Journal of *Marine Science and Engineering*

MDPI

Article

A Side-Scan Sonar Image Synthesis Method Based on a Diffusion Model

Zhiwei Yang [1], Jianhu Zhao [1,*], Hongmei Zhang [2], Yongcan Yu [1] and Chao Huang [1]

[1] Institute of Marine Science and Technology, School of Geodesy and Geomatics, Wuhan University, Wuhan 430079, China; 2022202140046@whu.edu.cn (Z.Y.)
[2] Department of Automation, School of Power and Mechanical Engineering, Wuhan University, Wuhan 430079, China
* Correspondence: jhzhao@sgg.whu.edu.cn

Abstract: The limited number and under-representation of side-scan sonar samples hinders the training of high-performance underwater object detection models. To address this issue, in this paper, we propose a diffusion model-based method to augment side-scan sonar image samples. First, the side-scan sonar image is transformed into Gaussian distributed random noise based on its a priori discriminant. Then, the Gaussian noise is modified step by step in the inverse process to reconstruct a new sample with the same distribution as the a priori data. To improve the sample generation speed, an accelerated encoder is introduced to reduce the model sampling time. Experiments show that our method can generate a large number of representative side-scan sonar images. The generated side-scan sonar shipwreck images are used to train an underwater shipwreck object detection model, which achieves a detection accuracy of 91.5% on a real side-scan sonar dataset. This exceeds the detection accuracy of real side-scan sonar data and validates the feasibility of the proposed method.

Keywords: diffusion model; sample augmentation; object detection; side scan sonar

Citation: Yang, Z.; Zhao, J.; Zhang, H.; Yu, Y.; Huang, C. A Side-Scan Sonar Image Synthesis Method Based on a Diffusion Model. *J. Mar. Sci. Eng.* **2023**, *11*, 1103. https://doi.org/10.3390/jmse11061103

Academic Editors: Tracianne B Neilsen, Haiqiang Niu and Rafael Morales

Received: 22 March 2023
Revised: 25 April 2023
Accepted: 21 May 2023
Published: 23 May 2023

1. Introduction

As humans continue to exploit marine resources, more and more detection techniques are being applied to underwater object detection [1], maritime search [2], marine engineering [3,4], and archaeological excavation [5]. Among these technologies, side-scan sonar systems are widely used for underwater object detection due to their low cost and high sweep and resolution [6]. Deep learning-based methods for object detection [7] have achieved performance that far surpasses traditional methods on a variety of public datasets. However, deep learning, especially deep convolutional neural network (DCNN) techniques, require a large number of representative samples to train object detection models [8]. Due to the high measurement cost and limited number of maritime events, the number of samples of side-scan sonar images is small and weakly representative. This limits the development of DCNN for underwater object detection.

To address the limited number of side-scan sonar image samples, some researchers have used image feature extraction and image transformation enhancement methods to increase the number of samples [9,10]. However, these generated samples are not sufficiently representative as they ignore variations in imaging conditions and environments. While solving the under-sample size problem, it does not help improve the accuracy of underwater object recognition, and the geometric transformation is prone to overfitting the recognition model. Additionally, many researchers have used style transfer and optical images to synthesize side-scan sonar images [11–14]. However, direct style transfer using optical images does not consider the side-scan sonar imaging mechanism, resulting in poorly represented single-style generated samples and limited improvement in object detection model performance. Huang et al. [15] proposed a comprehensive sample augmentation method for side-scan sonar targets, backgrounds, textures, resolutions, and noise using a

wreck as an example. However, this method is tedious and requires a lot of time for data collection and cleaning. In addition to sample augmentation using optical images and style transfer methods, various deep generation models have also achieved better results in the field of sample augmentation [16,17]. Generative adversarial networks (GAN) [18], autoregressive models [19], normalizing flows [20], and variational auto-encoders (VAEs) [21] have generated numerous high-quality image samples. Bore et al. [22] implemented a side-scan sonar map simulation for a specific measurement environment using conditional adversarial generative networks. However, this method requires measured seafloor topography at the corresponding location and side-scan sonar images for training the generative model, which is a demanding condition to implement. Jiang et al. [23] proposed a semantic image synthesis model based on adversarial generative networks that can quickly generate a completely new image based on a hand-drawn semantic segmentation map and any real side-scan sonar target image. However, the generated images are limited in terms of style and representation due to the limitations of the masks.

With the advancement of correlated iterative generation models, denoising diffusion probability models have shown their ability to produce samples comparable to GANs [24,25]. Diffusion models [26] aim to transform the prior data distribution into random noise that matches the Gaussian distribution and then gradually correct the transformation to reconstruct a completely new sample with the same prior data distribution. However, this requires several iterations to generate a high-quality sample. For denoising diffusion probabilistic models (DDPM) [27], its generation process approximates the reverse of the forward diffusion process and generally requires thousands of iterations. Diffusion models tend to take more time in sample generation compared to GANs. For this reason, we propose a side-scan sonar sample generation method based on the denoising diffusion implicit models (DDIM) [28] generation model with the introduction of an accelerated encoder [29] in this paper. This reduces the number of sampling steps in the diffusion model and accelerates image generation.

2. Methods

2.1. Diffusion Mode

The diffusion model [26] gradually changes the image into a Gaussian noise image by defining a forward process that continuously adds noise and then gradually denoises the Gaussian noise by defining an inverse process to obtain the sampled image. Both processes are defined in DDPM as a parametrized Gaussian Markov chain [27]. In the forward sampling process, the training data is assumed to satisfy the distribution $x_0 \sim q(x)$. The forward process sequentially adds Gaussian noise to the samples sequentially at T time steps.

$$q(x_t|x_{t-1}) = \mathcal{N}(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t I) \tag{1}$$

$$q(x_{1:T}|x_0) = \prod_{t=1}^{T} q(x_t|x_{t-1}) \tag{2}$$

The variance used for each step is $\{\beta_t \in (0,1)\}_{t=1}^{t}$ denotes the learning variance for the different steps that satisfy $\beta_1 < \beta_2 < \ldots < \beta_T$. Eventually, if T *is* large enough, the final obtained x_T then completely loses the features of the original data and becomes a random noise [25].

Its inference distribution depends on the edge distribution $q(x_t|x_0)$, rather than acting directly on the joint distribution $q(x_{1:T}|x_0)$. This indicates that DDPM, a hidden variable model, can have many inference distributions to choose from, as long as the inference distribution satisfies the edge distribution. For this reason, in the DDIM [28] the inference distribution is redefined in:

$$q_\sigma(x_{1:T}|x_0) = q_\sigma(x_T|x_0)\prod_{t=2}^{T} q(x_{t-1}|x_t, x_0) \tag{3}$$

At this point $q_\sigma(x_T|x_0) = \mathcal{N}(\sqrt{\alpha_T}x_0, (1-\alpha_T)\mathbf{I})$, and for all $t \geq 2$, to satisfy.

$$q_\sigma(x_{t-1}|x_t, x_0) = \mathcal{N}(x_{t-1}; \sqrt{\alpha_{t-1}}x_0 + \sqrt{1-\alpha_{t-1}-\sigma_t{}^2}\frac{x_t - \sqrt{\alpha_t}x_0}{\sqrt{1-\alpha_t}}, \sigma_t^2\mathbf{I}) \tag{4}$$

where the variance σ_t^2 is a real number, the inference distribution $q_\sigma(x_{1:T}|x_0)$ defined in Equation (3) has satisfied the marginal distribution $q_\sigma(x_t|x_0) = \mathcal{N}(x_t; \sqrt{\alpha_t}x_0, (1-\alpha_t)I)$ [28], and the mean value of $q_\sigma(x_{t-1}|x_t, x_0)$ is also defined as a combined function dependent on x_0 and x_t, at which point the generation process can be optimized according to the optimization method in DDPM.

