Research, Society and Development, v. 13, n. 7, e1613746226, 2024
(CC BY 4.0) | ISSN 2525-3409 | DOI: http://dx.doi.org/10.33448/rsd-v13i7.46226

A Comparison of U-Net with Conditional Generative Adversarial Networks and Cycle-Consistent Adversarial Networks for real seismic data interpolation: Tupi Field

Uma comparação de U-Net com Redes Gerativas Adversariais Condicionais e Redes Adversariais com Ciclo Consistente para interpolação de dados reais: Campo Tupi

Una comparación de U-Net con Redes Adversariales Generativas Condicionales y Redes Adversarias Consistentes en Ciclos para interpolación de datos reales: Campo Tupi

Received: 06/17/2024 | Revised: 06/24/2024 | Accepted: 06/25/2024 | Published: 06/28/2024

Jaime Andres Collazos Gonzalez
ORCID: https://orcid.org/0009-0002-1539-8018
Federal University of Rio Grande do Norte, Brazil
E-mail: collazos.jaime.andres@gmail.com
Katerine de Jesus Rincon Perez
ORCID: https://orcid.org/0000-0003-0592-3149
Federal University of Rio Grande do Norte, Brazil
E-mail: katerinerinconp@gmail.com
Tiago Barros
ORCID: https://orcid.org/0000-0001-9665-2238
Federal University of Rio Grande do Norte, Brazil
E-mail: joaomedeiros@fisica.ufrn.br
Gilberto Falkembach Corso
ORCID: https://orcid.org/0000-0003-1748-4040
Federal University of Rio Grande do Norte, Brazil
E-mail: gfcorso@gmail.com
João Medeiros de Araújo
ORCID: https://orcid.org/0000-0001-8462-4280
Federal University of Rio Grande do Norte, Brazil
E-mail: joaomedeiros@fisica.ufrn.br

Abstract
Deep learning models have been used to improve seismic trace interpolation in recent years. Encode-Decode models, such as U-Net, have been implemented to solve interpolation problems. The success of U-Net in seismic interpolation has inspired us to test the U-net also as an image generator for further Generative Adversarial Network (GAN) interpolation models. The objective of the present paper is to compare the performance of U-Net inside the GAN models for seismic interpolation: the U-Net alone and two GAN models, the conditional GAN (cGAN) and the Cycle Consistent GAN (CycleGAN), both using the U-Net as a generator inside their workflow. We test the methodologies for two scenarios: regular and irregular interpolation. All tests were performed in a real dataset from the Tupi Field which belongs to the Brazilian pre-salt region. A comparison of the statistical metrics shows that cGAN performs better than CycleGAN and the U-Net alone in most cases. The computational training time of the cGAN model, for all interpolation scenarios, is better than the CycleGAN. Finally, the cGAN training time is comparable to the training of the U-Net alone.
Keywords: Seismic interpolation; U-Net; CGAN; CycleGAN; Comparison; Real data.

Resumo
Nos últimos anos, modelos de aprendizagem profunda têm sido usados para melhorar a interpolação de traços sísmicos. Especialmente, modelos Encode-Decode, como U-Net, foram implementados para resolver problemas de interpolação. O sucesso da U-Net na interpolação sísmica nos inspirou a testar a U-net também como um gerador de imagens para outros modelos de interpolação de Rede Adversarial Gerativa (GAN). O objetivo deste artigo é comparar o desempenho da U-Net dentro dos modelos GAN para interpolação sísmica: o U-Net sozinho, e dois modelos GAN: o GAN condicional (cGAN) e o GAN de Ciclo Consistente (CycleGAN), ambos usando o U-Net como gerador dentro de seu fluxo de trabalho. Testamos as metodologias para dois cenários: interpolação regular e irregular. Todos os testes foram realizados em um conjunto de dados reais do Campo de Tupi, que pertence à região do pré-sal brasileiro. Uma comparação das métricas estatísticas mostra que o cGAN tem um desempenho melhor do que o CycleGAN e o U-Net

sozinho. O tempo de treinamento computacional do modelo cGAN, para todos os cenários de interpolação, é melhor que o CycleGAN. Finalmente, o tempo de treinamento do cGAN é comparável ao treinamento apenas da U-Net.
Palavras-chave: Interpolação sísmica; U-Net; CGAN; CycleGAN; Comparação; Dados reais.

Resumen
En los últimos años, los modelos de aprendizaje profundo se han utilizado para mejorar la interpolación de trazas sísmicas. Se han implementado modelos de codificación y decodificación, como U-Net, para resolver problemas de interpolación. El éxito de U-Net en la interpolación sísmica nos ha inspirado a probar U-Net también como generador de imágenes para otros modelos de interpolación de redes generativas adversariales (GAN). En el presente artículo el objetivo es comparar el desempeño de U-Net dentro de los modelos GAN para interpolación sísmica.: solo U-Net y dos modelos GAN: GAN condicional (cGAN) y GAN de Ciclo Consistente (CycleGAN), ambos utilizando U-Net como generador dentro de su flujo de trabajo. Probamos las metodologías para dos escenarios: interpolación regular e irregular. Todas las pruebas se realizaron en un conjunto de datos reales del campo Tupi, que pertenece a la región pre-sal brasileña. Una comparación de las métricas estadísticas muestra que cGAN tiene un mejor rendimiento que CycleGAN y U-Net solo. El tiempo de entrenamiento computacional del modelo cGAN, para todos los escenarios de interpolación, es mejor que el de CycleGAN. Por último, el tiempo de entrenamiento de cGAN es comparable al entrenamiento de solo U-Net.
Palabras clave: Interpolación sísmica; U-Net; CGAN; CycleGAN; Comparación; Datos reales.

1. Introduction

The acquisition of seismic data is a complex issue due to various factors, such as limited access to certain geographical areas, equipment malfunctions, and environmental conditions. These factors can lead to information gaps in the data, and to mitigate this problem, different interpolation methods can be applied (Yilmaz, 2001). Furthermore, seismic data interpolation is usually associated with physical or economic constraints (Porsani, 1999). We cite some interpolation methods used in the industry: filters, wave equations, transform domains, and low-rank theory. Prediction-filter-based methods involve the convolution of seismic data with filters, mainly the f-x domain seismic traces interpolation method (Spitz, 1991). We also cite a method for post-stack 3D data using low-rank matrix completion using information from local events and dips to complete meaningful structures (Ma, 2013).

