weights, $l(\cdot)$ represents the loss function, $G(\cdot)$ represents the l_1 norm penalty function, and p represents the penalty function for $G(\gamma)$ and $l(\lambda)$

$$Loss = \sum_{(x,y)} l(f(x, W)) + \rho \sum g(\gamma) + \rho \sum g(\lambda), \tag{7}$$

To further suppress less significant channels or pixels, the C3NAM module is proposed, as illustrated in Figure 4. When the feature map is inputted to the C3NAM module, one branch first passes through a 1×1 convolutional layer to adjust the channel number of the input feature map, mapping the channel number to a smaller unit, thereby reducing computational complexity and improving computational efficiency. Afterward, the input feature map passes through the NAMBottleNeck module, where NAMBottleNeck is formed by inserting the NAM module after the first Convolutional Batch Normalization with SiLU (ConvBNSiLU) in the BottleNeck. In the NAMBottleNeck, the input feature map first undergoes a 1×1 convolution operation, reducing the number of channels by half to decrease model complexity. Then, the NAM module applies normalization and channel weights to the input feature map, allowing the network to dynamically choose features to focus on in the target region. Learning the relationships between features enhances the model's attention and expressive capability towards input features, thereby improving the performance and generalization ability of the detection algorithm. Afterward, the feature map undergoes a 3×3 convolution operation to double the channel count, thereby increasing the network's receptive field and obtaining higher-level feature representation. Another branch goes through only a ConvBNSiLU module, conducting a convolution operation on the input feature map and undergoing batch normalization and activation function processing to extract global feature information. Finally, concatenate the feature map passed through the NAMBottleNeck module with the feature map from the other branch along the channel dimension, keeping the channel dimension unchanged and forming a new feature map. Then, pass through the ConvBNSiLU operation again and output the feature map. The purpose of the C3NAM module allows the model to flexibly adjust its attention to different channels, enhance the role of weight contribution factors in feature extraction, improve the model's expression capability on specific channels, and help the model better capture critical information in the image.

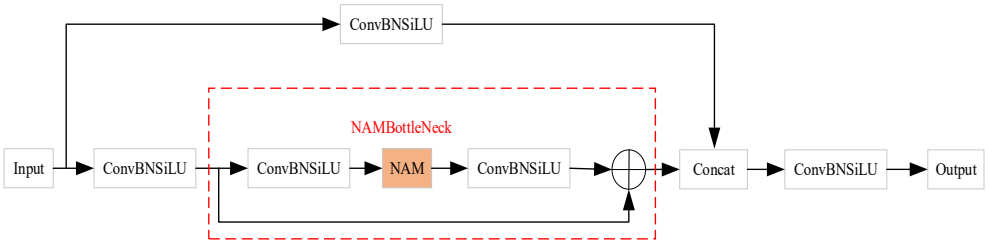


Figure 4. C3NAM structure graph.

C3NAM module flexibly adjusts the focus on different channels by applying a weight sparsity penalty to the attention module, thus improving the algorithm's generalization ability and its extract capability to feature information at different scales. Using scaling factors to reflect the richness of channel feature information, improves the algorithm's feature representation capacity on specific channels and enhances the extract capability of the weight contribution factors in extracting multi-scale feature information, which aids the algorithm in better capturing critical information within the image.

3.3. B-SPPCSPC

The Spatial Pyramid Pooling and Cross Stage Partial Channel (SPPCSPC) is composed of a feature pyramid pooling layer and small residual structures, with residual connections introduced before the Spatial Pyramid Pooling (SPP) layer. In this architecture, the output generated by the convolutional module of SPP is connected with the output from the residual structure, effectively integrating multi-scale features generated by SPP. The addition of residual connections provides additional feature information, significantly enhancing the network's capability of the perception and interpretation of image features. However, due to the adhesive nature of foam, during the treatment of leachate in the A/O pool, the upward and downward movement of the foam line may cause some foam to remain in its original position when it sinks and still maintains contact with the foam in the pool. This can result in the model mistakenly identifying the foam as still being at a higher position, leading to false detections of the foam line. Simultaneously, due to the obstruction of buildings within the A/O pool, there also may be instances of missed detections or false positives in the foam line. Therefore, to address the issue of susceptibility to interference from other backgrounds during the foam line detection process, the B-SPPCSPC module is proposed to reconstruct the spatial pyramid in the backbone network. On one hand, by adding pooling layers, strengthening the feature information extraction capability of a single branch and improving the model's perceptual awareness. On the other hand, by reducing ConvBNSiLU operations, retaining more shallow-layer feature information from the input feature map, increasing the number of pooling layers, and then preserving the newly downsampled feature map. The B-SPPCSPC module is illustrated in Figure 5.

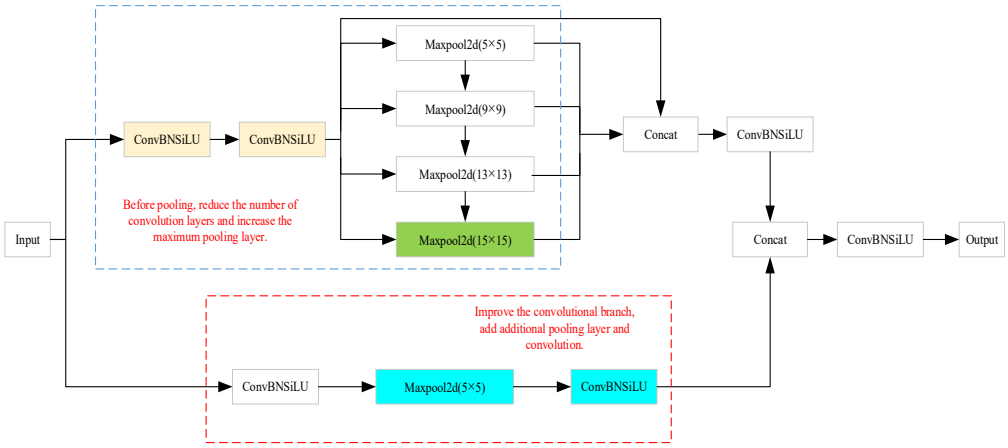


Figure 5. B-SPPCSPC structure graph.

