

Article

Real-Time Detection of Slug Flow in Subsea Pipelines by Embedding a Yolo Object Detection Algorithm into Jetson Nano

Weiliang Qiao ^{1,2} , Hongtongyang Guo ¹, Enze Huang ¹, Xin Su ¹, Wenhua Li ^{1,*}  and Haiquan Chen ¹

¹ Marine Engineering College, Dalian Maritime University, Dalian 116026, China; xiaoqiao_fang@dmlu.edu.cn (W.Q.); ghty@dmlu.edu.cn (H.G.); 1013253045hez@dmlu.edu.cn (E.H.); xin5231@dmlu.edu.cn (X.S.); chenapec@dmlu.edu.cn (H.C.)

² School of Maritime and Economics, Dalian Maritime University, Dalian 116026, China

* Correspondence: lwh992@dmlu.edu.cn

Abstract: In the multiple-phase pipelines in terms of the subsea oil and gas industry, the occurrence of slug flow would cause damage to the pipelines and related equipment. Therefore, it is very necessary to develop a real-time and high-precision slug flow identification technology. In this study, the Yolo object detection algorithm and embedded deployment are applied initially to slug flow identification. The annotated slug flow images are used to train seven models in Yolov5 and Yolov3. The high-precision detection of the gas slug and dense bubbles in the slug flow image in the vertical pipe is realized, and the issue that the gas slug cannot be fully detected due to being blocked by dense bubbles is solved. After model performance analysis, Yolov5n is verified to have the strongest comprehensive detection performance, during which, mAP_{0.5} is 93.5%, mAP_{0.5:0.95} is 65.1%, and comprehensive mAP (cmAP) is 67.94%; meanwhile, the volume of parameters and Flops are only 1,761,871 and 4.1 G. Then, the applicability of Yolov5n under different environmental conditions, such as different brightness and adding random obstructions, is analyzed. Finally, the trained Yolov5n is deployed to the Jetson Nano embedded device (NVIDIA, Santa Clara, CA, USA), and TensorRT is used to accelerate the inference process of the model. The inference speed of the slug flow image is about five times of the original, and the FPS has increased from 16.7 to 83.3.

Keywords: subsea engineering; flow assurance; deep learning; Yolo; slug flow; Jetson Nano



Citation: Qiao, W.; Guo, H.; Huang, E.; Su, X.; Li, W.; Chen, H. Real-Time Detection of Slug Flow in Subsea Pipelines by Embedding a Yolo Object Detection Algorithm into Jetson Nano. *J. Mar. Sci. Eng.* **2023**, *11*, 1658. <https://doi.org/10.3390/jmse11091658>

Academic Editors: Fausto Pedro García Márquez and Coro Gianpaolo

Received: 8 August 2023
Revised: 18 August 2023
Accepted: 23 August 2023
Published: 24 August 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

With the continuous improvement and innovation of subsea engineering technology, oil and gas exploitation is gradually shifting from land/coastal waters to ocean areas [1]. Deep sea oil and gas exploitation has also become an important part of marine resource development [2]. In the field of the subsea oil and gas industry, it is widely observed that multiple phases, such as oil, gas, water, or even impurities, are involved [3]. Therefore, when oil and gas energy is extracted and transported, there are usually several situations, such as a two-phase flow of gas–liquid and liquid–liquid and a three-phase flow of gas–liquid(oil)–liquid(water). Sometimes when solid impurities are contained, there will be a two-phase flow of gas–solid and liquid–solid, and so on. Among them, gas–liquid two-phase flow is a common fluid flow state in subsea oil and gas transportation [4]. In deep sea waters, the seabed usually has complex landforms. If rich oil and gas resources are explored in this water area, the pipe shape of the mixed transportation pipeline laid on the seabed will be forced to present a different style [5]. The most common pipe types are horizontal pipe [6], riser [7], and inclined pipe [8]. However, regardless of the pipe type, when the two-phase flow of gas–liquid flows in the pipeline, and the gas flow velocity and liquid flow velocity are just in a certain range, slug flow may occur [9]. Especially for the riser, when the air flow is large, slug flow occurs easily. The slug flow is the liquid slug and the gas slug in the pipeline that flows in a way that alternately fills the pipeline [10]. Sometimes when slug flow occurs, a large number of dense bubbles in the pipe may be observed [11].

However, slug flow is harmful to pipelines [12]. The gas slug will have high pressure, and a gushing phenomenon will occur at a certain position in the pipeline, and the pressure will be released instantly, which may lead to much vibration in the pipe together with the impact of the liquid slug [13]. The high pressure will also affect the fatigue degree of the pipeline [14]. Dense bubbles are attached to the pipe wall, and a large area of rupture often occurs. The impact of bursting bubbles will also cause corrosion to the pipeline, and the pipeline will be damaged. Therefore, to strengthen the monitoring of slug flow in pipes and optimize pipeline design, it is necessary to develop a real-time detection technology for slug flow.

1.1. Related Studies

The generation and evolution of slug flow are very complicated processes [15]. Many studies have been conducted to explore the flow mechanism and evolution process of slug flow [16,17]. The research methods are roughly divided into experiments [18,19], simulations [20,21], and their combination [22]. Both experiment and simulation are important ways to gain a deep understanding of the behavior and features of slug flow. Certain conditions are required for the formation and development of slug flow. Deendarlianto et al. [23] and Dinaryanto et al. [24] carried out slug flow experiments in horizontal transparent pipes, based on which the initiation mechanism of slug flow is investigated, and the effect of the superficial velocities of gas and liquid on the transition of slug flow is analyzed. In addition, the slug flow transition mechanism is also an important research entry point. Zhang et al. [25] studied the upward slug flow transition mechanism in a vertical pipe to obtain the influence mechanism of the size and shape of the bubble on the bubble flow rate ratio. Based on this, a new transition criterion from bubble flow to slug flow is established, which provides a strong reference for the cognition of slug flow. The features of the fluids in the process of oil and gas development are very complex. In addition to the superficial velocity of each phase fluid, the factors that have a greater impact on slug flow also have viscosity. Naidek et al. [26] focused on exploring the correlation between bubble velocity, slug frequency, slug length, and liquid viscosity. Cao et al. [27] used the relevant data from the elbow collected in the experiment as the basic data of the numerical simulation. The flow features of the slug flow at the pipe elbow are studied by comparing the experiment with the numerical simulation. Schmelter et al. [28] performed numerical simulations using data on the flow velocities of each phase in six large horizontal pipes. The effectiveness of the CFD model in the numerical prediction of slug flow is verified.

Nowadays, computer vision technology has developed to be more and more powerful and intelligent and has been widely used in various industries [29,30]. It can replace people's subjective consciousness and reduce labor costs to achieve special needs in different applications of visual scenarios. In the field of computer vision, image processing techniques and object detection algorithms represent two crucial branches of technology that have found extensive applications. Wang et al. [31] employed traditional image processing techniques, such as the Canny operator and Douglas–Peucker algorithm to accomplish shape detection and attribute recognition of contaminants within maritime hydraulic oil. Huang et al. [32] applied recursive filtering and MSRCR image enhancement algorithms for preprocessing foreign object images in coal mining transportation. Subsequently, they deployed MSAM-Yolov5 on the NVIDIA Jetson Xavier NX platform to achieve real-time detection of large foreign objects on coal mine conveyor belts. Zhao et al. [33] changed the backbone part of Yolov7 to GhostNet, the neck part was replaced by BiFPN, the CBAM attention module was introduced into the model, and LW-Yolov7 was constructed to realize the lightweight detection of corn seedlings. Slug flow identification is an important part of gas–liquid two-phase flow pattern identification technology. Previously, many scholars have conducted relevant research on the intelligent identification method of gas–liquid two-phase flow [34,35]. Image data can fully reflect the features of slug flow. Common types of slug flow images are photographic images from high-speed cameras and 2D or 3D images reconstructed using tomography equipment. Flow pattern images are used as basic

data by many researchers to identify flow patterns using deep learning or machine learning methods. The identification of slug flow is included in these studies. Xu et al. [36] combined the two methods of machine learning and deep learning. The classification module in ResNet50 is replaced with a Support Vector Machine (SVM). Efficient classification and identification of gas–liquid two-phase flow patterns are realized. The feature extraction ability of deep neural networks and the classification ability of SVM are fully utilized in this method. Qiao et al. [37] improved ResNet by embedding CBAM and ECA attention modules into ResNet50. The recognition performance of the gas–liquid two-phase flow regime is improved. Nie et al. [38] applied four classic convolutional neural network (CNN) models to the flow pattern recognition. And, the recognition performance of different models is analyzed. The feasibility of the CNN model to replace the subjective judgment of the human eye for flow pattern recognition was verified. Banasiak et al. [39] reconstructed the flow state with 3D electrical capacitance tomography (ECT) and then used two algorithms to classify and identify the reconstructed images, namely SVM and FCM (Fuzzy C-Means). Shibata et al. [40] used the time strip method to disassemble and reorganize the transition process from bubble flow to slug flow into sequence images and then input them into CNN for training. Finally, the recognition of the transitional flow pattern was realized. Li et al. [41] used ECT technology to obtain flow pattern reconstruction images in the jumper and used CNN for training and classification. The CNN model they adopted is the advanced EfficientNet, which is very rare in flow pattern recognition methods. Moreover, after the optimizer is replaced by Adam, it is proved that the proposed model has stronger flow recognition performance than other classical models.

1.2. Innovative Contribution

On the one hand, for the method of using artificial intelligence (AI) computer vision technology to identify the flow pattern image, the classification model is used, i.e., the image is classified to judge the flow pattern. But, if the means of image acquisition are to use a camera to take pictures of the flow pattern, gas slugs in slug flow are likely to be blocked by other bubbles. The images captured in this case will be difficult to correctly identify in the classification model. On the other hand, the currently used models are relatively complex, with a large scale and a large number of parameters, making it difficult to carry out real-time identification and monitoring in the actual industrial production environment. Generally, both data and models need to be sent to the cloud for inference, and then the results are returned. The main contributions of this study can be summarized as follows:

- The Yolo object detection algorithm was first applied to the field of slug flow detection technology. High-precision detection of gas slug and dense bubbles in slug flow is realized. The problem where gas slug in slug flow could not be correctly identified due to being obscured by other bubbles has been resolved.
- The model with the strongest overall performance is deployed to the Jetson Nano embedded device. It is also the first time that it is based on a low-power embedded device and combined with object detection algorithms to detect slug flow in real-time. Then, TensorRT is used to accelerate the model inference process, and the real-time inference detection speed of the slug flow video is greatly increased. The detection capability of the embedded devices provides the basis for simultaneous real-time detection and related analysis of multiple locations.

The remainder of the paper is organized as follows: Section 2 introduces the algorithms, datasets, and operating conditions for model training deployment. The training results of the model and the running effect of the model deployed on the embedded device are analyzed and discussed in Section 3, and the study is concluded in Section 4.

2. Materials and Methods

2.1. Yolov3 and Yolov5 Algorithm

Currently, many object detection algorithms based on deep neural networks have been proposed [42] with the development of AI-related technologies. And, it has been applied to various fields, aiming to use the object detection algorithm in computer vision technology to solve related problems in various industries. Yolo (you only look once) is a classic object detection algorithm based on a deep neural network [43]. It does not have a fixed model, it has many series, such as Yolov1-v8, and in each series, there are different versions. The practical application value of the object detection algorithm is to accurately identify the object category in terms of image or video; as a result, the object location would be marked. The advantage of the Yolo series algorithm is that it not only has considerable identification efficiency and accuracy, but more importantly, it must have a high speed, especially the inference speed after the model is deployed, because Yolo is a one-stage object detection algorithm [44]. Good identification efficiency and high detection speed can meet some special requirements in different scenarios. For example, in some scenarios, real-time identification and detection of video stream data are required. Simpler models and higher identification and detection speed are also extremely important for deploying object detection algorithms to embedded devices because the hardware capabilities of embedded devices are limited compared to cloud servers and fixed computers [45]. In this study, two series of Yolo object detection algorithms, Yolov3 and Yolov5, are applied to the detection of slug flow. Both Yolov3 and Yolov5 were created and published by Ultralytics. So far, although the Yolo algorithms of the v3 and v5 series are not the latest, they are relatively mature and widely used.

Yolov3 is an improved object detection algorithm based on Yolov2, and the detection accuracy and speed have been improved [46]. In Yolov3, the pooling layer and the fully connected layer are removed. The purpose of the pooling layer is generally to downsample the feature map, but in Yolov3, a convolutional layer with a step size of 2 is used instead of the pooling layer. The purpose is to effectively prevent the loss of feature information caused by the pooling layer. It means Yolov3 in the overall model structure, except for some necessary upsampling layers and neck layers, is using convolutional layers for feature extraction and feature integration. The DarkNet-53 network is adopted in the backbone part of the basic Yolov3. The DarkNet-53 network borrows the idea of the ResNet [47] network, and the residual connection structure is introduced [48]. On the side of the feature extraction process, a data connection route is reintroduced, and the original feature information is directly connected to the next layer to prevent the problems of gradient disappearance and gradient explosion caused by deeper neural networks. To perform feature extraction more efficiently [49], Yolov3 also has other derivative versions, such as Yolov3-tiny and Yolov3-spp. Yolov3-tiny is different from the classic Yolov3 in that it continues to use the maximum pooling layer for downsampling; the purpose is to discard the identification accuracy to reduce the complexity of the model, thereby improving the identification speed. As with Yolov3-spp, a SPP (spatial pyramid pooling) module is added in the neck part. The SPP module is a special module composed of four parallel branches, which contain three max-pooling layers and a residual connection structure [50].

