remote sensing

Article

SACuP: Sonar Image Augmentation with Cut and Paste Based DataBank for Semantic Segmentation

Sundong Park , Yoonyoung Choi and Hyoseok Hwang *

Department of Software Convergence, Kyung Hee University, Yongin 17104, Republic of Korea; sundong@khu.ac.kr (S.P.); choiyy0313@khu.ac.kr (Y.C.)
* Correspondence: hyoseok@khu.ac.kr

Abstract: In this paper, we introduce Sonar image Augmentation with Cut and Paste based DataBank for semantic segmentation (SACuP), a novel data augmentation framework specifically designed for sonar imagery. Unlike traditional methods that often overlook the distinctive traits of sonar images, SACuP effectively harnesses these unique characteristics, including shadows and noise. SACuP operates on an object-unit level, differentiating it from conventional augmentation methods applied to entire images or object groups. Improving semantic segmentation performance while carefully preserving the unique properties of acoustic images is differentiated from others. Importantly, this augmentation process requires no additional manual work, as it leverages existing images and masks seamlessly. Our extensive evaluations contrasting SACuP against established augmentation methods unveil its superior performance, registering an impressive 1.10% gain in mean intersection over union (mIoU) over the baseline. Furthermore, our ablation study elucidates the nuanced contributions of individual and combined augmentation methods, such as cut and paste, brightness adjustment, and shadow generation, to model enhancement. We anticipate SACuP's versatility in augmenting scarce sonar data across a spectrum of tasks, particularly within the domain of semantic segmentation. Its potential extends to bolstering the effectiveness of underwater exploration by providing high-quality sonar data for training machine learning models.

Keywords: forward-looking sonar image; data augmentation; semantic segmentation; deep learning

check for updates

Citation: Park, S.; Choi, Y.; Hwang, H. SACuP: Sonar Image Augmentation with Cut and Paste Based DataBank for Semantic Segmentation. *Remote Sens.* **2023**, *15*, 5185. https://doi.org/10.3390/rs15215185

Academic Editor: Andrzej Stateczny

Received: 18 September 2023
Revised: 25 October 2023
Accepted: 27 October 2023
Published: 31 October 2023

1. Introduction

Deep neural networks (DNNs) have achieved remarkable growth with the emergence of AlexNet [1], with convolutional neural networks (CNNs), and leading to VGG [2] with deeper layers and ResNet [3] with much deeper layers. Through the development of DNNs, transformers [4] have emerged in natural language processing and led to bidirectional encoder representations from transformers (BERT) [5] and GPT-3 [6]. In computer vision, great progress has been made with the emergence of regions with CNN features (R-CNN) [7] and you only look once (YOLO) [8] in the detection area, as well as in the emergence of fully convolutional networks (FCNs) [9], U-Net [10], and Mask R-CNN [11] in the segmentation area. In the field of computer vision with DNNs, perception, such as classification, detection, and segmentation, has been a major contributor to the realization of visual intelligence in robots as well as autonomous vehicles. Among these, semantic segmentation plays an essential role in unmanned mobile vehicles and remote sensing, as it provides detailed location and class information about the environment sensed by the mobile agent [12,13].

The application of DNNs has advanced the performance of unmanned underwater exploration. In particular, it is mainly used to find objects using sonar in the water, such as by detection and segmentation [14,15]. Underwater environments make detecting objects using cameras quite difficult due to light absorption and scattering [16]. Therefore, sensing in underwater environments relies on sonar sensors that use sound waves to detect and locate underwater objects, which are particularly useful in underwater environments where light is not transmitted well.

However, achieving performance improvements using DNNs in sonar-based applications lags behind the advances seen in camera-based systems [17]. This discrepancy arises from the inherent challenges associated with collecting data in underwater environments using sonar technology, primarily due to accessibility issues that result in limited datasets. Furthermore, the specialized equipment required for sonar-based data collection significantly escalates the costs and complexity of underwater research. The resulting insufficient data can constrain the performance of the model and potentially lead to overfitting.

Similarly, with the increasing size of DNN architectures in recent years, demand for larger datasets is increasing [18]. Therefore, the creation of high-quality datasets has become a significant research area to improve the performance and capabilities of DNNs. The benefits of having a large and diverse dataset are enormous, including avoiding overfitting and improving models' generalization ability. Because there is a limit to obtaining labeled data in real life, to obtain additional data a number of studies have recently been conducted to improve the generalization of models using small amounts of data such as data adaptation [19], few-shot learning [20], and data generation [21].

However, because these methods utilize data from other domains, it is difficult to obtain performance improvements when data in the sonar domain are insufficient. On the other hand, data augmentation [22] is effective in the sonar domain because it expands data using existing sonar data. Data augmentation has become an important technology in recent years for addressing the challenge of limited data in deep learning. Commonly used data augmentation methods include random flipping, rotation, and scaling of images. Methods such as CutOut [23] to drop out data and CutMix [24] to mix data can be used to improve the robustness of the model. In addition, generative adversarial networks (GANs) [25] have emerged as a powerful tool for data augmentation, creating new and realistic data samples similar to the original data. In remote sensing, augmentation methods such as random spatial shuffles [26] and manifold augmentation [27] have been studied recently. Data augmentation can significantly improve the generalization and robustness of deep neural network models by increasing the size and diversity of available datasets.

While traditional data augmentation is an effective method in image-based semantic segmentation, traditional data augmentation methods designed for camera images have certain limitations when applied to sonar imaging. Sonar images work on fundamentally different principles from visual cameras, as they are based on using sound waves rather than light. This results in a sonar image characterized by high noise and low resolution compared to visual images [16]. Moreover, sonar images are created on a depth basis, unlike cameras. In addition, when applying traditional augmentation to sonar images, shadows of sonar images can cause problems with the direction of the object, which can consequently impair the stability of the data. This means that traditional augmentation methods are not suitable for sonar images, and that data augmentation technologies specialized for sonar images should be explored. These methods should consider the characteristics of sonar imaging, such as depth-based characteristics of data, noise levels, and shadow effects. By designing augmentation methods tailored to these unique characteristics, it is possible to improve the robustness and generalization capabilities of DNN models in sonar-based semantic segmentation tasks. This means that traditional augmentation methods are not suitable for sonar images, and that data augmentation methods specific to sonar images should be explored to ensure reliable and accurate data and to improve the robustness and generalization of DNN models. These limitations highlight the need for specialized data augmentation methods tailored to sonar imaging that can improve the robustness and generalization of DNN models applied to this domain.

Recently, many studies have been conducted on sonar image segmentation. Several methods exist to directly improve sonar image segmentation and segmentation performance through data augmentation. One way to improve segmentation is morphological reconstruction combined with the level set method (MRLSM) [28], which improves upon the level set method [29] to perform sonar image segmentation. Deep learning-based methods include feature pyramid U-Net with attention (FPUA) [30], which features an improved semantic segmentation network architecture. Recently, following the introduction of the segment anything model (SAM) [31], there have been attempts to apply SAM to sonar [32]. Another approach that has improved sonar image segmentation performance through data augmentation is augmentation using the pix2pix image translation method [33] for sonar image segmentation [34].

In this paper, we propose Sonar image Augmentation with Cut and Paste based DataBank for semantic segmentation (SACuP). Our goal is to overcome the limitations of existing data augmentation methods when applied to the sonar domain. We seek to solve the problem of existing augmentation techniques being difficult to use in the sonar context. In order to increase the diversity of augmentation and improve the performance of segmentation, data were extracted by object and stored in DataBank. The DataBank images were attached based on the collected real data and shadows were artificially generated to augment them realistically. The proposed pipeline consists of several steps. First, we use images and masks to extract the backgrounds and objects from images. Second, we inpaint the background from which the object was removed. Third, we find a random location in the background at which to insert the object and insert it in such a way that it does not overlap. Finally, to create a realistic image, we artificially draw shadows generated from sonar images. Using this pipeline, we aim to improve the performance and generalization of deep neural network models applied to sonar tasks such as semantic segmentation. The contributions of this paper are as follows:

- We propose a novel data augmentation pipeline for sonar imaging applications that uses a cut-and-paste-based DataBank approach for segmentation operations.
- Our proposed method creates a DataBank using only existing images and masks, requiring no additional work and preserving the characteristics of sonar noise.
- We show that the proposed method improves performance when using only real data for training as well as when using other augmentation methods.

2. Related Works

2.1. Semantic Segmentation

Semantic segmentation is a computer vision task involving dividing an image into several regions or pixel-wise classifications with predefined category labels. Recently, many semantic segmentation methods using deep learning have been studied. Long et al. [9] proposed full convolution networks, the beginning of semantic segmentation based on deep learning. Most semantic segmentation models are based on FCNs, which are networks that enable end-to-end pixel-wise classification through a complete convolutional layer. Ronneberger et al. [10] proposed a U-shaped design that combines a contracting encoder and an extended decoder path. Currently, U-Net is a widely adopted architecture for image segmentation. In the ongoing pursuit of more advanced semantic segmentation methods, the field has transitioned to the DeepLab family of models. Chen et al. [35] proposed DeepLabV3, which is a semantic segmentation model that employs atrous convolutions to capture multi-scale features and achieve accurate object boundaries. Furthermore, Chen et al. [36] proposed DeepLabV3+, which has an encoder–decoder structure, by adding a decoder to DeepLabv3 and applying the depthwise separable convolution to the atrous spatial pyramid pooling and decoder modules. Various encoders, such as VGG [2], ResNet [3], and EfficientNet [37], are used as encoders in semantic segmentation models.