When the input feature map enters the B-SPPCSPC module, first, one branch undergoes ConvBNSiLU operation and then adds a maximum pooling layer with Kernel_size = 5, stride = 1, and padding = 4 to perform pooling operations on the smaller target of the input feature map, thereby achieving compression and feature extraction for feature map. Afterward, a ConvBNSiLU operation is applied again, involving convolution on the input feature map followed by batch normalization. Subsequently, introducing the non-linear characteristics by the activation function ensures that the generated feature map retains more shallow-level feature information. In the other branch, the input feature map has undergone four iterations of C3NAM modules and ConvBNSiLU modules for feature information extraction before entering the B-SPPCSPC module, and it already possesses a sufficient richness of deep-level feature information. Therefore, to retain a sufficiently rich amount of shallow information, we first reduced the number of convolutional layers to extract local features in the image. Subsequently, the feature map will undergo individual and sequentially max-pooling operations with pooling scales of 5, 9, 13, and 15, respectively. Among them, by adding a max-pooling layer with Kernel_size = 15, stride = 1, and padding = 8, it captures deep feature information of the image during sequential pooling. When pooling separately, it performs pooling operations on the retained shallow features of the feature map to preserve more shallow information. By applying different pooling operations to the input feature map, feature information of different scales is extracted. Then, the feature maps from different scales are adaptively fused through concatenation, forming a multi-scale feature map, which achieves a feature-level combination of local and global features at the scale of the feature map. Subsequently, channel-wise 1×1 convolutional operations are applied to fuse the multi-scale feature map in the channel dimension, further extracting rich feature information. Finally, by concatenating the feature maps output from the two branches and subjecting them to ConvBNSiLU operations, the final output is obtained.

The B-SPPCSPC module captures features at different scales by employing pooling kernels of various sizes. Adding additional max-pooling operations to expand the model's receptive field allows for the capture of a broader range of contextual information, thereby enhancing the extraction capability of multi-scale feature information of the foam line. By reducing the Conv operations to retain some shallow features, leveraging the detail information of the original image to combine with deep features after downsampling, which facilitates the combination of shallow and deep feature information of the input feature map. This method enhances the feature representation capability of the foam line, providing rich information for the detection task. Additionally, it improves the algorithm's inference capability and resistance to background interference.

3.4. Loss Function

In the YOLOv5x bounding box regression loss, the adopted loss function is the Complete Intersection over Union (CIOU) [41]. It comprehensively considers the overlap area, central point distance, and aspect ratio, leading to a more stable regression of predicted bounding boxes. Compared to the Intersection over Union (IOU) loss function [42], CIOU not only introduced the minimum enclosing rectangle of the predicted and ground truth boxes, but also augments a penalty term, which was used for the aspect ratio of the bounding boxes. The function is expressed as shown in Formulas (8), (9), and (10):

$$LOSS_{CIoU} = 1 - IoU + \frac{\rho^2\left(b, b^{gt}\right)}{c^2} + \alpha v, \tag{8}$$

$$\alpha = \frac{v}{(1 - IoU) + v}, \tag{9}$$

$$v = \frac{4\left(\arctan\frac{\omega^{gt}}{h^{gt}} - \arctan\frac{\omega}{h}\right)^2}{\pi^2}, \tag{10}$$

where α represents a positive balancing parameter, $\rho^2\left(b, b^{gt}\right)$ represents the square of the distance between the centers of the predicted and the truth bounding boxes, c represents the minimum of the diagonal lengths of the predicted box and the ground truth box, and the aspect ratio ν denotes the relative proportion of the height and width of the predicted bounding box compared to the truth bounding box.

CIOU incorporates ν as a penalty term into the bounding box loss function. While this can expedite the convergence of the regression process for predicted boxes, in the regression process of predicted boxes, if the aspect ratio of the predicted and the truth boxes shows a linear relationship, it results in the inability of the predicted box's height and width to simultaneously increase or decrease, which hinders the continuation of regression optimization.

The Efficient Intersection over Union (EIOU) loss function, building upon the CIOU penalty term, separates the contribution factors of the aspect ratio of the predicted and truth boxes. Then, calculate the length and width separately for both the predicted and truth boxes. The calculation formula is shown in Equation (11):

$$L_{EIOU} = L_{IOU} + L_{dis} + L_{asp} = 1 - IOU + \frac{\rho^2\left(b, b^{gt}\right)}{c^2} + \frac{\rho^2\left(\omega, \omega^{gt}\right)}{C_\omega^2} + \frac{\rho^2\left(h, h^{gt}\right)}{C_h^2}, \quad (11)$$

where C_ω represents the width of the minimum enclosing rectangle for the predicted and truth bounding boxes. C_h represents the height of the minimum enclosing rectangle for the predicted and truth bounding boxes. It can be seen that the EIOU function consists of the loss of overlap L_{IOU}, the loss of center point L_{dis}, and the loss of side length L_{asp} for both the predicted and ground truth boxes. The first parts of the EIOU loss remain unchanged, while the side length loss is obtained by splitting the aspect ratio loss term, minimizing the difference between the width and height of the predicted box and the ground truth box. This approach avoids the issue where the penalty term of CIOU becomes ineffective when the predicted bounding box's length and width are satisfied $\left[\left(\omega = k\omega^{gt}, h = kh^{gt}\right)\middle| k = R^+\right]$. It also addresses the problem of length and width being unable to simultaneously increase or decrease, thereby accelerating the convergence speed of the predicted box and improving the precision of predicted box regression.

To further optimize the issue of category imbalance in the bounding box regression task, reducing the number of predefined anchor boxes to mitigate the impact of low-quality anchor boxes on the predicted boxes regression optimization process, this study employed the FocalL1 Loss to enhance the EIOU Loss. The calculation formula for FocalL1 Loss is shown in Equation (12).

$$L_f(x) = \begin{cases} -\frac{\alpha x^2(2\ln(\beta x)-1)}{2}, 0 < x \le 1, \frac{1}{e} \le \beta \le 1 \\ -\alpha \ln(\beta) x + C, x > 1, \frac{1}{e} \le \beta \le 1 \end{cases}, \quad (12)$$

where C is a constant, set $C = \frac{2\alpha \ln \beta + \alpha}{4}$ to ensure $L_f(x)$ has continuous values at $x = 1$, β is a parameter controlling the shape of the focal loss curve, α is used to normalize the gradients of low- and high-quality anchor boxes corresponding to different cases of β, and x represents the bounding box regression loss.