The overall architecture of Yolov5 still uses the network structure of the previous Yolo model, including Input, Backbone, Neck, and Prediction [51]. The data are preprocessed and fed into the neural network in the Input section. Performing deep feature extraction on images in Backbone, the bidirectional structure of FPN + PAN is adopted in the Neck part of Yolov5 to transfer and integrate feature semantic information of different scale layers [52]. Mosaic data augmentation technology, adaptive anchor technology, and adaptive image scaling technology are used in Yolov5. The common purpose is to improve the detection effect of Yolov5 on objects so that the model can have stronger adaptability when facing different application scenarios [53]. In Yolov5, the loss function used to measure the prediction difference of the model and drive the weight update is composed of three loss functions, namely classification loss, localization loss, and confidence loss [54]. In this study,

the Yolov5 v6.0 version is used. Different from the previous Yolov5 version, the Focus module is canceled, and the convolutional layer with the Batchnormalization and SiLU activation function is used instead. The activation functions in other convolutional layers are all replaced by SiLU [55]. The original SPP module was removed and replaced by SPPF. The forward direction of the main body of the SPPF module is serial, which is different from the parallel branches of the SPP module. The calculation amount of the module is reduced, and the calculation speed is improved [56]. In the v6.0 version, Yolov5 is also divided into different model styles according to the complexity of the model and the number of modules, mainly including Yolov5n, Yolov5s, Yolov5m, Yolov5l, etc. The difference between them lies in the number of C3 modules and the change in the number of feature map channels during the convolutional layer operation. Figure 1 shows the Yolov5n model structure. Yolov5n has fewer parameters and has certain advantages in inference speed.

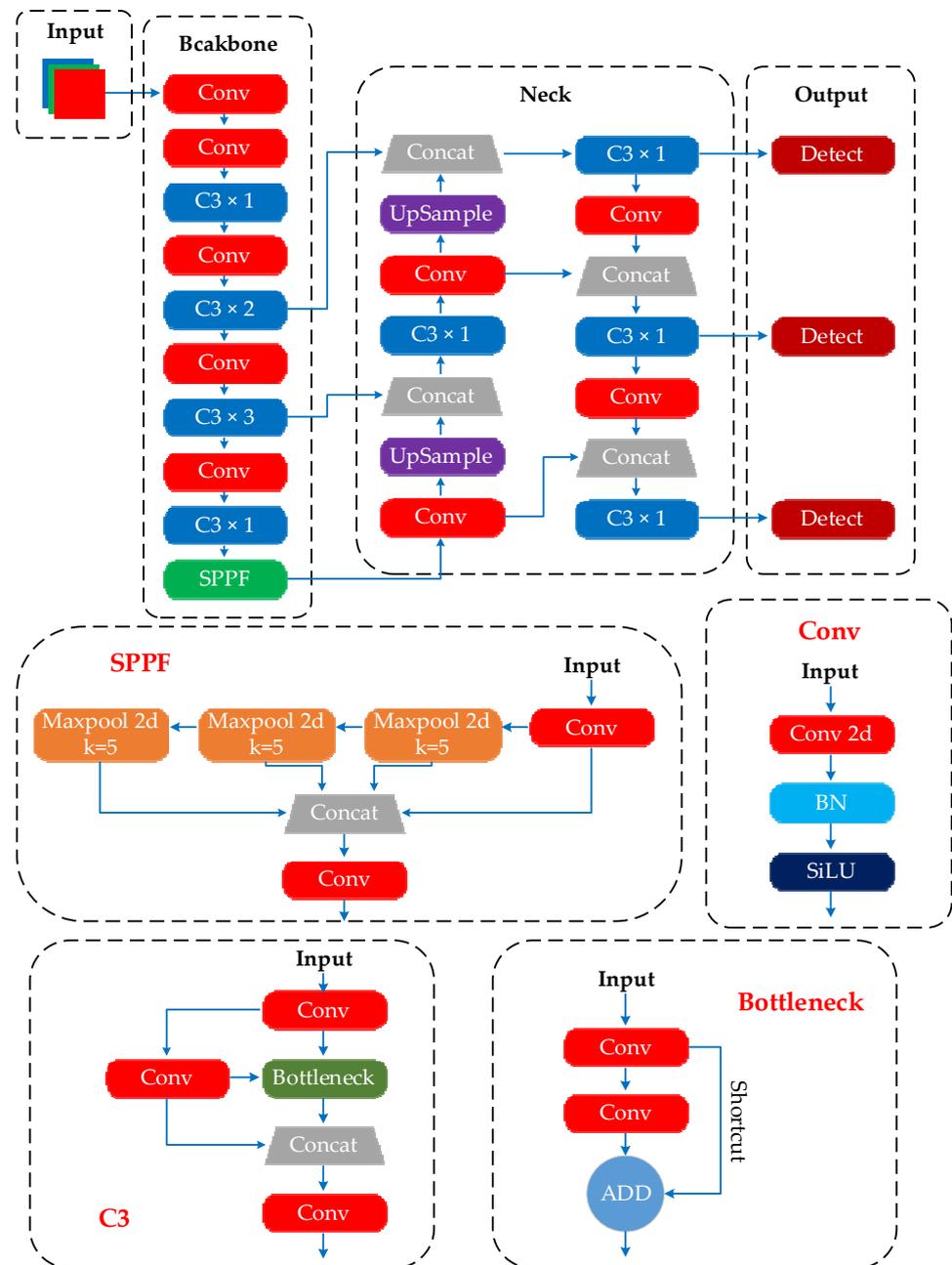


Figure 1. The overall structure of the Yolov5n model.

2.2. Data Acquisition and Preprocessing

2.2.1. Data Acquisition

To validate the feasibility of Yolov3 and Yolov5 object detection algorithms for slug flow detection, this study selected a public video of a gas–liquid two-phase flow in the riser [57]. The video is cut into images frame by frame, and valuable images are selected to build a dataset as the data for model training and validation. The dataset contains 733 images of slug flow in a vertical pipe. To mitigate the impact of human subjectivity on data balance, the approach of random sampling without replacement is employed. This involves selecting 70% of the image data for training purposes, while the remaining 30% is designated as the validation set. To make the features of the gas slug and dense bubbles more obvious in the slug flow image, the size of the image is cropped to $300 \times 750 \times 3$. There is no transparency property in the image. The channel dimension attributed to the image is RGB three-channel. Figure 2 shows part of the slug flow image data. When slug flow occurs, the sizable gas and liquid slugs alternate, leading to a rapid increase in pipeline pressure. This fluctuation in gas content can result in the emergence of numerous bubbles at times. Similarly, a series of problems, such as increased pressure drop, unstable flow, corrosion, and wear, can be caused by a large number of dense air bubbles. Therefore, it is of great engineering value to efficiently detect the slug flow, especially gas slug and dense bubbles.

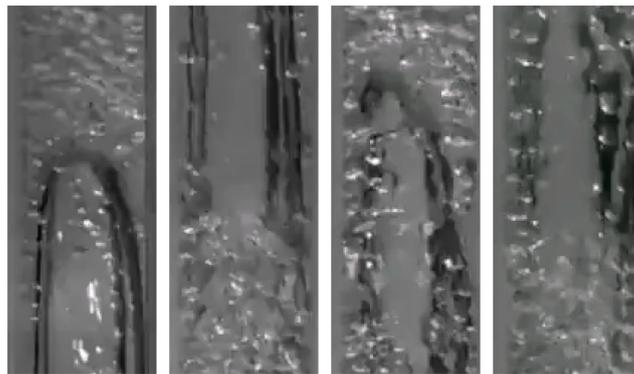


Figure 2. Slug flow in a vertical pipe.

2.2.2. Data Augmentation and Annotation

Mosaic data augmentation is a special form of data augmentation in Yolov5, which is different from traditional data augmentation methods [58]. First, the original images in the four datasets are randomly scaled, randomly cropped, etc., and then, the randomly generated new image data are spliced into a combined image in a random distribution manner. This combined image is used as data in the model input. In addition, the label and annotation information of the image will also be converted along with the operation in the data enhancement to correspond to the new combined image data. The size of the combined image is $320 \times 320 \times 3$. Data preprocessing using mosaic data augmentation technology has the following advantages:

- (1) Data diversity can be greatly enhanced. The model can learn more image data features from different perspectives. The generalization ability of the trained model is improved.
- (2) Mosaic data enhancement can simulate the occlusion and overlapping scenes between objects to a certain extent, which helps the model to solve the detection defects of occlusion and overlapping objects. It can adapt the model to more complex application scenarios and improve the stability of the model.
- (3) If the effectiveness of the trained models is roughly the same, mosaic data augmentation can be employed to decrease the dataset size because the data are more fully utilized. Similarly, if the original data remain unchanged and the mosaic data en-

hancement technology continues to be used, the occurrence of overfitting during the training process will be effectively suppressed, and the model performance can be further improved.

After the datasets were allocated, Labelme data labeling software was used to label the image data. Figure 3 shows an example of the Labelme software interface and data labeling process. Label information includes the coordinates information and object category. The red label indicates gas slug, and the green label indicates dense bubbles. After each image is annotated, the annotated information will be saved in JSON file format. The annotation files are then converted into the standard COCO data format, corresponding to the image data. The dataset and label set are fully constructed. Table 1 shows the quantity of images within the dataset and the number of labeled instances.

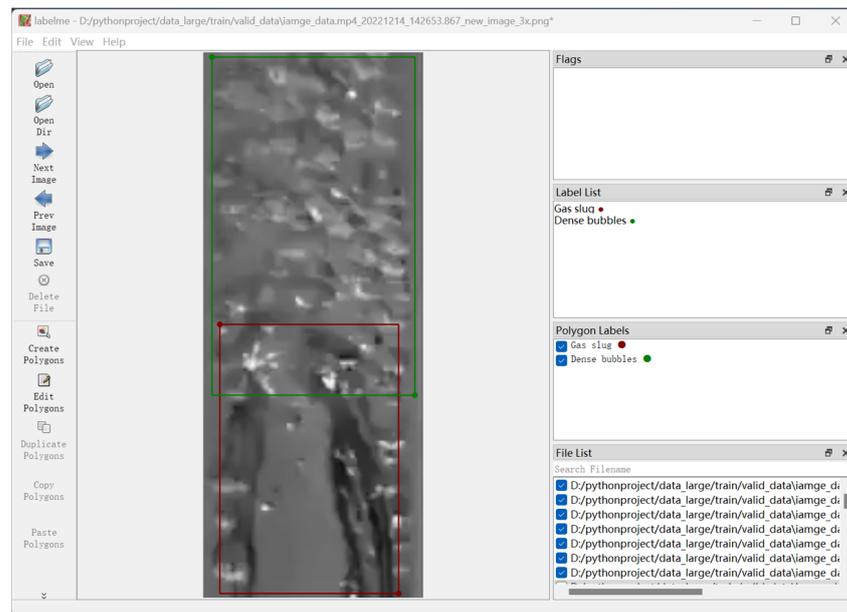


Figure 3. Using Labelme software for data annotation.

Table 1. Images and instances quantity in the dataset.

Dataset	Number of Images	Number of Gas Slug Instances	Number of Dense Bubbles Instances
Training	513	520	573
Validation	220	228	200
Total	733	748	773

Better training data distribution and instance labels are the prerequisites for the model to have stronger inference accuracy. Figure 4 shows the visualization results of data instances in the training set, including the number of instances, size ratio, location distribution, and size distribution. From Figure 4a, it can be concluded that the number of instances of gas slug and dense bubbles in the training set is roughly balanced, and the data labels quantity is even. Figure 4b shows that the proportion of label sizes of instances is also average. Figure 4c,d shows the center location and size distribution of instances. The darker the color, the more frequently the represented instance occurs. Since the rectangular shape of the pipe defines where the objects to be detected are located, the locations and size distributions of instances are relatively concentrated. However, in the special application scenario of a gas–liquid two-phase flow-mixed transportation, it can also meet the detection requirements.

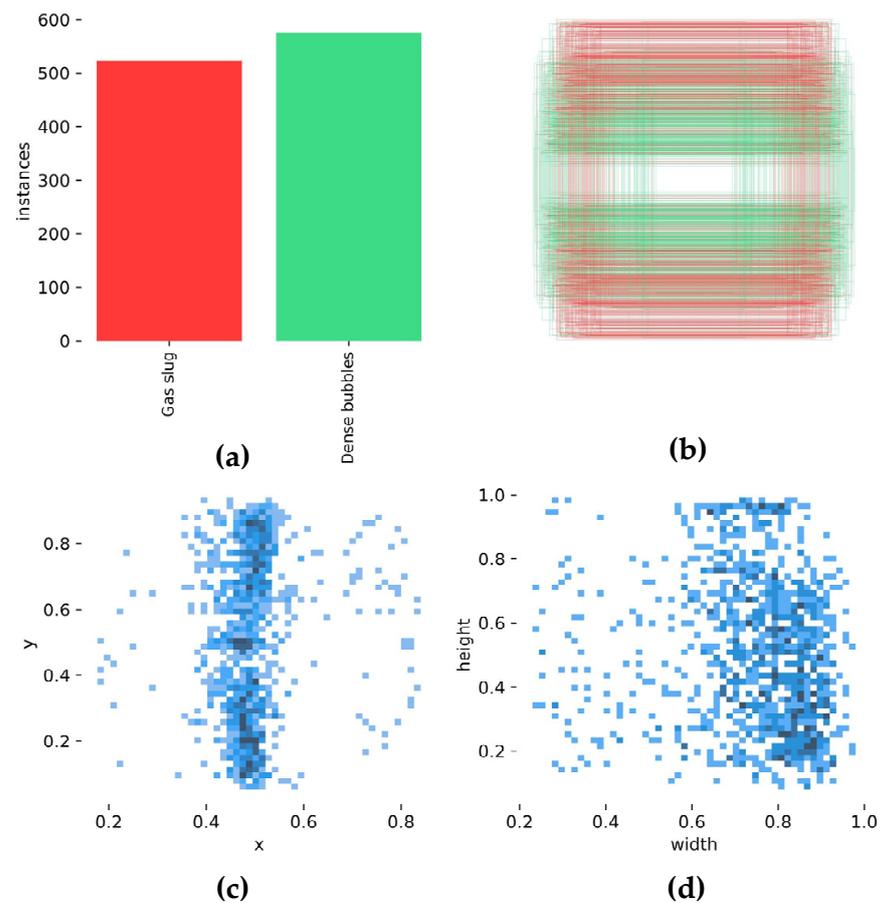


Figure 4. Instances of the training dataset: (a) The number of instances; (b) the labeling ratio of instances; (c) the location distribution of instances; and (d) the size distribution of instances.