2.2. Data Augmentation in 2D Images

Data augmentation is a method in deep learning that combats overfitting by expanding the training dataset, especially when data are limited. The quality of the dataset directly influences model training, leading to ongoing research into data augmentation. Image data augmentation techniques include both traditional approaches and deep learning-based approaches. Traditional methods include flipping, color jittering, random cropping, rotation, and translation [38]. Furthermore, methods such as CutOut [23] and CutMix [24] can increase the generalization and robustness of models. CutOut, proposed by DeVries and Taylor, is a regularization method that randomly masks inputs to improve the robustness of convolutional neural networks and improve overall model performance. Yun et al. introduced CutMix, a method that surpasses CutOut in efficiently utilizing information. By cutting portions of learning images and integrating parts from other images, CutMix enhances data diversity and boosts generalization. Traditional augmentation methods have a lot of information available in the data, but the data may not be well utilized. Zhang et al. introduced ObjectAug [39], a method designed for semantic segmentation that emphasizes augmenting object units. Additionally, recent studies have actively explored copy-and-paste methods [40,41], which offer a straightforward yet potent approach to data augmentation. However, the ObjectAug and copy-and-paste methods cannot reflect sonar characteristics. As deep learning has evolved, there has been an increase in the variety of model architectures available, which in turn has led to a greater diversity of methods for data augmentation. Deep learning-based data augmentation methods include methods such as generative adversarial networks (GANs) [25], variational autoencoders (VAEs) [42], and neural style transfer [43]. These methods can generate new samples of data or modify existing ones. However, deep learning-based data augmentation methods do not work well on small datasets with insufficient amounts of data.

2.3. Data Augmentation in Sonar Images

As deep learning is increasingly applied to a wider range of fields, the domains of perception tasks such as object detection and semantic segmentation are expanding to include new types of data, including sonar images. This has led to a need for large amounts of sonar image data to improve model performance and stability. Gathering sonar image datasets is challenging due to limitations in equipment and personnel which hinder the collection of extensive data. To tackle this problem, researchers are delving into methods for augmenting sonar image data. While traditional image augmentation techniques such as flipping, scaling, rotation, and shifting can be applied to generate sonar images, there are geometric constraints associated with using these conventional methods. This means that dedicated methods for augmenting the sonar image is required. One of the methods to overcome geometric constraints is to use simulators to generate diverse datasets. This approach is a more efficient way to obtain data than collecting it directly from the real world. There are several methods [44–48] of underwater sonar image data augmentation using simulators such as Gazebo [49]. In addition, other methods [50,51] use generative models such as GANs. Lee et al. [34] proposed a method of sonar image augmentation using image-to-image translation [33]. However, these methods have limitations, as simulation-based techniques necessarily involve a gap between simulated and real-world data. In particular, GAN-based methods remain sensitive to data quality, and show a tendency to fall into local minima.

3. Methods

In this section, we propose SACuP, a method of augmentation that preserves the characteristics of sonar images. Our method creates a DataBank by extracting background images, object images, and information to exploit small amounts of data. Each object's position and shadow information are calculated to create one item of statistical information for each class. When the DataBank is built, it can combine background and object images based on realistic statistical information to generate a realistic dataset that retains the characteristics of the source domain. The proposed pipeline consists of four steps: DataBank generation, background inpainting, object insertion, and shadow generation. The first step is to separate objects and backgrounds from source images with ground truth labels, then extract and store the information used to generate the data. Second, while extracting the object, it needs to be inpainted in the background with a hole. Then, the object and background are synthesized at random locations to obtain an image and a corresponding mask. Finally, to increase the realism of the synthetic image, artificial shadows are added to make it more closely resemble the source domain. The overall method is illustrated in the Figure 1. Code is available at https://github.com/AIRLABkhu/SACuP, accessed on 1 September 2023.

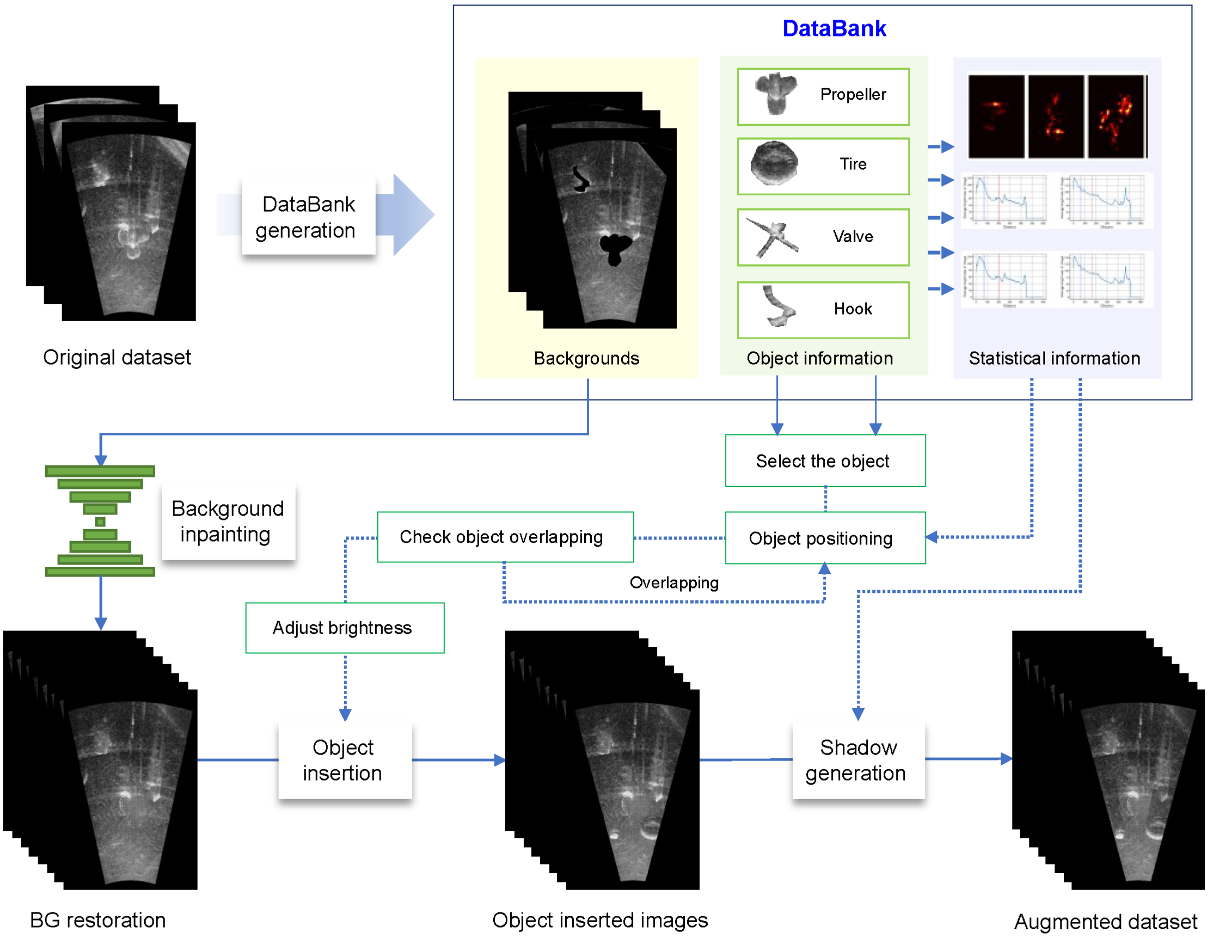


Figure 1. Overview of sonar image augmentation with cut and paste-based DataBank procedure.

3.1. DataBank Generation

The DataBank serves as a repository for components essential in generating synthetic data, encompassing background images, object images, and statistical information such as location and shadow information. DataBank generation is divided into two processes, namely, image extraction and information extraction. Image extraction uses images and semantic segmentation masks to separate objects and backgrounds. In the information extraction process, statistical information is collected, specifically, the location and shadow information of the extracted object. The process of DataBank generation is illustrated in the Figure 2.

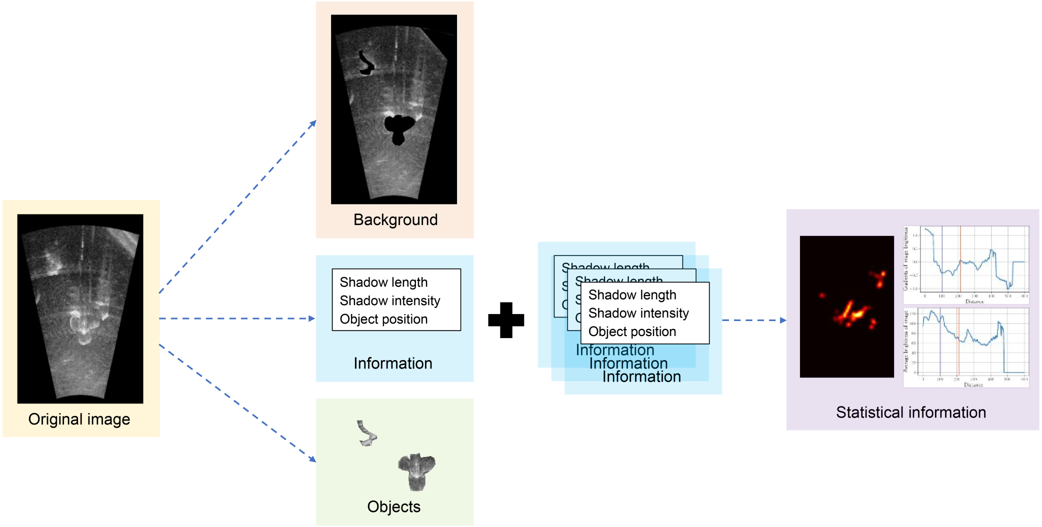


Figure 2. The process of DataBank generation. Objects and backgrounds are separated using images and ground truth, and information from extracted objects is collected to obtain statistical information.

3.1.1. Images Extraction

Datasets for semantic segmentation typically include images paired with ground truth annotations, which are called semantic segmentation masks. This ground truth allows for the direct extraction of foreground objects and backgrounds from the images. Although semantic segmentation provides class-type information, objects with the same values can have different shapes. Therefore, if the masks are adjacent to each other, we identify them as independent objects and extract them. This consolidation ensures that the extracted objects accurately represent distinct entities within the image. Instead of only extracting the individual objects, we apply a dilation operation to expand their boundaries to encompass more of the surrounding background. With each dilation iteration, this region gradually grows larger, providing a buffer around the objects. These extracted and dilated objects are then saved to the DataBank for each class separately, facilitating subsequent processing and analysis.