The Focal_EIOU Loss weights the EIOU loss function by IOU, separating high-quality anchor boxes from low-quality ones from a gradient perspective. This ensures that the regression process of predicted boxes focuses on high-quality anchor boxes, that is, those predicted boxes with a relatively high IOU compared to the ground truth box. The formula is shown in (13):

$$LOSS_{Focal\_EIOU} = IOU^\gamma LOSS_{EIOU}, \quad (13)$$

where γ is the parameter that controls the degree of outlier suppression. When the IOU between two bounding boxes increases, the loss value increases with the increase in IOU. The gradient values for low-quality anchor boxes decrease, while the gradient values for high-quality anchor boxes increase accordingly.

Accurate bounding box regression can ensure that both small-scale and large-scale targets can be accurately detected. The Focal_EIOU Loss function dynamically adjusts gradient values based on the quality of anchor boxes and by reducing the gradient values of low-quality anchor boxes to reduce their interference with model training. Selecting anchor boxes with high gradient values accurately reflects the relationship between predicted boxes and ground truth boxes, thus mitigating sample imbalance issues to some extent, and improving the accuracy of predicting box regression. For objects with significant size variations, this algorithm can also more flexibly adjust the predicted boxes, thereby better adapting to multi-scale features.

3.5. Optimizing Detection Layer

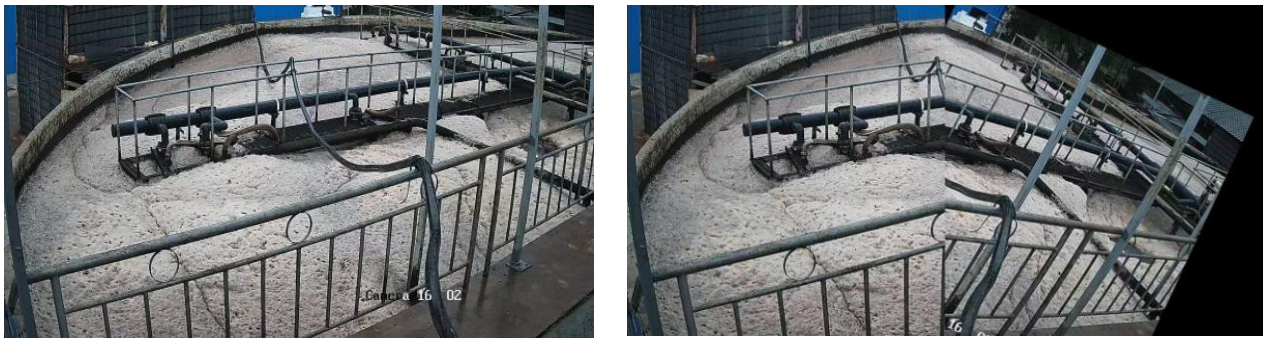
Due to the irregular and constantly changing nature of the foam line at the A/O pool edge, and considering that each segment of the foam line that needs to be detected occupies a relatively small portion of the image, the amount of feature information contained in each segment of foam line is also little. This will make the detection layer easily overlook crucial information in smaller areas, leading to a decrease in detection accuracy. Assuming the input image size is 640 × 640, after upsampling by 8 times, 16 times, and 32 times, the generated detection scales are 20 × 20, 40 × 40, and 80 × 80, respectively. The corresponding receptive field sizes are 32 × 32, 16 × 16, and 8 × 8 pixel regions. These scales are, respectively, responsible for detecting targets at deep, medium, and shallow levels. Considering the characteristics of the foam line, the scale of detection for deep-level targets was set to be consistent with that of the medium detection level. By reconstructing the deep-level detection layer to narrow the detection scale receptive field, the algorithm can be made to utilize both deep semantic information and shallow detailed features more effectively. This enables the algorithm to better capture foam line features in smaller regions, enhancing the perceptual ability for feature information in smaller regions, and thereby improving the algorithm's adaptability to multi-scale features of the foam line. By employing two medium detection scales to capture the feature information of smaller regions, the understanding and expressive capability of detail information in the image is enhanced, especially for the areas in the image that occupy smaller spaces but contain rich feature information. Additionally, it improves the extraction capability of multi-scale features of the foam line and detection accuracy. This adjustment, while ensuring detection effectiveness, aims to extract more information from smaller targets, thereby enhancing the model's inference speed.

4. Experiments and Results

4.1. Data Source and Preprocessing

The foam line dataset used in this study is derived from the leachate treatment renovation project at the municipal solid waste landfill site, and all data are captured by installed cameras during the production process. Due to the rectangular nature of the detection boxes in YOLOv5x, the foam line on the right side of the original image exhibits a relatively flat, horizontal orientation. This characteristic leads to the inability of the algorithm to detect the foam line in the right section of the image during the detection process. Therefore, to enhance the accuracy of foam line detection, before training, it is chosen to cut the image at the inflection point where the foam line appears approximately horizontally, choosing the image's upper-left corner of the image as the origin, with width as the horizontal axis. This point is marked as x_0. Cut the image along that point. In the following content, the left part refers to the image with horizontal coordinates less than x_0, and the right part refers to the image with horizontal coordinates greater than x_0. To achieve a similar inclination in both the left and right sections of the image, the right section is rotated clockwise. The rotation angle is selected within the range of 25 to 45 degrees, ensuring that both sections display a comparably inclined angle. Then, splice the right part and left part of the image at the midpoint between the edge of the A/O pool corresponding to the cutting point and the foam line. Parts (a) and (b) in Figure 6 represent the original image and the processed

image, respectively. Simultaneously, to increase the training samples, offline augmentation techniques such as rotation, salt-and-pepper noise, Gaussian blur, darkening, brightening, downward translation, and horizontal flipping were applied to the existing images, resulting in 4200 augmented images.



(**a**) (**b**)

Figure 6. Original image (**a**) and the stitched image (**b**).