The diffusion process is adding noise to the data and the inverse process is a denoising process. If the true distribution of each step of the inverse process $q(x_{t-1}|x_t)$ is known, then starting from a random noise $x_T \sim \mathcal{N}(0, \boldsymbol{I})$ and gradually denoising it will generate a true sample, so the inverse process is also the process of data generation.

Based on the above principles, the optimization process is constructed using a neural network, using the neural network ε_θ to predict the noise, and then, according to the form of $q_\sigma(x_{t-1}|x_t, x_0)$, in the generation phase, the generation can be divided into three parts, where one is generated by the predicted $x_0 = \frac{x_t - \sqrt{1-\alpha_t}\epsilon_\theta(x_t,t)}{\sqrt{\alpha_t}}$, the second is generated by the part pointing to $x_t = \sqrt{1-\alpha_{t-1}-\sigma_t^2}.\epsilon_\theta(x_t, t)$, and the third is the random noise $\sigma_t\epsilon_t$; at this time, ϵ_t is the noise not related to x_t, and then from the predicted x_0, x_t and the random noise $\sigma_t\epsilon_t$ can be generated from x_t x_{t-1}:

$$\begin{aligned} x_{t-1} &= x_0 + x_t + noise \\ &= \sqrt{\alpha_t - 1}\left(\frac{x_t - \sqrt{1-\alpha_t}\epsilon_\theta(x_t,t)}{\sqrt{\alpha_t}}\right) + \sqrt{1-\alpha_{t-1}-\sigma_t^2}.\epsilon_\theta(x_t, t) + \sigma_t\epsilon_t \end{aligned} \tag{5}$$

During the generation process, variations in the random noise $\sigma_t\epsilon_t$ introduce significant uncertainty in the target generation. At this point, ϵ_t is a noise unrelated to x_t. The difference in the value taken for σ_t determines the value of the random noise.

$$\sigma_t^2 = \eta.\sqrt{(1-\alpha_{t-1})/(1-\alpha_t)}\sqrt{(1-\alpha_t/\alpha_{t-1})} \tag{6}$$

When $\eta = 1$, at this time, $\sigma_t^2 = \beta_t$, and the generation process at this time is affected by random noise; when $\eta = 0$, at this time there is no random noise in the generation process; it is a deterministic process once the initial random noise x_T is determined, and there is no random noise in the generation process, and then the generation result can be determined.

2.2. DPM-Solver

In the generative process, the model sampling has to start from pure noise and then keep denoising step by step to finally obtain the target image. DDPM needs to sample 1000 steps to get a higher quality image [27]. The DDIM also needs to sample at least 50 to 100 steps to obtain a higher quality image [28] to increase the sample generation speed. In this paper, the DPM-Solver is integrated into DDIM. The DPM-Solver [29] is an efficient solver specifically designed for diffusion models, which does not require any additional training and can obtain very high quality samples in only 10 to 15 steps, which can greatly improve the model generation speed.

DDIM does not have an explicit forward process in the training process to this point, and a shorter sampling step can be defined compared to DDPM [28]. That is, a subsequent $\tau = [\tau_1, \tau_2, \ldots, \tau_S]$ of length S is sampled from the original sampling sequence $[1, \ldots, T]$. The forward generation process of x is defined as a Markov chain and satisfies $q(x_{\tau_i}|x_0)\mathcal{N}(x_t; \sqrt{\alpha_{\tau_i}}x_0, (1-\alpha_{\tau_i})I)$. The generation process is shown in the Figure 1.

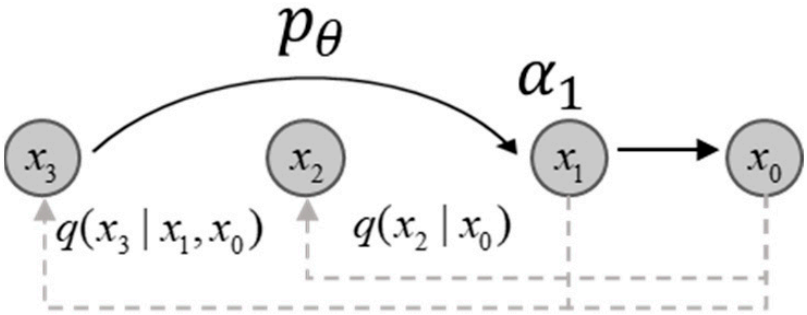


Figure 1. Graphical model for accelerated generation, where $\tau = [1, 3]$.

Then the generative process can also be replaced by a reverse Markov chain of subsequences τ. The generative process then becomes

$$x_{\tau_i - 1} = \sqrt{\alpha_{\tau_i} - 1}\left(\frac{x_{\tau_i} - \sqrt{1 - \alpha_{\tau_i}}\epsilon_\theta(x_{\tau_i}, \tau_i)}{\sqrt{\alpha_{\tau_i}}}\right) + \sqrt{1 - \alpha_{\tau_i - 1} - \sigma_{\tau_i}^2}.\epsilon_\theta(x_{\tau_i}, \tau_i) + \sigma_{\tau_i}\epsilon \quad (7)$$

The DPM-Solver [21] is based on the semi-linear structure of the diffusion model and by computing the linear terms in the ODE [30] in an exact and analytic way,

$$\frac{dx_t}{dt} = f(t)x_t + \frac{g^2(t)}{2\sigma_t}\epsilon_\theta(x_t, t) \quad (8)$$

$$x_t = e^{\int_s^t f(\tau)d\tau}x_s + \int_s^t \left(e^{\int_s^t f(r)dr}\frac{g^2(\tau)}{2\sigma_\tau}\epsilon_\theta(x_\tau, \tau)\right)d\tau \quad (9)$$

x_t is the exact solution at the time of t, and the remaining integral term is a complex integral with respect to time. This integral can be computed to compute all known terms as much as possible, approximating only the neural network part and minimizing discretization errors by the maximum procedure of

$$x_{t_{i-1} \to t_i} = \frac{\alpha_{t_i}}{\alpha_{t_{i-1}}}\widetilde{x}_{t_{i-1}} - \alpha_{t_i}\sum_{n=0}^{k-1}\hat{\epsilon}_\theta{}^{(n)}(\hat{x}_{\lambda_{t_{i-1}}}, \lambda_{t_{i-1}})\int_{\lambda_{t_{i-1}}}^{\lambda_{t_i}} e^{-\lambda}\frac{\left(\lambda - \lambda_{t_{i-1}}\right)^n}{n!}d\lambda + \mathcal{O}(h_i^{k+1}) \quad (10)$$

Among them, $\lambda_t = \log(\alpha_t/\sigma_t)$, and the derivation of DPM-solver proves that DDIM corresponds to the first-order ODE solver of diffusion ODE [30,31], while the DPM-Solver gives the corresponding higher-order solver that allows about 10 steps of sampling to reach a sampling comparable to the 1000 steps of DDPM. To improve the model generation speed, we use the third-order form of the DPM-Solver to obtain the generation results of the diffusion model.

2.3. Model Structure

The diffusion model uses the UNet model [32,33] structure to implement the process of diffusion through the encoder–decoder structure. The model builds the network structure by using the Resnet Block module [34] and uses the attention mechanism to regulate the model output. At the encoder structure, the model uses a convolutional model for downsampling. In the structure of the decoder, upsampling is performed using the interpolate function to amplify the details of the image with reduced image information loss. The network structure of the diffusion model is shown in Figure 2.

