

Universidade de São Paulo  
Instituto de Astronomia, Geofísica e Ciências Atmosféricas  
Departamento de Geofísica

Maurílio Ferreira Salgado

**Redes Neurais Convolucionais como  
Ferramenta de Interpolação de Dados  
Sísmicos**

São Paulo

2022



Maurílio Ferreira Salgado

# **Redes Neurais Convolucionais como Ferramenta de Interpolação de Dados Sísmicos**

Dissertação apresentada ao Departamento de Geofísica do Instituto de Astronomia, Geofísica e Ciências Atmosféricas da Universidade de São Paulo como requisito parcial para a obtenção do título de Mestre em Ciências.

Área de Concentração: Geofísica

Orientador: Prof. Dr. Carlos Alberto Moreno  
Chaves

Coorientador: Prof. Dr. Roberto Hirata Júnior

São Paulo

2022



*Dedico esta dissertação às duas mulheres mais importantes da minha vida, à minha esposa Ana Lúcia e à minha mãe Vera Lúcia.*



# Agradecimentos

Primeiramente gostaria de agradecer à pessoa que primeiro me incentivou a estudar e assim plantou a semente de tudo o que a educação me proporcionou, à minha amadíssima mãe, Vera Lúcia.

Agradeço muitíssimo a minha amada esposa Ana Lúdia, que não só me incentivou como esteve ao meu lado nessa caminhada pelo mestrado e pela pandemia.

Agradeço aos meus orientadores, Prof. Dr Carlos Alberto Moreno Chaves e Prof. Dr Roberto Hirata Júnior pelo acompanhamento constante do trabalho desenvolvido, pelas sugestões e liberdade de ação e pelas rápidas respostas às solicitações, dúvidas e problemas;

Agradeço ao meu grande amigo, Daniel Balbino, em breve Dr. Daniel Balbino, por todas as conversas, consultorias, dicas e sugestões que tanto me ajudaram e guiaram na realização deste trabalho;

Agradeço a meu amigo e colega de Petrobras, Dr. Bruno Dias Pereira, que só não foi meu coorientador por restrições burocráticas, mas que ainda sim me aconselhou e fez sugestões de grande valor.

Agradeço aos velhos amigos e à família que nos ajudam a manter a sanidade em tempos tão adversos.

Agradeço aos novos amigos feitos durante a realização deste mestrado, cuja convivência, apesar de diminuída pela pandemia, foi bastante agradável e colaborativa, em especial ao meu conterrâneo Klaus.

Agradeço ao Henrique Bueno dos Santos, que mesmo sem me conhecer, dedicou um tempo para me ajudar com o MADAGASCAR.

Agradeço ao relator Prof. Dr. Marcelo Belentani de Bianchi pela leitura e revisão de uma versão preliminar desse texto e pelas sugestões dadas.

Agradeço à equipe do Vision Lab do IME-USP, que foi sempre muito solícita e atenciosa às minhas constantes importunações.

Agradeço a meu amigo Lucas Martins, que muito me ajudou com o processo de transferência dos dados utilizados neste trabalho.

Agradeço à minha empregadora Petrobras, principalmente ao meu amigo e gerente imediato, Luciano Nicolini Toledo, que autorizaram a realização do mestrado, permitindo que dedicasse parte do meu tempo para este projeto.

Agradeço à FAPESP, projeto 15/22308-2.

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001.

*“By far, the greatest danger of Artificial Intelligence is that people conclude too early that they understand it.”*

Eliezer Yudkowsky

*“The question is not whether intelligent machines can have any emotions, but whether machines can be intelligent without any emotions.”*

Marvin Minsky



# Resumo

Redes neurais convolucionais estão com uso crescente na geofísica, destacadamente no imageamento sísmico. As aplicações variam desde classificação de fácies, segmentação, interpretação automática de falhas e horizontes a melhoria da resolução sísmica, filtragem de ruídos, reconstrução de dados, dentre outras. Neste trabalho usamos redes neurais convolucionais profundas, treinadas de forma competitiva (*Generative Adversarial Networks - GANs*), para fazer a interpolação de dados sísmicos migrados e empilhados, que sofriam de lacunas geradas por obstrução durante o processo de aquisição sísmica. Esse é um problema recorrente em aquisições do tipo *towed streamer*, que afeta dados antigos e novos e prejudica a interpretação sísmica da área afetada. Dessa forma, avaliamos modelos treinados com quatro métricas distintas, além na norma  $L1$ , que constitui parte da função objetivo da rede neural. Os dados da área afetada foram reconstruídos, mantendo a coerência entre as linhas adjacentes das imagens reconstruídas, sendo que essa análise da coerência não foi encontrada na literatura. Os resultados obtidos nesta dissertação foram melhores do que os obtidos com o método tradicional de interpolação, com funções de predição de erro (PEFs) em todas as métricas de avaliação, sendo que o coeficiente de correlação de Pearson médio entre as imagens interpoladas e as imagens originais, para uma lacuna centralizada com largura fixa de  $124 \times 256$ , foi de 0,65, para o nosso método, e de 0,35, para o método de PEFs. Aos compararmos a mesma métrica com resultados previamente publicados na literatura, tivemos uma correlação média de 0,68 para uma lacuna que perfazia 48,4% da imagem, enquanto que o resultado da literatura possui uma correlação média de 0,49 para uma lacuna de 30% da imagem.

**Palavras chave:** GANs, Redes Neurais convolucionais, Interpolação Sísmica



# Abstract

Convolutional neural networks are increasingly used in geophysics, especially in seismic imaging. Applications range from facies classification, segmentation, automatic fault interpretation, and horizons to improved seismic resolution, noise filtering, data reconstruction, among others. This work uses deep convolutional neural networks, trained in an adversarial way (*Generative Adversarial Networks* - GANs), to interpolate a migrated and stacked seismic data, which suffer from gaps generated by obstruction during the seismic acquisition process. This is a recurring problem in acquisitions of towed streamer types, which affects old and new data and impairs seismic interpretation of the affected area. We evaluate the trained models with four distinct metrics besides the  $L1$  norm, which is part of the objective function of the neural network. We are able to reconstruct the data in the affected area, maintaining the coherence between the adjacent lines in the reconstructed images, and this analysis of coherence was not found in the literature. The results obtained in this dissertation are better than the traditional interpolation method of prediction error functions (PEFs) in all evaluation metrics, and the average Pearson correlation coefficient between interpolated images and original images, for a centralized rectangular gap with a fixed width of 124x256, is 0.65, for our method, and 0.35, for the PEFs method. Comparing the same metric with previously published results, we have an average correlation of 0.68, for a gap that made up 48.4% of the image, while the literature result has an average correlation of 0.49, for a gap of 30% of the image.

**Key words:** GANs, Convolutional Neural Networks, Seismic Interpolation



# Lista de Figuras

1.1	Mapa de cobertura de uma aquisição sísmica antes do <i>undershooting</i> . . . .	24
1.2	Mapa de cobertura de uma aquisição sísmica depois do <i>undershooting</i> . . .	25
1.3	Seção sísmica mostrando lacunas . . . . .	26
2.1	Analogia neurônio- <i>perceptron</i> . . . . .	28
2.2	Exemplo do filtro convolucional Sobel . . . . .	32
3.1	Resultado do trabalho de Halpert (2019) sobre melhoria de resolução. . . .	38
3.2	Resultado do trabalho de Zhang et al. (2019) sobre melhoria de resolução.	39
3.3	Resultado do trabalho de Waldeland et al. (2018) sobre classificação de fácies.	40
3.4	Resultado do trabalho de Liu et al. (2019) sobre classificação de fácies. . .	41
3.5	Resultado do trabalho de Oliveira et al. (2018) sobre interpolação de dados sísmicos . . . . .	42
3.6	Resultado do trabalho de Mandelli et al. (2019) sobre interpolação de dados sísmicos . . . . .	44
3.7	Resultado do trabalho de Wei et al. (2021) sobre interpolação de dados sísmicos . . . . .	45
4.1	Exemplo de uma linha completa, sem o problema de ausência de dados . . .	48
4.2	Exemplo de um <i>patch</i> com máscara ao lado da imagem original . . . . .	49
4.3	Arquitetura da rede ShiftNet . . . . .	50
4.4	Funcionamento da camada de deslocamento (shift-layer) . . . . .	51
4.5	Esquema pictórico das camadas da Rede Discriminadora <i>D</i> . . . . .	52
5.1	Seção sísmica mostrando lacunas - reprodução da Figura 1.3 . . . . .	59

5.2	Exemplo de resultado para o modelo treinado com máscara retangular com 68 <i>pixels</i> de largura. À esquerda está a imagem real mascarada, no centro-esquerdo a imagem real e não corrompida, na centro-direita a falsificação gerada e, à direita, a diferença entre a imagem real e a interpolada. . . . .	60
5.3	Exemplo de resultado para o modelo treinado com máscara retangular com 96 <i>pixels</i> de largura. À esquerda está a imagem real mascarada, no centro-esquerdo a imagem real e não corrompida, na centro-direita a falsificação gerada e, à direita, a diferença entre a imagem real e a interpolada. . . . .	61
5.4	Exemplo de resultado para o modelo treinado com máscara retangular com 124 <i>pixels</i> de largura. À esquerda está a imagem real mascarada, no centro-esquerdo a imagem real e não corrompida, na centro-direita a falsificação gerada e, à direita, a diferença entre a imagem real e a interpolada. . . . .	61
5.5	Loss do modelo com máscara centralizada de 68 <i>pixels</i> . . . . .	62
5.6	NRMS do modelo com máscara centralizada de 68 <i>pixels</i> . . . . .	63
5.7	Pearson <i>R</i> do modelo com máscara centralizada de 68 <i>pixels</i> . . . . .	63
5.8	SSIM do modelo com máscara centralizada de 68 <i>pixels</i> . . . . .	64
5.9	PSNR do modelo com máscara centralizada de 68 <i>pixels</i> . . . . .	64
5.10	Loss do modelo com máscara centralizada de 96 <i>pixels</i> . . . . .	65
5.11	NRMS do modelo com máscara centralizada de 96 <i>pixels</i> . . . . .	65
5.12	Pearson <i>R</i> do modelo com máscara centralizada de 96 <i>pixels</i> . . . . .	66
5.13	SSIM do modelo com máscara centralizada de 96 <i>pixels</i> . . . . .	66
5.14	PSNR do modelo com máscara centralizada de 96 <i>pixels</i> . . . . .	67
5.15	Loss do modelo com máscara centralizada de 124 <i>pixels</i> . . . . .	67
5.16	NRMS do modelo com máscara centralizada de 124 <i>pixels</i> . . . . .	68
5.17	Pearson <i>R</i> do modelo com máscara centralizada de 124 <i>pixels</i> . . . . .	68
5.18	SSIM do modelo com máscara centralizada de 124 <i>pixels</i> . . . . .	69
5.19	PSNR do modelo com máscara centralizada de 124 <i>pixels</i> . . . . .	69
5.20	Exemplo dos recortes feitos na linha 335, centralizados na lacuna maior (em vermelho) e na lacuna menor (em azul). . . . .	72
5.21	Linha 331 antes e depois da interpolação usando o modelo treinado. . . . .	73

