A First Comparison of GANs and Diffusion Models for Generating Sidescan Sonar Images

Yannik Steiniger[1] and Benjamin Lehmann[2]

[1]German Aerospace Center, Institute for the Protection of Maritime Infrastructures, 27572 Bremerhaven, Germany
[2]City University of Applied Sciences Bremen, 28199 Bremen, Germany

Contact author: Yannik Steiniger, Fischkai 1, 27572 Bremerhaven, Germany,
yannik.steiniger@dlr.de

Abstract: *We investigate generative deep learning models, namely a R3GAN and a diffusion model (DM), for the generation of synthetic sidescan sonar images. Both models are able to generate snippets with realistic object and shadow geometries as well as pixel distributions. Since sonar data from specific objects is scarce, the synthetic images are used to augment a training dataset for object classification. This augmentation can increase the performance of a shallow convolutional neural network (CNN) as well as a state-of-the-art vision transformer. In our analysis, images from R3GAN show a higher variability but images from DM lead to a better performance of the classification models improving the balanced accuracy of the CNN by 12%.*

Keywords: *Deep Learning, Generative Models, Diffusion Models, Synthetic Sonar Data, Sonar Imagery*

1. INTRODUCTION

With the analysis of sonar images being increasingly shifted towards the use of deep learning models the need for sufficient training datasets grows. Compared to other computer vision applications, like autonomous driving, there exists no large, publicly available dataset of sonar images for a broad range of object classes. Furthermore, gathering and annotating real world data with a high degree of variability is cumbersome and expensive. It requires specialized equipment as well as trained staff. Thus, research on generating synthetic sonar images gained more attention in recent years. Applications range from highly realistic but computational slow models based on the wave equation and ray tracing [1] over Fourier-domain wavefield rendering approaches [2] to faster but less transparent deep learning based generative models, such as generative adversarial networks (GAN) [3, 4]. In the past few years diffusion models (DM) [5] have been established as another standard in the computer vision domain for generating synthetic images, especially to generate natural RGB images. This alternative approach to GANs does not suffer from training instabilities, like mode collapse, and can produce more detailed images, but typically requires more training data. DMs have recently been applied to enhance the resolution of sonar images which led to an improved classification performance [6].

In this work we, for the first time, train a GAN as well as a DM to generate realistic sidescan sonar images of multiple object classes. We consider the classes *Tire*, *Cylinder* and *Wreck*. For these object classes less than 30 training samples are available. The aim of these synthetic images is to augment a dataset of real sonar images for the training of a classifier. We use a basic convolutional neural network (CNN) as well as a modern vision transformer (ViT) as classifiers. In our study we compare the images generated by the GAN and DM based on qualitative measures, e.g., realistic object and shadow shapes, and on their ability to increase the performance of the classifiers when being used to augment the training dataset. Furthermore, we investigate the variability of the methods by successively increasing the portion of synthetic images in the training dataset and using a clustering based approach.

2. METHODS AND MATERIALS

2.1. SIDESCAN SONAR DATASET

The sidescan sonar images used to train and evaluate the generative models as well as the classification models were collected during several measurement campaigns between 2019 and 2020. An *Edgetech 2205* system operating at 850 kHz mounted on an *ATLAS ELEKTRONIK SeaCat* AUV was used to gather this data. The sonar data was processed with an experimental processing chain to form grayscale intensity images. Snippets of relevant objects were manually extracted to form the dataset. Similar to previous work [7] this dataset consists of the classes *Tire*, *Rock*, *Cylinder*, *Wreck* and *Background*. In Table 1 the number of samples per class in the training and test dataset are listed. As it can be seen from the number of snippets, the classes *Tire*, *Cylinder* and *Wreck* are underrepresented. These will be the objects for which the generative models should generate additional synthetic data. Since multiple snippets of the same object exist, the training and test split was setup such that snippets from the same object are only in one dataset. Due to the limited amount of data, snippets from the three underrepresented classes were split nearly 50/50 while adhering to the aforementioned constrain.