2.3. Model Training and Deployment

2.3.1. Training Environment and Parameter Settings

The model training process in this research was carried out in the Windows 11 operating system. The running memory of the operating system is 16 G. To accelerate the Yolo model training process, we formulated the GPU to accelerate the calculation of the neural network. The GPU model is the NVIDIA GeForce RTX3050 laptop GPU, VRAM 4 G. The type and model of the computer language interpreter are Python 3.9 (Python Software Foundation, Amsterdam, The Netherlands), and the deep learning framework is PyTorch (FAIR, Palo Alto, CA, USA), version 1.10.0.

In this study, seven Yolo models were used for training and validation on the slug flow image dataset, including Yolov3, Yolov3-tiny, Yolov3-spp, Yolov5n, Yolov5s, Yolov5m, and Yolov5l. The transfer learning methodology is applied, wherein the weight parameters of a pre-trained model, trained on the COCO dataset, serve as the initial parameters for training in this study. Table 2 shows the setting of key parameters during the training process. The optimizer uses SGD. To stabilize the weight parameter update during model training, the learning rate decay strategy of cosine annealing was used in this study. The initial learning rate was set to 0.001 and the overall decay rate was 0.01. Momentum was set to 0.937, and the weight decay was set to 0.0005; the purpose was to prevent the occurrence of overfitting. Under the capability of the GPU, the batch size was 8. For the model to be fully trained, the max epoch was set to 200.

Table 2. The configuration of primary training parameters.

	Parameters
Optimizer	SGD
Momentum	0.937
Weight decay	0.0005
Initial learning rate	0.001
Decay rate	0.01
Image size	320
Batch size	8
Max epoch	200

2.3.2. Model Performance Evaluation Metrics

The detection performance and robustness of the model need to be judged by important evaluation indicators. Moreover, it is more convincing to evaluate different models with the same indicators. The comprehensive performance of several models was compared using object detection algorithms, which were more commonly used and the main evaluation criteria in this study. Evaluation metrics included precision, recall, average precision (*AP*), $mAP_{0.5}$, and $mAP_{0.5:0.95}$. mAP is an average metric commonly used by object detection models to evaluate detection precision. To be more comprehensive when comparing model performance, an evaluation metric comprehensive mAP ($cmAP$) was added in this study, and different weights were assigned to $mAP_{0.5}$ and $mAP_{0.5:0.95}$ to obtain a comprehensive evaluation index. Among them, when the intersection over union (*IoU*) is 0.5, the average precision under different recalls is $mAP_{0.5}$, and *IoU* means the overlap ratio between the boundary box of the prediction (*pr*) and the ground truth (*gt*). The related expressions are defined as:

$$IoU = \frac{Area_{pr} \cap Area_{gt}}{Area_{pr} \cup Area_{gt}} \tag{1}$$

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

$$AP = \int_0^1 Precision \times Recall dr \tag{4}$$

$$mAP = \frac{\sum_1^C AP}{C} \tag{5}$$

$$mAP_{0.5:0.95} = \frac{1}{10}(mAP_{0.5} + mAP_{0.55} + \dots + mAP_{0.9} + mAP_{0.95}) \tag{6}$$

$$cmAP = 0.1 \times mAP_{0.5} + 0.9 \times mAP_{0.5:0.95} \tag{7}$$

where *TP* is the true positives, *FP* is the false positives, *FN* is the false negatives, and *C* is the number of categories.

2.3.3. Embedded Deployment and Operating Environment

Although personal computers (PC) or various servers have powerful computing capabilities and can quickly infer and detect image data or video stream data in neural networks, devices, such as PC and servers, are less mobile and not suitable for being developed into a product. If the data are sent to the cloud for processing and calculation, the processing time will be lengthened, and it cannot be applied to special industrial scenarios that require real-time detection, such as video surveillance detection, lane line detection in driverless driving, etc. During the flow pattern detection procedure within the submarine pipeline, and because the fluid flow is sometimes very fast, the detection method with a high frame rate and real-time detection are very important. In this study, the selected detection models were compared for comprehensive performance, and the

best model was deployed to the Jetson Nano embedded device to develop a mobile and real-time slug flow detection method. Jetson Nano is an AI-embedded edge computing development device [59]. Figure 5 and Table 3 show the basic hardware composition and hardware model of Jetson Nano, respectively.

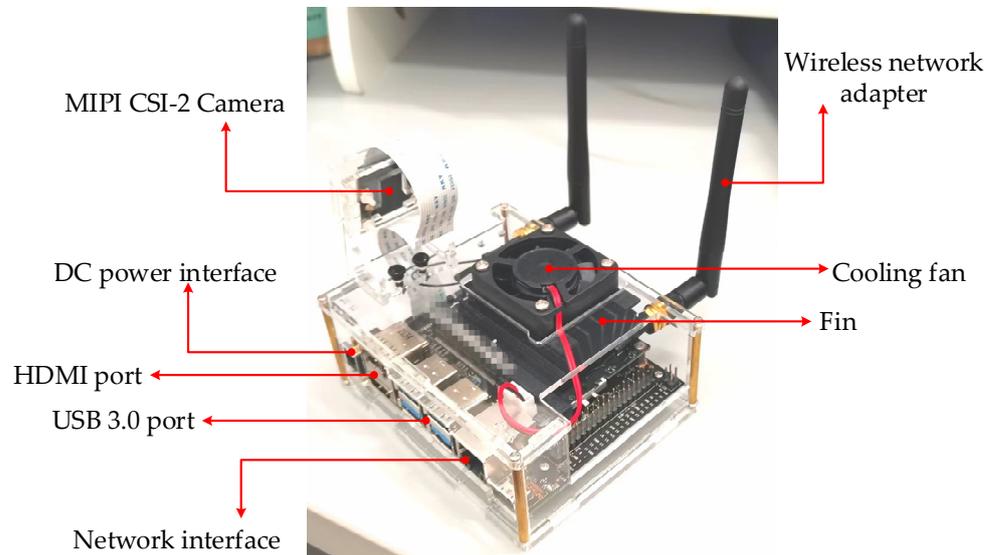


Figure 5. Jetson Nano embedded device.

Table 3. Jetson Nano hardware types.

Hardware Names	Specifications
CPU	Quad-Core ARM® Cortex®-A57 MPCore Processor
GPU	NVIDIA Maxwell™ architecture with 128 NVIDIA CUDA® cores
Memory	4 GB 64-bit LPDDR4

In order to facilitate the deployment of corresponding inference models on Jetson Nano, the operating system is Ubuntu 18.04, and the deep learning combination development toolkit is Jetpack 4.6.1 launched by NVIDIA. The interpreter is Python 3.6.9, the deep learning framework is PyTorch 1.8.0, Torchvision 0.9.0. Graphics-accelerated computing framework is CUDA 10.2, cuDNN 8.2. Figure 6 shows the main flow of the deployment of the slug detection model. Using the prepared dataset, with the pre-training model, the model is trained on a high-performance computer to obtain a new model for slug flow detection. The new model is then deployed to the Jetson Nano embedded device. The TensorRT acceleration tool is used to accelerate the model inference process, and the version of TensorRT is 8.2.1. Finally, high-speed and high-precision detection of slug flow is realized.

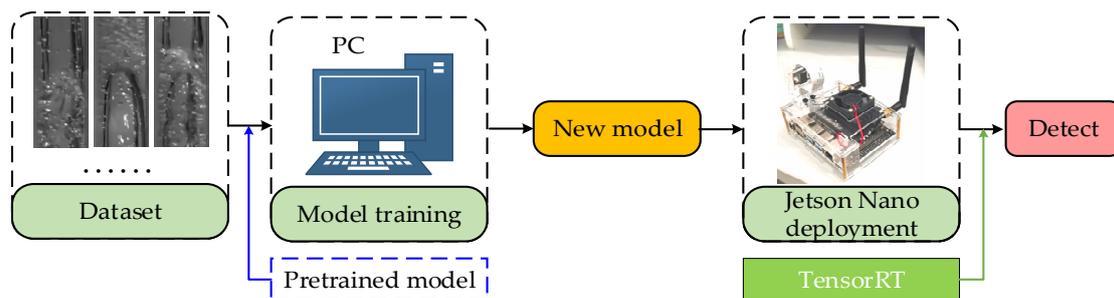


Figure 6. Embedded deployment process.

3. Results and Discussion

3.1. Training Results Analysis

In the process of training different models, after each epoch training, the model's performance will be assessed using the validation set data. Strictly, the validation set data has never been encountered by the model, so the performance indicators obtained after inferring and testing the validation set are convincing. The inference results of the validation set also reflect the model's capacity for generalization. The batch size is set to 8. Figure 7 shows the pattern of a batch of data in the training set when it is input to the model. Due to the use of mosaic data enhancement technology for image data, each datum in the batch is not a separate image, but a spliced image after image processing. And, the annotation information will also be scaled and spliced to ensure that the instance coordinates correspond to the image size. The objective remains centered on enhancing the model's robust generalization capability. In the model validation phase, the training of the model and the update of the weight parameters will not be affected by the setting of the batch size. So, in order to speed up the model validation, the batch size of the validation set is set to 16. Figure 8 shows the validation process of the Yolov5n model for a batch of data in the validation set. Table 4 and Figure 9 clearly show the detection effect of different versions of the Yolov5 model on gas slug and dense bubbles instances in the validation set. The highest detection precision of the gas slug and dense bubbles is 97.9% and 86.3%, respectively, with both using the Yolov5s model. Yolov5m has the highest detection cmAP for the gas slug, reaching 73.46%. On the contrary, Yolov5s is the lowest at 66.87%. For dense bubbles, Yolov5l's cmAP is the highest, reaching 65.57%, and Yolov5s is the lowest, at 62.50%. In general, the detection effect of the gas slug is better than that of dense bubbles. The reason is that in image data the features of the gas slug are more obvious than dense bubbles. And, when detecting slug flow, it is most important to detect gas slug accurately and efficiently. But, for one of the categories, the difference in cmAP of the four models is not large, which means that the average detection accuracy is similar. To compare the comprehensive performance of different models, we continued to compare the overall average detection performance of the models, including the indicators of precision, recall, and mAP. The complexity of the model was also analyzed.

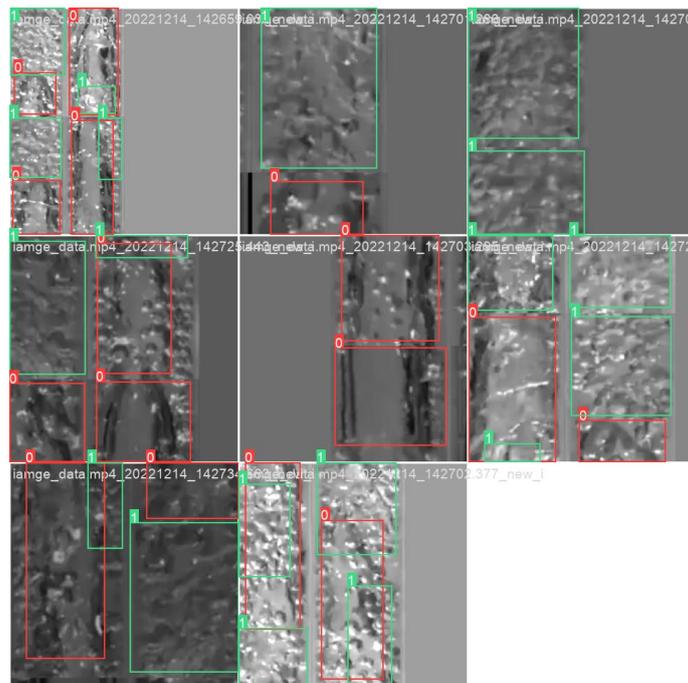


Figure 7. Batch data in the training dataset.