6.1	Imagens de teste na parte rasa. À esquerda está a imagem real mascarada, no centro-esquerdo a imagem real e não corrompida, na centro-direita a falsificação gerada e, à direita, a diferença entre a imagem real e a interpolada.	76
6.2	Imagens de teste na porção média. À esquerda está a imagem real mascarada, no centro-esquerdo a imagem real e não corrompida, na centro-direita a falsificação gerada e, à direita, a diferença entre a imagem real e a interpolada. . . . .	76
6.3	Imagens de teste na porção profunda. À esquerda está a imagem real mascarada, no centro-esquerdo a imagem real e não corrompida, na centro-direita a falsificação gerada e, à direita, a diferença entre a imagem real e a interpolada. . . . .	77
6.4	Imagens de teste mostrando armadilhas de interpretação. . . . .	77
6.5	Comparação entre as linhas 330 e 331 após a reconstrução. . . . .	79
6.6	Comparação entre as linhas 331 e 332 após a reconstrução. . . . .	80
6.7	Comparação entre as interpolações por PEF e pela GAN para o raso . . .	82
6.8	Comparação entre as interpolações por PEF e pela GAN para a profundidade média . . . . .	83
6.9	Comparação entre as interpolações por PEF e pela GAN para o profundo .	84



## Lista de Tabelas

2.1	Resumo da função objetivo adversarial . . . . .	35
5.1	Resultados do teste dos modelos para a época 20 . . . . .	71
5.2	Resultados do teste dos modelos para a última época . . . . .	71
6.1	Resultados da comparação PEF-GAN . . . . .	81
6.2	Resultados da comparação com a literatura . . . . .	85



# Sumário

1. <i>Introdução</i> . . . . .	21
1.1 Organização da Dissertação . . . . .	21
1.2 Definição do Problema . . . . .	21
1.3 Visão Geral do Projeto . . . . .	25
2. <i>Fundamentos Teóricos</i> . . . . .	27
2.1 Redes Neurais . . . . .	27
2.2 Aprendizagem . . . . .	29
2.3 Redes Neurais Convolucionais . . . . .	31
2.4 GANs . . . . .	33
3. <i>Revisão Bibliográfica</i> . . . . .	37
3.1 Redes Neurais Aplicadas em Imagens Sísmicas . . . . .	37
3.2 Interpolação de Dados Sísmicos . . . . .	42
4. <i>Base Experimental</i> . . . . .	47
4.1 Dados Utilizados . . . . .	47
4.2 Rede Geradora . . . . .	49
4.3 Rede Discriminadora . . . . .	52
4.4 Funções Objetivo . . . . .	53
4.5 Treinamento Adversarial . . . . .	53
4.6 Métricas de Desempenho . . . . .	54
4.7 Método Comparativo . . . . .	56
4.8 Implementação . . . . .	56

5. Resultados . . . . .	59
5.1 Experimentos . . . . .	59
5.2 Fase de Testes . . . . .	70
5.3 Estudo de Caso . . . . .	71
6. Análises . . . . .	75
6.1 Avaliação Qualitativa . . . . .	75
6.2 Coerência entre as Linhas . . . . .	78
6.3 Avaliação Quantitativa . . . . .	81
7. Conclusões . . . . .	87
7.1 Dos Resultados . . . . .	87
7.2 Do Trabalho Futuro . . . . .	88
Referências . . . . .	89
Apêndice . . . . .	95
A. Resumo Expandido Publicado no 17 <sup>o</sup> Congresso Internacional da Sociedade Brasileira de Geofísica . . . . .	97

## Introdução

### *1.1 Organização da Dissertação*

Esta dissertação é composta de sete capítulos. No primeiro capítulo, trazemos a organização da dissertação, a apresentação do problema que buscamos resolver e um resumo da proposta de trabalho e dos resultados alcançados. No segundo capítulo, descrevemos os fundamentos teóricos das redes neurais para melhor compreensão do que foi realizado. O terceiro capítulo apresenta uma revisão das principais aplicações de redes neurais convolucionais na área de geofísica, com destaque para os trabalhos relacionados à interpolação de dados sísmicos. No quarto capítulo, mostramos a base experimental, descrevendo o dado, as redes neurais utilizadas, as métricas de acompanhamento, o modo de treinamento, um método tradicional para fins comparativos e também detalhes da implementação do código. O quinto capítulo apresenta os experimentos realizados e os resultados obtidos, enquanto o capítulo seis traz uma análise desses resultados tanto no aspecto qualitativo quanto no quantitativo e também uma comparação com outro método de interpolação e com a literatura. Por fim, no capítulo sete, concluímos nossa análise com um resumo do que foi feito e discutimos as possibilidades de trabalhos futuros.

### *1.2 Definição do Problema*

Na exploração e produção de petróleo, o método geofísico que possui maior destaque é a sísmica de reflexão. Sua predominância se deve à alta resolução, alta acurácia e excelente penetração em profundidade, de modo que um poço exploratório dificilmente é perfurado sem uma imagem sísmica para auxiliar em sua locação (Telford et al., 1990).

O imageamento por sísmica de reflexão, especificamente aquele voltado às bacias se-

dimentares produtoras de petróleo, evoluiu muito ao longo da última década, atingindo níveis de qualidade cada vez melhores devido ao avanço tanto das técnicas de processamento quanto das de aquisição sísmica.

Simplificadamente, a sísmica de reflexão se utiliza de uma fonte sonora, a qual emite energia na forma de ondas acústicas que se propagam no meio físico, sendo refletidas nas interfaces de camadas geológicas com impedâncias acústicas diferentes. Essas interfaces são denominadas de refletores sísmicos e a polaridade do sinal depende desse contraste de impedâncias. O sinal refletido é captado pelos receptores, sendo gravado para posterior processamento e formação da imagem sísmica interpretável, ou seja, que pode ser usada para entender a geologia da subsuperfície amostrada pela aquisição. A imagem sísmica final, portanto, é, grosso modo, uma sequência vertical de refletores sísmicos com polaridade ora positiva (interface com camada de menor impedância acima da camada e maior impedância), ora negativa (interface com camada de maior impedância acima da camada e menor impedância), como pode ser visto na Figura 1.3.

Na aquisição *offshore*, as tecnologias de OBN (*Ocean Bottom Nodes*) e OBC (*Ocean Bottom Cables*), cujos receptores ficam no assoalho marinho, vêm crescendo em utilização, principalmente para monitoramento sísmico 4D pela sua excelente repetibilidade, além de trazerem informação em todos os azimutes, registrarem a onda S e possibilitarem maiores *offsets* (Jouno et al., 2020). Na aquisição do tipo *towed streamer* (navio arrastando os cabos com os receptores) destaca-se a *broadband*, que melhora o conteúdo de frequências (Haavik e Landrø, 2015). A escolha entre os tipos de aquisição depende de diversas considerações como a profundidade do reservatório alvo, complexidade geológica, grau de repetibilidade necessário e custo. As aquisições do tipo *streamer* são mais baratas e acabam sendo usadas para reservatórios mais rasos e com geologia menos complexa enquanto que OBN e OBC são mais caras e mais adequadas para geologia complexa e reservatórios mais profundos.

No processamento sísmico, houve avanços importantes na construção dos modelos de velocidades, principalmente com o uso do FWI (*Full Waveform Inversion*) (Penna et al., 2019), dos algoritmos de migração RTM (*reverse time migration*) (Penna et al., 2019), do uso das PSFs (*point spread functions*) (Wrona et al., 2020), remoção de múltiplas internas (Krueger et al., 2019), que tem o efeito combinado de reduzir a quantidade de ruídos, aumentar a precisão do posicionamento dos refletores e a resolução do dado, gerando uma imagem sísmica de altíssima qualidade.

Toda essa evolução foi acompanhada simultaneamente de um grande aumento do poder computacional, permitindo que técnicas antes inviáveis pelo custo computacional, como o FWI, pudessem ser aplicadas em escala comercial. Essa mesma base de capacidade computacional, aliada à grande disponibilidade de dados da atual era da informação também sustentou o florescimento dos algoritmos de aprendizagem de máquinas conhecidos como redes neurais convolucionais (CNNs).

Apesar de todos os avanços de processamento e aquisição, nem sempre é possível aplicar as melhores técnicas, seja por motivos financeiros ou de outra natureza. Além disso, alguns problemas de aquisição, como obstrução por plataforma e receptores defeituosos, ainda não estão completamente resolvidos por técnicas tradicionais e continuam com campo aberto para pesquisa.

Nesta dissertação, o foco se concentra no problema específico do imageamento causado por uma obstrução na aquisição sísmica *offshore* do tipo *streamer* (navio com fonte puxando cabos de hidrofones). Esse problema normalmente acontece quando a aquisição é realizada após a instalação do sistema produtor de petróleo, seja ele uma plataforma fixa ou um FPSO (*Floating Production Storing and Offloading*). A presença dessas estruturas limita a área manobrável do navio que puxa os cabos em uma aquisição do tipo *streamer*, gerando ausência de amostragem da área em subsuperfície abaixo da plataforma.