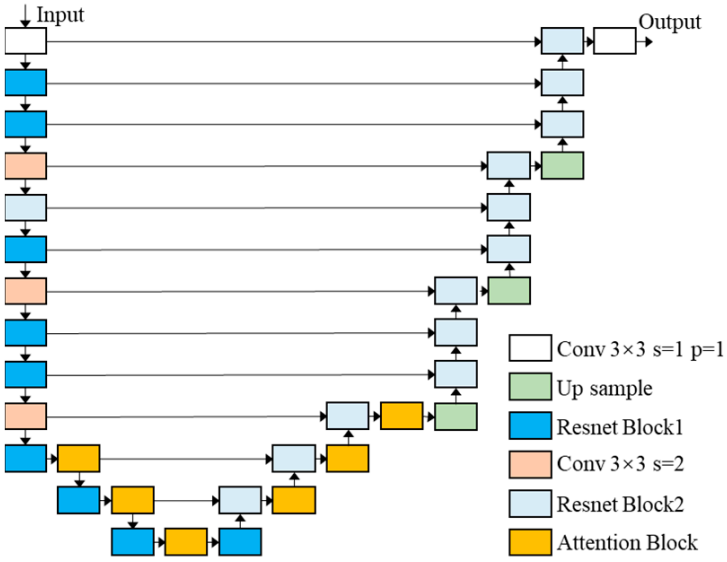
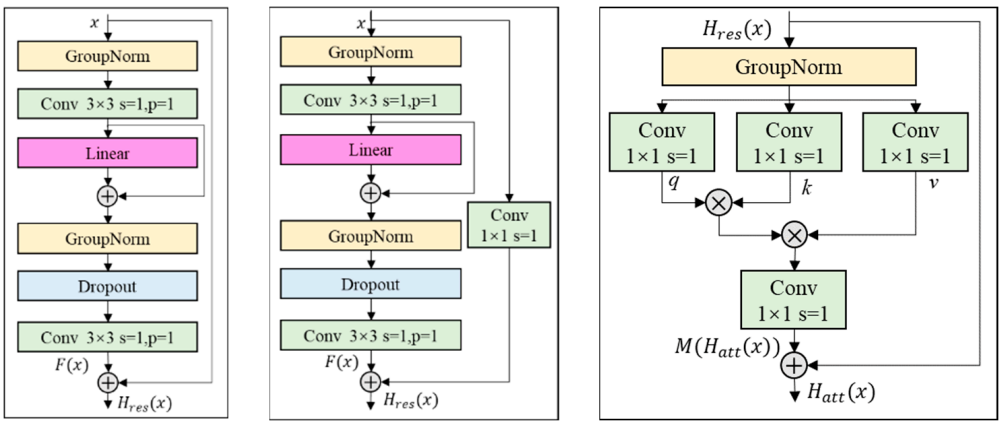


Figure 2. Network structure of diffusion model.

The Resnet Block1 module uses a typical residual structure and is used for feature extraction, while the Resnet Block2 module is used for feature fusion. The model loses image information and decreases image resolution when downsampling, and then allows the low-resolution image containing high-level abstract features to become high-resolution while retaining the high-level abstract features when upsampling and then does a concatenate operation with the high-resolution image of the low-level surface features on the left side, thus retaining more dimensional information. However, when doing fusion in the Resnet Block2 module, the dimensionality of the module input feature map and the feature map after two convolutions are different, so it is necessary to perform dimensionality reduction in the shortcut branch 1×1 convolution structure. The schematic diagram of the module structure is shown in the Figure 3.



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

Figure 3. Network module structure diagram. (**a**) Resnet Block1; (**b**) Resnet Block1; (**c**) Attention Block.

In the Resnet Block, the output of the residual network is expressed as

$$H_{res}(x) = x + F(x) \tag{11}$$

Here, x is the input of the residual module, $F(x)$ is the output of the residual backbone network, and $H_{res}(x)$ is the output of the residual module. During downsampling and upsampling, the attention mechanism is added after the Resnet Block, and the resulting

attention map is superimposed with the backbone feature map in a similar manner to residual learning, and the output at this point can be represented as follows.

$$H_{att}(x) = M(H_{res}(x)) + H_{res}(x) \tag{12}$$

$M(H_{res}(x))$ is the output result of the attention module, at which time the effective features in the output feature map can be enhanced by superimposing $H_{res}(x)$ and finally by superimposing the attention module in different downsampling and upsampling processes to gradually improve the expressive power of the network.

During model training, to better predict the noise, the model computes the error between the output noise of the diffusion model and the true noise as loss and updates the parameters in the UNetModel structure by backpropagation.

$$Loss(\theta) = \left(\epsilon - \epsilon_\theta(\sqrt{\alpha_t}x_0 + \sqrt{1-\alpha_t}\epsilon, t)\right)^2 \tag{13}$$

The Gaussian distribution ϵ_θ is the noise predicted by the neural network model for the generative process of model denoising, and the goal of the diffusion model training is to learn the mean squared error between Gaussian noise ϵ and ϵ_θ.

2.4. Model Training and Sample Generation Process

The goal of diffusion model training is to learn the inverse of the forward process, i.e., to train the probability distribution $p_\theta(x_{t-1}|x_t)$. By traversing backward through the training process, new data samples can be regenerated x_0. The overall training and generation process of the model is shown in the Figure 4.

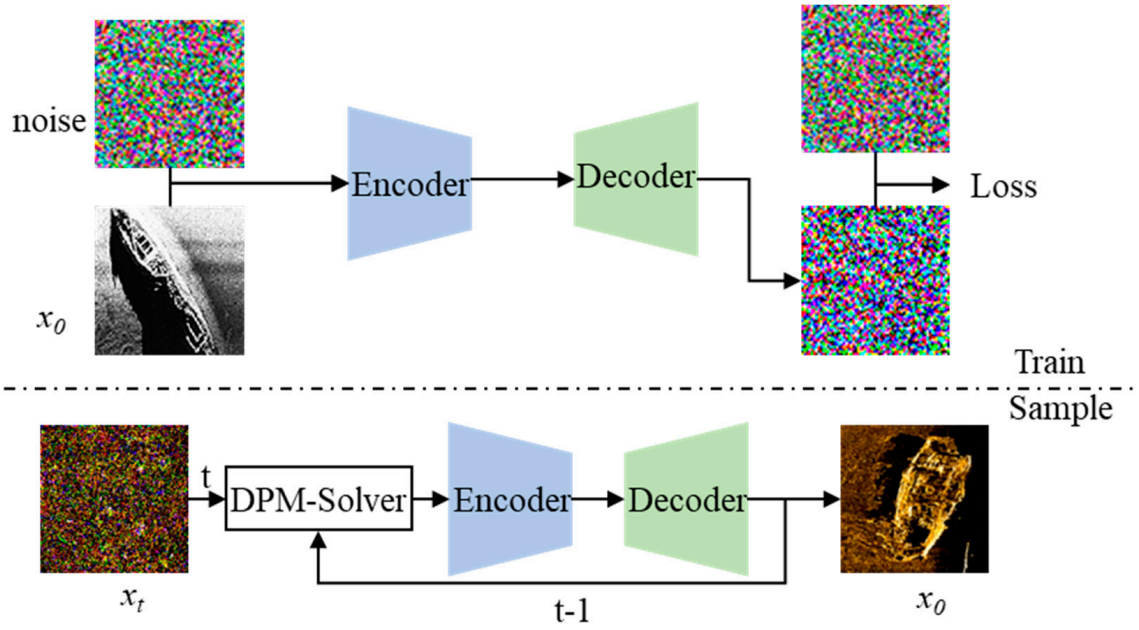
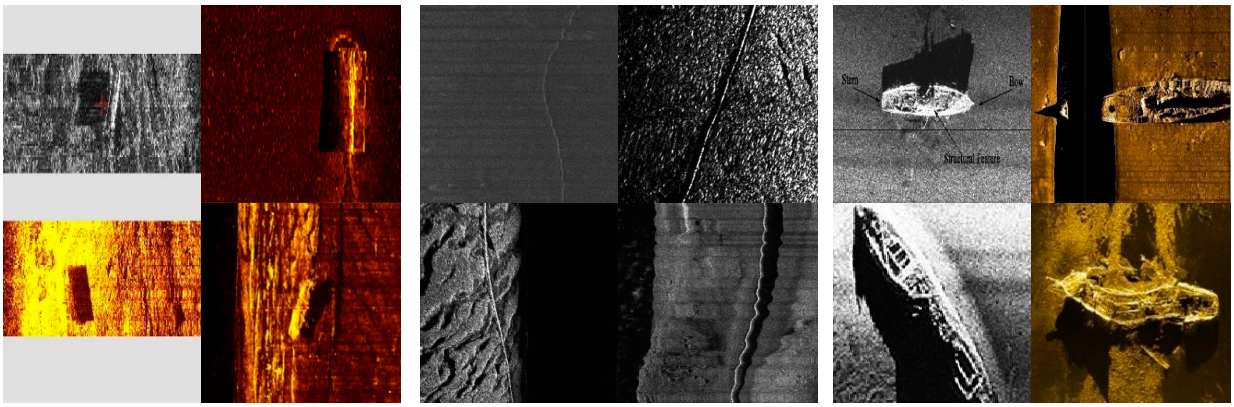