3.1.2. Information Extraction

During the object extraction process, the object's center point is initially determined. This center point's coordinates are then used to generate a heatmap for each object class, which is subsequently stored in the DataBank. Simply separating objects from the background overlooks shadows, which are essential properties of sonar images. To address this, we extract shadow information concurrently to restore these shadowed regions. Considering that shadows typically extend away from the sonar source, we initially transform the polar coordinate system used by the sonar into an orthogonal coordinate system. This transformation results in shadows appearing as square-shaped regions. To ensure that shadows are discernible in the noisy sonar images, we overlap all the shadows of the objects. Because the objects in the same class are all similar in shape and size, we incorporate information from the objects in order to more accurately estimate the shadows. We then define the shadow area to the point where the gradient is the largest, where the brightness value changes the most as the shadow ends. Because non-shadow areas can be included at this stage, only darker images are overlapped when comparing the previously obtained shadow area with the background to obtain the shadow area again. Using this refined shadow area, we compute various shadow characteristics, such as shadow length, shadow intensity, and the standard deviation of sonar noise. These shadow-related data are stored in the DataBank for further analysis. This process excludes the chain, valve, and wall classes, which have few shadows, and the shampoo-bottle and standing-bottle classes, which are already masked by shadows in the dataset.

3.2. Background Inpainting

We address the challenge of filling the void left after removing objects from sonar images using pix2pix [33], a kind of conditional GAN that uses images and labels as inputs to generate images via unsupervised learning. Our primary hurdle was finding an appropriate background to replace the removed object in the absence of reliable ground truth data. To tackle this issue, we devised a novel strategy. Instead of relying on the original mask and paired image, we introduced a new mask that did not overlap with the previous one. We then extracted the background regions devoid of the object and paired them with the original background image. This approach allowed us to generate a fitting background for the vacant area left by the removed object. Importantly, this method does not necessitate precise ground truth data, as it teaches the model how to generate suitable backgrounds for object-free areas. An example of the background inpainting is illustrated in the Figure 3.

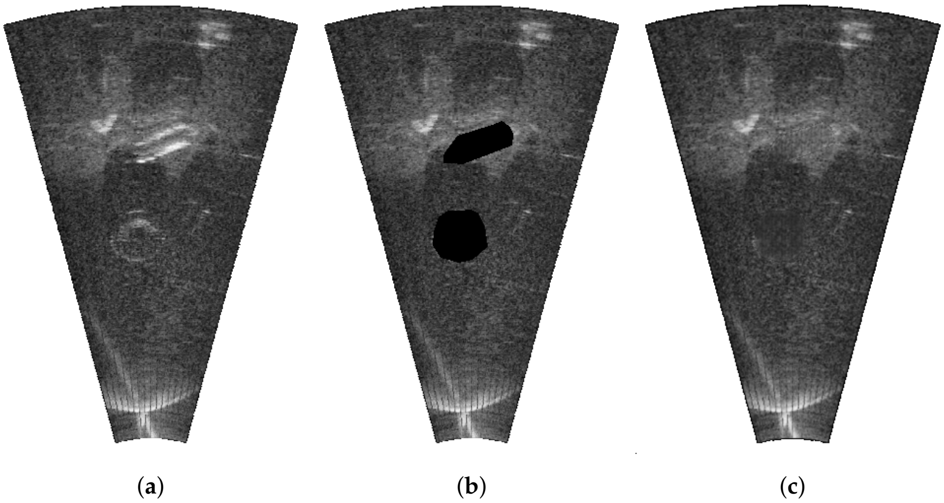


Figure 3. Example of the background inpainting step: (**a**) the original image, (**b**) background with holes after removing objects, and (**c**) the inpainted background with filled hole.

3.3. Object Insertion

Object insertion consists of four steps: selecting the object, object positioning, checking object overlapping, and adjusting the brightness. In the first step, objects are randomly selected from the DataBank based on the proportions of the classes. Next, object positioning is performed according to the distribution using the heatmap information of the class in the DataBank. Object overlapping is checked to ensure that there are no impossible images in which objects overlap with other objects. If objects overlaps, the object positioning step is repeated. In the last step, the brightness of the object is adjusted according to the brightness of the background to ensure that the brightness of the object to be added and the brightness of the target background are the same.

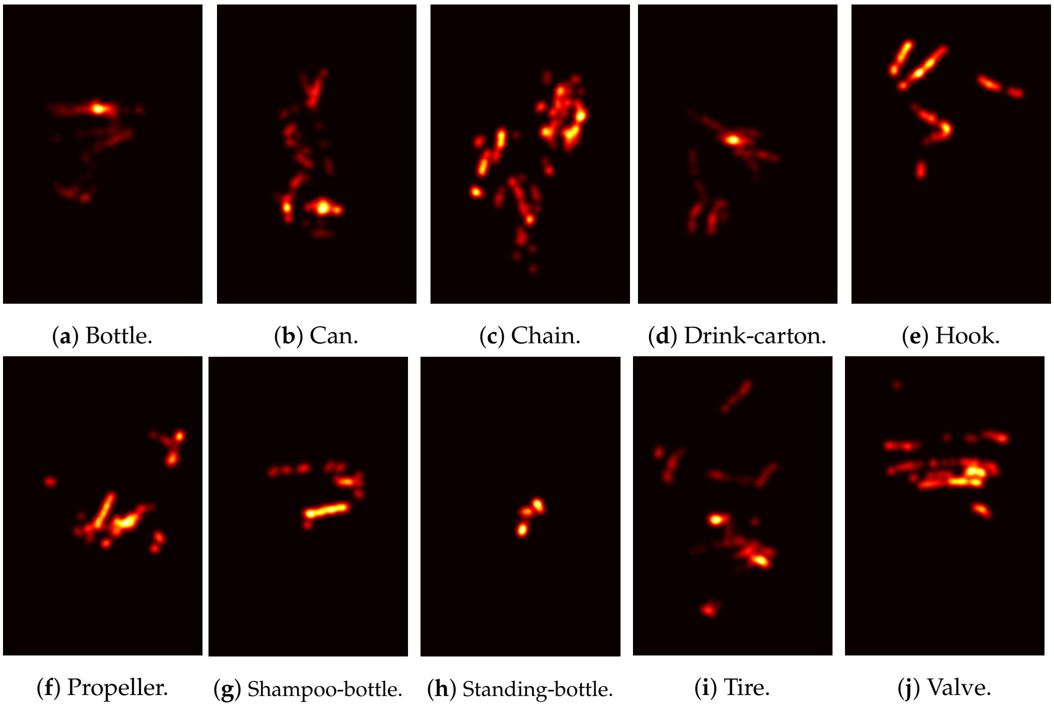
3.3.1. Selecting the Object

The object selection process consists of two steps, namely, selecting a class and selecting an instance. Each class that makes up a dataset has a different number of classes. To maintain this, we randomize and select the classes according to the ratio of classes in the dataset.

Before adding an object to a background, we first choose the class to be generated. The probability of selecting a given class can be adjusted to address imbalances in the dataset. This probability adjustment addresses class imbalance issues by assigning higher probabilities to classes that are underrepresented in the original dataset that is being augmented. When the class has been determined based on these probabilities, the next step is to randomly select an object from the DataBank that belongs to the chosen class. This process helps maintain diversity in the generated scenes while ensuring that objects are selected in a balanced and controlled manner according to the desired class distribution.

3.3.2. Object Positioning

The DataBank, contains a heatmap representing the positional distribution for each class. This heatmap is derived by analyzing the position of objects in existing datasets and constructing a probability distribution based on the heatmap generated for individual objects. To clarify, these heatmaps are essentially probability distributions generated for each object class. When selecting a two-dimensional position, we rely on these class-specific heatmaps to ensure that the chosen positions align with the likelihood of objects appearing in the sonar image. This process carries out a random selection guided by the probability of the heatmaps. The heatmap is shown in Figure 4.


(**a**) Bottle. (**b**) Can. (**c**) Chain. (**d**) Drink-carton. (**e**) Hook.

(**f**) Propeller. (**g**) Shampoo-bottle. (**h**) Standing-bottle. (**i**) Tire. (**j**) Valve.

Figure 4. Heatmaps representing the probability distributions for each object class. The position of the object is randomly selected depending on the probability in the heat map.

3.3.3. Check Object Overlapping

This step checks whether the inserted object overlaps with other objects. In the previous step, the positioning of objects in the background is random; thus, there is a possibility of objects overlapping. Before inserting the object, we temporarily generate a mask to verify whether it overlaps with the masks of other objects in the background where it is being placed. If the masks do not overlap, the test is passed. However, if there is overlap, the test fails. In the event of an unsuccessful test, the process reverts to the object positioning step and all steps are repeated until a synthesized image that meets the criteria is achieved. This iterative approach ensures that, while objects may overlap due to random placement, there are mechanisms in place to detect and prevent such occurrences, thereby enhancing the quality of the generated data.

3.3.4. Adjusting Brightness

In this process, when the brightness of the object is different from the brightness of the target background to be added, the brightness is adjusted to bring it into harmony with the background. The pasted object is partially overlapped because the surrounding background was extracted together through dilation. The average brightness of this overlapping background is calculated and the brightness of the object is adjusted based on the difference. To adjust the brightness of the surrounding background except for the object, we select I, the surrounding background of the object to be pasted, as follows:

$$I = \{O_{extracted} | M = 0\}, \tag{1}$$

where $O_{extracted}$ is the extracted object image and M is the mask. Next, the surrounding background B of the target background is selected as follows:

$$B = \{O_{background} | M = 0\}, \tag{2}$$

where $O_{background}$ is the target background image and M is the mask. The average of the surrounding background of the object to be pasted is

$$\mu_I = \frac{1}{H}\frac{1}{W}\sum_{i \in H}\sum_{j \in W} I(i,j), \tag{3}$$

where H is the height of the image and W is its width. The average of the surrounding background of the target background is

$$\mu_B = \frac{1}{H}\frac{1}{W}\sum_{i \in H}\sum_{j \in W} B(i,j). \tag{4}$$

The difference d between the two means is calculated as follows:

$$d = \mu_B - \mu_I. \tag{5}$$

We obtain $O_{adjusted}$ by adjusting the brightness:

$$O_{adjusted} = O_{extracted} + d. \tag{6}$$

The shampoo-bottle and standing-bottle class images are shadows; thus, when adjusting brightness they lose information when it becomes too dark and are unrealistic when it becomes too bright. Therefore, for these classes we adjust the brightness based on the ratio instead of the difference. The ratio r between the two means is calculated as follows:

$$r = \frac{\mu_B}{\mu_I}. \tag{7}$$

We obtain $O_{adjusted}$ by adjusting the brightness:

$$O_{adjusted} = O_{extracted} \times r. \tag{8}$$

After all the processes are completed, the object is pasted to the target background. The object is alpha-blended using a Gaussian filter to harmonize with the surrounding background. The final combined Synthetic Image is

$$Synthetic\ Image = \alpha \times O_{adjusted} + (1 - \alpha) \times O_{background}, \tag{9}$$

where α is the weight of the transparency produced by the Gaussian distribution. The process of adjusting brightness is illustrated in the Figure 5.