O primeiro ato de mitigação costuma ser feito na própria aquisição de dados, através da técnica de *undershooting* (Hill, 1986), em que um navio fonte atira o canhão de ar em um lado da obstrução e um navio com receptores adquire o sinal do outro lado. Apesar de bastante útil, essa técnica não é capaz de recuperar os ângulos de reflexão menores. Johann et al. (2006) relata que 6 FPSOs foram rotacionados para minimizar os buracos de aquisição devido à presença dessa infraestrutura no complexo de Marlim, zona com vários campos produtores de petróleo na Bacia de Campos.

Esse é um problema enfrentado tanto em dados mais antigos (Johann et al., 2006), como em aquisições mais recentes (Sano et al., 2020). Para ilustrá-lo, a Figura 1.1 e a Figura 1.2 mostram a cobertura de uma aquisição *streamer* antes do *undershooting* e depois dele. Na Figura 1.2 é possível observar que o *undershooting* ameniza o problema, mas como os autores colocam: “O mapa de cobertura no entorno das plataformas depois de um total de seis linhas de *undershooting* obviamente mostra uma zona incompleta relacionada ao

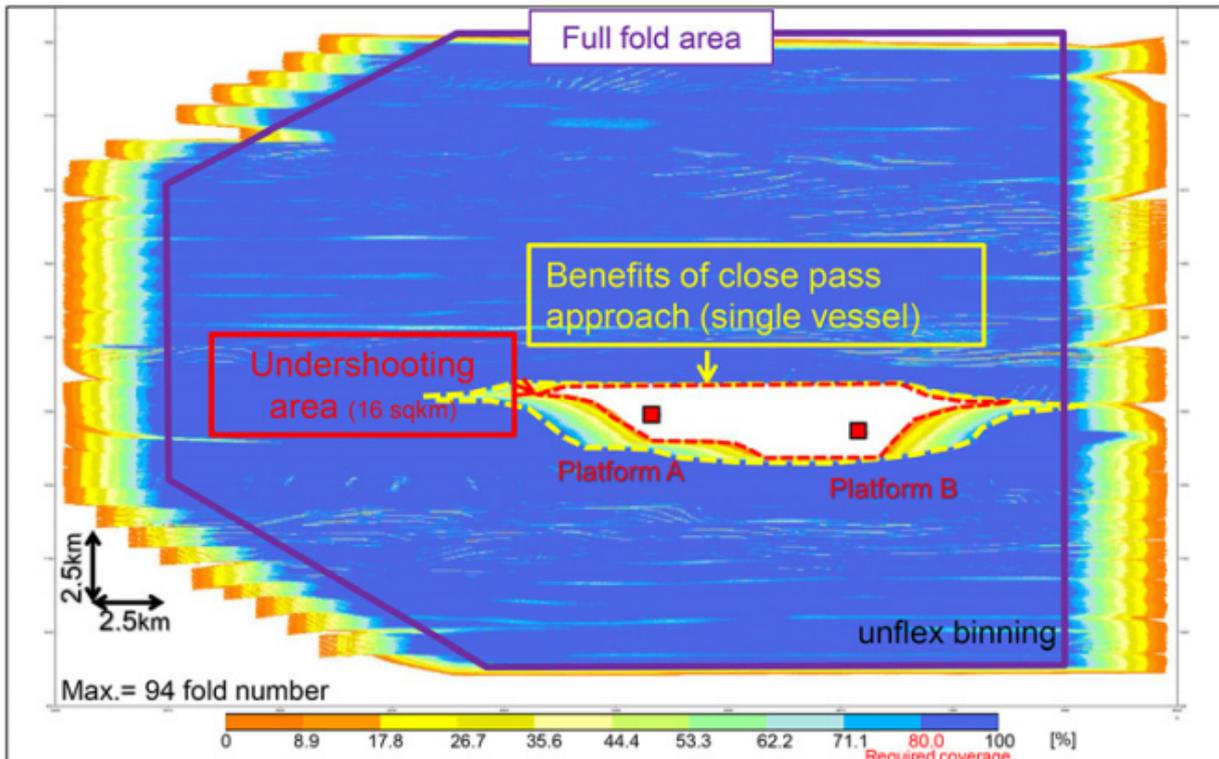


Figura 1.1: Mapa de cobertura de uma aquisição sísmica tipo *streamer* mostrando a ausência de informação sob a região com plataformas produtoras de petróleo. As cores violeta e azul mostram as maiores coberturas enquanto que as cores amarelo e laranja mostram as menores e onde está em branco não há amostragem em subsuperfície. Em vermelho tracejado temos a região que necessita do *undershooting* e em amarelo tracejado a região beneficiada por uma passagem pelo navio bem próxima da plataforma (Sano et al., 2020).

dado de *offset* (afastamento fonte-receptor) curto...” (Sano et al., 2020, tradução nossa<sup>1</sup>).

O efeito em seção sísmica dessa deficiência de *offsets* curtos é uma imagem que fica praticamente sem informação nas porções mais rasas, pois ali os *offsets* de médios a longos sofrem bastante com o efeito de *stretching* na correção de NMO (*normal moveout*), sendo assim silenciados (*muted*). Portanto, na parte rasa, há uma ausência de dados formando buracos, como mostrado na Figura 1.3.

Demonstrada a persistência do problema, mesmo após a aplicação de técnicas como o *undershooting*, cabe à etapa de processamento sísmico mitigar ainda mais os efeitos, fazendo a efetiva fusão e adequação do dado de *undershooting* ao levantamento principal e também através de técnicas usuais como a regularização do dado e a migração. No entanto, o fato é que: “se utilizamos algoritmos de imagem sísmica convencionais, que se

<sup>1</sup> Original: “The binning coverage map around the facilities after a total of six undershooting lines shows an obvious missing zone related to the near offset data”

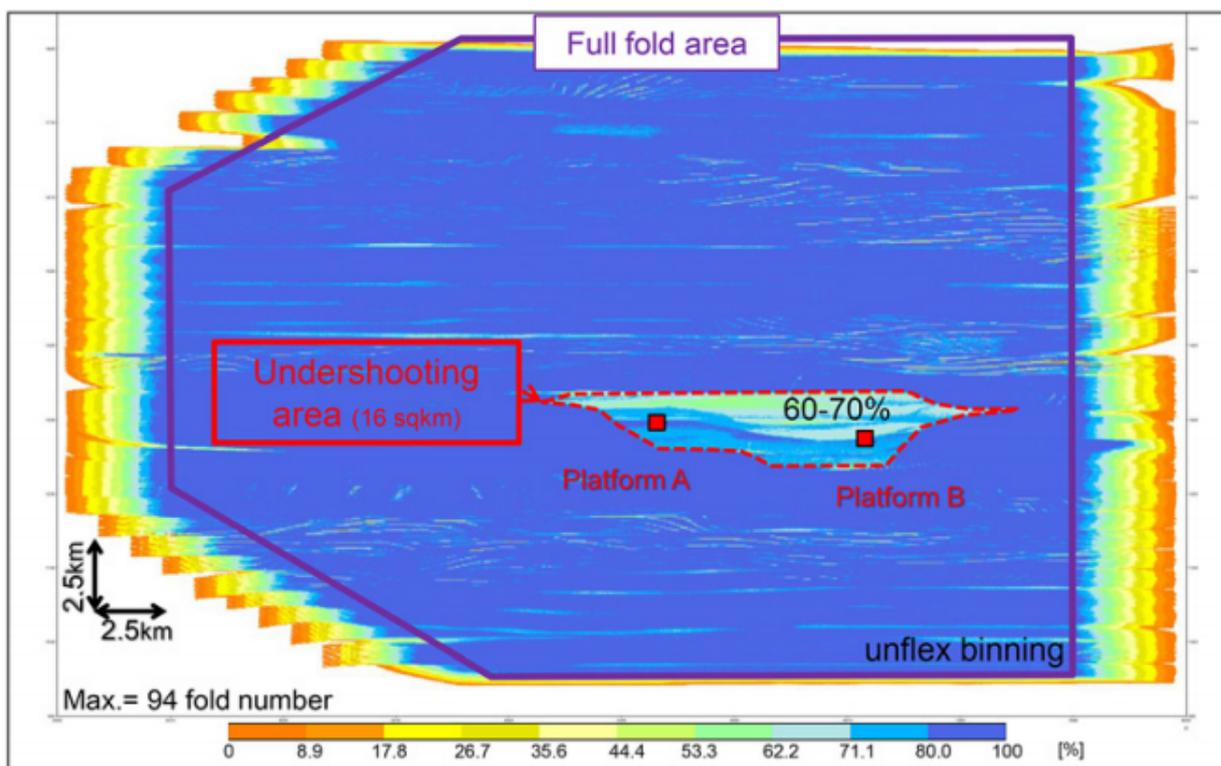


Figura 1.2: Mapa de cobertura mostrando a recuperação de informação gerada pela técnica de *undershooting* na região interna do tracejado em vermelho. Notar que a cobertura, apesar do *undershooting* não chega ao mesmo nível da parte desobstruída (Sano et al., 2020).

baseiam apenas nos dados da reflexão primária, artefatos de migração serão inevitáveis” (Nath e Verschuur, 2020, tradução nossa<sup>2</sup>). Nesse mesmo artigo, os autores propõem usar as reflexões múltiplas para mitigar os problemas de obstrução.

### 1.3 Visão Geral do Projeto

O problema de defeitos de imagem gerados pela ausência de amostragem de uma área em subsuperfície devido à presença de plataforma ainda não possui uma solução satisfatória. Assim, a proposta deste projeto é usar redes neurais convolucionais para remediar a ausência de informação em um dado sísmico. A parte da imagem sísmica sem problemas de iluminação, que possui características semelhantes à parte incompleta, é usada para treinar a rede neural convolucional para que essa aprenda a estatística e o estilo do dado sísmico, sendo capaz de gerar o dado de maneira verossímil na porção defeituosa.