Figure 4. Schematic diagram of the training and generation process of the diffusion model.

In the training process, the network performs model update by predicting Gaussian noise, and the generation process is the inverse process of training. Firstly, a noisy image conforming to the Gaussian distribution is randomly generated, and the diffusion model trained with discrete time tags is wrapped into a diffusion generation model accepting continuous time series as input by the DPM-Solver accelerated encoder, and a fast decoder for the diffusion model is built, after which, by step-by-step denoising optimization, image details are gradually added, and then a high-quality target image is generated.

3. Experimental Validation

In the process of side-scan sonar underwater object detection, factors such as image quality, target shadow, target background (topography and terrain), noise, and resolution can affect the accuracy of object detection. To verify the performance of the diffusion model in side-scan sonar sample augmentation, we designed a variety of comparison experiments. A diffusion model training set with three types of targets: shipwreck, container and pipe are built. The backgrounds for each class of targets were made as diverse as possible, containing different terrain and landform information. The training set samples contain various noise and resolution information. A total of 314 wrecks, 98 containers, and 503 pipelines were collected, part of which is shown in Figure 5. The hardware configuration used for model training was an Intel® Xeon® E5-2650 v4@2.20 GHz CPU and a GeForce GTX 1080Ti GPU. The software environment used is Pytorch 1.6.0, Cuda 10.1 and Python 3.7 on a Windows 10 operating system.



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

Figure 5. This is a sample training set for a part of the diffusion model. (**a**) Container; (**b**) pipelines; (**c**) shipwreck.

3.1. Evaluation Metrics

Image generation mainly evaluates the quality of generated images in terms of sharpness, diversity of features, and structural similarity. In this paper, we choose to evaluate the quality of generated images in terms of Fréchet inception distance (FID), kernel maximum mean discrepancy (MMD), peak signal-to-noise ratio (PSNR), learned perceptual image patch similarity (LPIPS), and structural similarity (SSIM). FID is used to calculate the distance between the real image set and the synthetic image set in the feature space. Firstly, the features of the two images are extracted separately using the inception network trained on the public large data set, then the feature space is modeled using the Gaussian model, and finally the distance is calculated based on the mean and covariance of the Gaussian model. MMD is based on the statistical test of maximum mean squared difference, which measures the similarity between two feature distributions by mapping the set of real images and the set of synthetic images to a kernel space with a fixed kernel function and then by computing the mean difference between the two distributions. PSNR can calculate the mean squared error between two images and then calculate the peak SNR to compare the training set and generated images for evaluation. SSIM is structurally similar, which defines structural information from the viewpoint of image composition as a property that reflects the structure of objects in a scene independently of luminance and contrast and models distortion as a combination of three different factors: luminance, contrast, and structure. The mean is used as an estimate of luminance, the standard deviation as an estimate of contrast, and the covariance as a measure of structural similarity. LPIPS is used to measure the difference between two images. The metric learns the inverse mapping of

the generated image to the real image, forcing the generator to learn the inverse mapping to reconstruct the real image from the fake image and prioritizing the perceived similarity between them, and LPIPS is more consistent with human perception. In this case, the value of SSIM ranges from 0 to 1, and the larger it is, the more similar the images are. The SSIM value is one if the two images are exactly the same. FID, MMD, and LPIPS are all smaller and better. The larger the PSNR value, the smaller the difference between the two images and the better the quality of the generated images.

3.2. Experimental Design and Image Generation

The image size set during the training of the diffusion model has a large impact on the image generation quality. The larger the image size, the more resources are consumed to train the model, and at the same time, the model needs a larger training set to learn more information. In order to verify the quality of the generated samples of the model under different image sizes, we set three different generated sample sizes during the model training, which are 64 × 64, 128 × 128, and 256 × 256. The model is saved every 5000 iterations during training and the corresponding training parameters are set for different training set model classes and different generation sizes. The parameters of each model are set as listed in Table 1.

Table 1. Diffusion model training parameters settings.

Group	Training Set Category	Batch Size	Image Size
T1	Shipwreck, Container, Pipeline	40	64 × 64
T2	Shipwreck	11	128 × 128
T3	Shipwreck	3	256 × 256

The GPU used in this training process is GeForce GTX 1080Ti, and because of the limited computing power of the graphics card, it is necessary to reduce the number of batches in training when training the large-size sample model. At the same time, the large-size sample generation needs to contain more images with more details and requires more iterations to ensure the quality of the generated samples.

Sample generation is performed using the model trained by T1, and the generation size of the image is 64 × 64, and the generation target category contains wrecks, containers, and pipelines. Some of the generated samples are shown in Figure 6.

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

Figure 6. This is a demonstration of the effect of multi-class image generation using the T1 model ((**a**) is the generated image of a container, (**b**) is the generated image of a pipe, and (**c**) is the generated image of a shipwreck).

As can be seen from Figure 6, the generated samples and the real side-scan sonar images have extremely strong similarity in style despite the small size of the generated

target. The morphology of the generated container samples is consistent with structural information, the sizes are diverse, and the relative positions of the target shadows and water column regions are consistent with side-scan sonar operation. For the generated pipeline samples, the target background presents different topographic and geomorphological information, the pipeline outline is clear, and the pipeline distribution on the seabed has extremely strong continuity, which is consistent with the pipeline characteristics. The resulting wreck samples are rich in the structure, attitude, and shape of the wreck targets.

Using the model trained by T2 for sample generation, the generation size is 128 × 128, the generation category contains only wrecks, and the generated samples are shown in Figure 7.

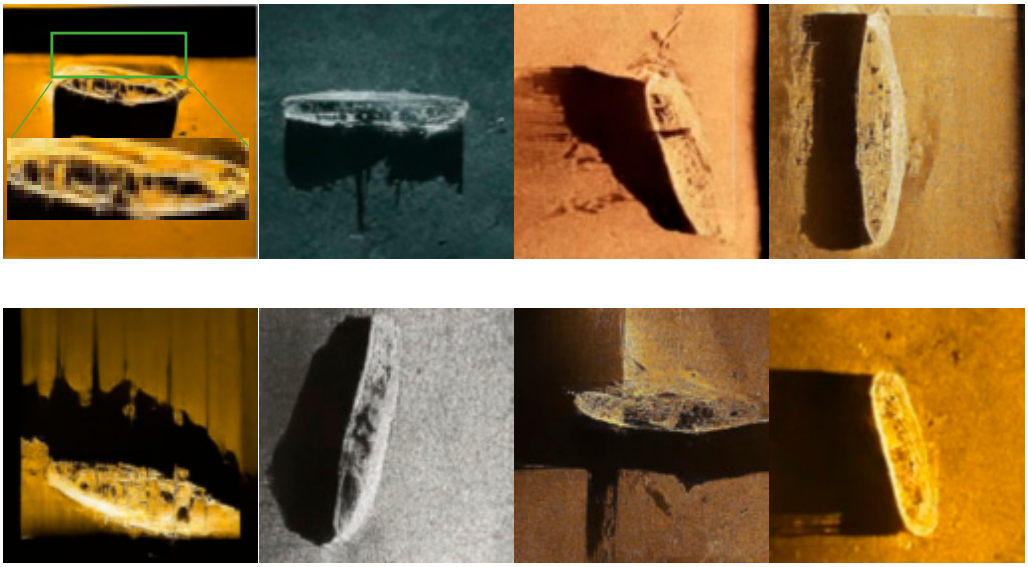


Figure 7. This is a 128 × 128 size image of the shipwreck target generated using the T2 model.