The experiments in this paper adopted a fully supervised learning approach, where images were manually annotated using the LabelMe image annotation tool. Taking the lowest and highest points within a certain segment of the foam line region in the image, the lowest and highest point coordinates were respectively assigned as the first and second coordinates of the annotation box. Respectively selecting the point on the A/O pool edge corresponding vertically to the highest and lowest point, and respectively assigning these points as the third and fourth coordinate of the annotation box. Using these four coordinates to construct an annotation box with the label "paomo", the remaining foam line areas on the image are annotated segment by segment. The annotated image is depicted as shown in Figure 7a. In the JSON file, adjust the coordinates of adjacent annotation boxes and respectively set the first and fourth coordinates of the subsequent annotation box equal to the second and third coordinates of the previous annotation box, which ensures the continuity of annotation boxes, reduces errors, and also maintains the continuity of the foam line and the A/O pool edge during detection. Finally, convert the JSON file to the XML file required by the model and modify the even indices label to "Pled". Therefore, there will be two types of labels for detection: "paomo" representing the foam line and "Pled" representing the A/O pool edge. Once the model training is completed, the annotation boxes for the two different labels will be separated, as depicted in Figure 7b. The connecting lines formed by the diagonal lines of annotation boxes belonging to the same label separately constitute the to-be-detected foam line and A/O pool edge.

(**a**) (**b**)

Figure 7. The labeled image (**a**) and the recognition result image (**b**).

4.2. Evaluation Metrics

The experimental environment was chosen to use the PyTorch learning framework with an NVIDIA GeForce RTX 3090-24GB GPU. The graphics card driver version was

530.30.02, and the processor was an AMD EPYC 7601 256G 64-core CPU. The CUDA version used was 12.1. The dataset is randomly divided into a training set and a validation set with a ratio of 9:1. The image input size was defined as 640 × 640, and the final learning rate was 0.1. Additionally, the training epochs were set to 300.

In the proposed algorithm, the following metrics are selected as evaluation criteria: Precision, Recall, True Positive (TP, the number of correctly detected frames for foam lines and A/O pool edges), False Negatives (FN, the number of missed detections for foam lines and A/O pool edges), the number of parameters, and Frames Per Second (Fps). Additionally, Average Precision (mAP) and mAP@.5:.0.95 are also chosen as evaluation metrics. The formulas for Precision, Recall, and mAP are as follows:

$$\text{Precision} = \frac{TP}{FP + TP} \times 100\%, \tag{14}$$

$$\text{Recall} = \frac{TP}{FN + TP} \times 100\%, \tag{15}$$

$$\text{mAP} = \frac{1}{n}\int_0^1 p(r)dr, \tag{16}$$

where FP represents the number of cases where foam lines and A/O pool edge detection boxes were detected but were not correctly identified. $p(r)$ represents the curve with precision as the vertical axis and recall as the horizontal axis; n represents the total number of classes, with n = 2.