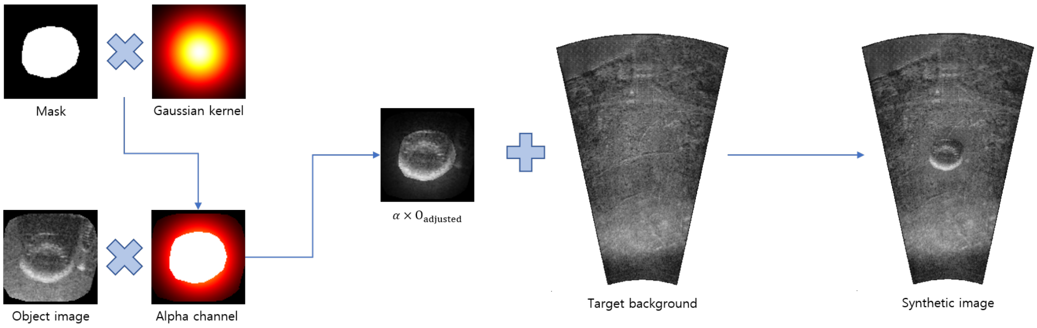
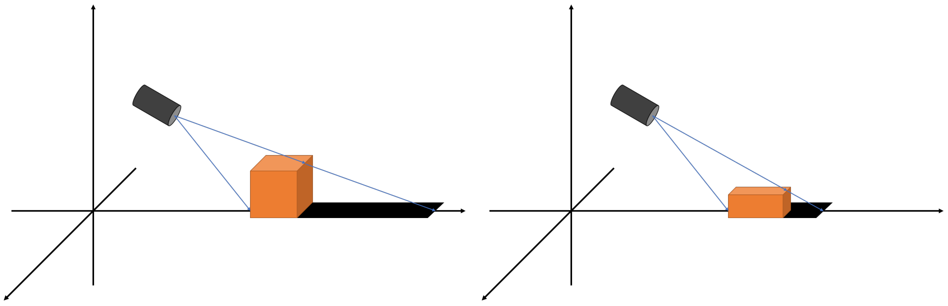


Figure 5. The process of adjusting the brightness and inserting it into the background. First, we create a Gaussian kernel for smooth insertion into the background. At this point, the object part is modified by overlapping the object mask to ensure that it is not lost by the Gaussian kernel. The object is then alpha-blended with the background using a Gaussian filter to create a synthetic image.

3.4. Shadow Generation

Because sonar images are tilted at a certain height and represented by converting data taken as if viewed from above, dark areas similar to shadows appear in blind spots where sound waves do not reach. Unlike the original dataset image, the inserted object has no shadow. To create a more realistic synthetic dataset, this step creates artificial shadows.

The direction of the shadow is created away from the sonar. Therefore, as shown in Figure 6, the length of the shadow varies depending on the object. The information in the semantic segmentation dataset, which has only images and masks, does not include the location and angle between the sonar and the object. Therefore, we calculate the position and angle between the sonar and the object, assuming that the lower middle point is the sonar's position, where the field of view of the sonar image converges into one place. To maintain the noise characteristics of the sonar, shadows are not created to have a value of zero. Shadows are created differently for each class using information stored in the DataBank. The DataBank stores the shadow length, shadow intensity, and standard deviation of the shadow noise. To artificially create shadows, we set the shadow area based on the shadow length in the DataBank in the direction away from the sonar then fill the area with Gaussian noise, with the shadow intensity as the mean and the standard deviation of the noise as the standard deviation. Because the shadow's intensity represents the rate at which it becomes darker than the surrounding background, multiplying it by the background darkens and preserves the sonar's characteristics. As in Figure 6, shadows are formed longer when an object has a long height. The comparison of shadow generation is shown in Figure 7.



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

Figure 6. Shadows appear differently depending on the height of the object: (**a**) when an object has a long height, the shadow appears long; (**b**) when an object has a short height, the shadow appears short.

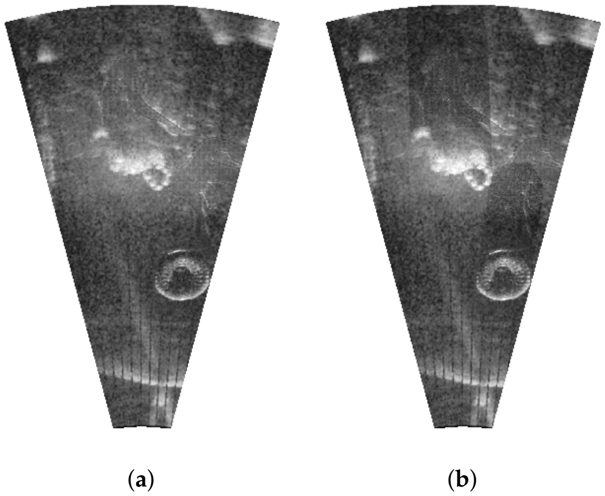


Figure 7. Comparison of (**a**) a synthetic image without a shadow and (**b**) a synthetic image with a generated shadow.

4. Experiments

4.1. Dataset

We used forward-looking sonar marine debris data [52] as the dataset. The data were collected by an ARIS Explorer 3000 [53] forward-looking sonar at frequencies of 3.0 MHz in an artificial water tank of 3 m width, 2 m height, and 4 m depth. This sonar has a field of view of 30° × 15° with 128 beams. The sonar was set to have a pitch angle between 15° and 30°. The images are 480 × 320 grayscale images, and the mask classifies each pixel by class. There are a total of twelve classes: background, bottle, can, chain, drink-carton, hook, propeller, shampoo-bottle, standing-bottle, wire, valve, and wall. The dataset contains a total of 1868 images and masks. We found 113 images and masks with incorrect or ambiguous masks; after excluding them, we used 1755 images and masks in our experiments. The data were randomly divided into 1053 images in the training set, 351 images in the validation set, and 351 images in the validation set with five folds.

We used real underwater sonar image (USI) datasets [34] for additional experiments. These data were collected be a Teledyne BlueView M900–90 forward-looking sonar at a frequency of 900 kHz. This sonar has a field of view of 90° × 20° and a range of 100 m. The collected underwater environments were of two types, reservoir and water pool; the water pool had a width of 10 m, a length of 12 m, and a depth of 1.5 m. Each environment was collected with two different conditions of sensitivity. This dataset contains only two classes, namely, backgrounds and objects. The dataset contains 800 training images and masks and 40 testing images and masks. We randomly divided these into a training set of 600 and validation set of 200 with four folds.

4.2. Experiment Setup

The the semantic segmentation performance of the proposed method was evaluated using the U-Net [10] model. U-Net is a U-shaped encoder–decoder structure. Various encoders are used in the U-Net architecture. We evaluated the proposed method based on U-Net with the ResNet-18 encoder [3]. In our experiments, the Adam optimizer [54] was used. The learning rate = 0.001, β_1 = 0.9, and β_2 = 0.999. The learning rate decreased exponentially by 0.99 per epoch. The weight of the network was initialized to He weight initialization [55]. The number of epochs in all experiments was 50. Augmentation was 1:1, and 1053 images were added to the training data. We considered the network with the highest score on the validation set to be the optimal model, and used this model for the test set.

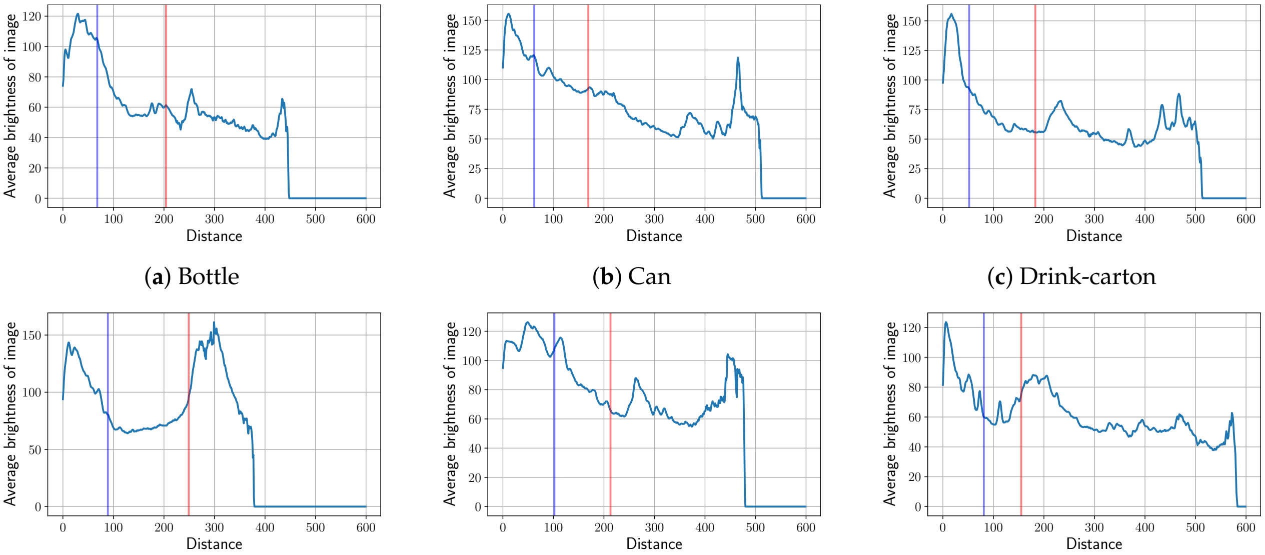
The baseline was the result when using only an existing unaugmented dataset. As comparison methods, we used traditional augmentation, CutOut [23], CutMix [24], ObjectAug [39], and Sim2Real [50], which are widely used methods. In traditional augmentation, rotation was within ±15°, scaling was 0.8 to 1.2, shifting was 3%, and flipping was horizontal. CutOut dropped out 25% of the image with 150 16 × 16 patches, while in CutMix 25% of the images were mixed with other images with one patch. ObjectAug was augmented in the same way as in traditional augmentation except that it was augmented in units of objects. In the Sim2Real method, we created a depth camera with the same field of view as sonar in the simulator, then CAD models were created to obtain depth images. For Sim2Real, we used contrastive learning for unpaired image-to-image translation (CUT) [56]. Pix2pix [33] could not be used because the two images of real sonar data and simulation data were not pairs. To preserve the structure of the simulation data while extracting the context of the sonar data, sim2real synthetic data were generated using patch-based contrastive learning.