As can be seen in Figure 7, the wrecks in the generated samples have richer texture details. An enlarged view of the wreck target shows that the generated wreck has a detailed structure with a reasonable distribution of shadow and the extremely strong correlation between the structure and location of the target, which is consistent with the side-scan sonar image mechanism.

Using the model trained by T3 for sample generation, the generation size is 256 × 256, the generation category contains only wrecks, and the generated samples are shown in Figure 8.

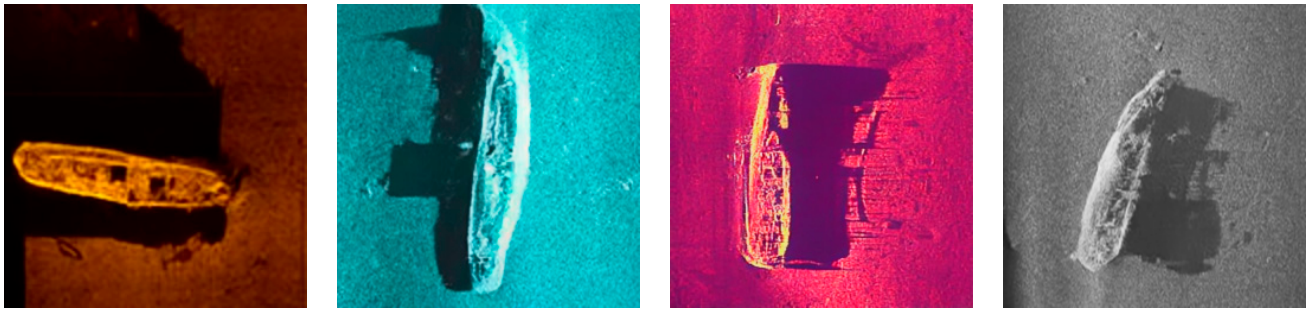


Figure 8. This is a 256 × 256 size image of the shipwreck target generated using the T3 model.

As can be seen from Figure 8, compared with the 64 × 64 and 128 × 128 size wreck targets, the 256 × 256 wreck has richer local details and clearer contours, and has a very high similarity to the real ship structure, and the geometric relationship between the generated

targets and shadows is consistent with the side-scan sonar imaging mechanism. The image background texture and noise are almost identical to the real side-scan sonar images, and the generated samples at this time visually meet the requirements of side-scan sonar sample augmentation.

3.3. Qualitative Analysis

Based on three sets of comparison experiments, FID, MMD, PSNR, SSIM, and LPIPS are computed for different classes and different sizes of generated images, where SSIM is better for values closer to 1, PSNR is better for larger values, and the remaining three metrics are better for smaller values. The final experimental results are shown in Table 2.

Table 2. Experimental results of different size models on FID, MMD, PSNR, SSIM and LPIPS.

Model	Target	FID	MMD	PSNR	SSIM	LPIPS
T1	Shipwreck	138.56	0.2357	11.1764	0.1753	0.3942
	Container	108.869	0.2324	10.0542	0.2512	0.426
	Pipeline	102.656	0.2343	15.1054	0.208	0.2921
T2	Shipwreck	153.59	0.1194	10.6517	0.1769	0.4415
T3	Shipwreck	153.75	0.0601	10.3241	0.18114	0.4969

As can be seen from Table 2, the FID evaluation index, T1, as a small size generation model, has higher generation quality. At the same generation size, the structure of containers and marine cables is simpler, and the distance between the real and synthetic images in the feature space is smaller. In the MMD index, the MMD score gradually decreases with the increase of the generated image size, which may be because the large size model has longer training time and learns more feature distributions of the real data. In the PSNR index, the container generation quality is better in T1. The scores of the remaining categories and T2 and T3 are more similar, which also proves that the generative quality of the generated images and the real images are more similar. In the SSIM score index, the scores achieved by the three models are lower, which indicates that there is a big difference in the structural similarity between the generated images and the real images. In the LPIPS index, it can be found that the scores of the three models are more similar and all of them are less than 0.5, which also indicates that there is a partial perceptual similarity between the generated image and the original image, and the generation effect is consistent with human perception.

3.4. Wreck Object Detection Model Training

To verify the feasibility and effectiveness of the generated samples in the training of the underwater object detection model. We use the DDIM+DPM-Solver to train the wreck sample generation model with a training set of 205 real side-scan sonar wreck samples. The wreck generation models of two sizes, 256 × 256 and 128 × 128, are trained respectively.

After that, two wreck sample generation models are used for wreck sample image generation. A total of 8765 augmented samples, 2971 of 256 × 256 size and 5794 of 128 × 128 size, were selected for the training of the side-scan sonar wreck object detection model. The test set is all real side-scan sonar images, which are not involved in the training process of the diffusion model. The labels of the training set and test set samples are automatically labeled by the neural network, and the control test groups are set as shown in Table 3.

For the above four groups of control experiments, the wreck object detection model was trained using YOLOv5 network, where batch size = 16 and input image size = 256 × 256. The rest are the default settings of the network. To verify generalization on augmented data, the model performance is evaluated using the object detection general metrics, prediction, recall, and mAP, and the model training results are shown in Table 4 for each group.

Table 3. Different control experimental settings.

Group	Training Set	Number	Test Set	Number
G1	SSS Images	205	SSS Images	81
G2	Generate images (128 × 128/256 × 256)	8765	SSS Images	81
G3	Generate image (128 × 128)	5794	SSS Images	81
G4	Generate image (256 × 256)	2971	SSS Images	81

Table 4. Results of different control tests.

Group	Training Set	Prediction	Recall
G1	SSS Images	0.888	0.84
G2	Generate images (128 × 128/256 × 256)	0.93	0.851
G3	Generate image (128 × 128)	0.929	0.702
G4	Generate image (256 × 256)	0.922	0.755

For G1, the model was trained with real side-scan sonar images, and the model accuracy is only 0.888 because of the limitation of the sample number, and the model performance is limited because of the limited number of samples in the training set. G2 is the model trained with total generated images, and at this time, the prediction, recall, and mAP all surpass the model trained with real SSS images, which verifies the feasibility of the method in this paper and effectiveness of the method in this paper, but for G3 and G4, the generated images of two sizes are trained separately, and the accuracy of the model decreases. The reason for this is that the generated data are automatically labeled by the network, which may lead to inaccurate labeling, and the data volume of the two sizes is small. Therefore, the model performance degrades when the two sizes are trained separately, while the large amount of data compensates for the impact of labeling errors on the model performance when they are trained together.

To further investigate the effectiveness of the wreck detection model trained by the generated data, 80 real SSS images were selected for detection using the above four groups of models, and the confidence of the detected wrecks was counted, and the results of the comparison experiments for each group are shown in Figure 9.