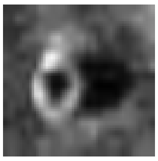
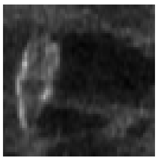
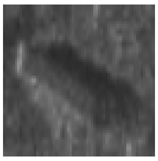
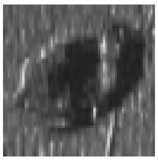
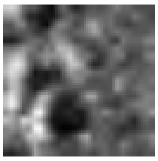
Object class	*Tire*	*Rock*	*Cylinder*	*Wreck*	*Background*
Number of training snippets	18	500	22	12	500
Number of test snippets	17	719	19	12	719
Example image					

Table 1: Overview of the sidescan sonar dataset.

2.2. GAN MODEL

A GAN is a deep learning model which can be used to generate synthetic images based on a provided training dataset. It consists of two sub-models, the generator G and the discriminator D, which are typically implemented as a CNN and trained in an adversarial manner. On the one hand, the discriminator learns to distinguish between images from the dataset and those from the generator. On the other hand, the generator is trained to transform a noise vector z into images similar to the training samples x in order to fool the discriminator. In general, this training procedure can be described by the loss function

$$\mathcal{L} = \mathrm{E}_{z\sim p_z} - \log\left(1 + \exp\left(-D(G(z))\right)\right) + \mathrm{E}_{x\sim p_{data}} - \log\left(1 + \exp\left(-D(x)\right)\right), \tag{1}$$

which the discriminator tires to maximize and the generator to minimize. Here p_z and p_{data} are the distributions of the noise and training data, respectively. A common problem when training a GAN is the so-called mode collapse. This is a state of the GAN in which the generator outputs the same image regardless of the input noise. Different loss functions, like the Wasserstein loss [8], or other training strategies like specific pre-training [9] have been developed to deal with the training difficulties of GANs.

In this work we use the state-of-the-art R3GAN [10] for generating synthetic sonar images. R3GAN uses a regularized relativistic GAN loss to over-come the training difficulties. This loss combines a relativistic pairing loss [11]

$$\mathcal{L} = \mathrm{E}_{z\sim p_z, x\sim p_{data}} - \log\left(1 + \exp\left(D(G(z)) - D(x)\right)\right) \tag{2}$$

with the zero-centred gradient penalties

$$R_1 = \frac{\gamma}{2}\mathrm{E}_{x\sim p_{data}} \left\|\Delta_x D\right\|^2 \tag{3}$$

and

$$R_2 = \frac{\gamma}{2}\mathrm{E}_{x\sim p_G} \left\|\Delta_x D\right\|^2 \tag{4}$$

where γ is a hyperparameter controlling the degree of regularization. The relativistic paring loss directly couples the real data $x \sim p_{data}$ and synthetic images $x \sim p_G$ in the training process while R_1 and R_2 regularize the gradient norm of D on the real and synthetic images, respectively. Both regularization term should ensure the convergence of the model. Furthermore, the networks of D and G are designed based on a modern ResNet architecture. Both networks consist of residual blocks and are setup symmetrically meaning that the generator upsamples the input through the resolutions 4×4, 8×8, 16×16 and 32×32 while the discriminator downsamples the image through the reverse order of resolutions. For a detailed overview about the architecture see [10]. We train R3GAN with its default parameters for 7813 iterations, which is equivalent to 250,000 images.