4.3. Statistical Analysis

We performed statistical analysis based on our generated synthetic dataset consisting of statistical information on the location of the object and the brightness distribution of the shadow. Statistical information obtained from overlapping objects on this dataset is shown in Figures 8 and 9. This information was used to obtain the shadow area; when the gradient obtains the maximum point, the local peak is obtained, as the area outside a certain range is an area unrelated to the shadow. The shadow information extracted from this dataset is shown in Table 1. Different classes have different lengths and different intensities of darkness.

Table 1. Shadow information; the shadow length is in pixels, the shadow intensity is the ratio at which it becomes darker compared to the surrounding background, and the standard deviation is from 0 to 255.

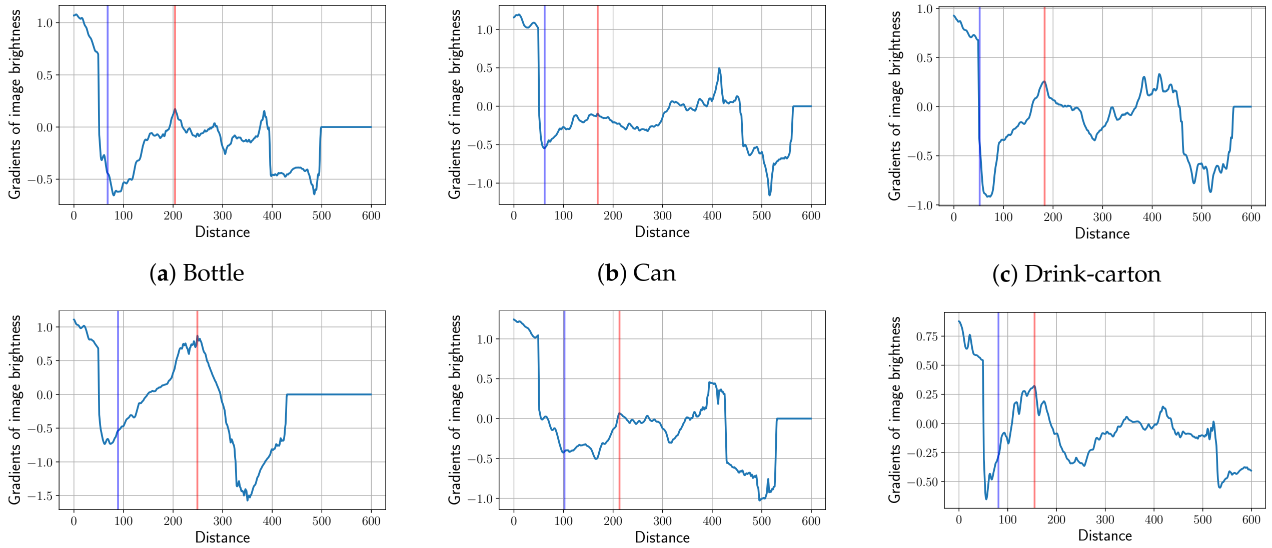
Object	Shadow Length	Shadow Intensity	Standard Deviation
bottle	76	0.88	21.19
can	112	0.94	25.28
drink-carton	122	0.81	21.04
hook	143	0.73	20.27
propeller	132	0.84	25.30
tire	90	0.75	27.51



(**a**) Bottle (**b**) Can (**c**) Drink-carton

(**d**) Hook (**e**) Propeller (**f**) Tire

Figure 8. Average brightness of images obtained by overlapping objects. The blue line is the average point at which the object ends, while the red line is the point estimated by the shadow.



(**a**) Bottle (**b**) Can (**c**) Drink-carton

(**d**) Hook (**e**) Propeller (**f**) Tire

Figure 9. Gradient of average brightness of an image obtained by overlapping objects. The average point at which the object ends is indicated by the blue line, while the shadow's estimated point is indicated by the red line.

Additionally, we analyzed the dataset and found that it was imbalanced. In particular, examples of certain classes, such as standing-bottle and shampoo-bottle, were very insufficient. Furthermore, smaller object classes such as standing-bottle, shampoo-bottle, hook, drink-carton, and valve had relatively insufficient numbers of pixels. On the other hand, the wall class had a very large number of objects and pixels. The number of objects for each class was randomly divided, and the number of pixels for each class is shown in Table 2.

Table 2. The number of objects and pixels in each class.

	The Number of Objects			**The Number of Pixels**		
Object	**Train**	**Test**	**Val**	**Train**	**Test**	**Val**
bottle	273	91	100	396,846	137,895	149,593
can	177	39	55	214,502	49,086	79,850
chain	185	71	62	728,342	277,067	326,391
drink-carton	185	74	75	129,130	55,511	57,594
hook	112	30	25	136,062	36,249	32,692
propeller	126	39	29	339,402	110,211	89,275
shampoo-bottle	42	27	27	81,300	60,334	53,563
standing-bottle	40	8	14	79,161	15,851	26,567
tire	371	127	113	1,020,156	331,350	300,764
valve	149	43	42	88,608	29,417	24,629
wall	554	179	186	2,207,620	673,196	626,694

4.4. Results of Comparison Experiment with Existing Methods

We used the Intersection over Union (IoU) for performance evaluation. IoU is expressed as follows:

$$IoU = \frac{area\ of\ overlap}{area\ of\ union} = \frac{|Prediction \cap GT|}{|Prediction \cup GT|} \times 100\ \%, \tag{10}$$

where GT is the ground truth. Because the experiment involved multiclass semantic segmentation, the mean IoU (mIoU), which is the average IoU of all classes, was used as the metric. The mIoU is expressed as a percentage.

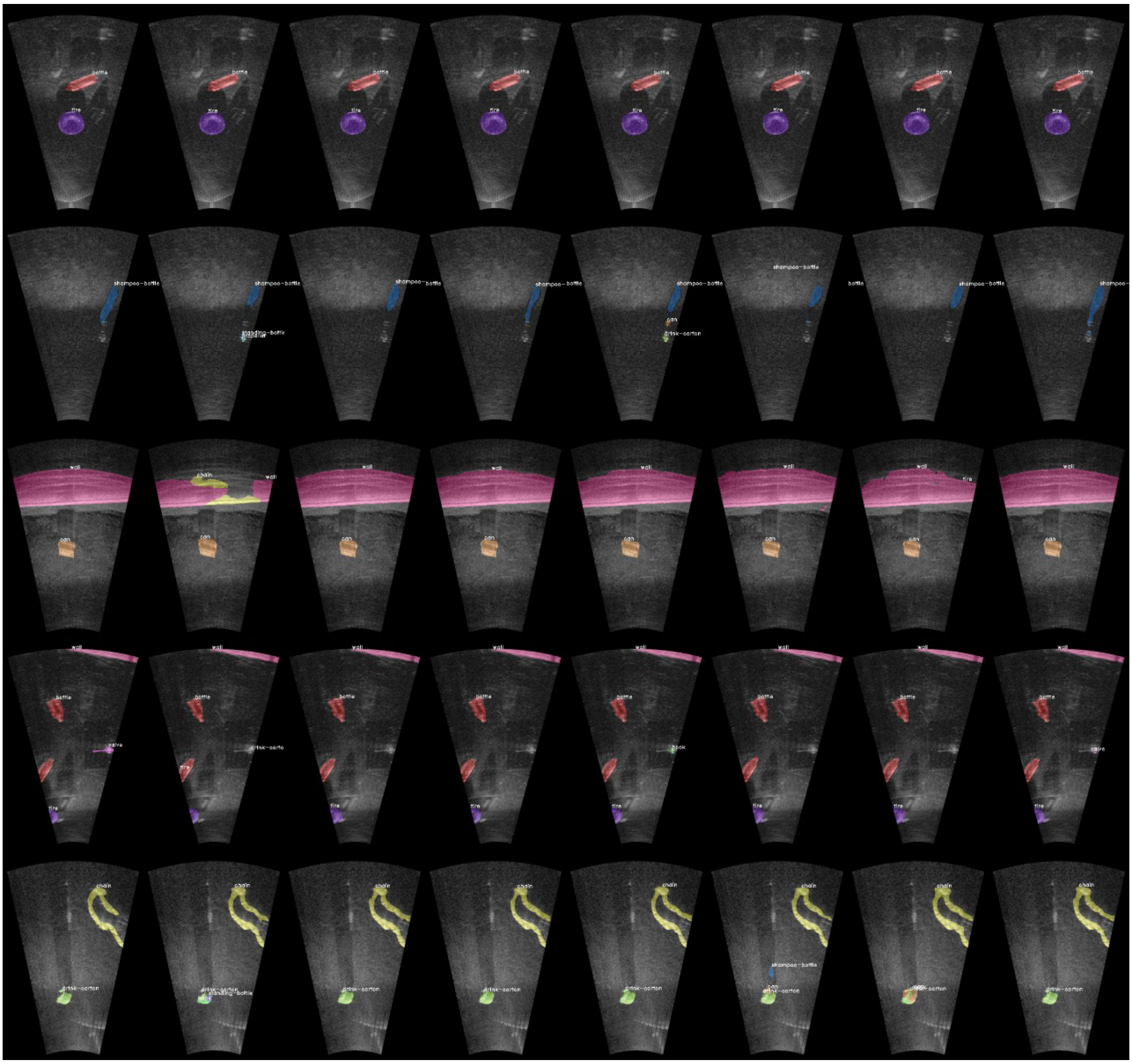
As shown in Table 3, the mIoU of the baseline was 75.14% and the mIoU of our approach was 76.24%, showing a performance improvement of 1.10%. The baseline had a higher mIoU than our approach in the case of the tire class; however, in all other classes our approach had a higher mIoU. Because the proportion of tire class objects in the dataset is high, as shown in Table 2, examples are more concentrated in the baseline that is not augmented, leading to bias. Augmentation increases the overall amount of data for each class, reducing the concentration on a specific class.