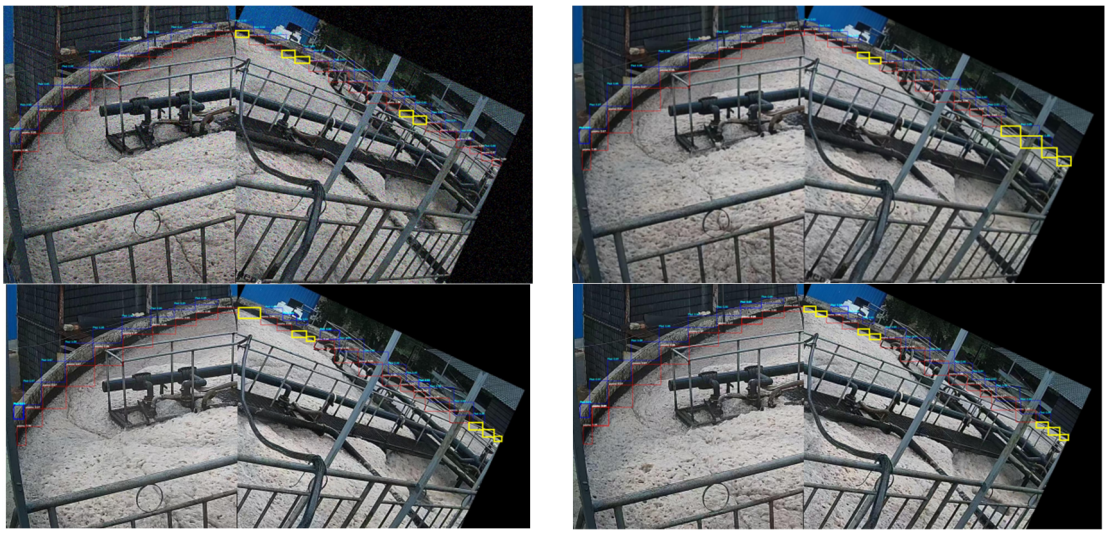
4.3. Result Analysis

4.3.1. Comparison Experiments

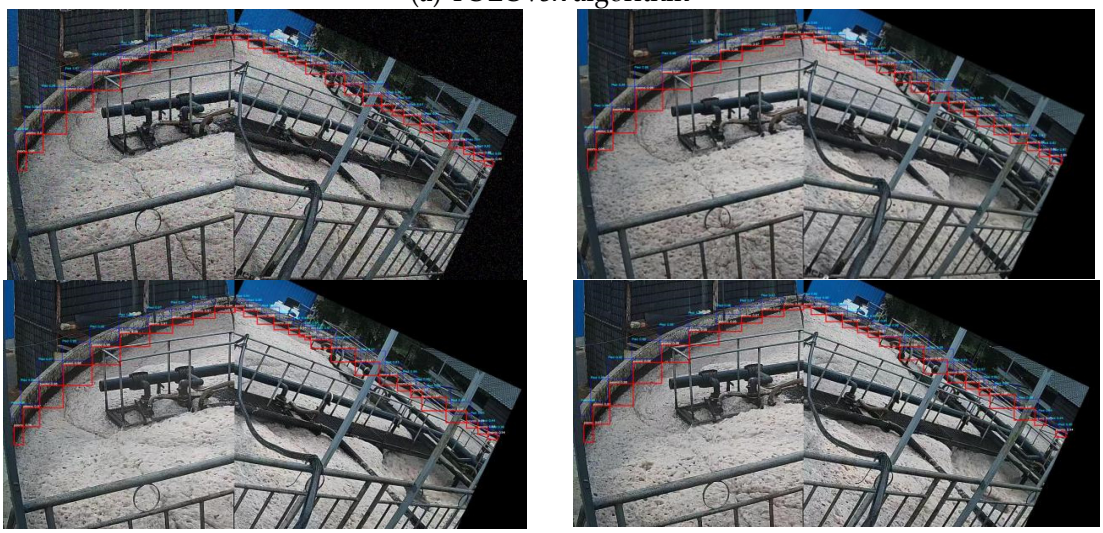
The experimental environment was chosen to use the PyTorch learning framework with an NVIDIA graphics card. Comparisons of results were conducted using images captured from the same frame, as illustrated in Figure 8. The label "paomo" designates the detected foam line region, and its connecting lines formed by the diagonal lines of annotation boxes constitute the foam line. The label "Pled" corresponds to the detected A/O pool edge region, and its connecting lines formed by the diagonal lines of annotation boxes constitute the A/O pool edge. The yellow box denotes areas where the foam line was not detected. It can be observed that the YOLOv5x detection results lead to curve omissions in both continuous and discontinuous regions, and the missed detection rate is high, which is easy to cause information loss. In contrast, the BYOLOv5x algorithm accurately and comprehensively shows an entire segment of the foam line. It connected all adjacent detection box diagonals with the same label to form the to-be-detected foam line. This approach reduces experimental errors, ensuring the integrity and continuity of the foam line.

To further validate the performance of BYOLOv5x, we trained SSD, Faster R-CNN, YOLOv5x, YOLOv5s, YOLOv5n, and YOLOX algorithms on the same training dataset for 300 epochs under identical experimental conditions and settings. The comparative results of various evaluation metrics are presented in Table 1. It can be observed that in the detection process of foam lines in the A/O pool, the performance of the BYOLOv5x algorithm is the best. Compared to YOLOv5x, the BYOLOv5x algorithm has an increase of 2.7 percentage points in Precision, an increase of 3.2 percentage points in Recall, an increase of 6 in TP, and a reduction of 4.8 in FN. Compared to YOLOX, YOLOv5s, and YOLOv5n, the Precision values of BYOLOv5x increased by 3, 3.6, and 13.2 percentage points, respectively. The Recall values increased by 4.6, 5.4, and 15.1 percentage points, respectively. TP increased by 8, 8, and 11 units, respectively, while FN decreased by 4.8, 5.3, and 9.3 units, respectively. Compared to Faster R-CNN and SSD, BYOLOv5x demonstrates a significantly improved detection speed. The optimized algorithm, despite an increase in parameter count, demonstrated improvements in all evaluation metrics. Moreover, the Fps for transmission increased by 21.7, 21.1, 16, 15.5, 8.6, and 5.6, respectively. The BYOLOv5x

algorithm maintained a detection frame rate of 26.2 frames per second, ensuring fluency and real-time performance during actual detection. The comparison curves for precision and recall are illustrated in Figure 9a,b, respectively. It can be observed that both the precision and recall of BYOLOv5x converge relatively quickly compared to YOLOv5x. Additionally, after reaching their peak values, both of them have no significant fluctuations, meeting the real-time and accuracy requirements for foam line detection.



(**a**) YOLOv5x algorithm



(**b**) BYOLOv5x algorithm

Figure 8. Comparison of detection results of YOLOv5x and BYOLOv5x.

Table 1. Comparison results.

Models	Precison/%	Recall/%	TP/Frame	FN/Frame	Fps	Parameter
SSD	68.4	66.3	21	38	4.5	50,175,361
Faster-RCNN	71.6	69.5	26	33	5.1	61,223,027
YOLOv5n	85.7	73.2	47	12.0	10.2	1,872,157
YOLOv5s	95.3	82.9	50	8.0	10.7	7,025,023
YOLOX	95.9	83.7	50	7.5	17.6	27,106,654
YOLOv5x	96.2	85.1	52	7.5	20.6	86,224,543
BYOLOv5x	98.9	88.3	58	2.7	26.2	126,983,623

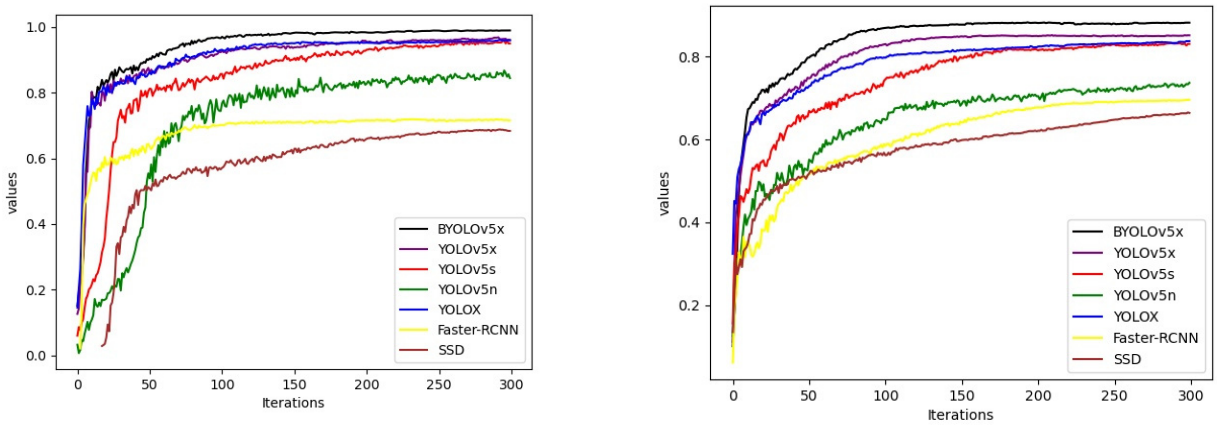


Figure 9. Precision comparison graph (**a**) and recall comparison graph (**b**).