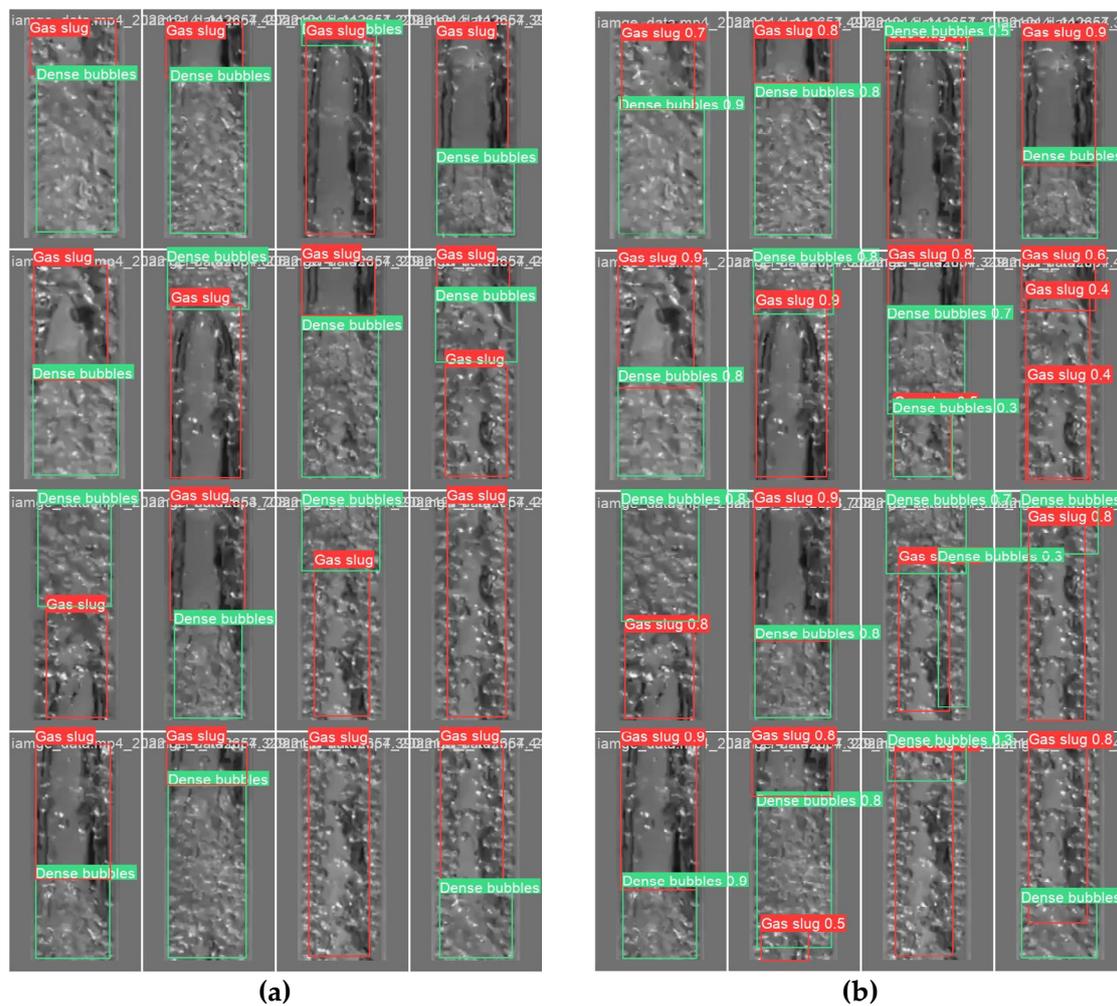


Figure 8. Validation process of a batch in the validation dataset: (a) labels; (b) validation results.

Table 4. Detailed indicators for different classes.

Class	Yolov5n			Yolov5s			Yolov5m			Yolov5l		
	Precision/ %	Recall/ %	cmAP/ %									
Gas slug	93.1	93.9	71.83	97.9	90.8	66.87	95.3	93.4	73.46	95.3	93.9	72.95
Dense bubbles	82.1	86.5	63.95	86.3	84.9	62.50	83.3	83.5	64.85	84.4	89.3	65.57

Table 5 shows the overall performance of Yolov5’s related models for the validation set data. The cmAP of Yolov5l is the highest, reaching 69.25%. However, compared with other models, the model complexity of Yolov5l is also the highest, with a parameter amount of 46113663 and Flops of 107.7 G. The performance of embedded devices is often inferior to that of fixed computers, so if such a complex and computationally intensive model is deployed to the device, the inference speed will be very slow, which is not conducive to the realization of applications in special scenarios. The cmAP of Yolov5n is 67.94%, which is only 1.31% lower than Yolov5l, and the precision and recall are only 2.2% and 1.4% lower than Yolov5l, respectively. However, the parameter amount and Flops of Yolov5n are only 3.8% of Yolov5l, which are 1761871 and 4.1G, respectively. Among the Yolov5 comparison models, the model complexity of Yolov5n is the lowest. Although the cmAP is a little lower compared with Yolov5m and Yolov5l, the difference of about 1% frequently does not significantly affect the model’s performance. For practical applications, it can also perform very well. Therefore, in the related comparison model of Yolov5, Yolov5n

has the most outstanding comprehensive performance. Figure 10a shows the change curves of some important indicators of the Yolov5n model during the training process, such as loss, precision, recall, and mAP. All loss indicators are gradually decreasing, and precision, recall, and mAP are gradually increasing. All metrics tend to stabilize in the later stages of training. There is no over-fitting phenomenon in the training process, and the model's capacity for generalization is assured. Precision and recall are a set of contradictory indicator parameters. Figure 10b shows the precision–recall (PR) curve. It reflects the relationship between precision and recall of the model with the best overall performance during the training process when the IoU threshold is 0.5. The area bounded by the PR curve and the *x*-axis and *y*-axis is $mAP_{0.5}$, with a value of 0.935.

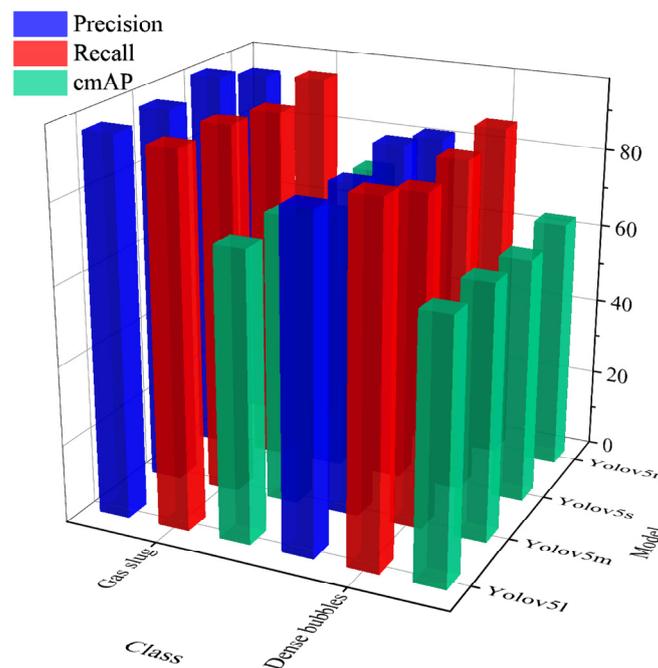


Figure 9. Yolov5 detection result indicators.

Table 5. Overall indicators comparison of the different models of Yolov5.

Model	Parameters	Flops/G	Precision/%	Recall/%	$mAP_{0.5}/%$	$mAP_{0.5:0.95}/%$	cmAP/%
Yolov5n	1,761,871	4.1	87.6	90.2	93.5	65.1	67.94
Yolov5s	7,015,519	15.8	92.1	87.9	94.2	61.4	64.68
Yolov5m	20,856,975	47.9	89.3	88.5	93.9	66.4	69.15
Yolov5l	46,113,663	107.7	89.8	91.6	94.1	66.5	69.25

Figure 11 illustrates the inference detection outcomes obtained from the Yolov5 series model applied to certain data. Yolov5s and Yolov5l are capable of detecting dense bubbles located along the edges of the image. The detection effect is good. The most important gas slug part can be detected by all models. But sometimes, the gas slug will be blocked by bubbles. Solely using a computer vision classification model makes it challenging to accurately identify the gas slug. Therefore, the strength of the object detection algorithm is that it can still exert a strong performance on partially obstructed objects. Yolov5l can completely detect the gas slug hidden behind the bubbles. Although other models do not fully detect a small part of the gas slug, most of the gas slug is also detected when it is obstructed by bubbles. Taking the image data in a validation set as an example, Figure 12 shows the feature map changes in Yolov5n in the process of inferring and detecting images. Each layer in the network is playing an important role in feature extraction, and all have the nature of learning and parameter updating. The deeper the feature extraction layer, the more abstract the obtained feature map, indicating that Yolov5n has strong feature

extraction and integration capabilities, providing a solid foundation for the work of the detection head. The input image consists of three channels ($c = 3$), representing the RGB three channels, respectively. In the first two Conv parts, the number of channels of the feature map is changed from 16 to 32, and the feature map will carry more channel feature information. With the process of feature extraction and feature integration, as the network deepens, the size of the feature map progressively decreases, which makes the receptive field of the feature map larger and larger. The 23rd layer is the last layer of the overall network, with the largest receptive field, and the corresponding detect head will be more sensitive to big objects. On the contrary, the depth of the 17th layer is lower, the receptive field is smaller, and the detect head will be more sensitive to small objects. Similarly, the detect head pair corresponding to the 20th layer is more focused on objects of medium size.

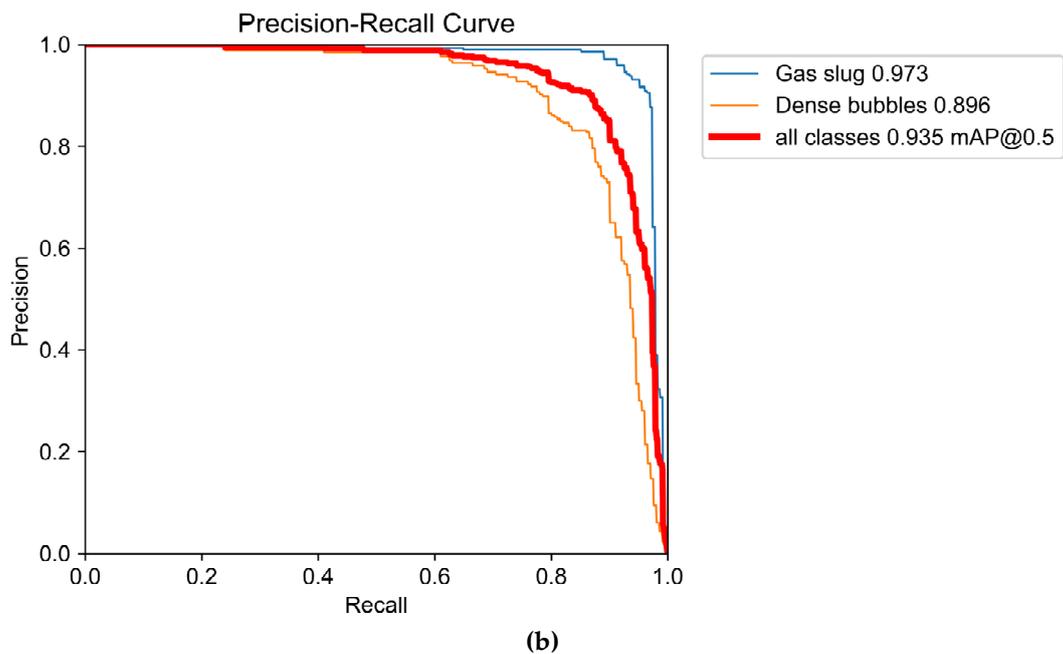
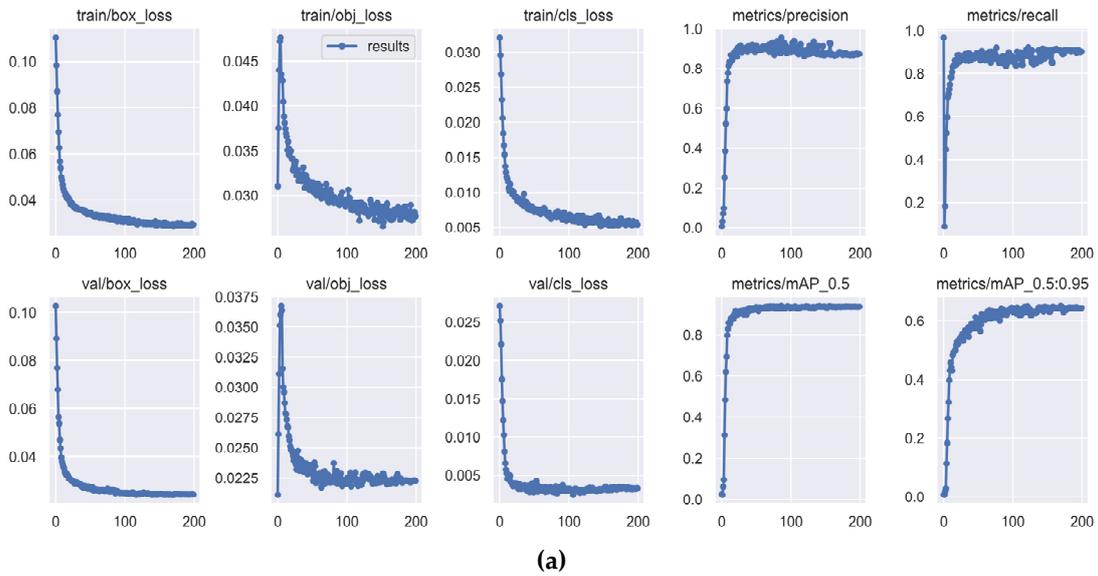


Figure 10. Change curve of YOLOv5n-related indicators: (a) training and validation curve; (b) precision–recall curve.