Table 3. mIoU when comparing each class with existing methods; the best IoU for each class is shown in bold.

Object	Baseline	TA	CutOut	CutMix	ObjectAug	Sim2Real	Ours
background	99.28	99.29	99.26	99.26	99.28	99.28	**99.29**
bottle	76.03	76.01	75.15	75.72	75.52	**79.30**	76.64
can	56.44	58.12	55.34	56.43	55.21	57.02	**58.99**
chain	63.48	63.44	62.00	61.83	62.35	62.96	**64.25**
drink-carton	73.75	74.65	72.44	74.31	73.31	74.30	**75.95**
hook	67.73	68.87	68.41	67.62	68.18	68.47	**69.41**
propeller	73.19	74.37	72.88	73.67	74.85	73.03	**74.89**
shampoo-bottle	78.07	**79.91**	78.18	78.51	79.47	78.88	78.61
standing-bottle	79.83	80.00	79.66	78.90	**82.67**	79.54	81.23
tire	**88.00**	87.64	87.63	87.49	87.65	87.61	87.92
valve	58.11	58.33	58.27	58.95	58.36	58.47	**59.56**
wall	87.74	**88.75**	88.24	88.07	88.38	88.31	88.17
mIoU	75.14	75.78	74.79	75.06	75.44	75.35	**76.24**

In our experimental results, our proposed method outperformed the other methods. Traditional augmentation achieved an mIoU of 75.78%, while our approach outperformed it by 0.46%. While raditional augmentation can improve data diversity, there is a limit to utilizing a variety of data to improve the actual diversity of data. Our method has a great advantage in diversity over traditional augmentation by cutting and pasting using various data from the DataBank. It can be difficult to overcome class imbalances with traditional augmentation, as it focuses on transforming the data. CutOut achieved an mIoU of 74.79%, while our approach exhibited superior performance with a 1.45% increase. CutMix achieved an mIoU of 75.06%, whereas our method was 1.18% higher. CutOut and CutMix improve the generalization of the model by cutting out parts of the images or replacing them with other images. However, this does not significantly improve the diversity of the data, and the model may have limitations in learning objects. ObjectAug achieved an mIoU of 75.44%, with our method surpassing it by 0.80%. ObjectAug focuses on performing data augmentation on object units, which increases object-level diversity; however, it has limitations in terms of the actual diversity of the data compared to our method using DataBank. Although ObjectAug has the advantage of augmenting on a per-object basis, our approach achieves improved performance by utilizing DataBank to make even more use of the data. Similarly, Sim2Real achieved an mIoU of 75.35%, with our approach outperforming it by 0.89%. Our method significantly improves data diversity using a variety of data from DataBank, while Sim2Real focuses on reducing domain differences between virtual and real environments, which can be limited in terms of data diversity due to relying on the diversity within the simulator. Moreover, Sim2Real requires a complex process to adapt the domain from the simulator to reality. In summary, our method performs better than other methods, significantly improving data diversity, reducing class imbalances, and showing improved adaptability to real-world environments. These results indicate its ability to achieve more accurate results in the semantic segmentation of sonar images.

When compared by class, the mIoU of small objects such as the can and drink-carton classes was the highest in our approach, and the mIoU of uncertain objects such as the chain, hook, propeller, and valve classes was the highest in our approach as well. These results indicate that our method allows segmentation models to better recognize and process ambiguous regions and fine object features. Figure 10 shows the results, indicating better segmentation of objects when using the proposed method.



(a) Ground truth (b) Baseline (c) TA (d) CutOut (e) CutMix (f) ObjectAug (g) Sim2Real (h) Ours

Figure 10. Segmentation results for different methods. At the top of the figure, the large clear bottle and tire objects are well segmented without augmentation. However, the smaller and more ambiguous objects are more difficult to segment. Notably, our proposed approach is effective at segmenting these objects.

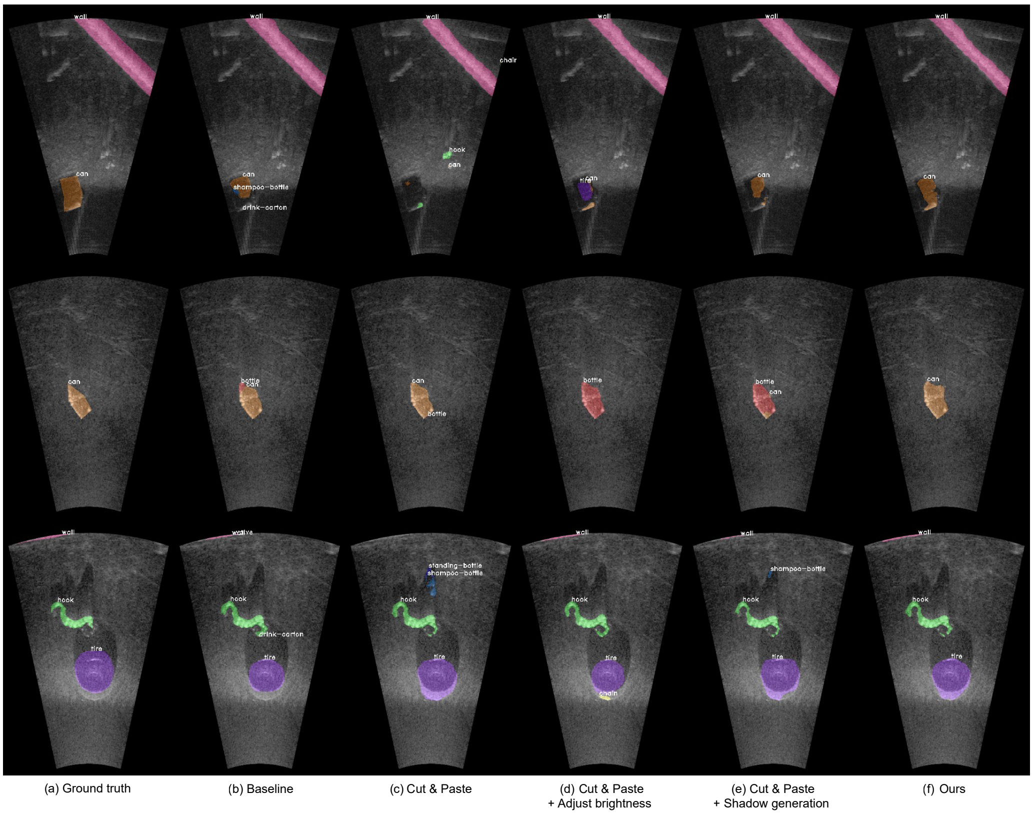
4.5. Ablation Study

We additionally conducted an ablation study to assess the impact of our various data augmentation methods on the performance of the semantic segmentation model. As shown in Table 4, with the cut-and-paste method, mIoU performance was improved by 0.65%. This method simply cuts and pastes images, allowing the model to learn various object placement and background combinations. This diversity can improve the generalization capabilities of the model and improve performance. This suggests that a simple data augmentation method can improve the performance of the semantic segmentation model.

When the cut-and-paste method and adjust brightness method were applied at the same time, mIoU performance was improved by 0.71%. This shows that adjusting objects to the brightness of the background can contribute to performance improvement when pasting them in various combinations. When the cut-and-paste method and shadow generation method were applied at the same time, mIoU performance was improved by 0.76%. This shows that augmenting data while maintaining the sonar's characteristics through the shadow generation method contributes to the performance improvement of the model. When all of the above methods were applied, the model showed the greatest improvement of 1.10%. In summary, our ablation study highlights that cut-and-paste data augmentation strategies that make the most of our data by creating a DataBank with fewer data and methods that maintain sonar characteristics both play a major role in improving model performance. The segmentation results in the ablation study are shown in Figure 11.

Table 4. mIoU results in the ablation study; the best mIoU for the model is shown in bold.

Model	Cut & Paste	Adjust Brightness	Shadow Generation	mIoU
Baseline				75.14
Cut & Paste	✓			75.79
Cut & Paste + Adjust brightness	✓	✓		75.85
Cut & Paste + Shadow generation	✓		✓	75.90
Ours	✓	✓	✓	**76.24**



(a) Ground truth (b) Baseline (c) Cut & Paste (d) Cut & Paste + Adjust brightness (e) Cut & Paste + Shadow generation (f) Ours

Figure 11. Segmentation results in the ablation study.

4.6. Effects of Augmented Ratio

In this section, we examine the impact of varying the augmentation ratio on model performance. Experiments were conducted with different augmentation levels: 0%, 50%, 100%, 200%, and 300% relative to the size of the training dataset. As shown in Figure 12, we observed a relationship between the augmentation ratio and mIoU improvement. Without augmentation (+0%), the mIoU remained at 75.14%. With 50% augmentation (+50%), we observed an increase to 75.79%. The most significant improvement was achieved with 100% augmentation (+100%), resulting in an mIoU of 76.24%. However, it is noteworthy that increasing the augmentation ratio beyond 100% did not consistently lead to further improvements. At a 200% augmentation ratio (+200%), the mIoU decreased slightly to 75.75%. Even with a 300% augmentation ratio (+300%), the mIoU only reached 75.83%, which was marginally lower than the +100% augmentation. These results suggest an optimal balance in the augmentation ratio, with a 1:1 augmentation-to-training dataset size being the most effective. These results show that data augmentation in excess of this ratio can adversely affect performance, as the model is trained to fit only augmented data and is inconsistent on real data. In summary, our findings underscore the importance of carefully calibrating the augmentation ratio, emphasizing that a 1:1 ratio of augmentation to training dataset size is optimal for enhancing model performance in this specific context.