<sup>2</sup> Original: “If we use conventional seismic imaging algorithms that rely on primary reflection data only, migration artefacts become unavoidable”

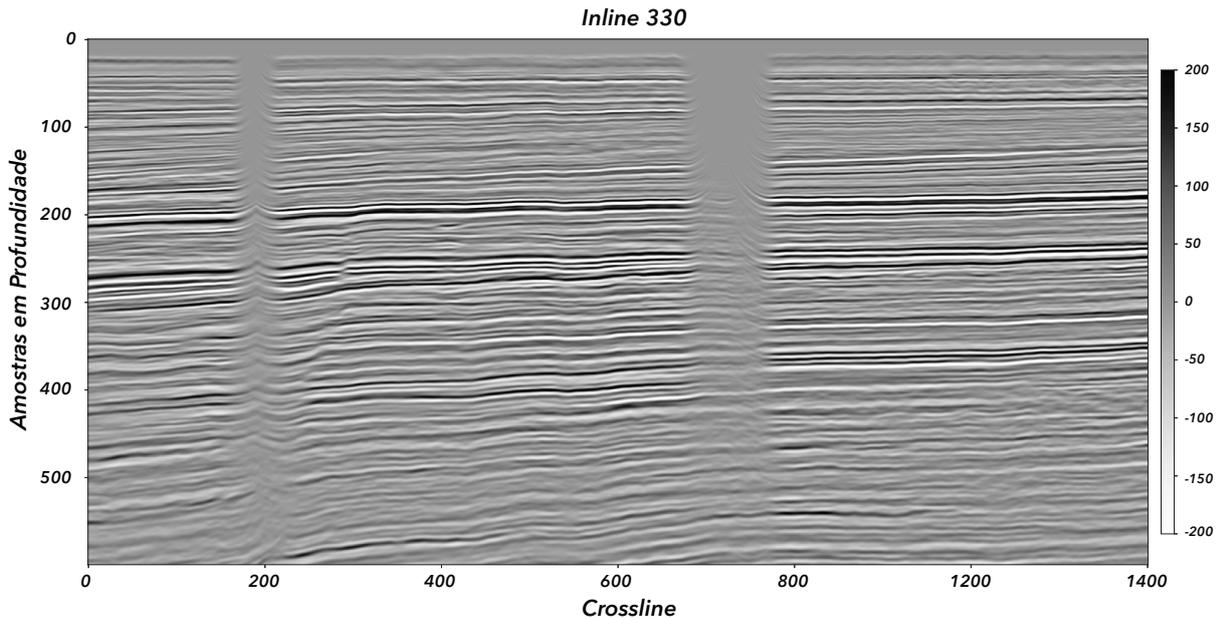


Figura 1.3: Seção sísmica mostrando lacunas na imagem, mais pronunciados na parte rasa. Na borda da região sem informação também se nota o efeito do operador de migração, “espalhando” o dado para cima. O eixo vertical é o número de amostras em profundidade e o eixo horizontal representa os traços (dimensão das *xlines*).

Nós encontramos na literatura uma arquitetura de rede neural adequada para a tarefa de interpolação de dados e que pudesse superar os resultados obtidos em trabalhos anteriores. Testamos diferentes arranjos de treinamento para verificar qual o mais adequado. Avaliamos o desempenho da rede sob diferentes métricas e comparamos com a literatura, obtendo um resultado melhor. Também escolhemos um método tradicional de interpolação para comparação, com a nossa rede sendo superior.

Além das análises quantitativas, realizamos uma apreciação qualitativa do resultado, principalmente do ponto de vista de interpretação sísmica, procurando evidenciar os pontos fracos do método e como ele pode gerar armadilhas para a interpretação. Uma análise ausente em trabalhos anteriores foi a avaliação da coerência das imagens interpoladas em linhas adjacentes do volume sísmico, uma vez que a interpolação acontece em imagens 2D e a rede neural pode produzir imagens bastante diferentes se a entrada for diferente.

O estudo realizado nesta dissertação foi inicialmente apresentado oralmente no XVII Congresso Internacional da Sociedade Brasileira de Geofísica, sendo que um artigo com a análise completa realizada neste trabalho será submetido ao periódico *Geophysics*.

# Fundamentos Teóricos

Este capítulo é dedicado a esclarecer alguns conceitos importantes ligados ao tema de redes neurais que estão relacionados às aplicações feitas nesta dissertação. Boa parte do que está aqui relatado utilizou como base o livro Deep Learning (Goodfellow et al., 2016) e também os sítios <https://machinelearningmastery.com/> e <https://towardsdatascience.com/>.

## 2.1 Redes Neurais

Antes de descrever o modelo utilizado, uma breve explicação sobre as redes neurais se faz necessária. As redes neurais surgiram na década de 40 (McCulloch e Pitts, 1943) e tiveram momentos de aquecimento e de esfriamento em termos publicações científicas. O momento atual é de explosão, com sua utilização transbordando para diversas áreas da ciência e da indústria, e a área de óleo e gás é uma que procura acompanhar essa tendência. Uma boa revisão sobre as técnicas pode ser encontrada em Lecun et al. (2015).

Apesar das redes neurais serem algoritmos antigos, de 2012 em diante, uma espécie de rede (a convolucional) passou a ter grande destaque devido à sua eficiência em algumas tarefas de visão computacional.

As redes neurais são algoritmos inspirados nos neurônios (Figura 2.1), sendo que um trabalho bastante importante nessa analogia foi o que criou o Perceptron (Rosenblatt, 1958), um classificador linear binário, que produz resposta 1 quando o resultado do produto de entradas  $x$  por pesos  $w$ , mais um viés  $b$ , é maior ou igual a zero e resposta zero se o resultado da operação citada é menor do que zero. Em linguagem matemática, tal processo pode ser descrito pela equação 2.1:

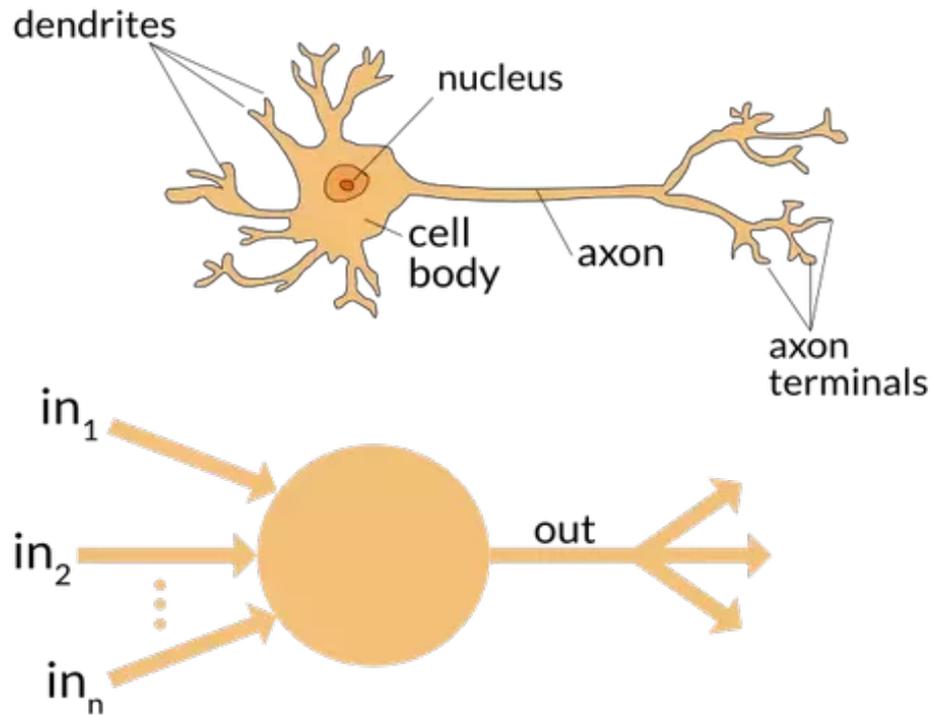


Figura 2.1: Analogia entre um neurônio e um Perceptron. O neurônio recebe os sinais pelos dendritos, que passam pelo corpo celular e são propagados através dos axônios. No perceptron o sinal chega (in) interage com os pesos do perceptron, o que seria o "corpo celular" e por fim sai (out) com o resultado. Fonte: <https://www.quora.com/What-is-the-differences-between-artificial-neural-network-computer-science-and-biological-neural-network>

$$f(x) = \begin{cases} 1, & \text{se } \vec{w} \cdot \vec{x} + b \geq 0 \\ 0, & \text{caso contrário.} \end{cases} \quad (2.1)$$

A analogia pode ser entendida como se um neurônio recebesse um estímulo  $x$ , que provocasse uma reação interna (o combinador linear  $\vec{w} \cdot \vec{x} + b$ ), sendo repassada como sinal ou não, a depender de seu valor. Esse sinal a ser propagado é definido pelo que costuma ser denominado de função de ativação, que no caso do Perceptron é uma função degrau, embora seja muito comum usar funções de ativação não lineares como a tangente hiperbólica e a função logística quando se deseja um sinal contínuo ao invés de binário em problemas de regressão, por exemplo. Uma função de ativação não linear bastante usada é a função ReLU (*Rectified Linear Unit*), que pode ser implementada com  $f(x) = \max(0, x)$  e é descrita na equação 2.2:

$$f(x) = \begin{cases} x, & \text{se } x > 0 \\ 0, & \text{caso contrário.} \end{cases} \quad (2.2)$$

Uma variante da ReLU, também bastante utilizada, é a *Leaky ReLU* (Maas et al., 2013), que ao invés de levar a zero tudo que é negativo, introduz uma reta, com pequena inclinação  $c$ , de modo que a nova função possa ser descrita pela equação 2.3:

$$f(x) = \begin{cases} x, & \text{se } x > 0 \\ cx, & \text{caso contrário.} \end{cases} \quad (2.3)$$

As funções não lineares citadas também tem a propriedade de serem diferenciáveis (exceto a ReLU e *Leaky ReLU* no ponto  $x = 0$ ), o que é importante para que algoritmos de aprendizado de retropropagação (*backpropagation*) alterem os pesos  $w$  dos neurônios quando uma rede de neurônios é treinada. A rede, portanto, nada mais é que associações desses neurônios, onde a saída de um alimenta o outro neurônio e, por fim, após a última camada de neurônios, a rede produz o seu resultado.

## 2.2 Aprendizagem

Para treinar uma rede, uma boa prática é separar os dados em 3 conjuntos distintos, sendo um conjunto para o treinamento em si, um conjunto para validação e outro conjunto para teste. Os únicos dados que são usados para atualização dos pesos da rede são os do primeiro conjunto. Os dados de validação servem para acompanhamento da evolução da rede e auxílio para definir um ponto de parada. Os dados de teste são apenas utilizados apenas ao final do treinamento para avaliação da rede.