Draw the curves of various evaluation metrics for BYOLOv5x, as shown in Figure 10. Observing Figure 10a, it is found that the algorithm quickly converges in the first 120 rounds, with a rapid decline in the loss value. As the number of training rounds increases, it tends to stabilize. Until the end of the training, there is no overfitting or underfitting, indicating a good training effect. Observing Figure 10b, it can be seen that the evaluation metrics Precision and Recall quickly converge in the first 100 rounds, while mAP@.5:.0.95 converges rapidly in the first 120 rounds. As the number of training rounds increases, they tend to stabilize. The accuracy of BYOLOv5x reached 98.9%, the recall rate reached 88.3%, and mAP@.5:.0.95 reached 87%, meeting the detection requirements for foam lines.

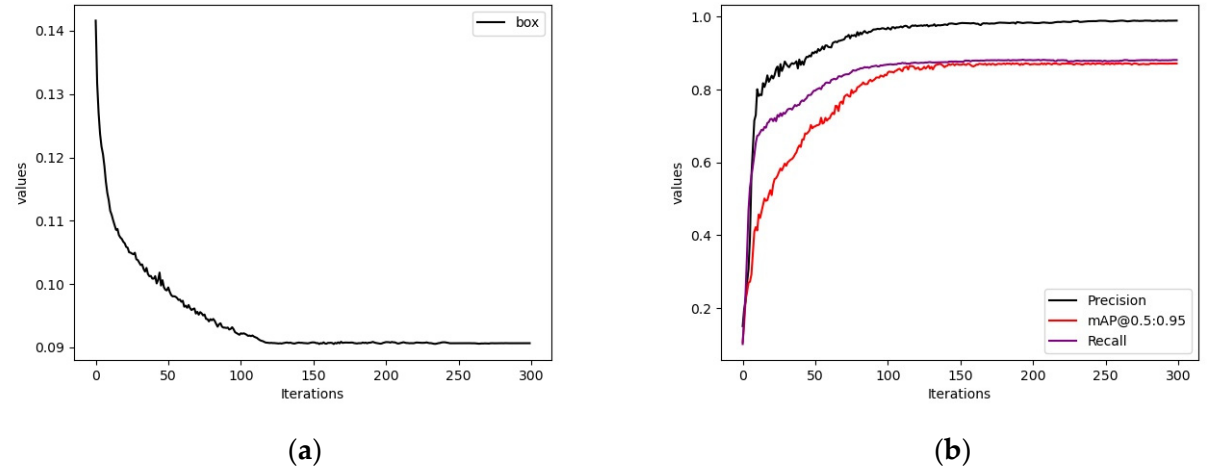


Figure 10. Loss function curve (**a**) and evaluation index curve (**b**).

Separately extract the diagonal coordinates of the detection boxes from the detection results of BYOLOv5x and YOLOv5x, and compare the detection curves formed by them with the actual position of the foam line, as shown in Figure 11. It can be observed that, compared to the actual position of the foam line, the YOLOv5x algorithm missed certain segments of the foam line during detection. If the foam line in this segment is in a stable state, in this case, the approximate state of the foam line in that segment can be represented by the line between adjacent points in that segment, and the impact on the results can be considered negligible. However, in actual scenarios, foam lines are constantly churning, and there is almost no stable state. Therefore, for the foam lines within the missing region, regardless of their length, it is challenging to accurately determine their specific conditions, and it very easily leads to misjudgment. At the same time, this also indirectly confirms

that a higher value of the evaluation metric TP corresponds to more accurate detection of target boxes, while a higher value of FN increases the likelihood of discontinuous foam line detections. The foam line detected by the BYOLOv5x algorithm is closer to the actual foam line, indicating a higher level of matching and stronger integrity and continuity. Consequently, the smaller the error, the higher the precision of the foam line detection.

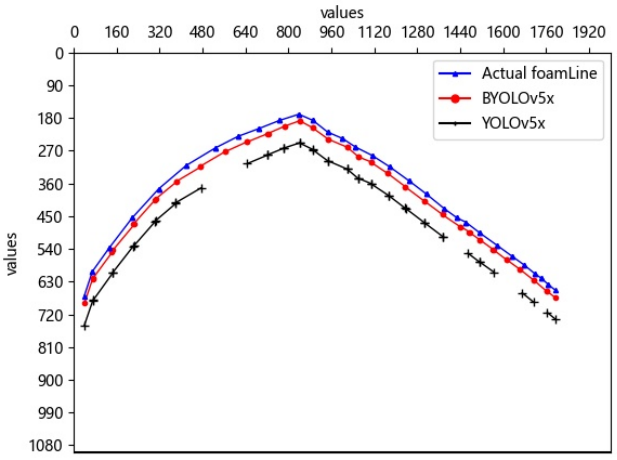


Figure 11. Foam line comparison.

4.3.2. Ablation Experiments

To better demonstrate the reliability of the proposed algorithm, respective improvements or compound improvements were made to YOLOv5x by adopting the proposed C3NAM module, B-SPPCSPC module, Focal_EIOU Loss function, and optimized detection layer scales. Table 2 presents the results of the experiments, '×' indicates that the method was not employed, while '√' indicates its adoption.

Table 2. Comparison of ablation experiment results.

Num	C3NAM	B-SPPCSPC	Focal_EIOU	Detection Layer Scale	P/%	R/%	mAP	mAP@.5:.0.95	Parameter
1	√	√	√	√	98.9	88.3	90.5	87.0	126,983,623
2	√	√	√	×	98.1	87.9	88.6	86.5	126,763,623
3	√	√	×	×	97.5	87.4	87.5	85.3	126,763,623
4	×	√	√	×	97.3	88.0	87.8	85.6	126,596,423
5	×	×	√	√	97.6	87.8	87.2	86.7	86,664,543
6	√	×	×	×	97.1	86.8	89.4	85.1	86,611,743
7	×	√	×	×	97.6	87.3	87.4	84.9	126,376,423
8	×	×	√	×	97.8	87.7	86.7	85.4	86,004,543
9	×	×	×	√	97.4	84.5	87.5	86.2	86,004,543
10	×	×	×	×	95.3	85.1	86.7	84.5	86,224,543