In general, deterministic methods of seismic data interpolation are used in seismic acquisition, given their effectiveness. A growing area in geophysics is seismic data interpolation using deep learning (DL) methods, especially with convolutional neural networks (CNN). In this study, we work with U-Net, a neural network proposed by (Ronneberger et al. 2015). for medical image segmentation. (Fang et al., 2021). used U-Net in seismic interpolation. In addition, generative adversarial network (GAN) models were introduced in the context of image recognition (Goodfellow et al, 2014). GAN models can also be used for seismic interpolation, as demonstrated by Duo et al, (2023), Chang et al, (2018) and Gonzalez et al, (2023). Several methodologies have been developed according to the situation regarding the GAN methodology. For example, the Conditional GAN (cGAN) generated images close to a specific target (Mirza, & Osindero, 2014). In the geophysical context, (Oliveira, Ferreira, Silva, & Vital Brazil, 2018) used cGAN to develop a seismic interpolation method. Another GAN-oriented model is the CycleGAN (Zhu et al., 2017), which proposes a new training model strategy. For the case of insufficient training datasets, (Kaur et al., 2021). employed CycleGAN for seismic interpolation.

All GAN models need a generator of new images and a discriminator to test the quality of the images. Otherwise, the U-Net is largely employed as an image generator. The success of U-Net in seismic interpolation (Fang et al., 2021), has inspired us to test the U-net also as an image generator for further GAN interpolation models. In this research, we present a comparison among U-Net, cGAN, and CycleGAN for seismic interpolation. To make clear, we compare three methods of seismic interpolation: the U-Net alone, and the cGAN and CycleGAN that use the U-Net as a generator inside their workflow. To compare the methods, we employ the Tupi field dataset. In our study, we use the U-Net network with a given set of hyperparameters together with cGAN and CycleGAN models to understand how the GAN model can improve the capacity of prediction of the

U-Net. Furthermore, we present some scenarios for regular and irregular interpolation and image examples to compare the DL methods. To quantify the comparison, we tabulate some metrics of pixel error, such as Mean Absolute Error (MAE), Mean Square Error (MSE), and signal-to-noise Ratio (SNR). Finally, we also estimate perceptual metrics such as Peak signal-to-noise ratio (PSRN) and Structural Similarity Index (SSIM).

1.1 Generative Networks

Several types of networks are used to make seismic interpolation, such as Generative Networks and the U-Net. In Figure 1 we show a schematic diagram of the neural network models used in this study. Here, we employ two distinct GAN frameworks, namely cGAN and CycleGAN, together with the U-Net as the generator, see Figure 1. To understand this point and the difference between U-net and network type GAN, first, it is important to know that GAN is a framework to train a generative model; this means that GAN has several components that can be replaced depending on the type of image to generate. In a GAN framework, we have a Generator G-model that creates news images, which, in this case, is the U-Net network. In addition, the GAN workflow has a discriminator D, a classification network that tries to identify real or fake images. During the training step, the G and D play a minimax play. G creates real images, and D identifies the fake images created for G. The loss GAN function is given for equation 1.

$$\min_{G}\max_{D} V(D,G) = E_x[logD(x)] + E_x \tag{1}$$

Where x is the target and z is the input. With this scheme, the generator produces a real image. Still, if we aim for a specific type of real image, such as interpolated data, it is necessary to modify the GAN framework, applying a conditional for the output. This configuration is called Conditional GAN (Figure 1c). We can see this variation in the loss function (equation 2)

$$\min_{G}\max_{D} V(D,G) = E_x[logD(x \vee y)] + E_x \tag{2}$$

Where y is the conditional or target, therefore, the generator generates new images similar to the target, and the discriminator *D* learns to identify a real target image.

On the other hand, Cycle GAN (CyleGAN) (Figure 1d) proposed by (Zhu et al, 2020), is a GAN framework that can be used when there are not sufficient datasets to train the DL model. This framework uses two generators G and F and two discriminators, D_y and D_x. The workflow train starts with the Generator G to generate an image $G(z \vee y)$ and then, with the output of G, it is used as the input of generator F to get an image F. The mean idea is to train the G to get an image close to y (target) from the input z, and with the F, it is to get an image close to z (input) from a $G(z \vee y)$ image. The loss function for CycleGAN is defined in equation 3.

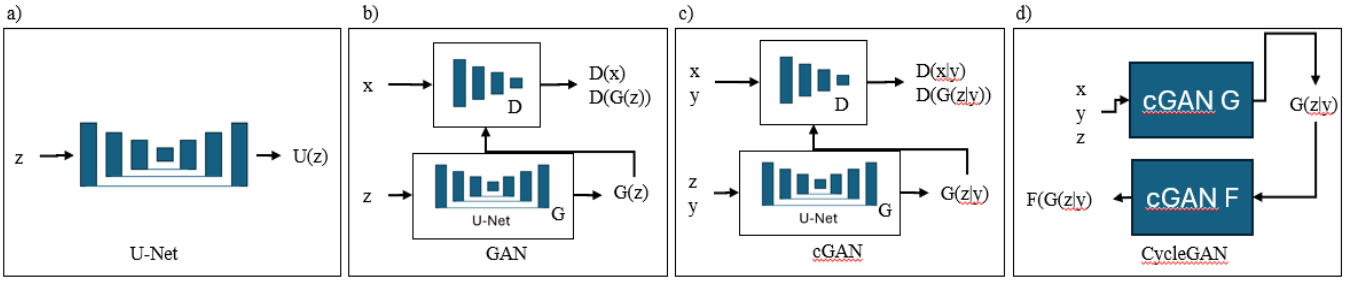
$$L\big(G,F,D_x,D_y\big) = L_{cGAN}(G,D_x) + L_{cGAN}\big(F,D_y\big) + \lambda L_{cyc}(G,F) \tag{3}$$

Where L_{cGAN} is the GAN loss function and L_{cyc} is the Cycle consistency loss, induced by the error of each generator to each cycle, and $\lambda$is the regularized parameter to the Cycle consistency loss. Finally, according to (Zhu et al, 2020), we can have an expression (4) similar to GAN and cGAN.

$$G^*F^* = \underset{G}{argmin}\underset{D}{max}L(G, F, D_x, D_y) \quad (4)$$

Where G and F play the minimax game with D_x andD_y to try to generate a real image, while each discriminator learns to identify real and fake images.

Figure 1 - Diagram of the neural network models used in this study.



Source: Authors (2024).

Figure 1 shows the General diagram of (a) simple U-Net networks and the frameworks: (b) GAN, (c) cGAN, and (d) CycleGAN. The U-Net network is a decoder-encoder model and can be used inside GAN models as a Generator. In this research, we use U-Net as a generator for the cGAN and CycleGAN. The cGAN is a variation of GAN with a conditional variable during the training by its turn the CycleGAN uses the cGAN in two times in a cycle during the training.

1.2 Error Metrics

Metrics are a valuable tool for quantifying image reconstruction results. The pixel-wise metrics are based on measuring the correspondence between the pixels of the original image and the reconstructed image. The neural network literature typically employs the Mean-Squared Error (MSE) or the Mean-Absolute Error (MAE) to measure the quality of an image reconstruction (Snell et al, 2017)., but also the Signal Noise Ratio (SNR). The SNR measures the proximity in amplitude between the target signal and the recovered signal. On the other hand, we applied perceptual metrics to extract structural information and texture, such as Peak Signal to Noise ratio (PSNR), which measures the quality of reconstructed data by comparing it to the original (Wang, Bovik, Sheikh, & Simoncelli, 2004a), and SSIM (Structural Similarity Index) which is focused on structural differences between two images (Wang, Simoncelli, & Bovik, 2004b).The SSIM rates the similarities between an original/reference image and a distorted/compressed image by considering factors such as brightness, contrast, and structure (Wang et al, 2004).