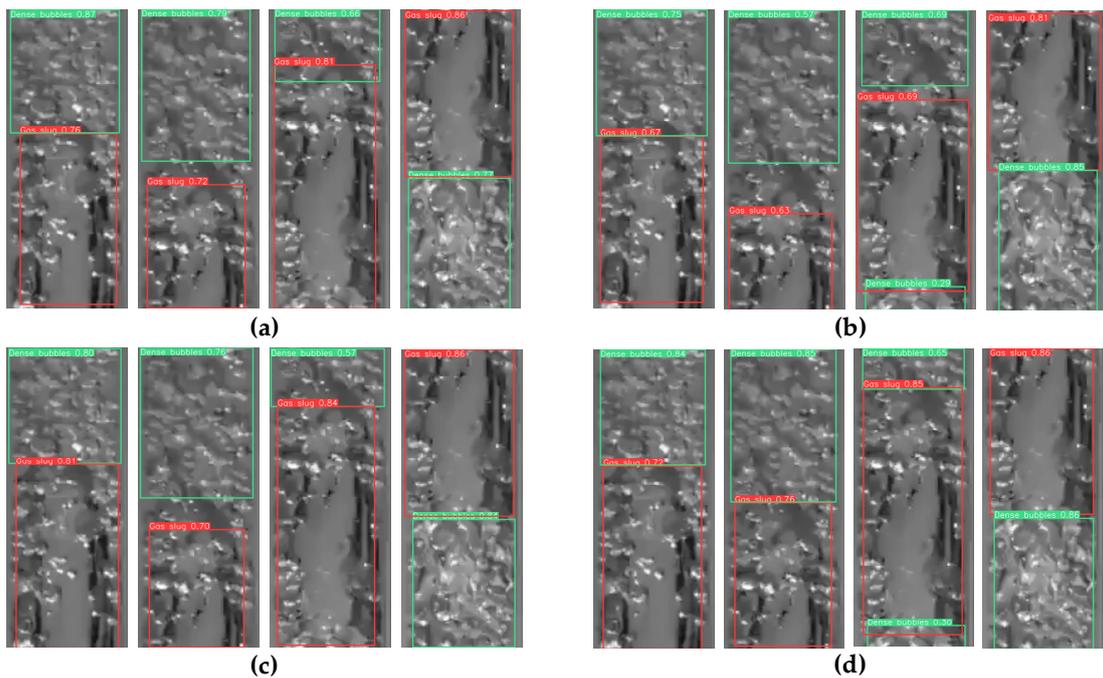


Figure 11. Detection results of Yolov5: (a) Yolov5n; (b) Yolov5s; (c) Yolov5m; (d) Yolov5l.

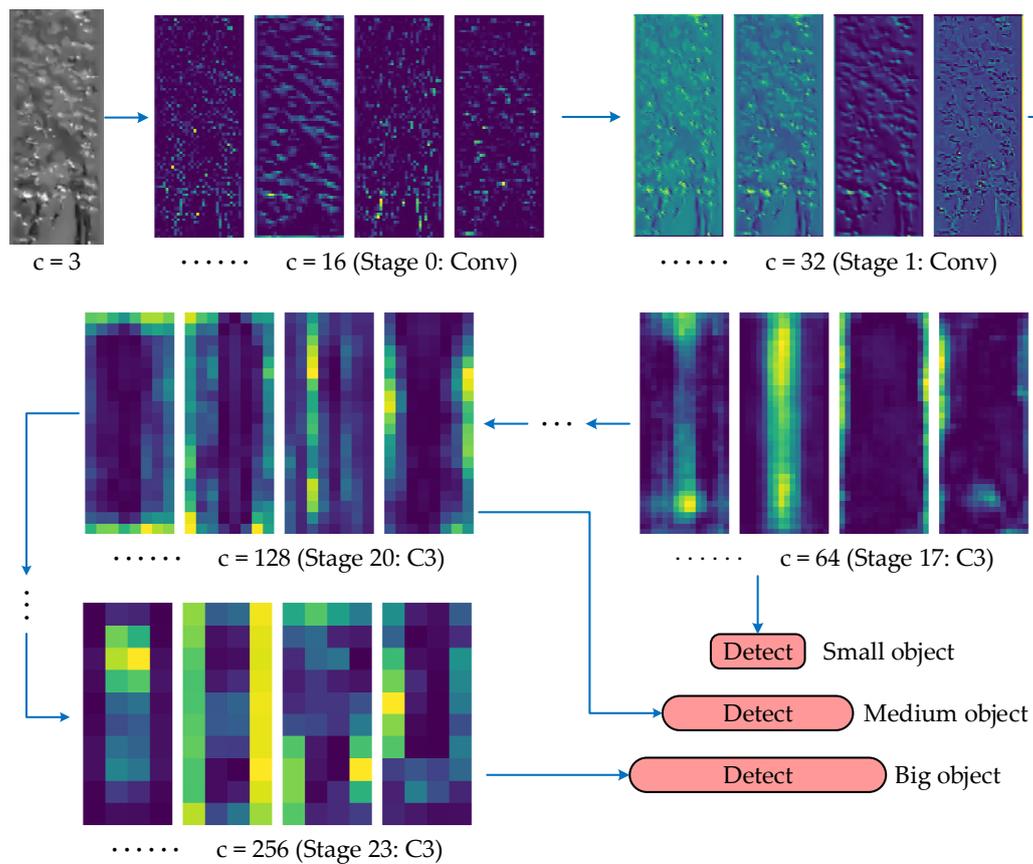


Figure 12. Visualization of feature maps in the Yolov5n inference process.

Similarly, the Yolov3 series of models also includes lightweight models and high-precision models. Table 6 shows the overall performance of different models in the Yolov3 series for inference on the validation set data. Figure 13 shows the detection results of

Yolov3 series models on slug flow images. In Yolov3, the lightest is Yolov3-tiny, with only 8,669,002 parameters. The volume of Flops is 12.9 G, about 80% of Yolov5s. Although the model of Yolov3-tiny is small, the overall performance of Yolov3-tiny is the worst among the models of the Yolov5 and Yolov3 series, $mAP_{0.5}$ and $mAP_{0.5:0.95}$ are the lowest, and $cmAP$ is only 62.52%. The $cmAP$ of Yolov3 and Yolov3-spp is 68.27% and 67.47%, respectively, which are basically the same as the performance of Yolov5n. However, the parameters of Yolov3 and Yolov3-spp are 35 times that of Yolov5n, and Flops are 38 times that of Yolov5n. Therefore, due to the huge model size of Yolov3 and Yolov3-spp, the inference detection speed is very low. When the $cmAP$ is roughly equal, the comprehensive performance is low, and Yolov3 and Yolov3-spp are not suitable for being deployed in embedded devices for functional development. Figure 14 shows the overall detection performance metrics for Yolov3 and Yolov5. In general, the different models of the Yolov3 and Yolov5 series show little difference in the detection effect of slug flow, which is mainly reflected in the mAP index. But, from the perspective of model complexity, Yolov5n has the least number of model parameters and Flops and is the model with the strongest comprehensive performance. Yolov5n lays the model foundation for the functional development of embedded devices.

Table 6. Comparison of performance among different Yolov3 models.

Model	Parameters	Flops/G	Precision/%	Recall/%	$mAP_{0.5}/%$	$mAP_{0.5:0.95}/%$	$cmAP/%$
Yolov3	61,502,815	154.6	91.1	91.0	95.0	65.3	68.27
Yolov3-tiny	8,669,002	12.9	89.0	90.4	92.4	59.2	62.52
Yolov3-spp	62,551,903	155.4	91.7	90.2	94.2	64.5	67.47

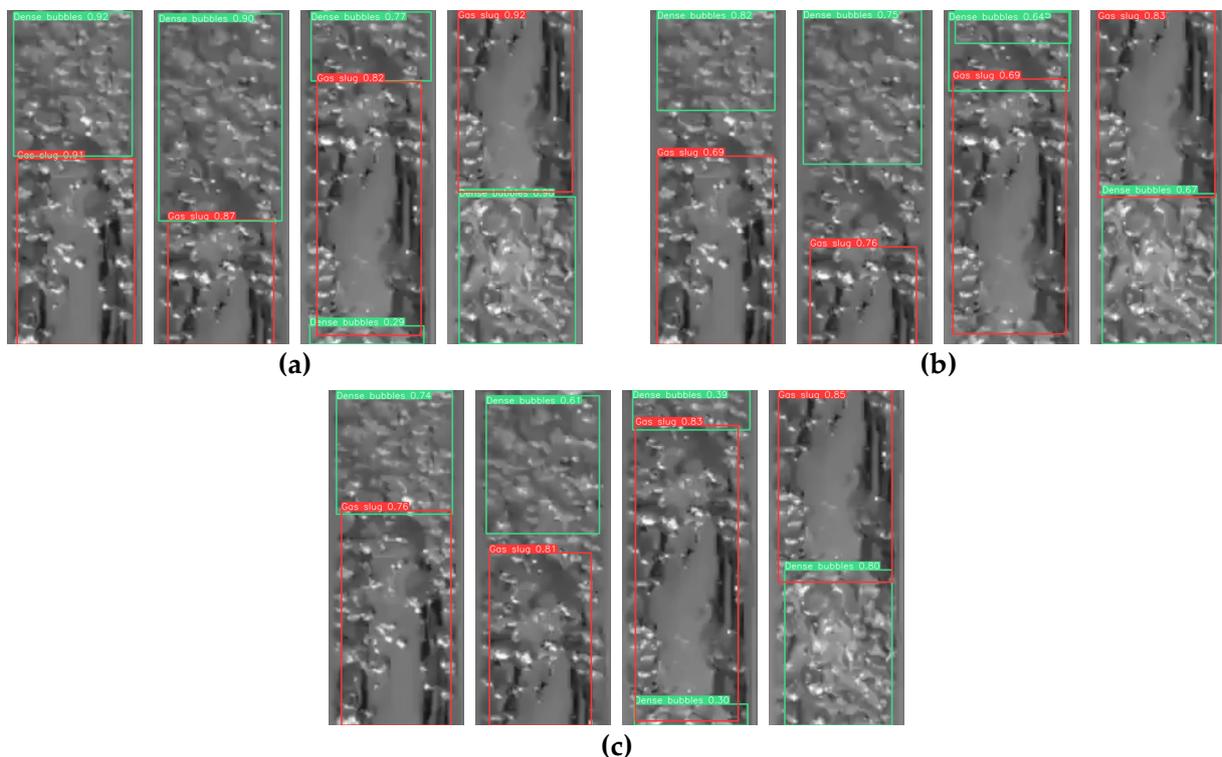


Figure 13. Detection results of Yolov3: (a) Yolov3; (b) Yolov3-tiny; (c) Yolov3-spp.

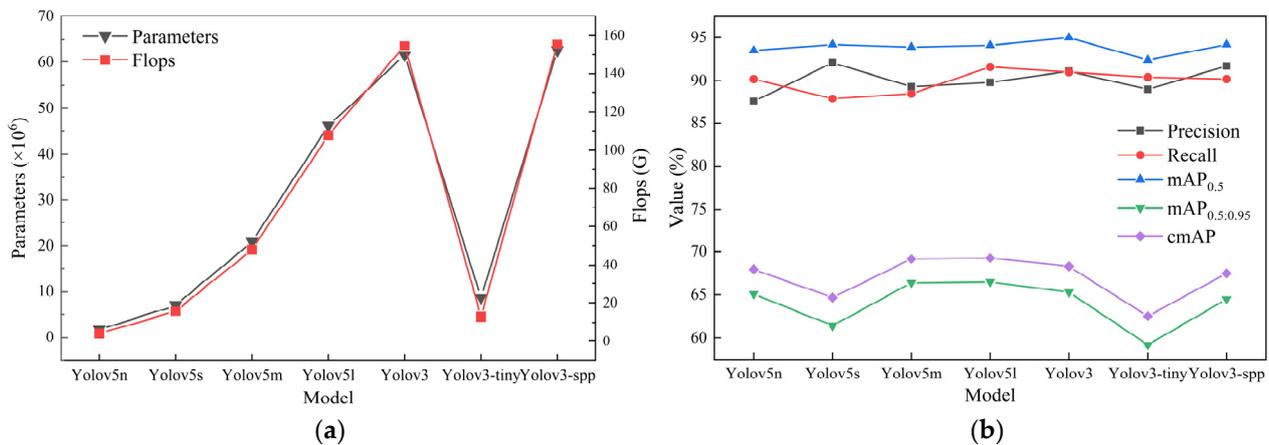


Figure 14. Overall detection performance indicators for YOLOv3 and YOLOv5: (a) comparing the complexity of the model; (b) model detection effect comparison.

Table 7 and Figure 15 provide comparison with previous research methods. The contrastive models encompass two one-stage object detection models: SSD and YOLOv5-bifpn. YOLOv5-bifpn is an enhancement over YOLOv5l, where the neck section is replaced with the bifpn structure. Additionally, the two two-stage object detection models are included: Faster R-CNN and Mask R-CNN. The backbone component for the SSD model is VGG16, whereas Faster R-CNN and Mask R-CNN employ ResNet50 as their backbone. Both VGG16 and ResNet50 are well-established deep convolutional neural networks renowned for their exceptional feature extraction capabilities. Analyzing the comparative results from Table 7 reveals that the cmAP for SSD and Mask R-CNN surpasses that of YOLOv5n, achieving 68.05% and 68.36%, respectively. Furthermore, the mAP_{0.5} and mAP_{0.5:0.95} metrics also outperform YOLOv5n due to the robust feature extraction and integration capabilities of SSD and Mask R-CNN. For Faster R-CNN, mAP_{0.5} is higher than YOLOv5n, reaching 93.8%. But, mAP_{0.5:0.95} is slightly lower than YOLOv5n at 65.1%. This led to a decrease in cmAP as well, which was 0.06% lower than YOLOv5n. For YOLOv5-bifpn, the primary differences are observed in mAP_{0.5:0.95} and cmAP, both of which are lower compared to YOLOv5n by 1.2% and 1.07%, respectively. When comparing the mAP metric, the recognition performance of the five models does not exhibit significant discrepancies. Minor differences in average precision are unlikely to heavily impact practical detection outcomes. However, considering model complexity, the distinction among the five models is substantial. Apart from YOLOv5n, SSD possesses the smallest parameter count at 34.31M, approximately 19 times that of YOLOv5n. Excluding YOLOv5n, the model with the lowest Flops is Faster R-CNN at 20.7 G, roughly five times that of YOLOv5n. In summary, compared to models employed in previous research, YOLOv5n demonstrates similarly excellent detection precision. Furthermore, YOLOv5n holds a significant advantage in terms of model complexity and computational load. This lays a strong foundation for high-speed slug flow detection and embedded deployment.