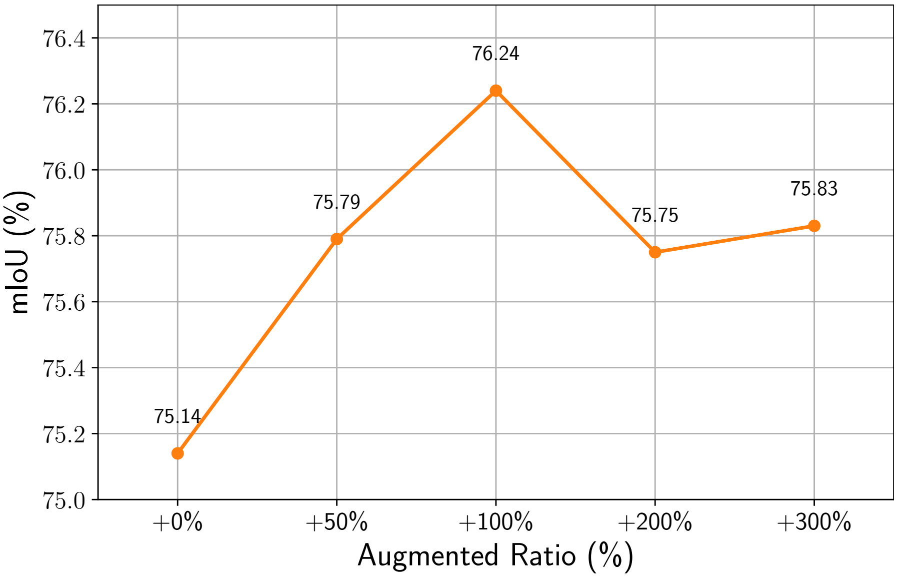


Figure 12. Effects of augmented ratio, showing the highest performance at +100%.

4.7. Application to Various Architectures

We assessed the effectiveness of our approach across different neural network architectures, specifically U-Net and DeepLabV3+ with the ResNet-18 and EfficientNet-B0 encoders. U-Net and DeepLabV3+ are widely used architectures in semantic segmentation tasks. U-Net already has an established reputation in semantic segmentation for its concise structure and outstanding performance, while DeepLabV3+ is a powerful architecture that provides a wide range of enhancements to segmentation tasks. ResNet-18 is a network that has already been validated for its reliability and performance, while EfficientNet-B0 combines good computational efficiency and performance. These encoders serve as the foundational structure of the model, which is essential for semantic segmentation tasks. As summarized in Table 5, we compared the mIoU performance of each architecture with and without our approach. In the case of U-Net with the ResNet-18 model, we observed a remarkable increase of 1.10% in mIoU. Similarly, the mIoU of the U-Net with EfficientNet-B0 model showed a significant improvement of 0.69%. The DeepLabV3+ with ResNet-18 model exhibited an enhancement of 0.47% in mIoU. Finally, the mIoU of the DeepLabV3+ with EfficientNet-B0 model increases by 0.92%. These findings underscore the robustness and adaptability of our approach, as it enhances the mIoU scores across various architectures. Our method improves semantic segmentation performance regardless of the specific neural network configuration used.

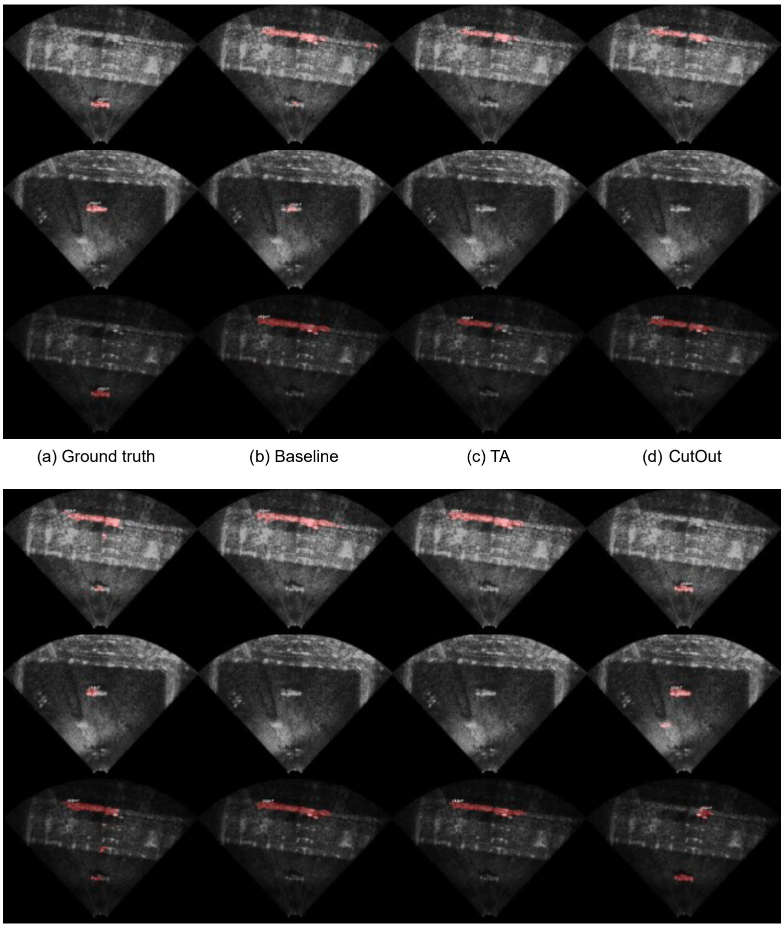
Table 5. Performance comparison of architectures; the best mIoU for each model is shown in bold.

Architecture	Encoder	Baseline	Ours
U-Net	ResNet-18	75.14	**76.24**
	EfficientNet-B0	75.71	**76.40**
DeepLabV3+	ResNet-18	75.98	**76.45**
	EfficientNet-B0	75.27	**76.19**

4.8. Results of Experiment on USI Datasets

We additionally experimented with USI datasets to verify the generalization ability and robustness of the proposed method. The Sim2Real experiment replaced synthetic USI datasets, as synthetic USI datasets generated with pix2pix were provided for augmentation in the USI datasets. The settings of the experiments were the same as before.

As shown in Table 6, the mIoU of the baseline was 79.59% and the mIoU of our approach was 84.19%, showing a performance improvement of 4.60%. The mIoU results for the other methods were as follows: traditional augmentation, 81.02%; CutOut, 81.88%; CutMix, 82.68%; ObjectAug, 79.98%; and the synthetic dataset, 78.24%. Thus, our approach achieved the highest mIoU. This experiment verifies that our method works well on datasets collected from different environments using different forward-looking sonars. Figure 13 shows the segmentation results on real USI datasets. Even in images with difficult segmentation, our method is better at finding objects.



(a) Ground truth (b) Baseline (c) TA (d) CutOut

(e) CutMix (f) ObjectAug (g) Synthetic (h) Ours

Figure 13. Segmentation results on real USI datasets; our method is better at finding objects in images that are difficult to segment.

Table 6. mIoU when comparing each class on real USI datasets; the best IoU for each class is shown in bold.

Object	Baseline	TA	CutOut	CutMix	ObjectAug	Synthetic	Ours
background	99.69	99.72	99.74	99.75	99.70	99.67	**99.78**
object	59.48	62.32	64.03	65.60	60.25	56.81	**68.60**
mIoU	79.59	81.02	81.88	82.68	79.98	78.24	**84.19**

5. Conclusions

In this paper, we have introduced a novel data augmentation pipeline designed specifically for sonar imagery. Previous studies have attempted to enhance semantic segmentation performance by applying transformations such as rotation and scaling to entire images or object units. However, these methods fall short of effectively leveraging the unique characteristics of shadowy and noisy sonar images. To address this limitation, we propose SACuP, a pipeline that utilizes a DataBank-based approach for augmenting sonar images while preserving their distinctive shadows and noise. Our approach focuses on augmenting images at the object unit level, thereby tailoring the augmentation process for segmentation tasks. No additional manual effort is needed, as the proposed approach leverages existing images and masks. This augmentation method significantly enhances semantic segmentation performance while faithfully retaining the sonar image's inherent features.

Extensive comparisons with existing augmentation methods show superior performance, with the mIoU of baseline being 75.14% and the mIoU of our approach 76.24%. Our ablation study highlighted the significant impact of various data augmentation methods on our semantic segmentation model. Notably, the cut-and-paste method improved mIoU by 0.65%, combining with brightness adjustment for a 0.71% improvement, and combining with shadow generation for a 0.76% increase. The most substantial boost, a total improvement of 1.10%, was achieved when all methods were applied together. A 1.10% increase can result in improved performance on small datasets, enabling more accurate detection of objects underwater, significantly reducing costs, and further improving performance in combination with other methods.

We anticipate that our approach will prove invaluable in augmenting insufficient sonar data across a range of tasks, particularly in the context of semantic segmentation. Its potential applications extend to enhancing the performance of underwater exploration, where high-quality sonar data are essential.

Author Contributions: Conceptualization, S.P. and H.H.; methodology, S.P. and H.H.; software, S.P. and Y.C.; validation, S.P. All authors have read and agreed to the published version of the manuscript.

Funding: This work was partly supported by a National Research Foundation of Korea (NRF) grant funded by the Korean government (MSIT) (NRF-2022R1C1C1008074) and by an Institute of Information and Communications Technology Planning and Evaluation (IITP) grant funded by the Korean government (MSIT) (No. RS-2022-00155911, Artificial Intelligence Convergence Innovation Human Resources Development (Kyung Hee University).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The forward-looking sonar marine debris dataset presented by Singh, D., & Valdenegro-Toro, M. and the dataset is available at https://github.com/mvaldenegro/marine-debris-fls-datasets, which was accessed on 7 August 2023. The USI dataset presented by Lee et al. and the dataset is available at https://doi.org/10.1371/journal.pone.0272602 which was accessed on 30 October 2023.