Figure 9a shows the detection results of model G1. The model was trained using the SSS wreck samples, and the detection accuracy of the model is more scattered, and the false detection rate is the highest due to the insufficient number of samples and limited representativeness of the training set, and the detection effect is poor. Figure 9b shows the detection results of model G2. The confidence of the model was trained using the total amplification samples for wreck detection and mostly stays around 0.9, with a relatively concentrated target confidence and the lowest false detection rate, and the model performs better. Figure 9c,d show the detection results of models T3 and T4; it is obvious that the confidence detected by the remaining two groups of group models is relatively scattered, and the confidence of detection is relatively low with a higher false detection rate. The generalization and effectiveness of the shipwreck detection model obtained by training on the total generated data is validated by comparison.

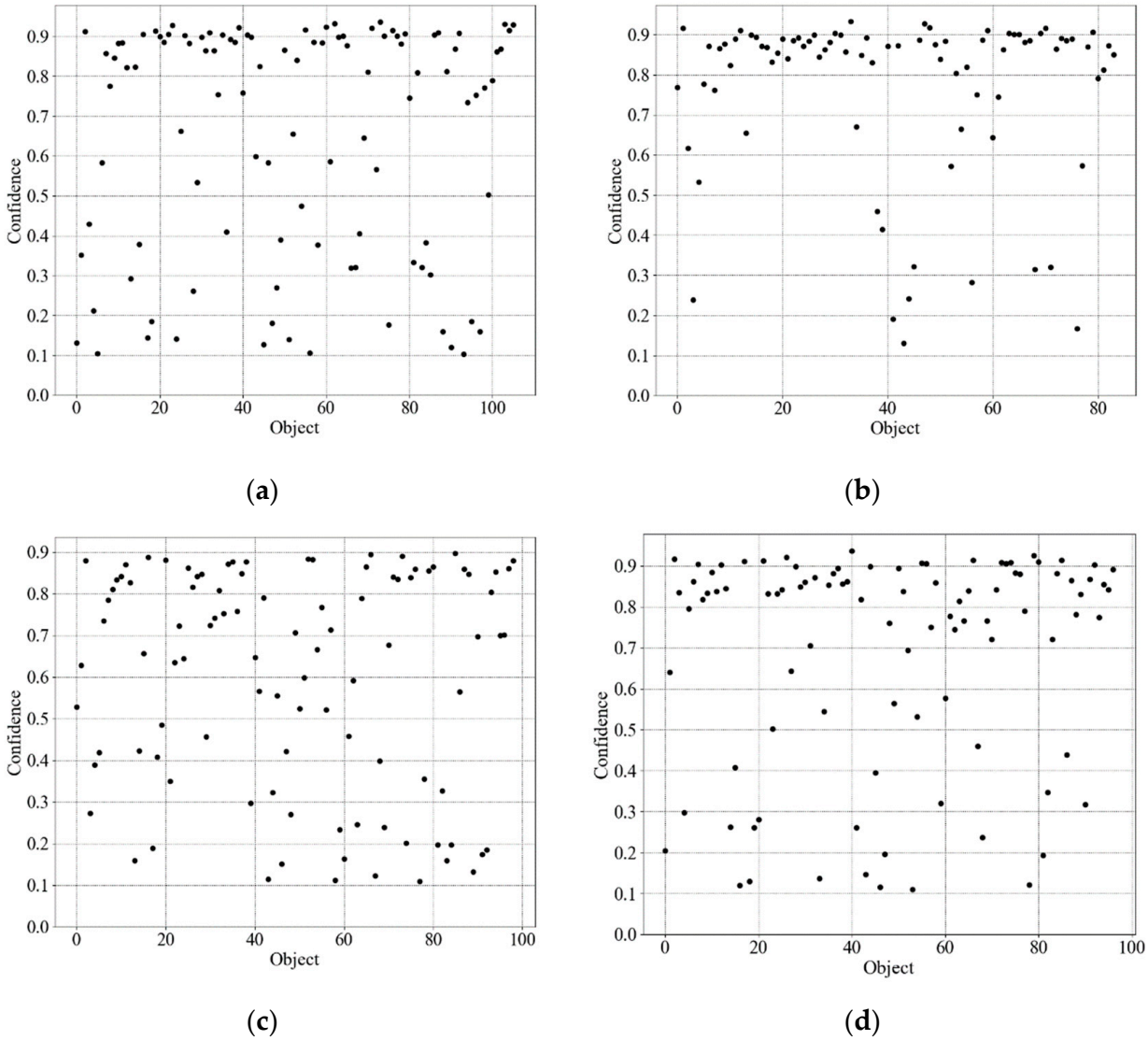


Figure 9. This is the detection result of 80 images of unfamiliar shipwrecks using four sets of models ((**a**) shows the detection results using the G1 model, (**b**) shows the detection results using the G2 model, (**c**) shows the detection results using the G3 model, and (**d**) shows the detection results using the G4 model).

4. Discussion

Deep-learning-based underwater object detection essentially involves extracting target features from a large training set. In this paper, we use a diffusion model to generate side-scan sonar images and perform the construction of a shipwreck detection model based on the generated samples. Although the generated data is visually extremely similar to the real data. However, we find that the amplified sample features (shape, pose, color, size, etc., of the wreck) are extremely similar to the training data. The amplified sample approximates a random combination between different features from multiple samples of the training data. This ensures to some extent the rationality of the generated images and enhances the diversity of the samples. In practical tasks, the larger the size of the generated images, the more training sets and resources are required and consumed. Therefore, we propose to build the corresponding generative model according to the actual requirements. In the generation process, although we take accelerated sampling, the sample generation speed of the diffusion model still needs to be further improved.

The generative process of the diffusion model starts from a pure noisy image and is optimized by successive sampling of the model to finally generate the target sample. In the

whole process, the generated images are affected by various factors, and the adjustment model input, random noise, and sampling steps can affect the generation results, so we hope to combine with the actual task when adopting our method, and we discuss the effects brought by different influencing factors on the generated samples separately.

4.1. Difference between Different Sampling Steps

The diffusion model starts from a pure noisy image, and then continuously denoises it step by step to finally obtain the target image. This process takes a lot of time. To this end, different sampling steps are set for the DDIM and DDIM+DPM-Solver, respectively, to compare the quality of generated images, and the generated samples are shown in Figure 10.

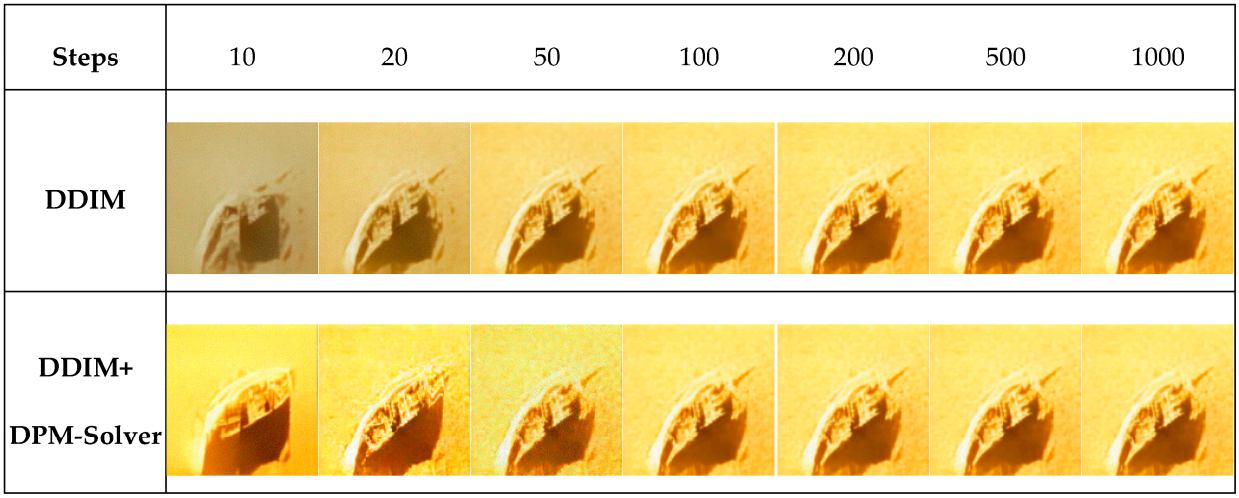


Figure 10. This is the effect of different sampling methods on the quality of the generated images.