Table 7. Comparison with previous studies.

Model	Parameters/M	Flops/G	mAP _{0.5} /%	mAP _{0.5:0.95} /%	cmAP/%
Yolov5n	1.76	4.1	93.5	65.1	67.94
SSD	34.31	38.56	93.7	65.2	68.05
Faster R-CNN	41.53	20.7	93.8	65.0	67.88
Mask R-CNN	43.75	71.8	94.1	65.5	68.36
Yolov5-bifpn	46.38	108.5	93.6	63.9	66.87

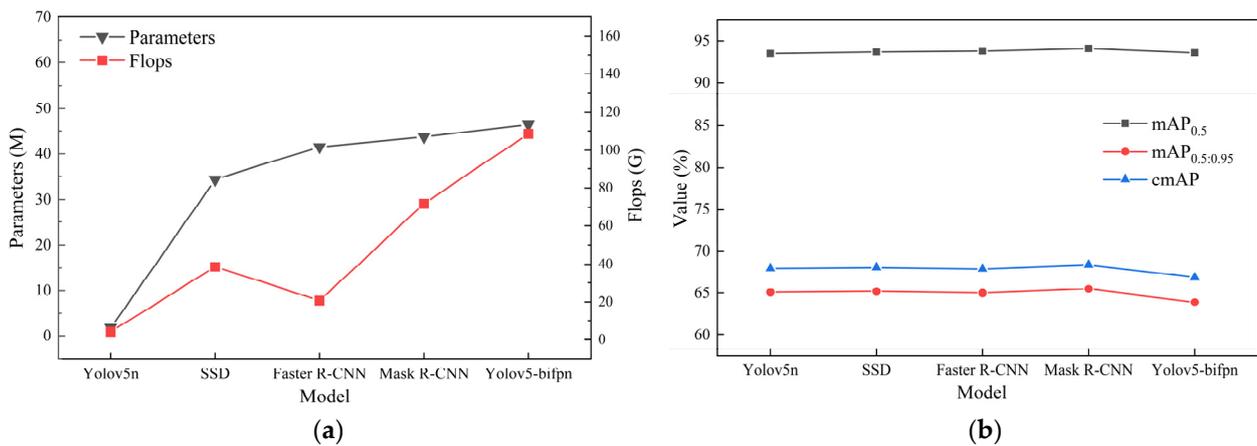


Figure 15. Performance comparison with previous methods: (a) comparing the complexity of the model; (b) model detection effect comparison.

3.2. Applicability and Limitation Analysis

3.2.1. Applicability Analysis

Mixed pipelines are often found in various environments, such as the seabed, where the brightness of light is subject to fluctuations. The features of the image are greatly influenced by the changes in brightness. There could potentially be variations in the object detection model’s performance when applied to images under different brightness conditions. Therefore, a correlation analysis on the applicability of the Yolov5n model with new weight parameters needs to be conducted. To simulate the changes in light brightness in the actual industrial production environment, brightness processing is performed on the validation set image data. The image data are subjected to brightness adjustments, with the brightness increased by a maximum of 45 units and decreased by a minimum of 75 units compared to the original brightness. The experimental conditions include eight groups, and the brightness scale for the comparison experiment is set at 15 units. The inference detection result of the updated Yolov5n model on the validation set image data under different brightness conditions is demonstrated in Figures 16 and 17 and Table 8. Within the brightness range from -45 to $+45$, the accurate detection of the position and overall size of the gas slug and even the detection of partially obstructed gas slug areas by bubbles is achieved using the Yolov5n model. Large areas of dense bubbles are also correctly detected, as depicted in Figure 16a–f. However, when the brightness is reduced by 60 or more units, the gas slug detection results deviate significantly from reality, with even the entire image being detected as a gas slug. The potential reason is that the features of the gas slug become blurred with a considerable decrease in brightness, resulting in confusing features similar to gas slugs formed by large dark areas in the image. In terms of result indices, the original image data demonstrate the most optimal detection performance. As the brightness gradually increases, there is a decrease in $mAP_{0.5}$, $mAP_{0.5:0.95}$, and $cmAP$ to varying degrees, while precision shows a gradual increase. Upon increasing the brightness by 45 units, the $cmAP$ is reduced by 3.37% to 64.57%, and precision increases by 3.2% to 90.8%. Conversely, as the brightness gradually decreases, all indicators start to decrease. The $cmAP$ remains above 60% until the brightness is reduced by 30 units, after which the model detection effect is significantly weakened. It can be deduced that the model’s performance is significantly impacted by the quality of the original training data. Despite the image data being processed by data enhancement technology before training, the features of the data are still based on the original data. If a large number of processed images are inferred, the detection performance will be affected, with the degree of influence depending on the deviation of the processed images from the original image data. The larger the deviation, the worse the detection effect, and vice versa. When the brightness changes, the $cmAP$ drops. The main impact is on the part of the image data that are more

difficult to detect. For the obvious and clear gas slug and dense bubbles, the model can still perform accurate inference and identification under varying brightness conditions.

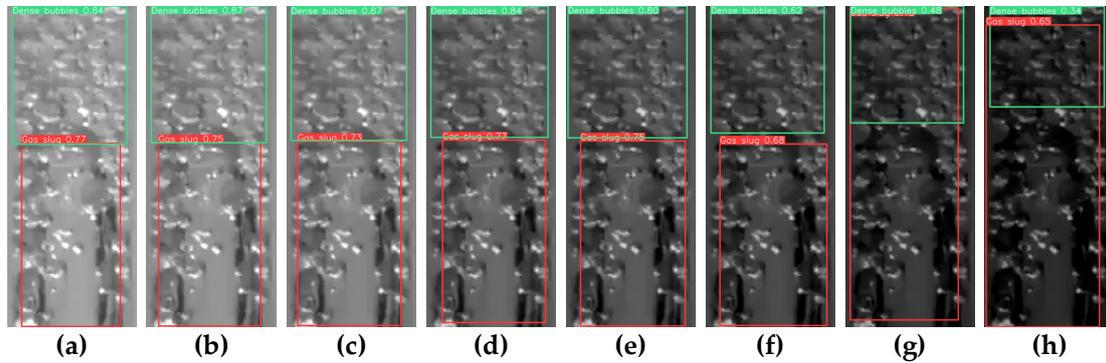


Figure 16. Yolov5n detection results in different brightness: (a) brightness + 45; (b) brightness + 30; (c) brightness + 15; (d) brightness - 15; (e) brightness - 30; (f) brightness - 45; (g) brightness - 60; (h) brightness - 75.

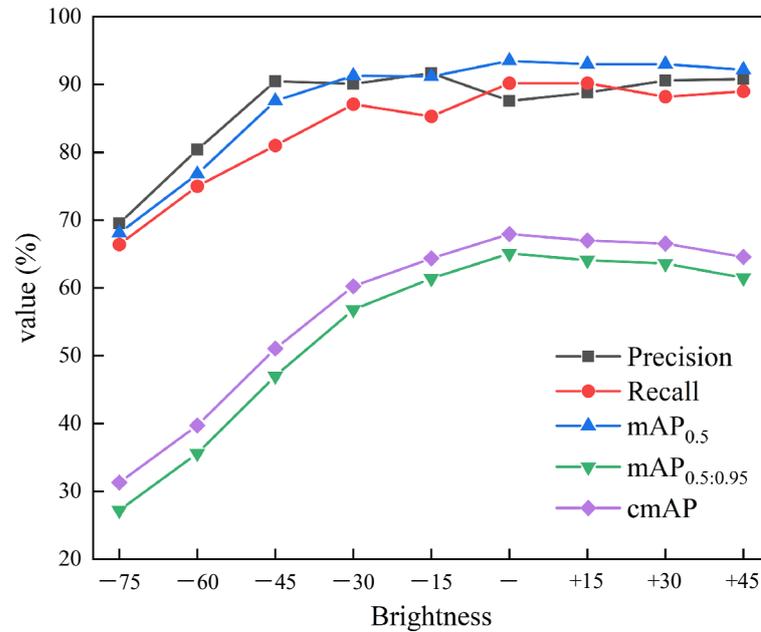


Figure 17. Changes in performance indicators of Yolov5n under different brightness levels.

Table 8. Yolov5n performance under different brightness conditions.

Brightness	Precision/%	Recall/%	mAP _{0.5} /%	mAP _{0.5:0.95} /%	cmAP/%
+45	90.8	89.0	92.2	61.5	64.57
+30	90.6	88.2	93.0	63.6	66.54
+15	88.8	90.2	93.0	64.1	66.99
(-)	87.6	90.2	93.5	65.1	67.94
-15	91.7	85.3	91.2	61.4	64.38
-30	90.1	87.1	91.3	56.8	60.25
-45	90.5	81.0	87.6	47.0	51.06
-60	80.4	75.0	76.8	35.6	39.72
-75	69.5	66.4	68.1	27.2	31.29

(-) represents the original dataset with constant brightness.

In the impact of the external environment, in addition to the impact caused by different light brightness, the obstruction of the image by the external environment is also very

common and cannot be ignored. For example, mixed transportation pipelines laid on the seabed, suspended solids and impurities in seawater, fish, dirt on pipeline walls, etc., will all obstruct the flow pattern image, thereby the features of the image will be affected. The obstruction environment in the actual environment is simulated by adding a circular occlusion at a random position to the image to analyze the detection applicability of the model when the image is obstructed. The radius of the randomly generated circular obstruction is between 0.05 and 0.1 of the width of the short side of the image. The obstruction numbers of the comparative experiments were three, five, and seven, respectively. Figures 18 and 19 and Table 9 show the detection effect and performance evaluation metrics of Yolov5n on the obstructed image dataset. In general, even if the gas slug and dense bubbles are obstructed in a distributed manner, those objects with clear features can be detected normally. But, if the obstructions are concentrated in a certain area, the inference process of the model will determine that area is a gas slug, as shown in Figure 18b. It is evident that the variability in obstruction positions introduces a random factor that influences the model's detection performance. For performance metrics, precision, recall, and mAP are all lower than the original data. The cmAP is lower than 60%, and the detection performance effect has been greatly affected. The reason is that the unique features of the gas slug and dense bubbles in the image may be affected by random obstructions. The more obstructions, the more severely the features are affected. For areas with less obstruction, the detection effect can still be maintained. For places with a lot of obstructions, the detection effect will be greatly weakened. A similar external environment will have an uncontrollable influence on the efficient detection of slug flow.

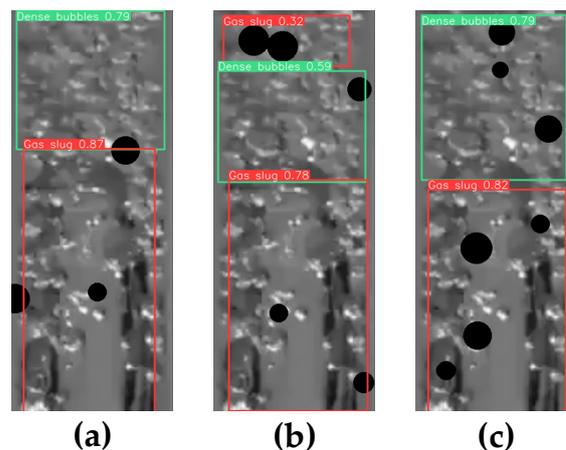


Figure 18. Yolov5n detection results with obstructions: (a) three obstructions; (b) five obstructions; (c) seven obstructions.

3.2.2. Limitation Analysis

The models discussed in this study have the advantages of high detection precision and fast inference speed. The model can still show excellent performance when there are small fluctuations in environmental conditions. But, the model still has deficiencies and limitations. In this study, the slug flow image data within the dataset were captured under the same conditions. Therefore, the size, brightness, and other formatting aspects of the images remain consistent. Despite employing mosaic data augmentation on the dataset, the overall structure of the dataset still closely resembles the original data. As inferred from the previous analysis and discussions, significant variations in image brightness or extensive occlusions negatively impact the model's detection performance. Performance metrics, such as precision, recall, and cmAP, experience substantial decreasing. Consequently, limitations are introduced to the model. Furthermore, the dataset comprises a mere 733 images. Typically, a larger dataset size contributes to the model's enhanced performance after training. This also constitutes a facet of the model's limitations.