From Table 2, comparing serial numbers 6 and 10, although the improvements of the C3NAM module led to an increase in parameter count, both precision and recall rates have improved, which has resulted in a significant improvement in detection efficiency. Serial number 7 improved the pooling layer through B-SPPCSPC. Despite a significant increase in the parameter count, the most noticeable improvement is observed in the recall, allowing the algorithm to correctly detect a greater number of positive instances. Serial number 8 employed the Focal_EIOU loss function, maintaining the mAP value and parameter count unchanged, resulting in a 2.5% increase in accuracy and a 2.6% increase in recall rate. Serial number 9, by optimizing the detection layer scale, improved the detecting object accuracy of smaller areas. Serial numbers 3, 4, and 5 represent composite experiments of improvement methods on YOLOv5x. Comparing with serial numbers 6, 7, 8, and 9, it can be observed that all the evaluation metrics have improved. Under the combination of

multiple improvements, the Recall and mAP@.5:.0.95 are superior to those achieved by individual improvement methods. Serial number 1 represents the improvement algorithm proposed in this paper, with an increase in precision of 3.77% and an increase in recall rate of 3.76%. Although there is a slight increase in parameter count, the increment falls within an acceptable range. The ablation experiments indicate that BYOLOv5x exhibits excellent feature extraction capabilities for frequently changing and irregular foam lines, enabling accurate detection of the foam line positions in the images.

5. Conclusions

In response to the challenges posed by traditional neural network-based object detection models in handling foam line scenarios, including low detection accuracy, slow speed, and difficulties in detecting rapidly changing regions, this paper proposed a real-time foam line online detection method called BYOLOv5x for A/O pool. To obtain the coordinates of the foam line, a new rectangular annotation method is employed to label both the foam line and A/O pool edge within the same region, so the to-be-detected foam line can be represented by the connecting lines formed by the diagonal lines of annotation boxes. Then, by proposing the C3NAM attention mechanism in the YOLOv5x algorithm, the capability of extracting foam line features is enhanced. The proposed BSPPCSPC module improves the model's pooling ability, addressing the issue of susceptibility to background interference in the foam line detection process. The introduction of the Focal_EIOU Loss function ameliorated the issue of class imbalance in model detection, enabling the model to pay more attention to the anchor boxes of high quality. Additionally, this way can also deliver more precise bounding box predictions. Furthermore, optimizing the detection layer scale improved the detection performance for smaller targets, enhancing foam line detection accuracy. This ensures that the foam line detection model meets the precision and real-time requirements in practical applications. However, this algorithm involves multiple image preprocessing steps, has a little large model framework, and lacks practical testing in complex scenarios such as nighttime conditions. In the future, there is a need to improve the image preprocessing methods to reduce the workload. Additionally, efforts should be directed towards model lightweighting and adapting the model for a wider range of practical scenarios.

Author Contributions: Conceptualization, Y.X., Y.W. and Y.G.; methodology, Y.X., Y.W. and Y.G.; investigation, Y.X. and Y.G.; resources, Y.X.; writing—original draft preparation, Y.X. and Y.W.; writing—review and editing, Y.X., Y.W. and Y.G. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Shanxi Province Free Exploration Basic Research Funding Project (grant number YDZJSX20231A044) and Shanxi Province Basic Research Plan General Program (grant number 2021030212321).

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

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

References

1. Zheng, J.F. Research on measures to improve the efficiency of A/O pool process treatment. *Man-Made Fibers* **2020**, *50*, 20–25.
2. Iwahashi, M.; Udomsiri, S. Water Level Detection from Video with Fir Filtering. In Proceedings of the 2007 16th International Conference on Computer Communications and Networks, Honolulu, HI, USA, 13–16 August 2007.
3. Lin, F.; Chang, W.-Y.; Lee, L.-C.; Hsiao, H.-T.; Tsai, W.-F.; Lai, J.-S. Applications of Image Recognition for Real-Time Water Level and Surface Velocity. In Proceedings of the 2013 IEEE International Symposium on Multimedia, Anaheim, CA, USA, 9–11 December 2013; pp. 259–262.
4. Chen, C.; Liu, Z.W.; Chen, X.S.; Luo, M.N.; Niu, Z.X.; Ruan, C. Technology of Water Level Automatically Extract based on Image Processing. *Water Resour. Informatiz.* **2016**, *1*, 48–55.
5. Zhou, H.; Zhong, S.D. Research on Water Level Monitoring Based on Image Processing. *Semicond. Optoelectron.* **2019**, *40*, 390–394+400.
6. Bao, J.; Tao, Q.C.; Zhang, P. Image Processing Based Water Level Detection Algorithm. *Hydropower Energy Sci.* **2015**, *33*, 96–99+210.