As can be seen from Figure 10, the targets generated by the DDIM+DPM-Solver are more detailed and of higher quality targets than the DDIM at sampling steps 10 and 20. The difference between the samples generated by the two methods gradually decreases when the number of sampling steps exceeds 50. By comparison, it can be found that sample generation can be accelerated using DDIM+DPM-Solver, providing the possibility to generate a large number of side-scan sonar target samples.

4.2. Effect of Different Noise Inputs on Sample Generation

The diffusion model sampling starts from a pure noise image. Two different random noises will produce different images, and a fused image will be generated by generating a new noise by spherical linear interpolation of the two random noises. In Equation (5), we can see that it $\sigma_t \epsilon_t$ represents the effect of random noise on the generation results, and in Equation (6), we can change the value of random noise by adjusting the value of η. In order to explore the effect of different noise inputs on sample generation, we want to shield the random noise, Settings $\eta = 0$, eliminating the effect of random noise in the generation process. The generated sample is shown in Figure 11.

Each row of images in Figure 11 represents a set of results with correlated noise output. it can be seen that the samples generated by the initial noise with correlations have some similarity. This is because we eliminate the effect of random noise in the generation process, and the generation process is deterministic once the initial noise is determined. Therefore, for correlated initial noise, the final generation effect is also of some relevance.

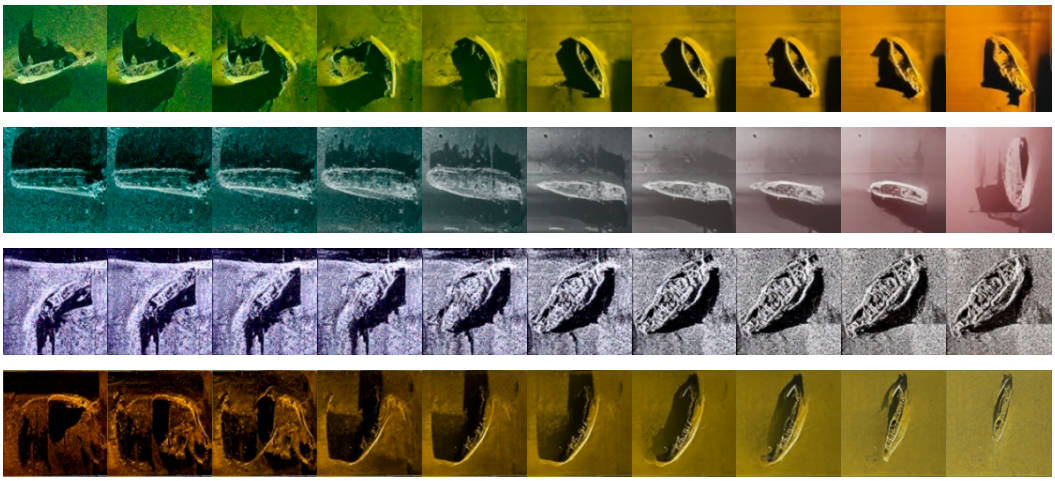


Figure 11. This is the output of the fusion noise in the diffusion model.

4.3. The Effect of Random Noise on Sample Generation

Diffusion model image generation is dominated by three parts, where for the random noise $\sigma_t \epsilon_t$, by setting different η can affect the value of σ_t, from which in turn affects the result of target generation. To further explore the impact of the change of random noise on the sample generation, the following control test was set for the different values of η. The generated results are shown in Figure 12.

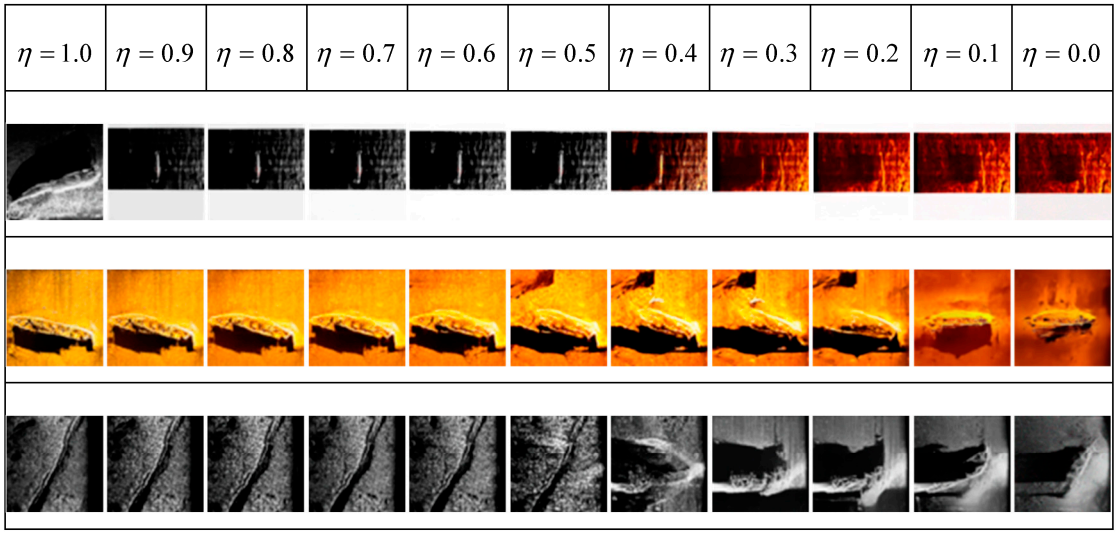


Figure 12. This is the effect of different η values on the generated samples.

As can be seen in the figure, the generated image changes locally at this point as the value of η changes. This is because the change in the value of η during the generation process affects the value of the random noise $\sigma_t \epsilon_t$ during the generation process. However, the sample generation also depends on the predicted x_0 and x_t. Therefore, the change in η will only have a local change on the sample.

5. Conclusions

In this paper, we present a method for generating samples of side-scan sonar images based on a diffusion model. The model generates entirely new samples with the same distribution based on a priori data distribution features using a diffusion model, and to improve the sample generation speed, we introduce an accelerated encoder. With our approach, a large number of side-scan sonar images with strong representations can be

generated quickly. Compared to other existing SSS image generation methods, our method does not require tedious data collection and data cleaning, and the implementation process is simple. Experimental results show that the generated data have excellent similarity to the real SSS images in terms of texture, noise, background, and resolution. At the same time, the sample generation results can be controlled by fine-tuning the model input and random noise during generation. Finally, the generated wreck samples and YOLOv5 network are used to train the underwater wreck detection model, and the generated data achieve high accuracy and low false-detection rate when comparing the SSS training results. Experimental results show that the problem of small number of samples and insufficient representation of SSS data can be effectively compensated using our method.

Author Contributions: Conceptualization, Z.Y.; methodology, Z.Y.; software, Z.Y.; validation, Z.Y., C.H. and J.Z.; formal analysis, C.H. and Y.Y.; investigation, Z.Y.; resources, Z.Y. and J.Z.; data curation, Z.Y.; writing—original draft preparation, Z.Y.; writing—review and editing, C.H. and Y.Y.; visualization, Z.Y.; supervision, C.H. and Z.Y.; project administration, J.Z. and Z.Y.; funding acquisition, J.Z and H.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by National Natural Science Foundation of China under grant 42176186 and the National Key R&D Program of China, grant number 2022YFC2808303.