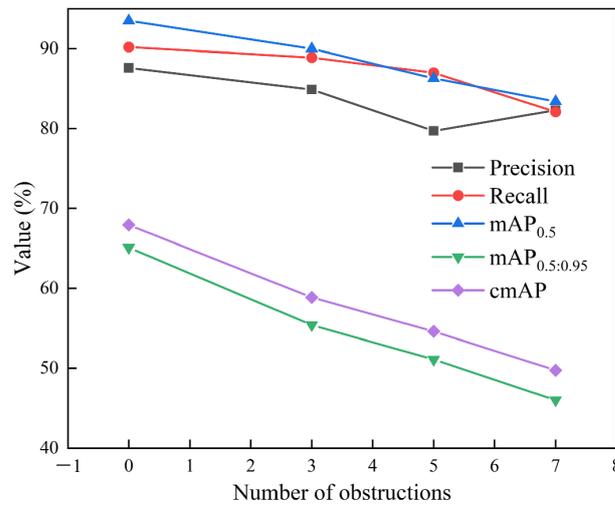


Figure 19. Changes in performance indicators of Yolov5n under different obstruction quantities.

Table 9. Yolov5n performance under different obstruction conditions.

Number of Obstructions	Precision/%	Recall/%	mAP _{0.5} /%	mAP _{0.5:0.95} /%	cmAP/%
0	87.6	90.2	93.5	65.1	67.94
3	84.9	88.9	90.0	55.4	58.86
5	79.7	87.0	86.3	51.1	54.62
7	82.3	82.1	83.4	46.0	49.74

The limitations imposed on the model by specific image data can be addressed through the following methods. Firstly, a significant increase in the quantity of image data can be achieved. Subsequently, within the augmentation of data volume, the diversity of images can be enhanced, for instance, incorporating a multitude of images with varying brightness levels, sizes, and occlusion patterns into the dataset. Employing data augmentation techniques further enhances data diversity, thus emulating real-world production environments. This approach strengthens the model’s generalization ability, enabling its weight parameters to accommodate a broader array of data styles. The updated model can adapt to a more extensive range of challenging production environments, fulfilling detection requirements across a wider spectrum of application scenarios.

3.3. Embedded Deployment Results of Yolov5n

In the previous analysis, for the slug flow image data, the Yolov5n model with the best comprehensive performance was obtained. It has a strong average detection accuracy. The most important thing is that the number of model parameters is very small; Flops is only 4.1 G, which provides the foundation and data guarantee for the embedded deployment of the model and the development of special detection application functions. To validate the practicality of the model for slug flow detection on embedded devices, we initially collected video data showcasing slug flow phenomena. The dimensions of this video data correspond to those of the image data used during model training, ensuring the model can fully leverage its inference and detection capabilities when applied to video data. Then, we deployed the trained Yolov5n model (.pt file format) in the Jetson Nano embedded device to perform inference detection on the slug flow video data. Figure 20 shows the detection effect of the Yolov5n model on slug flow video data in Jetson Nano. Since the image in the video data is the same as the image data in the dataset of the previous training model, and the size is also the same, the detection accuracy is the same as that in the training results analysis. This aspect of the research is dedicated to assessing the detection speed of the Yolov5n model on the Jetson Nano platform because the realization and development of detection functions in special scenarios are directly affected by the detection speed of

embedded devices. During Yolov5n’s inference and detection of slug flow video data, the average preprocessing, inference, and NMS (Non-Maximum Suppression) time for each image are 2.1 ms, 50.2 ms, and 8.6 ms, respectively, a total of about 60 ms. The FPS of video inference is about 16.7. Such inference is very slow, and the basic requirements of industrial production cannot be met. Therefore, it is very necessary to accelerate the inference of the model in Jetson Nano.

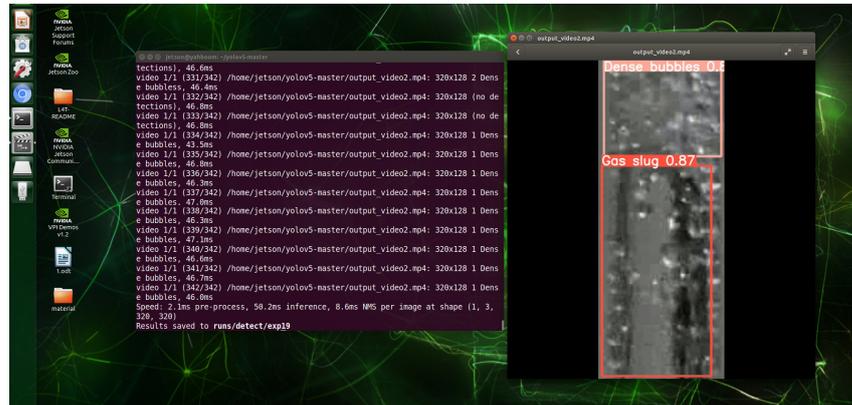


Figure 20. The detection performance of Yolov5n deployed on Jetson Nano.

TensorRT is an integrated tool designed for deep learning inference acceleration, which can convert the Yolov5n model into a TensorRT dedicated model and combine it with NVIDIA GPU to accelerate the inference process of the model. First, the Yolov5n model (.pt) is converted into another new format (.wts) using a file written in the Python language. Then, the serializable file is generated by continuing to execute the CMake command. Also, it is important that the new model format (.wts) must be converted to a final format (.engine) that TensorRT can read. Finally, the video data are inferred by reading the model file (.engine). Figure 21 shows the inference detection effect after TensorRT acceleration. In the detection results, category 0 represents the gas slug, and category 1 represents dense bubbles. During inference, the gas slug and dense bubbles can be accurately detected. Importantly, however, the total time of inference processing for each image in the video data is reduced to about 12 ms, and the FPS reaches about 83.3. Therefore, after using TensorRT to accelerate model deployment, the inference speed is greatly improved, which is five times that before the acceleration. The power of TensorRT for inference acceleration is proven. This method provides a new idea and method for real-time detection of slug flow.

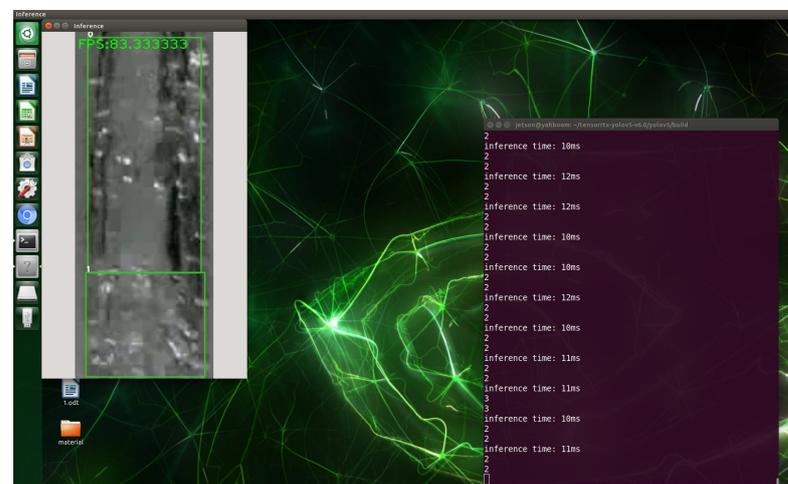


Figure 21. Detection effect after acceleration with TensorRT.

4. Conclusions

During the process of mixed transportation of subsea gas–liquid two-phase flows, the presence of slug flows can often result in damage to pipelines and related production equipment due to high pressure and vibrations. Accurate detection of slug flows is imperative. Gas slugs within the slug flow are sometimes concealed by a large number of bubbles, and the determination of slug flow patterns can also be influenced by subjective human perception. In this study, the Yolo object detection algorithm is introduced for the first time to slug flow identification. Based on experimental results, all seven models within the YoloV3 and YoloV5 series achieved high-precision detection of gas slugs and dense bubbles within the slug flow. The issue of gas slugs being undetectable due to bubble occlusion has been effectively resolved. Notably, when compared to previous research, the YoloV5n model demonstrated superior inference performance with the fewest parameters and Flops, totaling 1,761,871 and 4.1 G, respectively. It exhibited exceptional detection accuracy, with $mAP_{0.5}$ reaching 93.5%, $mAP_{0.5:0.95}$ at 65.1%, and $cmAP$ reaching 67.94%. In contrast, YoloV5n displayed the most comprehensive detection performance for slug flows. Furthermore, YoloV5n exhibited robust applicability under simulated conditions involving changes in image brightness and occlusion levels. This suggests that the trained YoloV5n model can adapt to varying real-world production environments within a certain range of environmental conditions, achieving proficient slug flow detection outcomes.

Additionally, due to the limitations in real-time performance posed by cloud servers and fixed computers, this study introduces for the first time the combination of the slug flow recognition algorithm with the Jetson Nano embedded device. The new YoloV5n model is embedded within the Jetson Nano and accelerated using the TensorRT tool during the inference process, resulting in a speed enhancement of approximately five times for the inference of slug flow video data. The achieved Frames Per Second (FPS) reached 83.3. The integration of the Yolo object detection algorithm with the Jetson Nano for slug flow detection enables real-time detection of slug flows with lower power consumption and fewer computational resources. Moreover, it offers the convenience of simultaneous deployment for multi-location detection. The proposed method not only lays the foundation for the development of relevant equipment for subsea slug flow detection but also provides fresh insights and guidance for efficient real-time detection of slug flows within subsea pipelines. However, this study does have limitations. For instance, in cases where significant changes occur in the production environment, image quality may be severely affected, resulting in a notable decline in slug flow detection accuracy. Future work should focus on enhancing the diversity of data within the dataset, including images with varying brightness levels and occlusions, to enable the model to adapt to a broader range of image scenarios. Furthermore, exploring faster inference methods for slug flow detection can potentially elevate detection efficiency.

Author Contributions: Methodology and formal analysis, W.Q., H.G. and W.L.; data collection and formal analysis, H.G. and E.H.; algorithm development, E.H., H.G. and W.L.; Investigation, X.S. and H.G.; Conceptualization, W.L., W.Q. and H.C.; Writing -original draft, H.G., W.L. and W.Q.; Supervision, H.C. and W.L.; Writing-review and editing, H.C. and W.Q.; Funding acquisition, W.Q. and W.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Postdoctoral Funding of China (2022M720626), National Natural Science Foundation of China (52271258), the Central Guidance on Local Science and Technology Development Fund of Liaoning Province (2023JH6/100100049), LiaoNing Revitalization Talents Program (XLYC2007092), 111 Project (B18009), and the Fundamental Research Funds for the Central Universities (3132023530,3132023510).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