No caso de aprendizado supervisionado, que é a forma de aprendizagem a ser utilizada neste trabalho, os pesos da rede são alterados de forma a atender uma função objetivo (*loss function*). A função objetivo, ou função de perda, é de extrema importância e é o guia último do aprendizado. Existem diversas funções objetivo possíveis, cada uma adequada a um determinado tipo de problema, como por exemplo a função quadrática, ou a função de perda entrópica cruzada, dentre outras (Janocha e Czarnecki, 2017).

Um importante aspecto da função objetivo é que ela resume o processo de aprendizagem em um único número. Alguém pode querer maximizar a função objetivo, ou minimizá-la, a

depende de qual problema se quer resolver, de forma que os pesos da rede vão se alterando no sentido de aumentar (no caso de maximização) ou diminuir (no caso de minimização) esse número.

Essa atualização dos pesos é feita através de um algoritmo de otimização. Existe uma vasta literatura sobre métodos de otimização, sendo que os mais usados são de primeira ordem (relacionados ao gradiente). Podemos citar, por exemplo, os métodos de gradiente descendente, como o *Stochastic Gradient Descent* (SGD), o *Root Mean Squared Propagation* (RMSProp) (Hinton, 2012) e o *Adaptive Movement Estimation* (Adam) (Kingma e Ba, 2015). De modo geral, o passo de atualização de cada peso leva em conta o gradiente e uma taxa de aprendizagem que modula o quanto será adicionado em determinada direção. Enquanto o primeiro método usa uma taxa de aprendizagem fixa, os dois últimos usam uma taxa variável (Goodfellow et al., 2016).

O que permite o uso desses métodos é o algoritmo de retropropagação (Rumelhart et al., 1986), que é uma forma de calcular (via a regra da cadeia) o gradiente da função objetivo com respeito aos parâmetros do modelo. O nome retropropagação é porque os gradientes vão sendo calculados a partir da função objetivo e se desdobrando no sentido inverso da rede, das últimas camadas até a camada inicial. A eficiência desse método reside no fato dele usar uma técnica de programação conhecida como Programação Dinâmica.

Podemos entender o mecanismo de aprendizado do seguinte modo:

- O dado de entrada é alimentado na rede, a qual produz um resultado;
- O resultado da rede é comparado com alguma referência através da função objetivo;
- Calcula-se o gradiente da função objetivo com respeito a cada parâmetro da rede (*backpropagation*);
- Atualizam-se os parâmetros segundo um método de otimização (i.e. SGD);
- Repete-se o procedimento.

Na realidade, o passo de calcular o gradiente pode ser realizado ao final da época de treinamento, na média da função objetivo sobre todo o conjunto de treino, como é no caso do método do gradiente descendente, mas o mais comum é que isso seja feito por batelada, em subconjuntos do conjunto de treinamento (*batches*), se aproximando mais do método

SGD que é quando o tamanho desse subconjunto é a unidade. O SGD costuma convergir mais rápido, no entanto, é computacionalmente mais intensivo, então um tamanho maior que 1 costuma ser usado para atingir um compromisso entre eficiência computacional e velocidade de convergência.

Uma vez iniciado o procedimento, é necessário definir um critério de parada. Normalmente, isso é feito com auxílio de um conjunto de dados de validação que é avaliado ao final de cada época (quando todo o conjunto de treinamento passou pela rede) de treinamento para acompanhar a capacidade de generalização da rede neural. Se as métricas do conjunto de validação começarem a piorar em relação ao conjunto de treinamento é sinal que a rede está deixando de aprender e passando a memorizar o conjunto de treinamento, configurando o chamado *overfitting*. Um bom critério de parada é no ponto mínimo da curva do conjunto de validação.

## 2.3 Redes Neurais Convolucionais

A rede convolucional surgiu para tratar principalmente imagens. Imagens possuem dependências espaciais entre seus *pixels* e temporais entre seus *frames*. A convolução é um filtro, ou uma transformação da imagem que depende de uma máscara matricial (núcleo) geralmente menor do que o tamanho da imagem, que é aplicado ponto a ponto na imagem de entrada, de forma a percorrê-la por completo.

Esse conceito pode ser melhor entendido através da Figura 2.2, que ilustra a aplicação de um filtro Sobel. A diferença é que, numa rede neural, os valores do filtro são aprendidos pela rede, e não previamente definidos. Uma camada convolucional é um conjunto de um ou mais filtros que são aplicados nos diversos canais da imagem de entrada. A sequência de camadas convolucionais gera uma rede convolucional. Os *pixels* e posteriormente as suas combinações (pelas máscaras de convolução) nas camadas seguintes da rede são denominados *features* (características) e o treinamento da rede faz com que os filtros realcem as características mais importantes para se resolver o problema proposto.

Existem diversas parametrizações possíveis para uma camada convolucional. As principais são o tamanho do filtro e a quantidade de filtros. O filtro se desloca da esquerda para a direita variando um pixel e, após passar por todas as colunas, desloca-se de cima pra baixo com um pixel de diferença para a próxima linha e a percorre da esquerda para

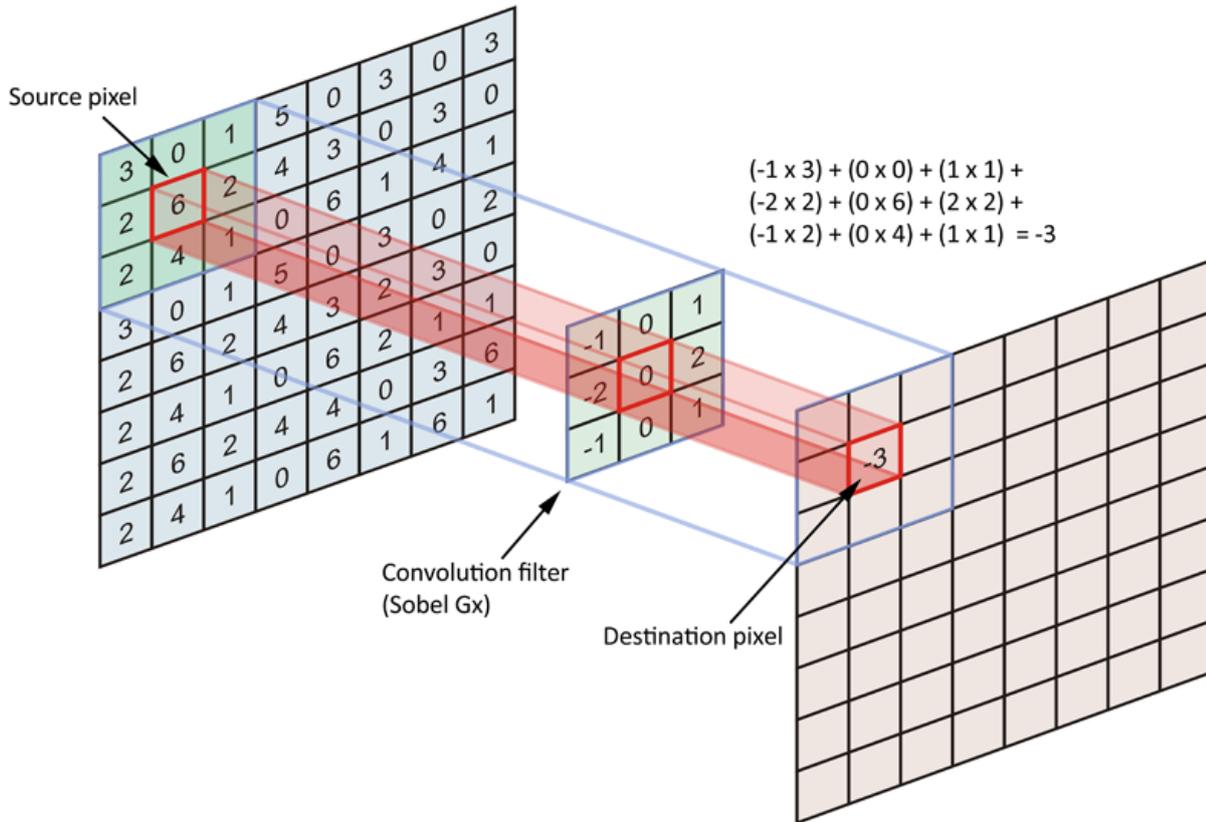


Figura 2.2: Exemplo de um filtro convolucional. O filtro em verde percorre a imagem em azul para formar o resultado em laranja. Fonte: <https://qastack.com.br/datascience/23183/why-convolutions-always-use-odd-numbers-as-filter-size>

a direita novamente, e assim por diante até que toda a imagem seja percorrida. Esse deslocamento de um *pixel* é o passo (*stride*), e geralmente é simétrico nas colunas e linhas. Esse deslocamento pode ser diferente de um *pixel*, de modo que o resultado final terá uma dimensão menor que a imagem inicial, além de diminuir o número de operações. Outro fator que afeta esse tamanho final são os efeitos de borda, sendo comum completar com zeros (*padding*) a borda da imagem antes de iniciar a convolução.

Para evitar o *overfitting*, é importante usar algum tipo de regularização. Duas formas famosas de regularização são o *dropout* e o *batch normalization* (Ioffe e Szegedy, 2015). O primeiro consiste em ignorar um percentual aleatório de neurônios em cada batelada do treinamento e é mais comum de ser usado em redes neurais do tipo MLP (*multi-layer perceptron*). Isso evita que um neurônio seja super ativado. O segundo consiste em normalizar (por batelada) o resultado de uma camada, de forma que a camada seguinte sempre receba entradas com a mesma distribuição. Existem algumas variantes desse segundo método,

como o *instance normalization* (Ulyanov et al., 2016), que diferem no modo de como a média e variância são calculadas para a normalização, i.e., no *instance normalization* a média e variância são calculadas em cada imagem do lote (*batch*) e não em todo o lote como no *batch normalization* (Ioffe e Szegedy, 2015).