Data Availability Statement: Access to the data will be considered by the authors upon request.

Acknowledgments: We would like to thank the editor and the anonymous reviewers for their valuable comments and suggestions that greatly improve the quality of this paper.

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

References

1. Yu, Y.; Zhao, J.; Gong, Q.; Huang, C.; Zheng, G.; Ma, J. Real-time underwater maritime object detection in side-scan sonar images based on transformer-YOLOv5. *Remote Sens.* **2021**, *13*, 3555. [CrossRef]
2. Cvikel, D.; Grøn, O.; Boldreel, L.O. Detecting the Ma'agan Mikhael B shipwreck. *Underw. Technol.* **2016**, *34*, 93–98. [CrossRef]
3. Xiaonan, C.; Minyan, L.; Longjun, H. Shipwreck statistical analysis and suggestions for ships carrying liquefiable solid bulk cargoes in China. *Proc. Eng.* **2014**, *84*, 188–194. [CrossRef]
4. Piccinelli, M.; Gubian, P. Modern ships voyage data recorders: A forensics perspective on the Costa concordia shipwreck. *Digit. Investig.* **2013**, *10*, 41–49. [CrossRef]
5. Ødegård, Ø.; Mogstad, A.A.; Johnsen, G.; Sørensen, A.J.; Ludvigsen, M. Underwater hyperspectral imaging: A new tool for marine archaeology. *Appl. Opt.* **2018**, *57*, 3214–3223. [CrossRef]
6. Greene, A.; Rahman, A.F.; Kline, R.; Rahman, M.S. Side scan sonar: A cost-efficient alternative method for measuring seagrass cover in shallow environments. *Estuar. Coast. Shelf Sci.* **2018**, *207*, 250–258. [CrossRef]
7. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Region-based convolutional networks for accurate object detection and semantic segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2016**, *38*, 142–158. [CrossRef]
8. Tan, C.; Sun, F.; Kong, T.; Zhang, W.; Yang, C.; Liu, C. A survey on deep transfer learning. In Proceedings of the Artificial Neural Networks and Machine Learning-ICANN 2018: 27th International Conference on Artificial Neural Networks, Rhodes, Greece, 4–7 October 2018; pp. 270–279.
9. Nayak, N.; Nara, M.; Gambin, T.; Wood, Z.; Clark, C.M. Machine learning techniques for AUV side-scan sonar data feature extraction as applied to intelligent search for underwater archaeological sites. *Field Serv. Robot.* **2021**, *16*, 219–233.
10. Nguyen, H.-T.; Lee, E.-H.; Lee, S. Study on the classification performance of underwater sonar image classification based on convolutional neural networks for detecting a submerged human body. *Sensors* **2019**, *20*, 94. [CrossRef]
11. Lee, S.; Park, B.; Kim, A. Deep learning from shallow dives: Sonar image generation and training for underwater object detection. *arXiv* **2018**, arXiv:1810.07990.
12. Huo, G.; Wu, Z.; Li, J. Underwater object classification in sidescan sonar images using deep transfer learning and semisynthetic training data. *IEEE Access* **2020**, *8*, 47407–47418. [CrossRef]
13. Ge, Q.; Ruan, F.; Qiao, B.; Zhang, Q.; Zuo, X.; Dang, L. Side-scan sonar image classification based on style transfer and pre-trained convolutional neural networks. *Electronics* **2021**, *10*, 1823. [CrossRef]
14. Li, C.; Ye, X.; Cao, D.; Hou, J.; Yang, H. Zero shot objects classification method of side scan sonar image based on synthesis of pseudo samples. *Appl. Acoust.* **2021**, *173*, 107691. [CrossRef]
15. Huang, C.; Zhao, J.; Yu, Y.; Zhang, H. Comprehensive Sample Augmentation by Fully Considering SSS Imaging Mechanism and Environment for Shipwreck Detection Under Zero Real Samples. *IEEE Trans. Geosci. Remote Sens.* **2021**, *60*, 1–14. [CrossRef]

16. Karras, T.; Laine, S.; Aittala, M.; Hellsten, J.; Lehtinen, J.; Aila, T. Analyzing and improving the image quality of stylegan. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 14–19 June 2020; pp. 8110–8119.
17. van den Oord, A.; Dieleman, S.; Zen, H.; Simonyan, K.; Vinyals, O.; Graves, A.; Kalchbrenner, N.; Senior, A.; Ka-vuk-cuoglu, K. WaveNet: A generative model for raw audio. *arXiv* **2016**, arXiv:1609.03499.
18. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. *Generative Adversarial Nets in Advances in Neural Information Processing Systems (NIPS)*; Curran Associates, Inc.: Red Hook, NY, USA, 2014; pp. 2672–2680.
19. van den Oord, A.; Kalchbrenner, N.; Kavukcuoglu, K. Pixel recurrent neural networks. *arXiv* **2016**, arXiv:1601.06759.
20. Rezende, D.J.; Mohamed, S. Variational inference with normalizing flows. *arXiv* **2015**, arXiv:1505.05770.
21. Kingma, D.P.; Welling, M. Auto-Encoding variational bayes. *arXiv* **2013**, arXiv:1312.6114v10.
22. Bore, N.; Folkesson, J. Modeling and Simulation of Side scan Using Conditional Generative Adversarial Network. *IEEE J. Ocean. Eng.* **2021**, *46*, 195–205. [CrossRef]
23. Jiang, Y.F.; Ku, B.; Kim, W.; Ko, H. Side-Scan Sonar Image Synthesis Based on Generative Adversarial Network for Im-ages in Multiple Frequencies. *IEEE Geosci. Remote Sens. Lett.* **2021**, *18*, 1505–1509. [CrossRef]
24. Brock, A.; Donahue, J.; Simonyan, K. Large scale GAN training for high fidelity natural image synthesis. *arXiv* **2018**, arXiv:1809.11096.
25. Karras, T.; Laine, S.; Aila, T. A Style-Based generator architecture for generative adversarial networks. *arXiv* **2018**, arXiv:1812.04948.
26. Sohl-Dickstein, J.; Weiss, E.; Maheswaranathan, N.; Ganguli, S. Deep unsupervised learning using nonequilibrium ther-modynamics. In Proceedings of the International Conference on Machine Learning, Lille, France, 6–11 July 2015; pp. 2256–2265.
27. Ho, J.; Jain, A.; Abbeel, P. Denoising diffusion probabilistic models. *arXiv* **2020**, arXiv:2006.11239.
28. Song, J.; Meng, C.; Ermon, S. Denoising diffusion implicit models. *arXiv* **2020**, arXiv:2010.02502.
29. Lu, C.; Zhou, Y.; Bao, F.; Chen, J.; Li, C.; Zhu, J. DPM-Solver: A Fast ODE Solver for Diffusion Probabilistic Model Sam-pling in Around 10 Steps. *arXiv* **2022**, arXiv:2206.00927.
30. Atkinson, K.; Han, W.; Stewart, D.E. *Numerical Solution of Ordinary Differential Equations*; John Wiley & Sons: Hoboken, NJ, USA, 2011; Volume 108.
31. Hochbruck, M.; Ostermann, A. Exponential integrators. *Acta Numer.* **2010**, *19*, 209–286. [CrossRef]
32. Ronneberger, O.; Fischer, P.; Brox, T. U-net: Convolutional networks for biomedical image segmentation. In Proceedings of the Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, 5–9 October 2015; Proceedings, Part III 18. Springer International Publishing: Berlin/Heidelberg, Germany, 2015; pp. 234–241.
33. Xiao, X.; Lian, S.; Luo, Z.; Li, S. Weighted res-unet for high-quality retina vessel segmentation. In Proceedings of the2018 9th International Conference on Information Technology in Medicine and Education (ITME), Hangzhou, China, 19–21 October 2018; pp. 327–331.
34. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.