References

1. Wang, J.G.; Meng, Y.; Han, B.Y.; Liu, Z.X.; Zhang, L.X.; Yao, H.Y.; Wu, Z.; Chu, J.W.; Yang, L.; Zhao, J.F. Hydrate blockage in subsea oil/gas flowlines: Prediction, prevention, and remediation. *Chem. Eng. J.* **2023**, *461*, 142020. [[CrossRef](#)]
2. Wang, Y.J.; Fan, J.C.; Wu, S.N. Research on the scheduling scheme of resource storage locations in deep-sea oil and gas exploitation. *Geoenergy Sci. Eng.* **2022**, *220*, 111214. [[CrossRef](#)]
3. Chang, X.P.; Song, Q.; Qu, C.J.; Li, Y.H.; Liu, J. Axial-transverse coupled vortex-induced vibration features of composite riser under gas-liquid two-phase internal flow. *Ocean Eng.* **2023**, *277*, 114163. [[CrossRef](#)]
4. Bao, Y.Y.; Ma, T.X.; Zhang, Y.; Wang, L. Influence of vibration on transient flow features of gas-liquid two-phase flow in inclined pipes. *Chem. Eng. Res. Des.* **2023**, *196*, 71–88. [[CrossRef](#)]
5. Hong, C.; Estefen, S.F.; Lourenco, M.I.; Wang, Y.X. A nonlinear constrained optimization model for subsea pipe route selection on an undulating seabed with multiple obstacles. *Ocean Eng.* **2019**, *186*, 106088. [[CrossRef](#)]
6. Zhang, K.; Chen, S.T.; Gong, Y.J.; Ning, D.Y.; Zhang, Z.M. Method to simulate transient pressure behaviors in subsea natural gas pipelines with a moving smart isolation device. *Eng. Fail. Anal.* **2022**, *144*, 106985. [[CrossRef](#)]
7. Nnabuife, S.G.; Tandoh, H.; Whidborne, J. Slug Flow Control in an S-shape Pipeline-Riser System using an Ultrasonic Sensor. *Digit. Chem. Eng.* **2022**, *2*, 100005. [[CrossRef](#)]
8. Zheng, Q.M.; Xu, Y.Q.; Zhang, P.; Bian, J.; Wang, F.H. Identification of gas-liquid two-phase flow regime in pipelines with low liquid holdup based on ResNet1D-34. *Flow Meas. Instrum.* **2022**, *88*, 102249. [[CrossRef](#)]
9. Li, W.S.; Guo, L.J.; Xie, X.D. Effects of a long pipeline on severe slugging in an S-shaped riser. *Chem. Eng. Sci.* **2017**, *171*, 379–390. [[CrossRef](#)]
10. Bordalo, S.N.; Morooka, C.K. Slug flow induced oscillations on subsea petroleum pipelines. *J. Pet. Sci. Eng.* **2018**, *165*, 535–549. [[CrossRef](#)]
11. Padrino, J.C.; Srinil, N.; Kurushina, V.; Swailes, D. Prediction of unsteady slug flow in a long curved inclined riser with a slug tracking model. *Int. J. Multiph. Flow* **2023**, *162*, 104410. [[CrossRef](#)]
12. Fadairo, A.; Obi, C.; Ling, K.; Rasouli, V.; Aboaba, O.; Gbadamosi, O. An improved model for severe slugging stability criteria in offshore pipeline-riser systems. *Pet. Res.* **2022**, *7*, 318–328. [[CrossRef](#)]
13. Zhao, X.Y.; Xu, Q.; Wu, Q.H.; Chang, Y.J.; Cao, Y.Q.; Zou, S.F.; Guo, L.J. Effect of high pressure on severe slugging and multiphase flow pattern transition in a long pipeline-riser system. *Exp. Therm. Fluid Sci.* **2023**, *148*, 110976. [[CrossRef](#)]
14. Mohammed, A.O.; Al-Kayiem, H.H.; Osman, A.B.; Sabir, O. One-way coupled fluid–structure interaction of gas–liquid slug flow in a horizontal pipe: Experiments and simulations. *J. Fluids Struct.* **2020**, *97*, 103083. [[CrossRef](#)]
15. Li, Y.C.; Hosseini, M.; Arasteh, H.; Toghraie, D.; Rostami, S. Transition simulation of two-phase intermittent slug flow features in oil and gas pipelines. *Int. Commun. Heat Mass Transf.* **2020**, *113*, 104534. [[CrossRef](#)]
16. Abdul-Majeed, G.H.; Firouzi, M. The suitability of the dimensionless terms used in correlating slug liquid holdup with flow parameters in viscous two-phase flows. *Int. Commun. Heat Mass Transf.* **2019**, *108*, 104323. [[CrossRef](#)]
17. He, K.; Qian, H.L.; Zhang, H.B.; Huang, S.M. Experimental study of ensemble phase splitting features of water-oil slug/droplet flow at a branching micro-T-junction. *Chem. Eng. Res. Des.* **2023**, *191*, 552–563. [[CrossRef](#)]
18. Farias, P.S.C.; Azevedo, L.F.A.; de Paula, I.B. Experimental investigation of slug initiation by controlled disturbances in gas-liquid horizontal pipe flow. *Exp. Therm. Fluid Sci.* **2023**, *148*, 110982. [[CrossRef](#)]
19. Baungartner, R.; Goncalves, G.F.N.; Loureiro, J.B.R.; Freire, A.P.S. Slug flows of gas and shear-thinning fluids in horizontal pipes. *Int. J. Multiph. Flow* **2023**, *165*, 104473. [[CrossRef](#)]
20. Dreiling, R.; Dubois, V.; Zimmermann, S.; Nguyen-Xuan, T.; Schreivogel, P.; di Mare, F. Numerical investigation of slug flow in pulsating heat pipes using an interface capturing approach. *Int. J. Heat Mass Transf.* **2022**, *199*, 123459. [[CrossRef](#)]
21. Plana, C.; Song, B.F.; Avila, M. Direct numerical simulation of two-phase pipe flow: Influence of the domain length on the flow regime. *Int. J. Multiph. Flow* **2021**, *144*, 103786. [[CrossRef](#)]
22. Li, W.H.; Zhou, Q.; Yin, G.; Ong, M.C.; Li, G.; Han, F.H. Experimental Investigation and Numerical Modeling of Two-Phase Flow Development and Flow-Induced Vibration of a Multi-Plane Subsea Jumper. *J. Mar. Sci. Eng.* **2022**, *10*, 1334. [[CrossRef](#)]
23. Deendarlianto; Rahmandhika, A.; Widyatama, A.; Dinaryanto, O.; Widyaparaga, A.; Indarto. Experimental study on the hydrodynamic behavior of gas-liquid air-water two-phase flow near the transition to slug flow in horizontal pipes. *Int. J. Heat Mass Transf.* **2019**, *130*, 187–203. [[CrossRef](#)]
24. Dinaryanto, O.; Prayitno, Y.A.K.; Majid, A.I.; Hudaya, A.Z.; Nusirwan, Y.A.; Widyaparaga, A.; Indarto; Deendarlianto. Experimental investigation on the initiation and flow development of gas-liquid slug two-phase flow in a horizontal pipe. *Exp. Therm. Fluid Sci.* **2016**, *81*, 93–108. [[CrossRef](#)]
25. Zhang, M.H.; Pan, L.M.; Ju, P.; Yang, X.H.; Ishii, M. The mechanism of bubbly to slug flow regime transition in air-water two phase flow: A new transition criterion. *Int. J. Heat Mass Transf.* **2017**, *108*, 1579–1590. [[CrossRef](#)]
26. Naidek, B.P.; Conte, M.G.; Cozin, C.; dos Santos, E.N.; Rodrigues, H.T.; da Fonseca, R.; da Silva, M.J.; Morales, R.E.M. Experimental study of influence of liquid viscosity in horizontal slug flow. *Exp. Therm. Fluid Sci.* **2022**, *141*, 110798. [[CrossRef](#)]
27. Cao, X.W.; Zhang, P.; Li, X.; Li, Z.; Zhang, Q.R.; Bian, J. Experimental and numerical study on the flow features of slug flow in a horizontal elbow. *J. Pipeline Sci. Eng.* **2022**, *2*, 100076. [[CrossRef](#)]
28. Schmelter, S.; Olbrich, M.; Schmeyer, E.; Bar, M. Numerical simulation, validation, and analysis of two-phase slug flow in large horizontal pipes. *Flow Meas. Instrum.* **2020**, *73*, 101722. [[CrossRef](#)]

29. Alsakka, F.; Assaf, S.; El-Chami, I.; Al -Hussein, M. Computer vision applications in offsite construction. *Autom. Constr.* **2023**, *154*, 104980. [[CrossRef](#)]
30. Sharma, H.; Kumar, H.; Mangla, S.K. Enablers to computer vision technology for sustainable E-waste management. *J. Clean. Prod.* **2023**, *412*, 137396. [[CrossRef](#)]
31. Wang, C.Y.; Yang, C.; Zhang, H.P.; Wang, S.Z.; Yang, Z.X.; Fu, J.G.; Sun, Y.Q. Marine-Hydraulic-Oil-Particle Contaminant Identification Study Based on OpenCV. *J. Mar. Sci. Eng.* **2023**, *10*, 1789. [[CrossRef](#)]
32. Huang, K.F.; Li, S.Y.; Cai, F.; Zhou, R.H. Detection of Large Foreign Objects on Coal Mine Belt Conveyor Based on Improved. *Processes* **2023**, *11*, 2469. [[CrossRef](#)]
33. Zhao, K.; Zhao, L.L.; Zhao, Y.A.; Deng, H.B. Study on Lightweight Model of Maize Seedling Object Detection Based on YOLOv7. *Appl. Sci. -Basel* **2023**, *13*, 7731. [[CrossRef](#)]
34. OuYang, L.; Jin, N.D.; Ren, W.K. A new deep neural network framework with multivariate time series for two-phase flow pattern identification. *Expert Syst. Appl.* **2022**, *205*, 117704. [[CrossRef](#)]
35. Roxas, R.; Evangelista, M.A.; Sombillo, J.A.; Nnabuike, S.G.; Pilario, K.E. Machine Learning Based Flow Regime Identification using Ultrasonic Doppler Data and Feature Relevance Determination. *Digit. Chem. Eng.* **2022**, *3*, 100024. [[CrossRef](#)]
36. Xu, H.; Tang, H.; Zhang, B.; Liu, Y. Identification of two-phase flow regime in the energy industry based on modified convolutional neural network. *Prog. Nucl. Energy* **2022**, *147*, 104191. [[CrossRef](#)]
37. Qiao, W.L.; Guo, H.T.Y.; Huang, E.Z.; Chen, H.Q.; Lian, C.P. Two-Phase Flow Pattern Identification by Embedding Double Attention Mechanisms into a Convolutional Neural Network. *J. Mar. Sci. Eng.* **2023**, *11*, 793. [[CrossRef](#)]
38. Nie, F.; Wang, H.; Song, Q.; Zhao, Y.; Shen, J.; Gong, M. Image identification for two-phase flow patterns based on CNN algorithms. *Int. J. Multiph. Flow* **2022**, *152*, 104067. [[CrossRef](#)]
39. Banasiak, R.; Wajman, R.; Jaworski, T.; Fiderek, P.; Fidos, H.; Nowakowski, J.; Sankowski, D. Study on two-phase flow regime visualization and identification using 3D electrical capacitance tomography and fuzzy-logic classification. *Int. J. Multiph. Flow* **2014**, *58*, 1–14. [[CrossRef](#)]
40. Shibata, N.; Miwa, S.; Sawa, K.; Takahashi, M.; Murayama, T.; Tenma, N. Identification of high-pressure two-phase flow regime transition using image processing and deep learning. *J. Nat. Gas Sci. Eng.* **2022**, *102*, 104560. [[CrossRef](#)]
41. Li, W.; Song, W.; Yin, G.; Ong, M.C.; Han, F. Flow regime identification in the subsea jumper based on electrical capacitance tomography and convolution neural network. *Ocean Eng.* **2022**, *266*, 113152. [[CrossRef](#)]
42. Chen, W.; Li, Y.; Tian, Z.J.; Zhang, F. 2D and 3D object detection algorithms from images: A Survey. *Array* **2023**, *19*, 100305. [[CrossRef](#)]
43. Situ, Z.; Teng, S.; Feng, W.E.; Zhong, Q.S.; Chen, G.F.; Su, J.H.; Zhou, Q.Q. A transfer learning-based YOLO network for sewer defect detection in comparison to classic object detection methods. *Dev. Built Environ.* **2023**, *15*, 100191. [[CrossRef](#)]
44. Ji, S.J.; Ling, Q.H.; Han, F. An improved algorithm for small object detection based on YOLO v4 and multi-scale contextual information. *Comput. Electr. Eng.* **2022**, *105*, 108490. [[CrossRef](#)]
45. Soom, J.; Pattanaik, V.; Leier, M.; Tuhtan, J.A. Environmentally adaptive fish or no-fish classification for river video fish counters using high-performance desktop and embedded hardware. *Ecol. Inform.* **2022**, *72*, 101817. [[CrossRef](#)]
46. Gong, C.P.; Li, A.J.; Song, Y.M.; Xu, N.; He, W.K. Traffic Sign Identification Based on the YOLOv3 Algorithm. *Sensors* **2022**, *22*, 9345. [[CrossRef](#)] [[PubMed](#)]
47. He, K.M.; Zhang, X.Y.; Ren, S.Q.; Sun, J. Deep Residual Learning for Image Identification. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Identification (CVPR), Las Vegas, NV, USA, 27–30 June 2016.
48. Huang, Y.Q.; Zheng, J.C.; Sun, S.D.; Yang, C.F.; Liu, J. Optimized YOLOv3 Algorithm and Its Application in Traffic Flow Detections. *Appl. Sci.* **2020**, *10*, 3079. [[CrossRef](#)]
49. Li, W.; Zhu, D.L.; Wang, Q. A single view leaf reconstruction method based on the fusion of ResNet and differentiable render in plant growth digital twin system. *Comput. Electron. Agric.* **2022**, *193*, 106712. [[CrossRef](#)]
50. Al Muksit, A.; Hasan, F.; Emon, M.F.H.B.; Haque, M.R.; Anwary, A.R.; Shatabda, S. YOLO-Fish: A robust fish detection model to detect fish in realistic underwater environment. *Ecol. Inform.* **2022**, *72*, 101847. [[CrossRef](#)]
51. Shi, J.T.; Yang, J.; Zhang, Y.T. Research on Steel Surface Defect Detection Based on YOLOv5 with Attention Mechanism. *Electronics* **2022**, *11*, 3735. [[CrossRef](#)]
52. Hong, W.W.; Ma, Z.H.; Ye, B.L.; Yu, G.H.; Tang, T.; Zheng, M.F. Detection of Green Asparagus in Complex Environments Based on the Improved YOLOv5 Algorithm. *Sensors* **2023**, *23*, 1562. [[CrossRef](#)] [[PubMed](#)]
53. Yao, J.L.; Fan, X.S.; Li, B.; Qin, W.L. Adverse Weather Object Detection Algorithm Based on Adaptive Color Levels and Improved YOLOv5. *Sensors* **2022**, *22*, 8577. [[CrossRef](#)] [[PubMed](#)]
54. Chen, S.F.; Yang, D.Z.; Liu, J.; Tian, Q.; Zhou, F.T. Automatic weld type classification, tacked spot identification and weld ROI determination for robotic welding based on modified YOLOv5. *Robot. Comput.-Integr. Manuf.* **2022**, *81*, 102490. [[CrossRef](#)]
55. Yu, F.Y.; Zhang, G.X.; Zhao, F.Y.; Wang, X.X.; Liu, H.; Lin, P.; Chen, Y.M. Improved YOLO-v5 model for boosting face mask identification accuracy on heterogeneous IoT computing platforms. *Internet Things* **2023**, *23*, 100881. [[CrossRef](#)]
56. Liu, G.; Hu, Y.X.; Chen, Z.Y.; Guo, J.W.; Ni, P. Lightweight object detection algorithm for robots with improved YOLOv5. *Eng. Appl. Artif. Intell.* **2023**, *123*, 106217. [[CrossRef](#)]
57. Shaban, H.; Tavoularis, S. Video: Zorbubbles (Producing flow regimes in air-water flow). In Proceedings of the 68th Annual Meeting of the APS Division of Fluid Dynamics, Boston, MA, USA, 22–24 November 2015. [[CrossRef](#)]

58. Chen, S.N.; Duan, J.F.; Wang, H.; Wang, R.Q.; Li, J.Z.; Qi, M.; Duan, Y.; Qi, S.L. Automatic detection of stroke lesion from diffusion-weighted imaging via the improved YOLOv5. *Comput. Biol. Med.* **2022**, *150*, 106120. [[CrossRef](#)]
59. Le, M.; Le, D.; Vu, H.H.T. Thermal inspection of photovoltaic modules with deep convolutional neural networks on edge devices in AUV. *Measurement* **2023**, *218*, 113135. [[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.