## 2.4 GANs

GANs (*Generative Adversarial Networks*) são “um novo modo de estimar modelos geradores através de um treinamento competitivo, em que dois modelos são treinados simultaneamente: o modelo gerador  $G$ , que captura a distribuição dos dados, e um modelo discriminador  $D$ , que estima a probabilidade que a amostra venha do dado de treinamento e não de  $G$ . O procedimento de treinamento de  $G$  é maximizar a probabilidade de  $D$  cometer um erro.” (Goodfellow et al., 2014; tradução nossa<sup>1</sup>).

O que o modelo gerador  $G$  faz é um mapeamento de um vetor de ruídos  $z$  para uma imagem  $i$ ,  $G : z \rightarrow i$ , de modo que um bom modelo gera imagens  $i$  que tenham a mesma distribuição das imagens de treinamento,  $d$ .

No artigo original, Goodfellow et al. (2014) apresentam a função objetivo como a função valor de um jogo de minimização e maximização com dois jogadores, conforme a seguinte equação:

$$\min_G \max_D L_{GAN} = \mathbb{E}_d[\log D(d)] + \mathbb{E}_z[\log(1 - D(G(z)))]. \quad (2.4)$$

Aqui, as esperanças  $\mathbb{E}_d$  e  $\mathbb{E}_z$  podem ser entendidas como médias no lote (*batch*) de treinamento, em seus respectivos domínios  $D$  e  $G$ .

Na prática, não se treinam as redes simultaneamente, mas alternadamente, de forma que para a rede  $D$  a função deve ser maximizada e é igual a:

$$L_D = \mathbb{E}_d[\log D(d)] + \mathbb{E}_z[\log(1 - D(G(z)))], \quad (2.5)$$

enquanto que para a rede  $G$  ela deve ser minimizada e o único termo que depende de  $G$  é

---

<sup>1</sup> Original: “a new framework for estimating generative models via an adversarial process, in which we simultaneously train two models: a generative model  $G$  that captures the data distribution, and a discriminative model  $D$  that estimates the probability that a sample came from the training data rather than  $G$ . The training procedure for  $G$  is to maximize the probability of  $D$  making a mistake.”

o segundo termo da 2.4, de modo que:

$$L_G = \mathbb{E}_z[\log(1 - D(G(z)))]. \quad (2.6)$$

Goodfellow et al. (2014) também sugerem uma modificação na função objetivo  $L_G$ , uma vez que no início do treinamento a rede D consegue distinguir facilmente entre imagem falsa e real de modo que o termo  $\log(1 - D(G(z)))$  satura. Como alternativa, eles propõe a maximização da seguinte equação:

$$L_{Gnonsat} = \mathbb{E}_z[\log(D(G(z)))]. \quad (2.7)$$

No entanto, a maximização das equações 2.5 e 2.7 implicaria alterações que fizessem o gradiente subir ao invés de descer. Então, a implementação é normalmente feita como um problema de classificação binária, com 0 representando a imagem falsa e 1 a imagem verdadeira, e minimiza-se a entropia cruzada binária (Binary Cross Entropy – BCE) dada pela seguinte equação:

$$BCE_n(x_n, y_n) = -w_n[y_n \cdot \log(x_n) + (1 - y_n) \cdot \log(1 - x_n)], \quad (2.8)$$

em que  $x_n$  é a predição da rede D,  $y_n$  é o gabarito,  $w_n$  é um peso a ser aplicado e  $n$  é o índice do par  $(x_n, y_n)$ , dentro do *batch* (lote) de treinamento. Assim, no caso de estar predizendo  $x_n = D(d)$ , uma imagem verdadeira ( $y_n = 1$ ), apenas o primeiro termo da equação 2.8 sobrevive e, no caso de estar predizendo  $x_n = D(G(z))$ , uma imagem falsa ( $y_n = 0$ ), apenas o segundo termo sobrevive. Apresenta-se, então, as duas imagens (falsa e verdadeira) em sequência à Rede D, e faz-se uma média dos dois termos da BCE com o peso  $w_n$ .

Já para a rede G, também é usada a equação 2.8, só que com a predição de imagem falsa da rede D como entrada  $x_n = D(G(z))$  e  $y_n = 1$  como gabarito, pois o objetivo dessa rede é “convencer” a rede D de que seus produtos são reais. O processo de implementação está sumarizado na tabela 2.1:

Tabela 2.1 - Resumo da função objetivo adversarial

Rede	Imagem	$x_n$	$y_n$	termo final
D	Real	$D(d)$	1	$-w_n \cdot \log(x_n)$
D	Falsa	$D(G(z))$	0	$-w_n \cdot \log(1 - x_n)$
G	Falsa	$D(G(z))$	1	$-w_n \cdot \log(x_n)$

Além da GAN original, é possível estender o conceito para uma cGAN (*conditional* GAN). Neste caso, a rede geradora aprende a mapear de um vetor de ruído  $z$ , mais um condicionante  $c$ , para a imagem  $i$ , de modo que  $G : \{z, c\} \rightarrow i$ . A função objetivo também é modificada e pode ser escrita como na equação abaixo:

$$\min_G \max_D L_{cGAN} = \mathbb{E}_{d,c}[\log D(d, c)] + \mathbb{E}_{z,c}[\log(1 - D(G(z, c)))]. \quad (2.9)$$



## Revisão Bibliográfica

### 3.1 *Redes Neurais Aplicadas em Imagens Sísmicas*

Pelo nível de desenvolvimento tecnológico atual, não existe uma solução completa para o problema das lacunas em dados sísmicos gerados por obstrução durante a aquisição sísmica. Uma vez que os métodos tradicionais ainda não o resolvem satisfatoriamente, técnicas não convencionais podem ser usadas para solucionar o problema.

Neste trabalho, fazemos uso de uma técnica não convencional e, para tanto, apresentamos um breve resumo histórico de seu desenvolvimento e aplicação na geofísica.

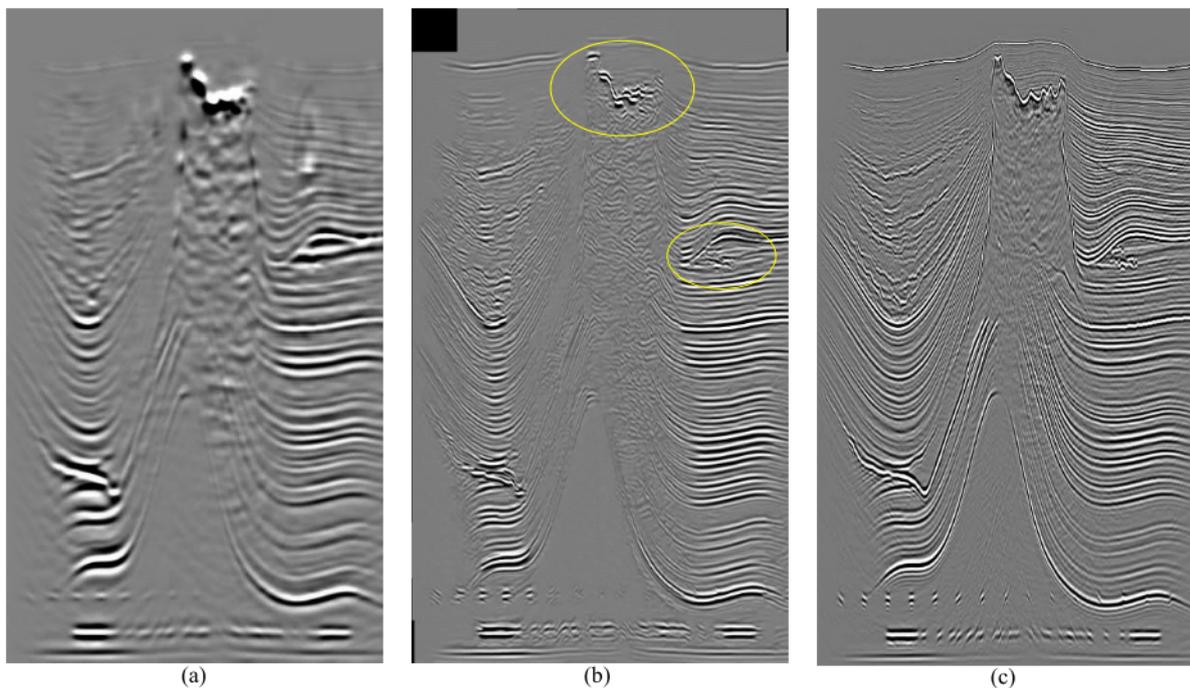
Em 2010, uma colaboração entre os grupos responsáveis pelo ImageNet e pelo Pascal VOC challenge criou o ILSVRC (*ImageNet Large Scale Visual Recognition Challenge*). Esse desafio usou um subconjunto das imagens presentes no banco de dados da ImageNet, o qual possuía 1000 categorias diferentes de imagens. Diversos grupos, empresas e pessoas competiram com variados tipos de algoritmo em algumas tarefas de reconhecimento visual. Em 2011, o erro na tarefa de classificação de imagens foi da ordem de 25%, caindo drasticamente para 16%, em 2012, quando a rede neural convolucional profunda chamada AlexNet (Krizhevsky et al., 2012) ganhou a competição.

Esse foi o gatilho para o ressurgimento das DCNNs (*deep convolutional neural networks*). Dali em diante, diversas arquiteturas de redes neurais convolucionais nasceram, como a GoogLeNet Inception (Szegedy et al., 2014) e a VGG (Simonyan e Zisserman, 2015), respectivamente, campeã e vice da competição de 2014; A ResNet (He et al., 2015), com suas conexões residuais, ganhou a competição em 2015. Com o sucesso da aplicação desses algoritmos, foram surgindo desafios focados em tarefas mais complexas, como localização de objetos em imagens (Redmon et al., 2016; He et al., 2017). Tal técnica também foi

sendo aplicada nas mais variadas áreas como carros autônomos, diagnósticos médicos, sensoriamento remoto, etc.

Uma das fronteiras mais promissoras e com grande aplicabilidade prática é a da visão computacional, principalmente na manipulação de imagens. Segmentação de imagens, classificação de imagens, localização de objetos, super-resolução, interpolação de *frames*, extrapolação de imagens, reconhecimento de padrões, enfim, todas essas tarefas estão sendo realizadas por DCNNs, com grande sucesso, e sua aplicabilidade na geofísica é imediata.