2. Methodology

The fundamental focus of this research is to compare the U-Net network used alone with the U-Net used as a generator inside cGAN and CycleGAN. Using real Tupi field data, the methodologies are tested for seismic interpolation in regular and irregular decimations. To achieve our goal, we first built a dataset with several scenarios. We pre-selected the train, validation, and test sets to ensure that all DL models employ the same training set. Then, we trained the U-Net, cGAN, and CycleGAN models. After training, we applied error metrics to compare the results.

Section 2.1 describes the field dataset used for this study. In section 2.2, we describe the four scenarios for the regular interpolation test and four scenarios for the irregular scenarios test and present some decimated image examples. Section 2.3 describes the workflow and hyperparameters for training the deep learning models. In section 2.4, we present the error pixel

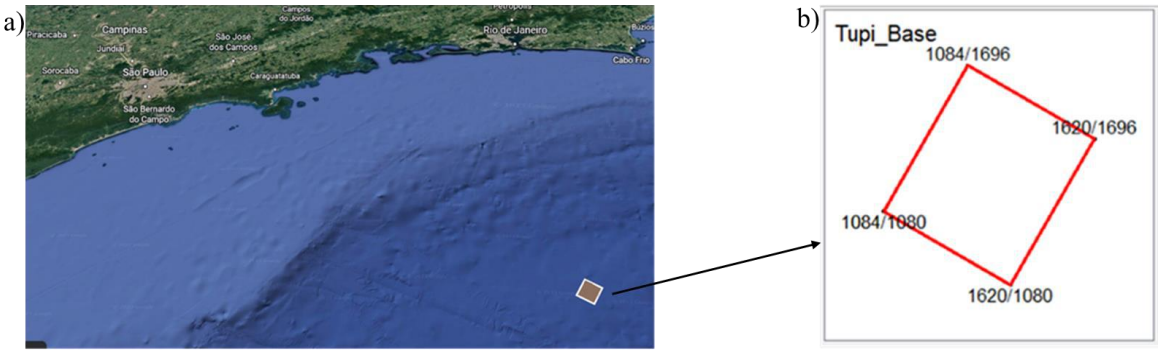
metric and perceptual error metric for each scenario. These initial values will help us to compare the result with the seismic data interpolated using U-Net and GAN models.

For this research, we apply the study case method (Pereira et al., 2018), identifying the capacity of U-Net for seismic interpolation, and its enhancement for a GAN model as a Generator. To conduct this study, we evaluate regular and irregular interpolation. First, we make four datasets for training each one of the models details in section 2.2; each dataset represents a specific scenario; before training the model, we measure the error pixel metric of each scenario with the decimated data. Then, we selected the best hyperparameters after a tuning process. With these hyperparameters, we train the U-Net, cGAN, and CycleGAN (Section 2.3). The quantity of epoch and the learning rate are the same for the Generators and discriminator with the GAN models. This method ensure that all models are trained with the same dataset and parameters. After the training, we measure the error metric for each scenario, presented in section 3. To identify the best result, we visually compare real and predicted image, in addition we numerically compare the error metrics. Finally, we perform a complete interpolation for all inline sections using the cGAN model, which is the model that presented the best error metric and visual results.

2.1 Dataset

This research uses post-stack data from the Tupi field in the Santos Basin, 400 km from Sao Paulo, Brazil. Figure 2a shows the location and shape of the field (Figure 2b). Tupi is the primary oil and natural gas-producing field in the pre-salt reservoir (*Boletim Mensal da Produção de Petróleo e Gás Natural*, 2020). The geometry of this dataset has a bin size of 25m × 25m and a depth sample rate of 5m. For this work, we take part in all post-stack sections, obtain 3D cube data composed of 490 Xline and 430 Inline, and take 800 samples from 2500m depth. Because we are using a 2D U-Net network, we use the 2D image for training. Therefore, we employ the Inline section as a dataset to train the DL model. To apply the DL methodology, 80% of the Inlines were used for training and 20% for testing. From the 80% training data, 75% was used in the training process, and 25% for validation.

Figure 2 - Geographic position and survey of Tupi field. (a) The brown rectangle corresponds to the Tupi field. (b) Survey with Inline and Xline extremes.



Source: Image extracted from Google earth software (2024).

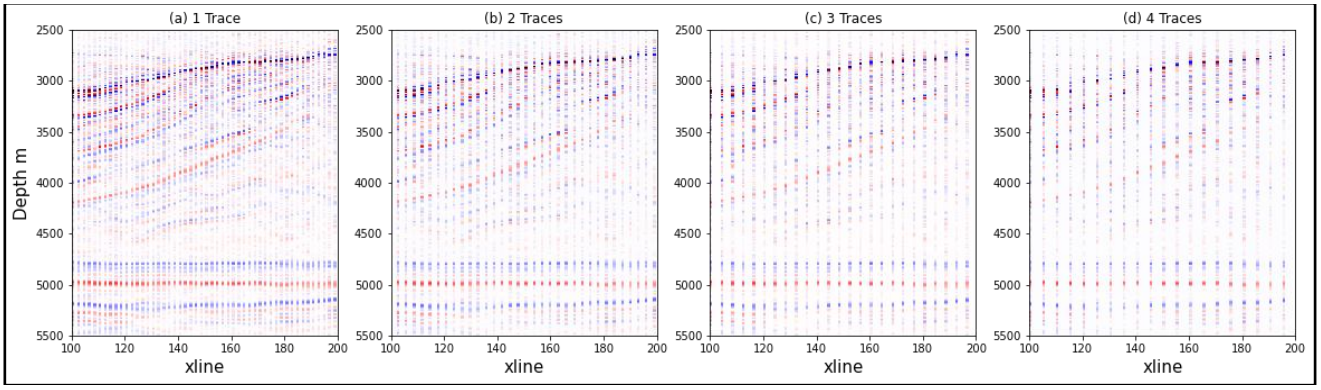
After splitting the dataset, we defined the Inline section as the target and the decimated Inline section as the input of the DL models. The decimated scenarios are defined in the next section. All DL models were trained with the same dataset, it means that the training, validation, and test data were selected with a random function just a single time, and then the same data was used to train, validate, and test each DL model in this study.

2.2 Data Scenarios

This research compares three different DL structures applied to seismic data interpolation. To ensure that the data is the same and make a real comparison, we carefully carried out the scenarios of the decimated data to ensure that the same ones are always used with the different DL models. Exploring the capacity of each DL model for post-stack seismic interpolation, we prepared two scenarios with regular and irregular decimation. Besides, each scenario presented several fractions of decimated traces. These scenarios were created to find which generative network performs best with seismic interpolation using U-Net.