2.3. DENOISING DIFFUSION PROBABILISTIC MODELS

Denoising Diffusion Probabilistic Models (DM) are a class of generative models to synthesize data, given a two-staged process. In a forward diffusion process a *clean* input image x_0 taken from the training dataset is gradually disturbed by adding Gaussian noise. After completing consecutive noising steps, also referred to as timesteps, the image content approaches an isotropic Gaussian noise while the information is lost in noise completely. In the reverse process, a neural network (typically a U-Net architecture) is trained to remove the added noise to unveil the original data or at least data that look similar to the original one. With this approach the network was supposed to learn the underlying pixel distribution in the datasets. The term pixel distribution refers to the spatial dependencies among neighbouring pixels (up to the complete image domain), rather than the statistics of individual pixels in isolation. The loss function to be minimized is the mean squared error (MSE) between the true noise ϵ and the predicted noise ϵ_θ added at timestep t

$$\mathcal{L} = \sum_t \mathrm{E}_{\epsilon \sim \mathcal{N}(0,I)} ||\epsilon - \epsilon_\theta(x_t, t)||^2, \tag{5}$$

where I is the identity matrix, x_t is the disturbed image and t is uniformly sampled from 1 to T, the maximum number of steps. The generation of synthetic images starts with isotropic Gaussian noise and all pixels not obeying the learned pixel distribution are removed respectively denoised.

The standard DM loss functions treat all timesteps equally, which may lead to bias towards early learning steps and tend to overfit to high signal-to-noise (SNR) examples, which is of special interest for sonar images which might suffer from noise. Therefore the P2-weighting was introduced [12, 13], which formulates a loss re-weighting function $w(t)$ based on the signal-to-noise ratio. It was proven to neglect early (less noisy) steps and promotes learning in later (more noisy) steps. Following widely adopted implementations the explicit weighting scheme

$$\mathcal{L}_{P2} = \sum_t w(t) \mathrm{E}_{\epsilon \sim \mathcal{N}(0,I)} ||\epsilon - \epsilon_\theta(x_t, t)||^2, \tag{6}$$

with

$$w(t) = \left(\frac{1}{1 + \mathrm{SNR}(t)} \right)^\alpha \tag{7}$$

is applied, where α is a hyperparameter, commonly set as $\alpha \in [1.0, 2.0]$. In all our experiments the training was started from scratch, where for each object class $\in$ {*Tire*, *Cylinder*, *Wreck*} one individual network was trained and used afterwards to generate synthetic object images.

Exp.	Tire		Cylinder		Wreck		Rock		Background	
	real	synth.	real	synth.	real	synth.	real	synth.	real	synth.
1	18	0	22	0	12	0	36	0	36	0
2	18	18	22	14	12	24	36	0	36	0
3	18	32	22	28	12	38	50	0	50	0
4	18	82	22	78	12	88	100	0	100	0
5	18	482	22	478	12	488	500	0	500	0

Table 2: Number of real and synthetic snippets per class in each experiment.

3. EXPERIMENTAL SETUP

The main purpose of generating synthetic sonar snippets is to augment a training dataset for a classification model at hand. Thus, the quality of the generated images will be measured indirectly in terms of performance improvements of the classification model. Especially, we are interested in how many synthetic snippets can be generated and added to the training dataset to still see noticeable improvements. In Table 2 the five experiments conducted in this work are listed, with increasing number of training snippets, where *Experiment 1* serves as the baseline without any synthetic data added. Based on previous work [9] we reduce the number of rock and background snippets to be 36 in order to balance the dataset. In *Experiment 2* we add synthetic snippets to the classes *Tire*, *Cylinder* and *Wreck* so that every class has 36 training snippets in total. Since this number is still very low compared to conventional deep learning dataset, we successively increase the number of real rock and background snippets as well as synthetic tire, cylinder and wreck snippets through *Experiments 3 – 5*. The total number of snippets per class is 50, 100 and 500, respectively.

We train two different classification models on the datasets described in Table 2, namely a shallow CNN used in previous work [7] and a state-of-the-art vision transformer (ViT) [14]. Our shallow CNN consists of three convolutional layers with 8, 16 and 32 kernel of size 3×3. The output of a convolutional layer is passed through a ReLU activation function, batch normalization and 2×2 max pooling. Features from the last convolutional layer are compressed using a fully connected layer with 100 neurons prior to the final output layer with five neurons. Dropout is added before both fully connected layers. All input images are scaled to match the input size of the network, which is 64×64 pixel. We train this CNN without any pre-training for 50 epochs with the Adam optimizer and a learning rate of 10^{-4}, which is reduced by a factor of 0.1 after every 20 epochs.