7. Xu, Z.K.; Feng, J.; Zhang, Z.Z.; Shu, X.C. An automatic water depth measurement method combined with convolutional neural network. *Small Microcomput. Syst.* **2019**, *40*, 793–797.
8. Cheng, S.; Zhao, K.; Zhang, S.; Zhang, D. Water Level Detection Based on U-net. *Acta Metrol.* **2019**, *40*, 361–366.
9. Xiao, Z.; Tao, Q.C.; Shen, J.J. A Video water-level Recognition Based on SSD Object Detect Network. *Mod. Comput.* **2019**, 60–64.
10. Huang, J.; Miao, H.; Li, L.; Wen, Y.; Xiao, C. Deep Visual Waterline Detection within Inland Marine Environment. *arXiv* **2019**, arXiv:1911.10498.
11. Wu, T.; Chu, Z.F.; Chen, C.; Zhu, J.Y. Study on image level recodnition based on grayscale stretching. *High Technol. Commun.* **2021**, *31*, 327–332.
12. Liao, Y.; Duan, Q.; Liu, J.; Zhou, H. Water line detection algorithm based on deep learning. *Comput. Appl.* **2020**, *40* (Suppl. S1), 274–278.
13. Shrestha, A.; Mahmood, A. Review of Deep Learning Algorithms and Architectures. *IEEE Access* **2019**, *7*, 53040–53065. [CrossRef]
14. Bengio, Y. Learning Deep Architectures for AI. *Found. Trends Mach. Learn.* **2009**, *2*, 1–127. [CrossRef]
15. Wong, A.; Shafiee, M.J.; Chwyl, B.; Li, F. GenSyth: A new way to understand deep learning. *Electron. Lett.* **2019**, *55*, 970–971. [CrossRef]
16. Meng, D.Y.; Sun, L. Some New Trends of Deep Learning Research. *Chin. J. Electron.* **2019**, *28*, 1087–1091. [CrossRef]
17. Gheisari, M.; Ebrahimzadeh, F.; Rahimi, M.; Moazzamigodarzi, M.; Liu, Y.; Dutta Pramanik, P.K.; Heravi, M.A.; Mehbodniya, A.; Ghaderzadeh, M.; Feylizadeh, M.R.; et al. Deep learning: Applications, architectures, models, tools, and frameworks: A comprehensive survey. *CAAI Trans. Intell. Technol.* **2023**, *8*, 581–606. [CrossRef]
18. Sit, M.; Demiray, B.Z.; Xiang, Z.; Ewing, G.J.; Sermet, Y.; Demir, I. A Comprehensive Review of Deep Learning Applications in Hydrology and Water Resources. *Water Sci. Technol. A J. Int. Assoc. Water Pollut. Res.* **2020**, *82*, 2635–2670. [CrossRef] [PubMed]
19. Li, K.Q.; Chen, Y.; Liu, J.C.; Mu, X.W. Survey of Deep Learning-Based Object Detection Algorithms. *Comput. Eng.* **2022**, *48*, 1–12.
20. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Columbus, OH, USA, 23–28 June 2014; pp. 580–587.
21. 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]
22. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 1137–1149. [CrossRef]
23. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 26 June–1 July 2016; pp. 779–788.
24. Redmon, J.; Farhadi, A. YOLO9000: Better, Faster, Stronger. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 6517–6525.
25. Redmon, J.; Farhadi, A. YOLOv3: An Incremental Improvement. *arXiv* **2018**, arXiv:1804.02767.
26. Bochkovskiy, A.; Wang, C.Y.; Liao, H.Y. YOLOv4: Optimal Speed and Accuracy of Object Detection. *arXiv* **2020**, arXiv:2004.10934.
27. Jocher, G.; Stoken, A.; Chaurasia, A.; Borovec, J.; Kwon, Y.; Michael, K.; Changyu, L.; Fang, J.; Skalski, P.; Hogan, A.; et al. *Ultralytics/Yolov5: v6.0-YOLOv5n 'Nano' Models, Roboflow Integration, TensorFlow Export, OpenCV DNN Support*; Zenodo: Geneva, Switzerland, 2021.
28. Ge, Z.; Liu, S.; Wang, F.; Li, Z.; Sun, J. YOLOX: Exceeding YOLO Series in 2021. *arXiv* **2021**, arXiv:2107.08430.
29. 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 Computer Vision–ECCV 2016, Amsterdam, The Netherlands, 11–14 October 2016; pp. 21–37.
30. Zhang, Z.; Guo, W.; Yu, W.; Yu, W. Multi-task fully convolutional networks for building segmentation on SAR image. *J. Eng.* **2019**, *2019*, 7074–7077. [CrossRef]
31. Chandran, P.; Zoss, G.; Gotardo, P.; Gross, M.; Bradley, D. Adaptive Convolutions for Structure-Aware Style Transfer. In Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 20–25 June 2021; pp. 7968–7977.
32. Liu, S.; Qi, L.; Qin, H.; Shi, J.; Jia, J. Path Aggregation Network for Instance Segmentation. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 8759–8768.
33. Chan, K.H.; Im, S.K.; Ke, W. Multiple classifier for concatenate-designed neural network. *Neural Comput. Appl.* **2022**, *34*, 1359–1372. [CrossRef]
34. Im, S.K.; Chan, K.H. Distributed Spatial Transformer for Object Tracking in Multi-Camera. In Proceedings of the 2023 25th International Conference on Advanced Communication Technology (ICACT), Pyeongchang, Republic of Korea, 19–22 February 2023; pp. 122–125. [CrossRef]
35. Collins, A.M.; O'dea, A.; Brodie, K.L.; Bak, A.S.; Hesser, T.J.; Spore, N.J.; Farthing, M.W. Automated Extraction of a Depth-Defined Wave Runup Time Series From Lidar Data Using Deep Learning. *IEEE Trans. Geosci. Remote Sens.* **2023**, *61*, 5700913. [CrossRef]
36. Gao, M.; Zou, Y.L.; Cao, X.W. Fabric defect detection based on improved YOLOv5 model. *Mod. Text. Technol.* **2023**, *31*, 155–163.
37. Li, K.; Ou, O.; Liu, G.B.; Yu, Z.F.; Li, L. Target Detection Algorithm of Remote Sensing Image Based on Improved YOLOv5. *Comput. Eng. Appl.* **2023**, *59*, 207–214.

38. Zhao, X.D.; Zhang, X.Y. Optimization Algorithm of Autonomous Target Recognition for Unmanned Vehicles Based on YOLOv5. *J. Ordnance Eng.* **2023**, *44*, 2732–2744.
39. Chen, Y.; Liao, F.; Huany, X.; Yang, J.; Gong, H. Multi-scale YOLOv5 solar cell defect detection. *Opt. Precis. Eng.* **2023**, *31*, 1804–1815. [CrossRef]
40. Li, H.J.; Kong, F.C.; Lin, Y. Infrared ship detection algorithm based on improved YOLOv5s. *Syst. Eng. Electron. Technol.* **2023**, *45*, 2415–2422.
41. Zheng, Z.; Wang, P.; Liu, W.; Li, J.; Ye, R.; Ren, D. Distance-IoU Loss: Faster and Better Learning for Bounding Box Regression. In Proceedings of the AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 27 January–1 February 2019; pp. 12993–13000.
42. Yu, J.; Jiang, Y.; Wang, Z.; Cao, Z.; Huang, T. UnitBox: An Advanced Object Detection Network. In Proceedings of the MM '16: Proceedings of the 24th ACM International Conference on Multimedia, Amsterdam, The Netherlands, 15–19 October 2016; pp. 516–520.