We performed regular decimation inline by inline to create the regular interpolation scenario. We developed four setups: the first with one trace interspersed and decimated, the second with two traces interspersed and decimated, and so on until four traces interspersed and decimated. Figure 3 depicts an example of regular decimation. We created four setups for the irregular scenario (Figure 4) to test the DL model. In this case, we randomly selected a percentage of traces to decimate: 30%, 40%, 50%, and 60%. To clarify, we did not set continuous traces in the scenarios; in some cases, we could have up to 294 traces decimated. In total, we created 8 decimated datasets: 4 for regular and 4 for irregular interpolation.

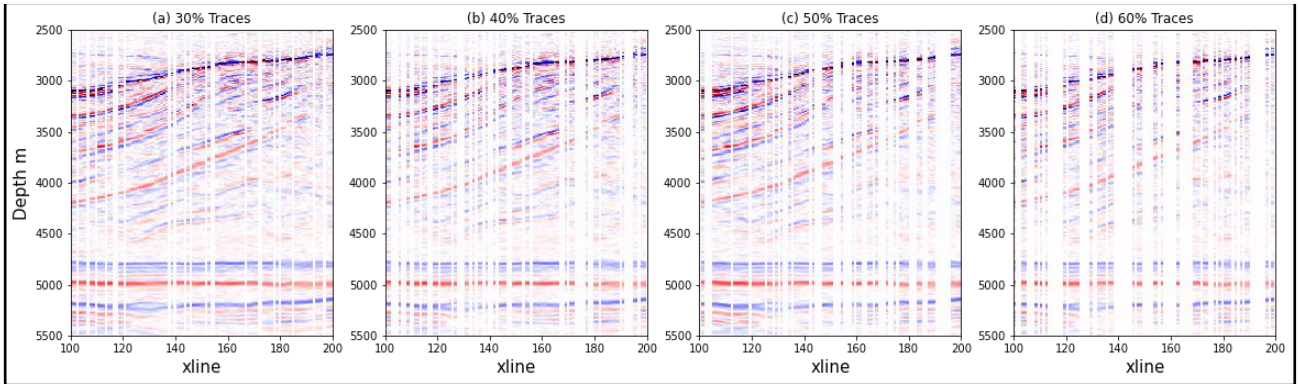
Figure 3 - Example of four scenarios for regular interpolation.



Source: Authors (2024).

In Figure 3, we show the four scenarios of seismic regular interpolation. Figure 3(a) is an example of Inline section with 1 trace decimated, in Figure 3(b) we have an Inline section with 2 traces decimated, Figure 3(c) is an example of an Inline section with 3 traces decimated, and Figure 3(d) an example with 4 traces decimated. We can observe in each scenario the loss information, but, in each image, it is also possible to find the geological events.

Figure 4 - Example of irregular interpolation scenarios.



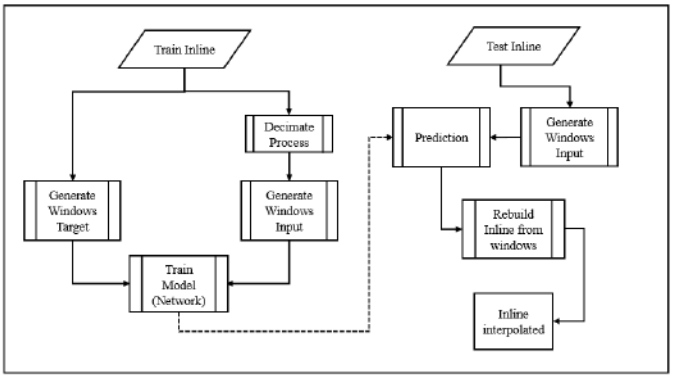
Source: Authors (2024).

In Figure 4, we show four scenarios for irregular interpolation, Figure 4(a) is an example of an Inline section with 30% of traces decimated, Figure 4 (b) an example of an Inline section with 240% of traces decimated, Figure 4(c) a case of 50% of traces decimated and in Figure 4(d) a case of 60% of traces decimated. In this last scenario, it is possible to find sections of missed traces higher than in the regular scenarios. This means, that even having a section with valuable information, there are sections without any information, making it more difficult to get a correct seismic interpolation.

2.3 Training U-Net, cGAN and CycleGAN

In this work, we use the U-Net network proposed by (Isola, Zhu, Zhou, & Efros, 2017). with some modifications. We fixed the input and output size to 256x256x1 and employed just one channel because the seismic image has only one seismic amplitude channel. Moreover, we normalized the input between -1 and 1. The general architecture of the U-Net model consists in five down-sample layers and five up-sample layers, with all output layers having a rectified linear (ReLU) activation function. However, the last layer has a Tanh activation function to produce an output between -1 and 1, similar to the input amplitude range. In our study, this U-Net description was used to perform interpolation and as the Generator in cGAN and CycleGAN interpolation frameworks. To train the U-Net network, we used a kernel of 3x3 for the convolutional operations with a stride of 2 and zero padding, a learning rate of 2x10-4, and trained for 50,000 epochs. All these hyperparameters were used for the generator G in the cGAN (Figure 1c) and the generators G and F (Figure 1d) in CycleGAN. The implementation of discriminators for cGAN and CycleGAN is the same as that proposed by (Isola et al, 2017)., which is called PatchGAN. This setup ensures that the U-Net has similar conditions during training and ensures that the predictions from U-Net make sense and allows a fair comparison between the GAN models.

Figure 5 - General workflow for training and prediction.



Source: Authors (2024).

Figure 5 shows the general workflow implemented for training and prediction. We used the Inline section to train the U-Net and the two GAN models. As previously mentioned, we built a decimated dataset scenario. A notable point is the difference in size between inline sections and the input of the U-Net. As shown in Figure 5, we split each inline section into windows of 256x256 for the training process. Therefore, this process is reversed at the prediction step., We rebuild the predicted windows for each inline from the testing dataset with a taper function to get a full interpolated section. The DL models U-Net, cGAN, and CycleGAN employ the same dataset to train, validate, and test. The decimation process involves regular and irregular decimation.

2.4 Metrics for Decimated Data

For each inline section of the testing dataset, we measured several types of errors grouped into pixel error metrics and perceptual metrics. To estimate pixel metrics error we used MAE, MSE, and SNR. We measured the error inline by inline and then calculated an average across all inline sections from the test. This process was repeated for each scenario of irregular and regular interpolation. For the perceptual metrics, we estimated PSNR and SSIM. This kind of measure provides non-local information, which means, it computes the error in pixel and spatial correlations. The process for calculating these metrics was the same as for the other metrics: inline by inline on the test dataset, and then we calculated the average for each regular and irregular interpolation scenario. Before showing the interpolation result metrics, we can see the result of the measurements carried out on the data without interpolating. Table 1 shows the values related to the pixel-wise metrics and Table 2 shows the values resulting from the perception metrics.