As an alternative method to verify the gain in performance we employ the original vision transformer with little modifications to account for sonar images and training data used. The core innovation in using ViT architectures is the replacement of traditional CNN architectures with transformer encoder architectures, well known from natural language processing. This milestone introduced a new performance area for image classification tasks. The image under consideration is split into fixed-size patches, flattened and embedded linearly into a vector. A token is added to represent the image's overall classification and processed by a transformer encoder stack, consisting of multi-head self-attention and feedforward layers, to be forwarded to a multi-layer perceptron head for final classification. In our experiments we made use of the ViT base model (ViT-B) with patch sizes of 16×16 pixels, with pre-trained weights from the ImageNet-1K dataset. For the different sonar datasets with varying cardinality in our experiments, we achieved comparable performances by employing 5 to 25 epochs, using the Adam optimizer and a learning rate of 10^{-5}. In the following we report the performance for the model trained for 20 epochs. In addition to that we reduced number of neurons in the classification MLP to 100 and the number of classification classes to 5 to match the sonar dataset.

We evaluate the classification performance of both models in all five experiments using the balanced accuracy

$$ACC_{\text{bal}} = \frac{1}{|\mathcal{C}|} \sum_{c \in \mathcal{C}} \text{recall}(c) \tag{8}$$

and the macro F1-score

$$F1_{\text{macro}} = \frac{1}{|\mathcal{C}|} \sum_{c \in \mathcal{C}} F1(c) = \frac{1}{|\mathcal{C}|} \sum_{c \in \mathcal{C}} \frac{2 \cdot \text{precision}(c) \cdot \text{recall}(c)}{\text{precision}(c) + \text{recall}(c)}. \tag{9}$$

Both metrics take an average over the different classes $\mathcal{C} = \{$*Tire*, *Rock*, *Cylinder*, *Wreck*, *Background*$\}$ and are thus suitable for the unbalanced test dataset. Note that neither during training of the generative models nor during training of the classification models images from this test dataset have been used.

To further study the data generated by the GAN and the DM we investigate the diversity in the synthetic images. For this we use a clustering based approach [7]. The clustering approach starts by selecting one generated image as the center of a cluster. Images whose pixel-wise Euclidean distance to this initial image is below a given threshold will be grouped to this cluster. In the next iteration an image that has not been assigned to a cluster serves as the center point of the next cluster and the process is repeated with all unassigned images. Finally, the number of clusters determines the diversity of the synthetic images, with a large number of clusters relating to a high diversity.