### Deep learning-enabled image enhancement



*Figura 3.1:* Exemplo 2D de melhoria com GAN. Depois da GAN processar recortes de baixa resolução da imagem de teste (a), o resultado reconstruído (b) possui uma melhoria na definição dos refletores e em sua interpretabilidade, especialmente nas áreas indicadas no topo do sal e no complexo de canais. O resultado da GAN é mais similar ao do gabarito (c) do que a da imagem de baixa frequência (Halpert, 2019).

Embora a imagem sísmica seja uma imagem de natureza diferente daquelas do ImageNet, com todas as suas particularidades, em essência, é uma imagem digital, com números positivos e negativos indicando a intensidade de seu monocal. Além da imagem sísmica, existem também os perfis de imagem ultrassônica, que são adquiridos em grande parte dos poços do pré-sal, enriquecendo bastante a interpretação geológica com os detalhes que

captura. Nesses dois tipos de dados geofísicos, onipresentes na indústria de óleo e gás, as técnicas “estado da arte” de DCNNs podem ser diretamente aplicadas, gerando um grande potencial de melhoria das imagens e automatização de algumas etapas de interpretação, como veremos a seguir.

Analisando a literatura recente, pode-se encontrar um aumento expressivo de artigos com aplicações desses algoritmos, e conferências como as da SEG (*Society of Exploration Geophysicists*) já reservam sessão específica para o tema de inteligência artificial e *big data*.

Embora as aplicações sejam variadas, pode-se destacar o seu uso para melhoria de resolução da imagem sísmica. Normalmente, aplica-se alguma arquitetura de GAN (*Generative Adversarial Networks*) (Goodfellow et al., 2014) treinada para gerar imagens de alta resolução a partir de imagens de baixa resolução. Em Halpert (2019), a imagem original foi usada como gabarito do treinamento da rede e imagens com a resolução artificialmente degradada foram a entrada da rede (Figura 3.1). Já em Zhang et al. (2019), o gabarito foi uma imagem de alta resolução gerada por extensão da banda adaptativa e a imagem original é que foi a entrada da rede (Figura 3.2). Infelizmente, ambos trabalhos citados sofrem de ausência de uma métrica quantitativa para analisar a qualidade dos resultados. No entanto, em Dutta et al. (2019) são propostas três métricas já existentes para a avaliação da qualidade dos modelos: *peak signal to noise ratio*, *structural similarity index* e *multiscale structural similarity index*.

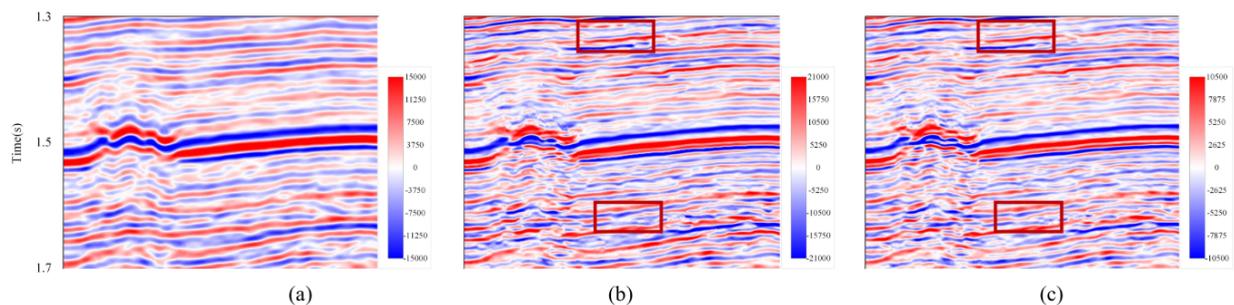


Figura 3.2: Figura mostrando o (a) dado sísmico original, o aumento de resolução através de um (b) método convencional e (c) usando o método de GAN dos autores Zhang et al. (2019).

Outro tema que vem sendo bastante explorado é o de classificação de fácies sísmicas. Waldeland et al. (2018) utilizaram redes neurais para classificar corpos de evaporitos em dados sísmicos (Figura 3.3), enquanto Zhao (2019) usou uma rede neural do tipo *encoder-decoder* para classificar uma seção sísmica 2D em 8 fácies diferentes. Já Liu et al. (2019)

classificaram fácies em um dado sintético usando diferentes formas de dados de entrada, obtendo melhor resultado com a rede treinada em um pequeno cubo sísmico (Figura 3.4).

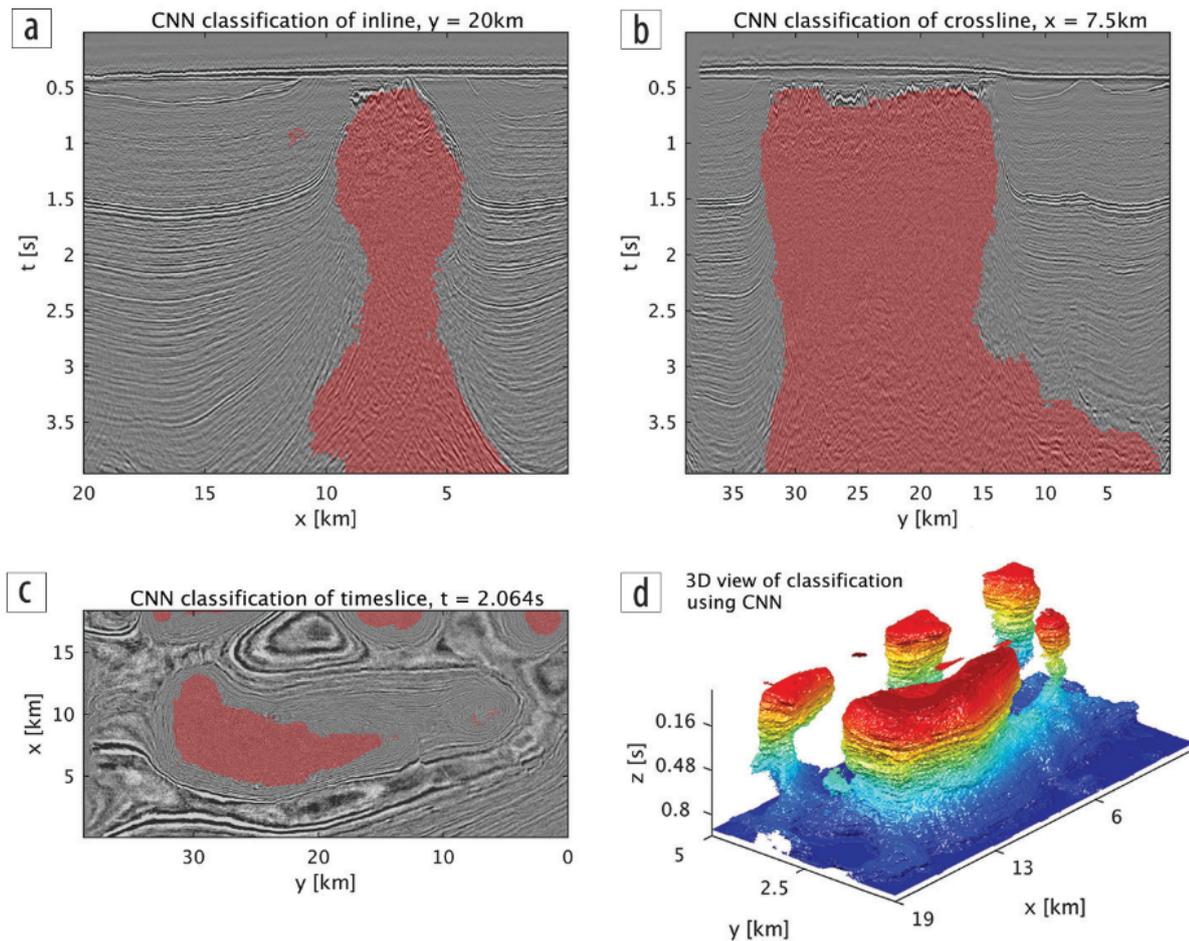


Figura 3.3: (a–c) O corpo de sal predito está marcado em vermelho. (d) O corpo de sal 3D completo predito com a cor indicando o tempo. Modificado de Waldeland et al. (2018).

A tarefa de classificação de fácies exige a interpretação humana das fácies para treinar as redes de modo supervisionado. Como essa tarefa pode ser bastante demorada e também sujeita a erros, muitos autores têm usado dados sintéticos, sendo que a geração dos mesmos, acompanhada da anotação da feição, também gerou um novo *front* de pesquisas. Wu et al. (2020) publicaram um modo de gerar dados sísmicos sintéticos com anotações de falhas e horizontes para treinamento de redes neurais convolucionais.

Algumas aplicações que são menos dependentes de anotações são a recuperação de dados perdidos, e a interpolação de dados, além da já citada melhoria de resolução. Siahkoohi et al. (2019) simularam uma aquisição do tipo OBN (*ocean bottom nodes*), com alguns receptores com problemas e usaram uma GAN para fazer a recuperação dos dados faltantes.