Table 1 - Pixel error metrics using each regular and irregular interpolation scenario.

Decimation	Scenario 1			Scenario 2			Scenario 3			Scenario 4		
	MAE	MSE	SNR	MAE	MSE	SNR	MAE	MSE	SNR	MAE	MSE	SNR
Regular	8.46E-03	4.81E-04	3.011	1.13E-02	6.41E-04	1.763	1.27E-02	7.22E-04	1.246	1.35E-02	7.69E-04	0.968
Irregular	5.06E-03	2.91E-04	5.253	6.77E-03	3.81E-04	4.020	8.36E-03	4.77E-04	3.072	1.03E-02	5.88E-04	2.140

Source: Authors (2024).

Table 1 shows the error metric of the test dataset for regular and irregular interpolation scenarios, these values help us to compare the results. In Table 1 we notice that the MAE and MSE have small values once the difference between decimated and target dataset is just in the missed traces, the other traces of decimated data are equal to the target traces, and the SNR values are low due to the decimated missed trace being taken as a noise.

Table 2 - Perceptual error metrics using each regular and irregular interpolation scenario.

Decimation	Scenario 1		Scenario 2		Scenario 3		Scenario 4	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Regular	39.63	0.913	38.38	0.868	37.87	0.843	37.59	0.827
Irregular	41.87	0.949	40.64	0.927	39.69	0.907	38.76	0.879

Source: Authors (2024).

Table 1 shows the perceptual error metric of the test dataset for regular and irregular interpolation scenario, these values will help us to compare the results. In Table 2, we notice that the PSNR values are close to 40db, we expect that after interpolation these values increase and the SSIM values should be closer to 1 after interpolation.

3. Results and Discussion

The results of this research can be divided into two parts: the first part discusses the error metrics for regular interpolation and the second part for irregular interpolation. Each section shows illustrative examples for all studied scenarios, followed by the error metrics.

3.1 Regular Interpolation Results

Figure 6 shows an illustrative example of the prediction result of the seismic interpolation DL models. The first column corresponds to the decimated input image, and the other columns are the prediction results for U-Net, cGAN, and CycleGAN. In the first row, we have the scenario with one trace decimated; we can see that all DL models make a satisfactory interpolation. However, Table 3 shows that cGAN shows the best metrics result, including perceptual metrics (Table 4). The second scenario, which has two decimated traces, only U-Net and cGAN produce valuable images; CycleGAN is not able to recover the missing information. In Tables 3 and 4, we can see that the pixel and perceptual metrics corroborate the visual inspection of the images in figure 6. For the last scenario with four decimated traces, only cGAN generates a good image. In this case, U-Net generates an image with some noise. This means that training U-Net within a framework like cGAN can, in fact, improve the interpolation of U-Net.

Table 3 - Metrics for pixel error, and regular interpolation.

Model	Scenairio 1 (1 Decimated trace)			Scenario 2 (2 Decimated traces)			Scenario 3 (3 Decimated traces)			Scenario 4 (4 Decimated traces)		
	MAE	MSE	SNR	MAE	MSE	SNR	MAE	MSE	SNR	MAE	MSE	SNR
U-Net	2.47E-03	4.47E-05	13.415	4.59E-03	1.13E-04	9.377	5.95E-03	1.77E-04	7.416	6.98E-03	2.18E-04	6.495
cGAN	2.97E-03	6.23E-05	11.997	5.06E-03	1.38E-04	8.511	6.34E-03	1.99E-04	6.902	7.91E-03	2.76E-04	5.480
CycleGAN	3.51E-03	8.04E-05	10.885	1.39E-02	7.52E-04	1.081	9.43E-03	3.86E-04	4.032	1.15E-02	6.19E-04	1.976

Source: Authors (2024).

Table 3 shows the pixel error metric for regular interpolation. The quantity of decimated traces is indicated in the header. The rows correspond to U-Net, cGAN, and CyleGAN DL models. If we compare the first row of Table 1 for each scenario, we find that the U-Net and cGAN have better metrics, in oppostion to the CycleGAN model.

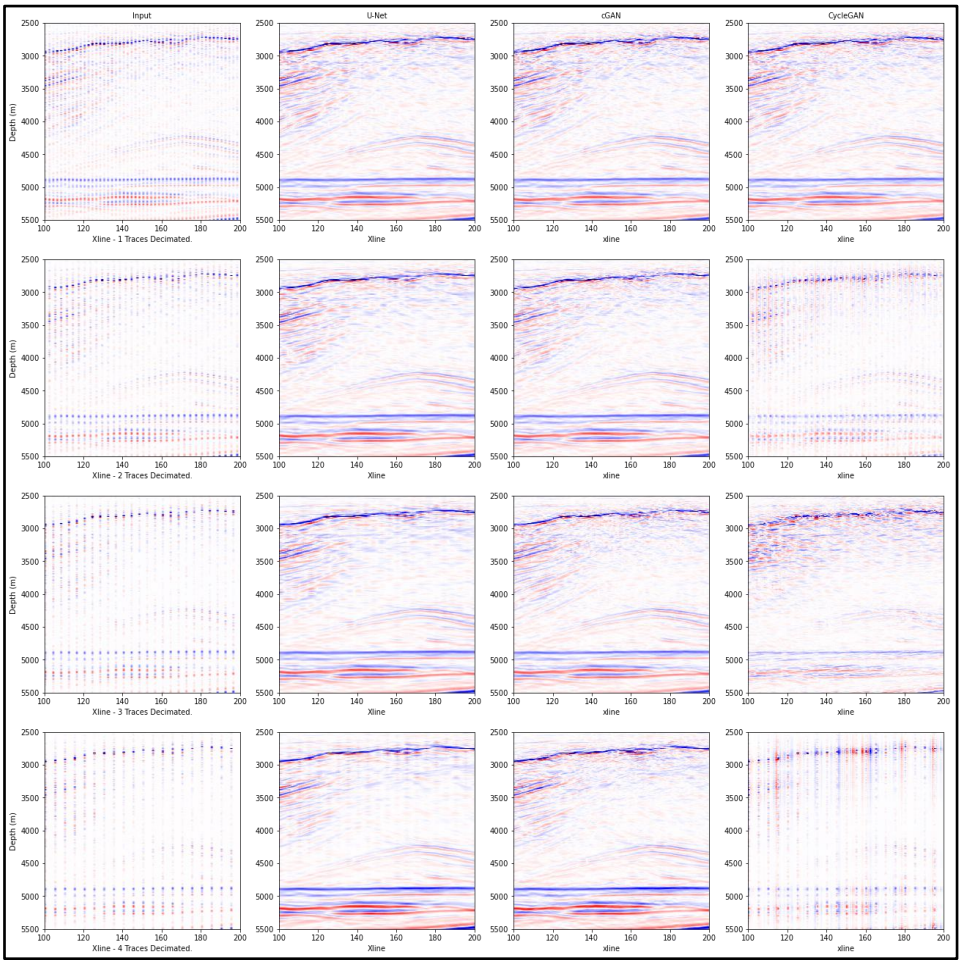
Table 4 - Perceptual metrics, regular interpolation.