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

References

1. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. *Adv. Neural Inf. Process. Syst.* **2012**, *25* , 1097–1105. [CrossRef]
2. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. In Proceedings of the International Conference on Learning Representations, San Diego, CA, USA, 7–9 May 2015.
3. 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.
4. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 5998–6008.
5. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv* **2018**, arXiv:1810.04805.
6. Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. Language models are few-shot learners. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 1877–1901.
7. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 580–587.
8. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
9. Long, J.; Shelhamer, E.; Darrell, T. Fully convolutional networks for semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 3431–3440.
10. 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, October 5–9, 2015, Proceedings, Part III 18*; Springer: Cham, Switzerland, 2015; pp. 234–241.
11. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2961–2969.
12. Cordts, M.; Omran, M.; Ramos, S.; Rehfeld, T.; Enzweiler, M.; Benenson, R.; Franke, U.; Roth, S.; Schiele, B. The cityscapes dataset for semantic urban scene understanding. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 3213–3223.
13. Minaee, S.; Boykov, Y.; Porikli, F.; Plaza, A.; Kehtarnavaz, N.; Terzopoulos, D. Image segmentation using deep learning: A survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, *44*, 3523–3542. [CrossRef] [PubMed]
14. Valdenegro-Toro, M. Object recognition in forward-looking sonar images with convolutional neural networks. In Proceedings of the OCEANS 2016 MTS/IEEE Monterey, Monterey, CA, USA, 19–23 September 2016; pp. 1–6.
15. Valdenegro-Toro, M. End-to-end object detection and recognition in forward-looking sonar images with convolutional neural networks. In Proceedings of the 2016 IEEE/OES Autonomous Underwater Vehicles (AUV), Tokyo, Japan, 6–9 November 2016; pp. 144–150.
16. Hansen, C.H. Fundamentals of acoustics. In *Occupational Exposure to Noise: Evaluation, Prevention and Control*; World Health Organization: Geneva, Switzerland, 2001; Volume 1, pp. 23–52.
17. Steiniger, Y.; Kraus, D.; Meisen, T. Survey on deep learning based computer vision for sonar imagery. *Eng. Appl. Artif. Intell.* **2022**, *114*, 105157. [CrossRef]
18. Sun, C.; Shrivastava, A.; Singh, S.; Gupta, A. Revisiting unreasonable effectiveness of data in deep learning era. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 843–852.
19. Wang, M.; Deng, W. Deep visual domain adaptation: A survey. *Neurocomputing* **2018**, *312*, 135–153. [CrossRef]
20. Wang, Y.; Yao, Q.; Kwok, J.T.; Ni, L.M. Generalizing from a few examples: A survey on few-shot learning. *ACM Comput. Surv.* **2020**, *53*, 1–34. [CrossRef]
21. Figueira, A.; Vaz, B. Survey on synthetic data generation, evaluation methods and GANs. *Mathematics* **2022**, *10*, 2733. [CrossRef]
22. Yang, S.; Xiao, W.; Zhang, M.; Guo, S.; Zhao, J.; Shen, F. Image data augmentation for deep learning: A survey. *arXiv* **2022**, arXiv:2204.08610.
23. DeVries, T.; Taylor, G.W. Improved regularization of convolutional neural networks with cutout. *arXiv* **2017**, arXiv:1708.04552.
24. Yun, S.; Han, D.; Oh, S.J.; Chun, S.; Choe, J.; Yoo, Y. Cutmix: Regularization strategy to train strong classifiers with localizable features. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 6023–6032.
25. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative adversarial nets. *arXiv* **2014**, arXiv:1406.2661.
26. Song, T.; Wang, Y.; Gao, C.; Chen, H.; Li, J. MSLAN: A Two-Branch Multidirectional Spectral–Spatial LSTM Attention Network for Hyperspectral Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 5528814. [CrossRef]
27. Sheng, Y.; Xiao, L. Manifold Augmentation Based Self-Supervised Contrastive Learning for Few-Shot Remote Sensing Scene Classification. In Proceedings of the IGARSS 2022-2022 IEEE International Geoscience and Remote Sensing Symposium, Kuala Lumpur, Malaysia, 17–22 July 2022; pp. 2239–2242.

28. Zhang, B.; Zhou, T.; Shi, Z.; Xu, C.; Yang, K.; Yu, X. An underwater small target boundary segmentation method in forward-looking sonar images. *Appl. Acoust.* **2023**, *207*, 109341. [CrossRef]
29. Gibou, F.; Fedkiw, R.; Osher, S. A review of level-set methods and some recent applications. *J. Comput. Phys.* **2018**, *353*, 82–109. [CrossRef]
30. Zhao, D.; Ge, W.; Chen, P.; Hu, Y.; Dang, Y.; Liang, R.; Guo, X. Feature Pyramid U-Net with Attention for Semantic Segmentation of Forward-Looking Sonar Images. *Sensors* **2022**, *22*, 8468. [CrossRef]
31. Kirillov, A.; Mintun, E.; Ravi, N.; Mao, H.; Rolland, C.; Gustafson, L.; Xiao, T.; Whitehead, S.; Berg, A.C.; Lo, W.Y.; et al. Segment Anything. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Paris, France, 2–3 October 2023; pp. 4015–4026.
32. Wang, L.; Ye, X.; Zhu, L.; Wu, W.; Zhang, J.; Xing, H.; Hu, C. When SAM Meets Sonar Images. *arXiv* **2023**, arXiv:2306.14109.
33. Isola, P.; Zhu, J.Y.; Zhou, T.; Efros, A.A. Image-to-image translation with conditional adversarial networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 1125–1134.
34. Lee, E.h.; Park, B.; Jeon, M.H.; Jang, H.; Kim, A.; Lee, S. Data augmentation using image translation for underwater sonar image segmentation. *PLoS ONE* **2022**, *17*, e0272602. [CrossRef]
35. Chen, L.C.; Papandreou, G.; Schroff, F.; Adam, H. Rethinking atrous convolution for semantic image segmentation. *arXiv* **2017**, arXiv:1706.05587.
36. Chen, L.C.; Zhu, Y.; Papandreou, G.; Schroff, F.; Adam, H. Encoder-decoder with atrous separable convolution for semantic image segmentation. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 801–818.
37. Tan, M.; Le, Q. Efficientnet: Rethinking model scaling for convolutional neural networks. In Proceedings of the International Conference on Machine Learning, PMLR, Long Beach, CA, USA, 9–15 June 2019; pp. 6105–6114.
38. Shorten, C.; Khoshgoftaar, T.M. A survey on image data augmentation for deep learning. *J. Big Data* **2019**, *6*, 60. [CrossRef]
39. Zhang, J.; Zhang, Y.; Xu, X. Objectaug: Object-level data augmentation for semantic image segmentation. In Proceedings of the 2021 International Joint Conference on Neural Networks (IJCNN), Shenzhen, China, 18–22 July 2021; pp. 1–8.
40. Ghiasi, G.; Cui, Y.; Srinivas, A.; Qian, R.; Lin, T.Y.; Cubuk, E.D.; Le, Q.V.; Zoph, B. Simple copy-paste is a strong data augmentation method for instance segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Virtual, 19–25 June 2021; pp. 2918–2928.
41. Illarionova, S.; Nesteruk, S.; Shadrin, D.; Ignatiev, V.; Pukalchik, M.; Oseledets, I. Object-based augmentation improves quality of remote sensing semantic segmentation. *arXiv* **2021**, arXiv:2105.05516.
42. Kingma, D.P.; Welling, M. Auto-encoding variational bayes. *arXiv* **2013**, arXiv:1312.6114.
43. Gatys, L.A.; Ecker, A.S.; Bethge, M. Image style transfer using convolutional neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 2414–2423.
44. Manhães, M.M.M.; Scherer, S.A.; Voss, M.; Douat, L.R.; Rauschenbach, T. UUV simulator: A gazebo-based package for underwater intervention and multi-robot simulation. In Proceedings of the OCEANS 2016 MTS/IEEE Monterey, Monterey, CA, USA, 19–23 September 2016; pp. 1–8.
45. DeMarco, K.J.; West, M.E.; Howard, A.M. A computationally-efficient 2D imaging sonar model for underwater robotics simulations in Gazebo. In Proceedings of the OCEANS 2015-MTS/IEEE Washington, Washington, DC, USA, 19–22 October 2015; pp. 1–7.
46. Cerqueira, R.; Trocoli, T.; Neves, G.; Joyeux, S.; Albiez, J.; Oliveira, L. A novel GPU-based sonar simulator for real-time applications. *Comput. Graph.* **2017**, *68*, 66–76. [CrossRef]
47. Cerqueira, R.; Trocoli, T.; Albiez, J.; Oliveira, L. A rasterized ray-tracer pipeline for real-time, multi-device sonar simulation. *Graph. Model.* **2020**, *111*, 101086. [CrossRef]
48. Choi, W.S.; Olson, D.R.; Davis, D.; Zhang, M.; Racson, A.; Bingham, B.; McCarrin, M.; Vogt, C.; Herman, J. Physics-based modelling and simulation of multibeam echosounder perception for autonomous underwater manipulation. *Front. Robot. AI* **2021**, *8*, 706646. [CrossRef]
49. Koenig, N.; Howard, A. Design and use paradigms for gazebo, an open-source multi-robot simulator. In Proceedings of the 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No. 04CH37566), Sendai, Japan, 28 September–2 October 2004; Volume 3, pp. 2149–2154.
50. Sung, M.; Kim, J.; Kim, J.; Yu, S.C. Realistic sonar image simulation using generative adversarial network. *IFAC-PapersOnLine* **2019**, *52*, 291–296. [CrossRef]
51. 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.
52. Singh, D.; Valdenegro-Toro, M. The marine debris dataset for forward-looking sonar semantic segmentation. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, QC, Canada, 11–17 October 2021; pp. 3741–3749.
53. SoundMetrics. ARIS Explorer 3000: See What Others Can't. Available online: http://www.soundmetrics.com/products/aris-sonars/ARIS-Explorer-3000/015335_RevD_ARIS-Explorer-3000_Brochure (accessed on 7 August 2023).
54. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.

55. He, K.; Zhang, X.; Ren, S.; Sun, J. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1026–1034.
56. Park, T.; Efros, A.A.; Zhang, R.; Zhu, J.Y. Contrastive learning for unpaired image-to-image translation. In *Proceedings of the Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IX 16*; Springer: Cham, Switzerland, 2020; pp. 319–345.

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.