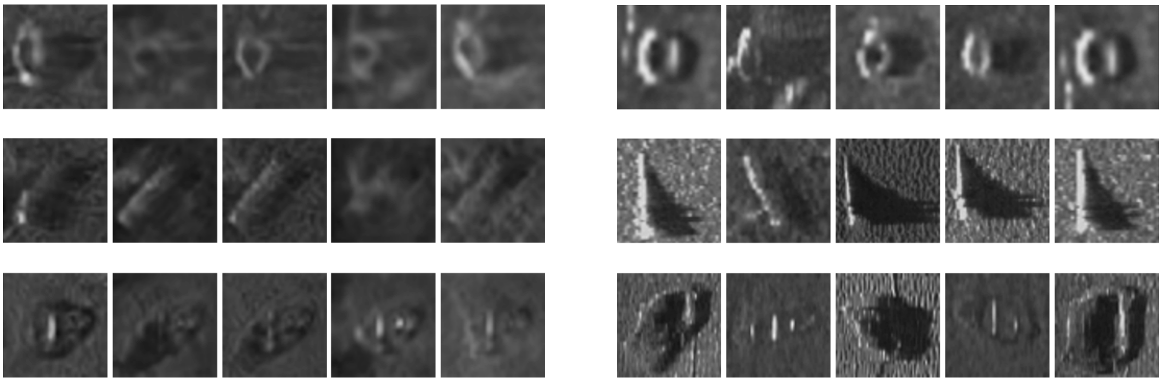
4. RESULTS

4.1. GENERATED IMAGES

Before investigating the influence on the classification performance we analyse the synthetic images generated by R3GAN and DM quantitatively. Figure 1 shows five random examples per class from both generative models. Both methods succeed in generating the required object shapes and realistic shadow geometries. The contrast in the images generated by DM is higher than the one in the images from R3GAN. Both models generate different snippets and do not suffer from any form of mode collapse. All cylinders generated by R3GAN in Figure 1 (a) have the same orientation, which is also over-represented in the remaining generated images. The training dataset, however, also contains examples with a different orientation (see Table 1). Thus, R3GAN seems not to be able to cover this variability. The images generated by DM show a strong similarity to the training images, e.g., the first generated wreck is similar to the example shown in Table 1.

4.2. CLASSIFICATION PERFORMANCE

Table 3 lists the classification performance measured in balanced accuracy and macro F1-score for the five experiments and images generated by R3GAN and DM. Note that since the