Model	Scenario 1 (1 Decimated trace)		Scenario 2 (2 Decimated traces)		Scenario 3 (3 Decimated traces)		Scenario 4 (4 Decimated traces)	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
U-Net	50.03	0.992	46.00	0.981	44.04	0.969	43.11	0.960
cGAN	48.62	0.990	45.13	0.978	43.52	0.966	42.10	0.953
CycleGAN	47.51	0.987	37.70	0.857	40.65	0.938	38.60	0.901

Source: Authors (2024).

Table 4 shows the perceptual metric for regular interpolation. The quantity of decimated traces is indicated in the header of the table. The rows correspond to U-Net, cGAN, and CyleGAN DL models. When we compare the first row of Table 1, the PSNR and SSIM values of U-Net and cGAN are better than CycleGAN, and even we observe that the CycleGAN has values similar to the decimated image.

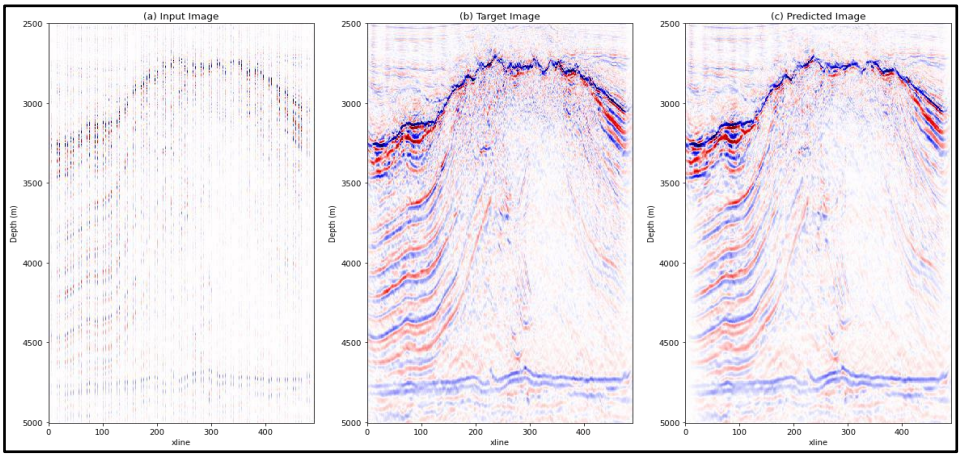
Figure 6 - Regular Interpolation prediction image for the studied DL models.



Source: Authors (2024).

In Figure 6 column 1 corresponds to the input image (Decimated Inline), column 2 to the prediction from U-Net, column 3 to the cGAN prediction, and column 4 to the CycleGAN prediction. Each row shows an example of the scenarios with 1, 2, 3, and 4 decimated traces. We illustrate a typical window selected from all inline sections.

Figure 7 - Example of cGAN model, an inline section for the regular interpolation scenario. (a) Input image with 4 traces decimated. (b) The target Inline section. (c) Predicted image.



Source: Authors (2024).

In the following, we show the results corresponding to an illustrative inline section of cGAN interpolation, this DL model presents the best results for both regular and irregular decimation. Figure 8 corresponds to an inline interpolation using an input with four decimated traces (Figure 7a). Moreover, Figure 7b corresponds to the target image, and Figure 7c to the predicted interpolated image. We notice that the predicted image recovers most of the missing traces.

3.2 Irregular Interpolation Results

The results for irregular scenarios Figure 8 are remarkably like regular scenarios. For 30% of traces decimated, U-Net and the GAN models produce reasonable interpolated images. This situation is also verified in Tables 5 and 6, where all metrics show similar values. However, the results differ significantly for a scenario of 40% and 50% decimation, especially for the CycleGAN model. Visually, U-Net and cGAN present comparable results even for 60% of traces decimated, see Tables 5 and 6. A comparative analysis of Tables 5 and 6 reveals that U-Net and cGAN results are very close, but cGAN performs better. Therefore, for irregular interpolation, similar to regular interpolation, using U-Net as a generator in a cGAN framework improves the prediction results for seismic trace interpolation.

Table 5 - Metrics for pixel error, irregular interpolation.

Model	Scenario 1 (30 Decimated traces)			Scenario 2 (40 Decimated traces)			Scenario 3 (50 Decimated traces)			Scenario 4 (60 Decimated traces)		
	MAE	MSE	SNR	MAE	MSE	SNR	MAE	MSE	SNR	MAE	MSE	SNR
U-Net	5.02E-03	1.65E-04	7.772	6.34E-03	2.16E-04	6.495	7.38E-03	2.54E-04	5.820	9.68E-03	3.81E-04	4.044
cGAN	4.74E-03	1.45E-04	8.363	6.05E-03	1.98E-04	6.886	7.82E-03	2.76E-04	5.489	9.72E-03	3.83E-04	4.019
CycleGAN	4.93E-03	1.61E-04	7.852	7.79E-03	3.50E-04	4.390	9.10E-03	4.13E-04	3.704	1.14E-02	5.49E-04	2.426

Source: Authors (2024).

Table 5 shows the pixel error metrics for irregular interpolation. The quantity of decimated traces is indicated in table header. The rows correspond to U-Net, cGAN, and CyleGAN DL models. We can compare the result with the second row of Table 1. For U-Net and cGAN we find the best results. CycleGAN shows the worst performance.

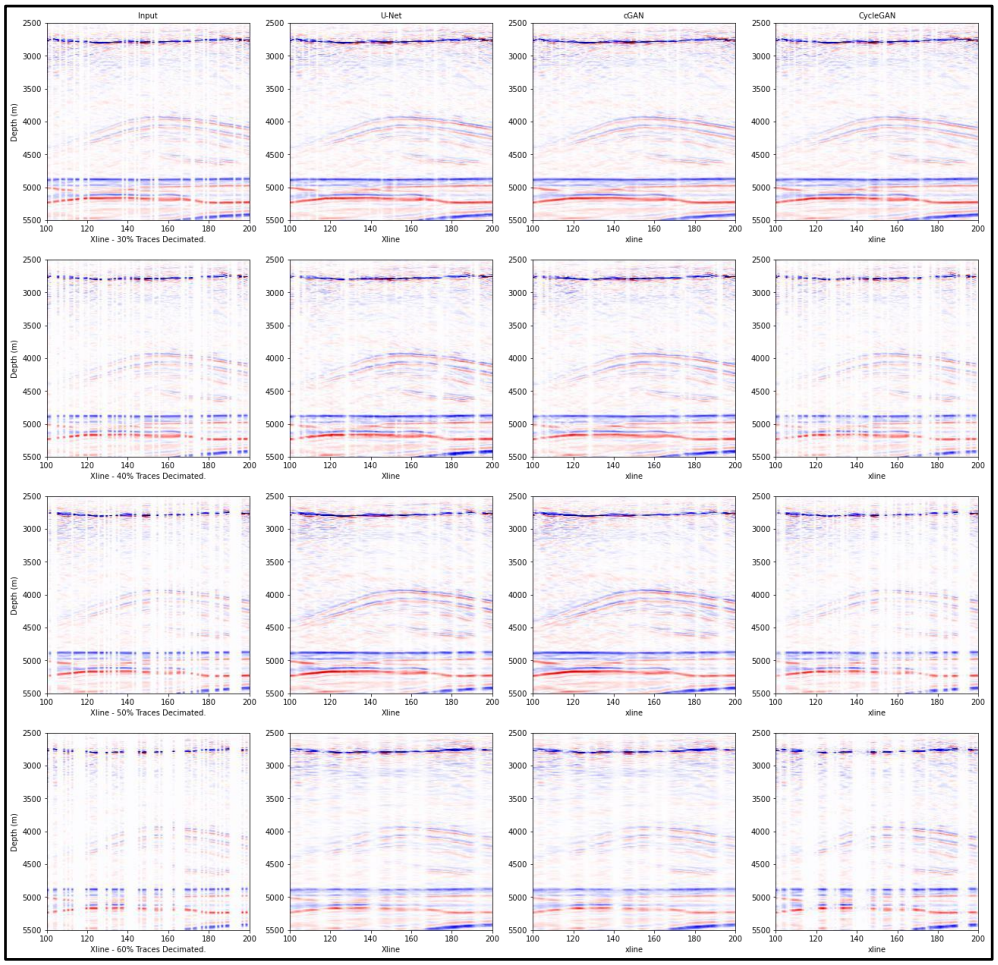
Table 6 - Perceptual metrics, irregular interpolation.

Model	Scenario 1 (30 Decimated traces)		Scenario 2 (40 Decimated traces)		Scenario 3 (50 Decimated traces)		Scenario 4 (60 Decimated traces)	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
U-Net	44.39	0.969	43.12	0.957	42.44	0.952	40.66	0.922
cGAN	44.98	0.974	43.51	0.962	42.11	0.952	40.64	0.921
CycleGAN	44.47	0.972	41.01	0.929	40.32	0.914	39.05	0.881

Source: Authors (2024).

Table 6 shows the perceptual metrics for irregular interpolation. The quantity of decimated traces is indicated in table header. The rows correspond to U-Net, cGAN, and CyleGAN DL models. Comparing the result with the second row of Table 2, we notice that U-Net and cGAN have values greater than CycleGAN, this means that the perception of an correctly interpolated image is better in U-Net and cGAN models.

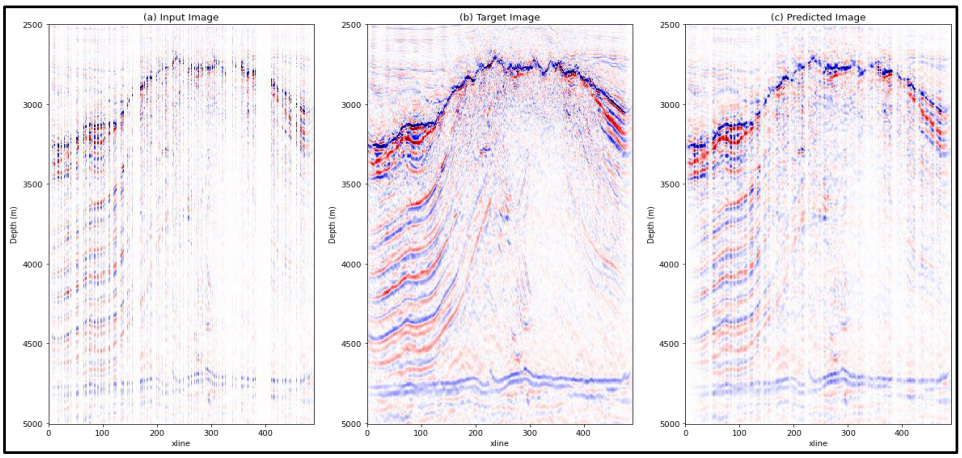
Figure 8 - Irregular Interpolation prediction image for the studied DL models.



Source: Authors (2024).

In Figure 8 column 1 corresponds to the input image (Decimated Inline), column 2 to the prediction from U-Net, column 3 to the cGAN prediction, and column 4 to the CycleGAN prediction. Each row shows an example of the scenarios with 30%, 40%, 50%, and 60% decimated traces. We illustrate a typical window selected from all inline sections.

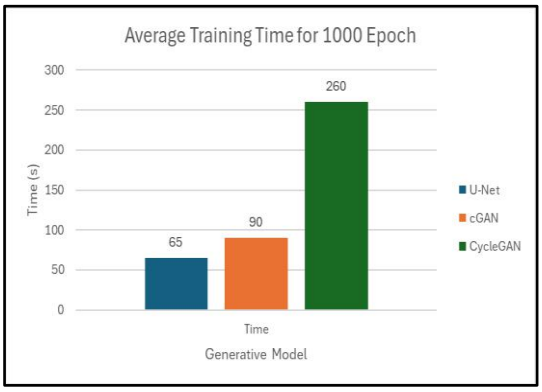
Figure 9 - Example of cGAN model, an inline section for the irregular interpolation scenario. (a) Input image with 60% of traces decimated. (b) The target inline section. (c) Predicted image.



Source: Authors (2024).

Figure 9 is an example of an irregular interpolation with 60% of traces decimated (Figure 9a). It follows, Figure 9b which is the target, and Figure 9c, which shows the predicted image. In this last case, we observe a significant gap in some regions with many missing traces; in this case, the cGAN model cannot fully recover the lost traces' information. However, increasing the number of training epochs could improve the model results. We have not performed such a task because the main idea of this study is to compare the performance of U-Net within GAN-type networks.

Figure 10 - Comparing the computational time for 1000 epochs for U-Net, cGAN, and CycleGAN.



Source: Authors (2024).

Finally, to compute the performance of the models, Figure 10 shows the training time for each model and a comparison among U-Net, cGAN, and CycleGAN. We noticed that U-Net has the lowest value and CycleGAN has the highest. In the case of CycleGAN, it is a framework that requires training on two generative models, so the training time increases. Therefore, for commercial purposes, using CycleGAN as an interpolation tool for large seismic acquisitions can be inefficient, and finding the best hyperparameters could be even more challenging. On the other hand, the training time for cGAN is close to U-Net, with better results. Therefore, using cGAN in interpolation problems is the best choice in real datasets.