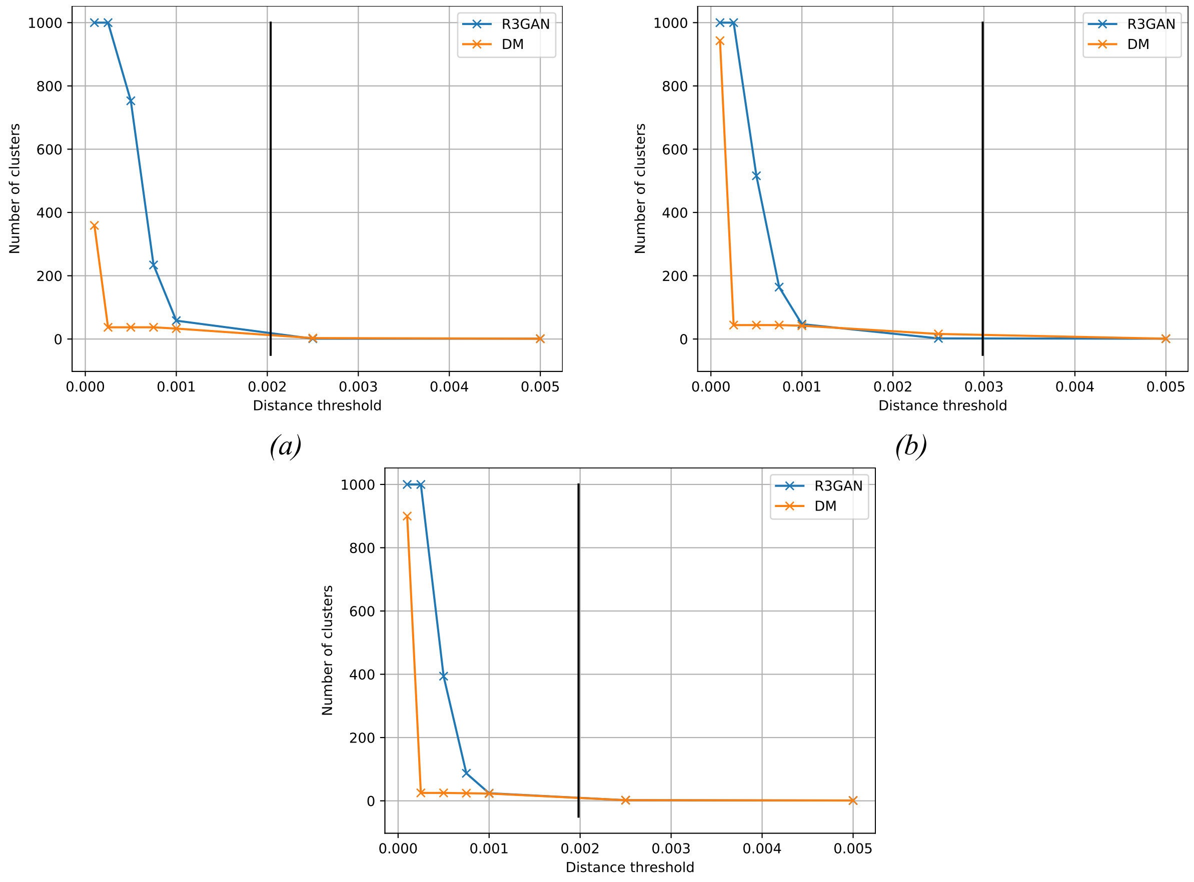
(a) *(b)*

Figure 1: Synthetic snippets generated with (a) R3GAN and (b) DM.

Exp.	**Generator**	**CNN**		**ViT**	
		ACC_{bal}	$F1_{\text{macro}}$	ACC_{bal}	$F1_{\text{macro}}$
1	-	0.493±0.033	0.423±0.015	0.287	0.348
2	R3GAN	0.506±0.044	0.440±0.018	0.249	0.216
3	R3GAN	0.531±0.031	0.457±0.023	0.322	0.249
4	R3GAN	0.528±0.032	0.461±0.021	0.402	0.331
5	R3GAN	0.443±0.047	0.395±0.016	0.408	**0.465**
2	DM	**0.559**±0.055	0.445±0.029	0.290	0.320
3	DM	0.534±0.034	0.460±0.022	0.293	0.493
4	DM	0.552±0.029	**0.469**±0.017	0.289	0.277
5	DM	0.481±0.050	0.430±0.033	**0.442**	0.440

Table 3: Classification performance of the CNN and ViT in the five experiments with increasing number of synthetic snippets. Best value marked in boldface.

capacity of the shallow CNN is too small to be pre-trained on ImageNet-1K it is initialized with random weights. Thus, we initialize and train the shallow CNN ten times and report the mean and standard deviation of the metrics in Table 3. *Experiment 1* serves as the baseline performance without any synthetic data. Through *Experiment 2* to *5* the number of synthetic images is successively increased (see Table 2). Except for *Experiment 5*, which adds over 470 synthetic samples per class, augmenting the training dataset leads to an improvement in balanced accuracy and macro F1-score for the shallow CNN. With images generated by R3GAN the performance even decreases in *Experiment 5* compared to the baseline. In contrast to this, the ViT required a large training dataset and only shows a clear improvement in balanced accuracy and macro F1-score in *Experiment 5*. For the shallow CNN images from DM lead to a better performance in the individual experiments than images from R3GAN. Also, for most experiments the images from DM lead to a better performance of the ViT. The best overall performance is achieved with the shallow CNN in *Experiment 4* with 100 samples per class and synthetic images generated by the DM. Adding the synthetic snippets increases the balanced accuracy by 12% and the macro F1-score by 11%.



(a)

(b)

(c)

Figure 2: Number of clusters of the synthetic snippets for class (a) Tire, (b) Cylinder and (c) Wreck.

4.3. DIVERSITY ANALYSIS

As described at the end of Section 2 we investigate the variability of the generative models by clustering. We generate 1,000 images of the classes *Tire*, *Cylinder* and *Wreck* with both models and perform the clustering separately for each class. Figure 2 shows the number of clusters for a varying distance threshold. Naturally, as the threshold decreases the number of clusters grows. The black line indicates the mean distance across all 1,000 images for each class as a reference. Based on the number of clusters R3GAN shows a higher variability than DM. Already for a low threshold of $2.5 \cdot 10^{-4}$ the number of clusters if below 50 while for R3GAN each image is still its own cluster. For the class *Tire* and a threshold of 0.001 R3GAN and DM produce 58 and 33 clusters, respectively, which improves over previous results of a transfer-learned GAN with 21 clusters [9].