4. Conclusion

The results of our study corroborate previous results showing that the U-Net is a good seismic interpolation method. In addition, we show that the cGAN together with the U-Net generator is a valuable seismic interpolator. For regular decimation, the cGAN framework shows good prediction accuracy. Comparing the statistical metrics, cGAN shows better values than CycleGAN and the U-Net alone in most cases. This fact indicates that cGAN recovers the information of decimated traces for both regular and irregular scenarios. Moreover, the computational training time of the cGAN model for several interpolation scenarios, both regular and irregular, is much better than the CycleGAN.

CycleGAN can be an option for seismic interpolation, but this method requires tuning the hyperparameters which is a time-consuming task, moreover, the training time for achieving accurate seismic interpolation is also challenging. On the other hand, using the same U-Net setup used for seismic interpolation, we can train the model within a cGAN framework and improve the precision of predictions. One important advantage of cGAN is that its training time is not higher than that of U-Net alone. Regarding the interpolation capacity, when U-Net was trained alone with regular decimation, it could only correctly recover information with up to 3 traces decimated and with irregular decimation up to 50% of traces. However, when the U-Net is trained within a cGAN framework, the interpolation is boosted, as we obtained superior quality images with four traces decimated for regular interpolation and accurate predictions for irregular interpolation even for 60% of traces decimated. To conclude, the cGAN model improves the precision of image prediction, increasing the capacity for seismic interpolation.

In future work, we plan to explore other GAN models. For instance, the Deep Convolutional GAN (DCGAN), and Super Resolution GAN (SRGAN). Additionally, we could research variations of the hyperparameters, as different hyperparameters might yield different results for the GAN model with U-Net, especially considering the hyperparameter of the framework used. Another topic we recommend is evaluating the models with alternative datasets with geological characteristics different from the Tupi field and even testing in the pre-stack domain. Finally, we are planning to use another setup of the U-Net model, with more layers or even with a ResNet-Unet model.

Acknowledgments

The authors gratefully acknowledge support from Shell Brazil through the "New computationally scalable methodologies for target-oriented 4D seismic in pre-salt reservoirs project" at Universidade Federal do Rio Grande do Norte and the strategic importance of the support given by ANP through the R&D levy regulation. This research was supported by NPAD (High-Performance Computing Center) at UFRN. We are grateful to the Brazilian Petroleum Agency ANP for granting us access to the seismic data, which allowed us to use it in our research. J. M. de Araújo and G.F. Corso acknowledged the CNPq Brazilian research agency for funding (grant no. 311589/2021-9 and 312763/2019-0).

References

Boletim Mensal da Produção de Petróleo e Gás Natural. (2020). Agência Nacional do Petróleo, Gás Natural e Biocombustíveis. https://www.gov.br/anp/pt-br/centrais-de-conteudo/publicacoes/boletins-anp/boletins/boletim-mensal-da-producao-de-petroleo-e-gas-natural

Chang, D., Yang, W., Yong, X., & Li, H. (2018). Generative adversarial networks for seismic data interpolation. *SEG 2018 Workshop: SEG Maximizing Asset Value Through Artificial Intelligence and Machine Learning, Beijing, China, 17–19 September 2018*.

Dou, Y., Li, K., Duan, H., Li, T., Dong, L., & Huang, Z. (2023). MDA GAN: Adversarial-learning-based 3-D seismic data interpolation and reconstruction for complex missing. *IEEE transactions on geoscience and remote sensing: a publication of the IEEE Geoscience and Remote Sensing Society*, *61*, 1–14. https://doi.org/10.1109/tgrs.2023.3249476

Fang, W., Fu, L., Zhang, M., & Li, Z. (2021). Seismic data interpolation based on U-net with texture loss. *Geophysics*, *86*(1), V41–V54. https://doi.org/10.1190/geo2019-0615.1

Gonzalez, J. C., Da Costa, C. A. N., Pinheiro, D., Perez, K. R., Gebre, M. G., De Araújo, J. M., & Lopez, J. (2023). *Seismic data interpolation with an iterative workflow and generative adversarial networks*. 84th EAGE Annual Conference & Exhibition.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2020). Generative adversarial networks. *Communications of the ACM*, *63*(11), 139–144. https://doi.org/10.1145/3422622

Isola, P., Zhu, J.-Y., Zhou, T., & Efros, A. A. (2017). Image-to-image translation with conditional adversarial networks. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Kaur, H., Pham, N., & Fomel, S. (2021). Seismic data interpolation using deep learning with generative adversarial networks. *Geophysical Prospecting*, *69*(2), 307–326. https://doi.org/10.1111/1365-2478.13055

Ma, J. (2013). Three-dimensional irregular seismic data reconstruction via low-rank matrix completion. *Geophysics*, *78*(5), V181–V192. https://doi.org/10.1190/geo2012-0465.1

Mirza, M., & Osindero, S. (2014). Conditional generative Adversarial Nets. En *arXiv [cs.LG]*. http://arxiv.org/abs/1411.1784

Oliveira, D. A. B., Ferreira, R. S., Silva, R., & Vital Brazil, E. (2018). Interpolating seismic data with conditional generative adversarial networks. *IEEE geoscience and remote sensing letters: a publication of the IEEE Geoscience and Remote Sensing Society*, *15*(12), 1952–1956. https://doi.org/10.1109/lgrs.2018.2866199

Pereira,A. S., Shitsuka, D., Parreira, F.,Shitsuka, R. (2018).*Metodologia da pesquisa científica*. Retrieved from https://repositorio.ufsm.br/.

Porsani, M. J. (1999). Seismic trace interpolation using half-step prediction filters. *Geophysics*, *64*(5), 1461–1467. https://doi.org/10.1190/1.1444650

Ronneberger, O., Fischer, P., & Brox, T. (2015). U-Net: Convolutional Networks for Biomedical Image Segmentation. En *arXiv [cs.CV]*. http://arxiv.org/abs/1505.04597

Snell, J., Ridgeway, K., Liao, R., Roads, B. D., Mozer, M. C., & Zemel, R. S. (2017). Learning to generate images with perceptual similarity metrics. *2017 IEEE International Conference on Image Processing (ICIP)*.

Spitz, S. (1991). Seismic trace interpolation in the *F-X* domain. *Geophysics*, *56*(6), 785–794. https://doi.org/10.1190/1.1443096

Wang, Z., Simoncelli, E. P., & Bovik, A. C. (2004a). Multiscale structural similarity for image quality assessment. *The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003*.

Wang, Zhou, Bovik, A. C., Sheikh, H. R., & Simoncelli, E. P. (2004b). Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing: A Publication of the IEEE Signal Processing Society*, *13*(4), 600–612. https://doi.org/10.1109/tip.2003.819861

Yilmaz, O. (2001). *Seismic Data Analysis: Processing, inversion, and interpretation of seismic data*. Society of Exploration Geophysicists.

Zhu, J.-Y., Park, T., Isola, P., & Efros, A. A. (2017). Unpaired image-to-image translation using cycle-consistent adversarial networks. En *arXiv [cs.CV]*. http://arxiv.org/abs/1703.10593