5. SUMMARY

In this work we have, for the first time, compared a GAN and a DM for the generation of synthetic sidescan sonar images. Both methods are able to create realistic highlight and shadow geometries of different object classes. Augmenting a training dataset for classification with the synthetic snippets leads to a performance improvement of a shallow CNN as well as a ViT.

The better performance is achieved with images from DM which can increase both balanced accuracy and macro F1-score by more than 10% for the CNN and more than 25% for the ViT. We anticipate an improvement, though not as significant, with a more comprehensive dataset that is more representative for a classification problem. We have also analysed the variability of the generated images, which is higher for R3GAN. The better quality and contrast of the images generated by the DM can be a reason for the better results in the classification analysis.

REFERENCES

[1] J. M. Bell, L. M. Linnett: "Simulation and analysis of synthetic sidescan sonar images", *IEE Proceedings - Radar, Sonar and Navigation* **144(4)**, **(1997)**.

[2] C. J. Sanford, B. W. Thomas, A. J. Hunter: "Fourier-domain wavefield rendering for rapid simulation of synthetic aperture sonar data", *IEEE Journal of Oceanic Engineering* **49(4)**, **(2024)**.

[3] Y. Steiniger, J. Stoppe, T. Meisen, D. Kraus: "Dealing with highly unbalanced sidescan sonar image datasets for deep learning classification tasks" in *Global OCEANS 2020 MTS/IEEE Singapore - U.S. Gulf Coast*, (virtual, 2020).

[4] M. Jegorova, A. I. Karjalainen, J. Vazquez, T. M. Hospedales: "Unlimited Resolution Image Generation with R2D2-GANs" in *Global OCEANS 2020 MTS/IEEE Singapore - U.S. Gulf Coast*, (virtual, 2020).

[5] J. Ho, A. Jain, P. Abbeel: "Denoising Diffusion Probabilistic Models" in *Advances in Neural Information Processing Systems*, **33**, **(2020)**.

[6] O. Bryan, T. Berthomier, B. D'Ales, T. Furfaro, T. S. F. Haines, Y. Pailhas, A. Hunter: "A diffusion-based super resolution model for enhancing sonar images" *Journal of the Acoustical Society of America* **157(1)**, **(2025)**.

[7] Y. Steiniger, J. Stoppe, D. Kraus, T. Meisen: "On the detection and classification of objects in scarce sidescan sonar image dataset with deep learning methods" in *7th Underwater Acoustic Conference and Exhibition*, (Kalamata, 2023).

[8] M. Arjovsky, S. Chintala, L. Bottou: "Wasserstein generative adversarial networks" in *Proceedings of the 34th International Conference on Machine Learning*, (Sydney, 2017).

[9] Y. Steiniger, D. Kraus, T. Meisen: "Generating synthetic sidescan sonar snippets using transfer-learning in generative adversarial networks", *Journal of Marine Science and Engineering* **9(3)**, **(2021)**.

[10] Y. Huang, A. Gokaslan, V. Kuleshow, J. Tompkin: "The GAN is dead; long live the GAN! A modern baseline GAN" in *The 38 Annual Conference on Neural Information Processing Systems*, (Vancouver, 2024).

[11] A. Jolicoeur-Martineau: "The relativistic discriminator: a key element missing from standard GAN" in *International Conference on Learning Representations*, (New Orleans, 2019).

[12] J. Choi, J. Lee, C. Shin, S. Kim, H. Kim, S. Yoon: "Perception Prioritized Training of Diffusion Models" in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, (New Orleans, 2022).

[13] C. Saharia, W. Chan, S. Saxena, L. Li, J. Whang, E. Denton, S. Ghasemipour, B. Ayan, S. Mahdavi, R. Lopes, T. Salimans, J. Ho, D. Fleet, M. Norouzi: “Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding”, (arXiv:2205.11487). arXiv. https://doi.org/10.48550/arXiv.2205.11487, **(2022)**.

[14] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, N. Houlsby: “An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale”, (arXiv:2010.11929). arXiv. https://doi.org/10.48550/arXiv.2205.11487, **(2010)**.