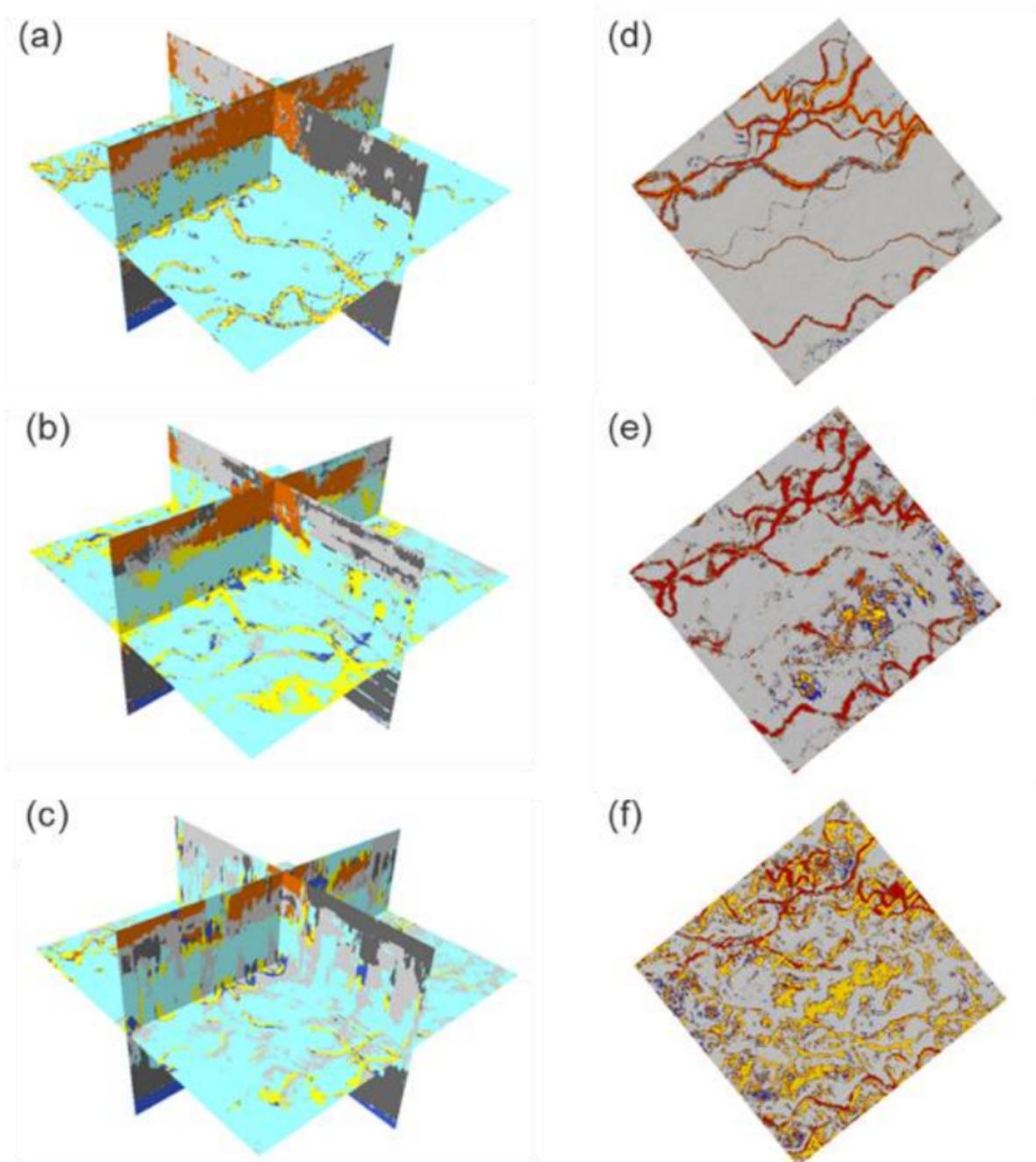


Figura 3.4: Modelo de litofácies 3D mostrando (a) gabarito, (b) predição da GAN, (c) predição da CNN e uma fatia em tempo mostrando (d) gabarito, (e) predição da GAN, e (f) predição da CNN (Liu et al., 2019).

Em editorial para a revista *Journal of Petroleum Engineering*, Carpenter (2019) descreve um trabalho ainda não publicado de completção de informação sísmica usando uma GAN.

### 3.2 Interpolação de Dados Sísmicos

No tema de interpolação de dados sísmicos, o qual é objeto de estudo desta dissertação, destacam-se os trabalhos de Oliveira et al. (2018), Mandelli et al. (2019) e Wei et al. (2021).

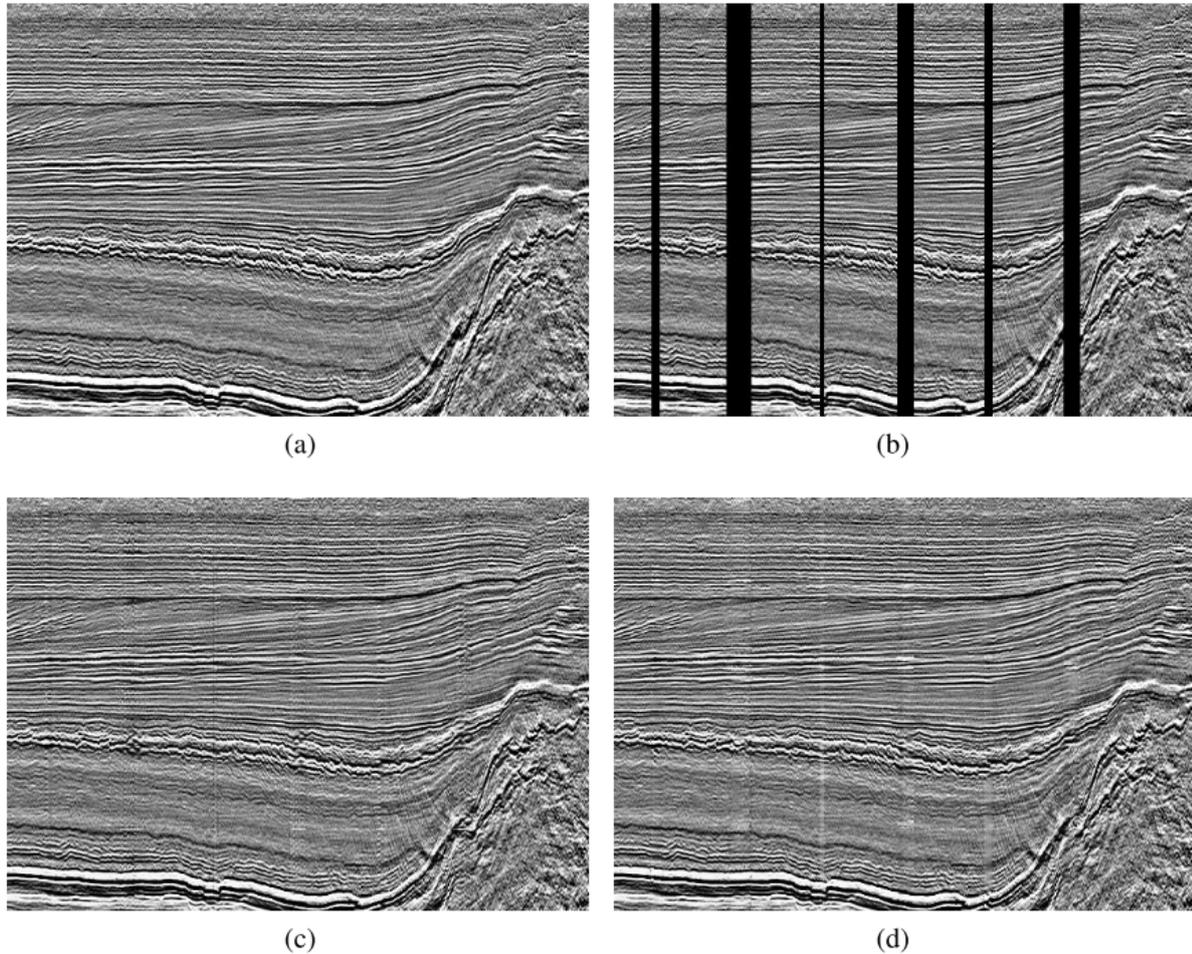


Figura 3.5: Resultados da interpolação dos traços sísmicos. (a) Imagem original, (b) Imagem com as lacunas inseridas, (c) resultado de uma única GAN e (d) resultado de uma GAN especialista para cada largura de lacuna (Oliveira et al., 2018).

Oliveira et al. (2018) utilizam a rede pix2pix (Isola et al., 2017) para a interpolação de dados pós-empilhamento, verificando que quanto maior a lacuna de dados a interpolar, menor o desempenho da rede. Em sua análise quantitativa do resultado, utilizaram o coeficiente de correlação  $R$  de Pearson (Pearson, 1895) para comparar a imagem original e a interpolada pela rede neural, obtendo uma correlação de 49% para uma lacuna de 30% da imagem (24 de 80 *pixels*). Também obtiveram melhor resultado utilizando redes especificamente treinadas em lacunas de tamanhos fixos, do que uma única rede treinada com lacunas de tamanho variável, pelo critério da correlação. Um exemplo do resultado

obtido está na Figura 3.5.

Mandelli et al. (2019) utilizaram a arquitetura U-net (Ronneberger et al., 2015) tanto para filtragem de ruídos quanto para interpolação de dados, inclusive para os dois casos ao mesmo tempo. Sua abordagem não utilizou treinamento adversarial (GAN), mas sim uma função objetivo quadrática, comparando o dado interpolado pela rede com o original. O treinamento ocorreu em imagens sintéticas no domínio do tiro e as lacunas foram geradas aleatoriamente. Os modelos foram treinados para quantidades diferentes de dados ausentes, até um máximo de 50%. Os autores utilizaram a razão sinal ruído (S/N) para avaliar os resultados mostrando que a performance decai quanto maior a quantidade de dado ausente, similar ao constatado por Oliveira et al. (2018). Para 10% de lacunas a S/N foi de 32.8dB e para 50% foi de 18.8dB. Um exemplo do resultado obtido está na Figura 3.6.

Wei et al. (2021) usam a rede pix2pix (Isola et al., 2017) para interpolação de dados no domínio do tiro, com treinamento feito em dados sintéticos. Uma inovação que fazem é o uso de uma camada na rede discriminadora que adiciona um ruído aleatório com distribuição gaussiana, que permite treinar uma rede geradora mais poderosa. O tamanho das imagens de entrada da geradora era de 128x128 e a rede discriminadora da arquitetura pix2pix pode ser alimentada com recortes dessa imagem, o que é denominado *patch* GAN. Os autores testaram diferentes tamanhos de *patches*, obtendo o melhor resultado com *patches* de tamanho 10x10. Também testaram qual o melhor conjunto de treinamento em termos de percentual de traços faltantes e descobriram que um conjunto combinado de 40% e 50% de traços ausentes a rede treinada obtém melhor S/N quando testada em dados com diferentes percentuais de lacuna de dados. Com o modelo treinado nessa configuração, aplicaram o mesmo em um dado real inserindo as lacunas artificialmente, com percentuais de 30%, 50% e 70%, obtendo uma S/N de 24,42, 23,59 e 17,87 dB, respectivamente, para as seções reconstruídas da Figura 3.7.

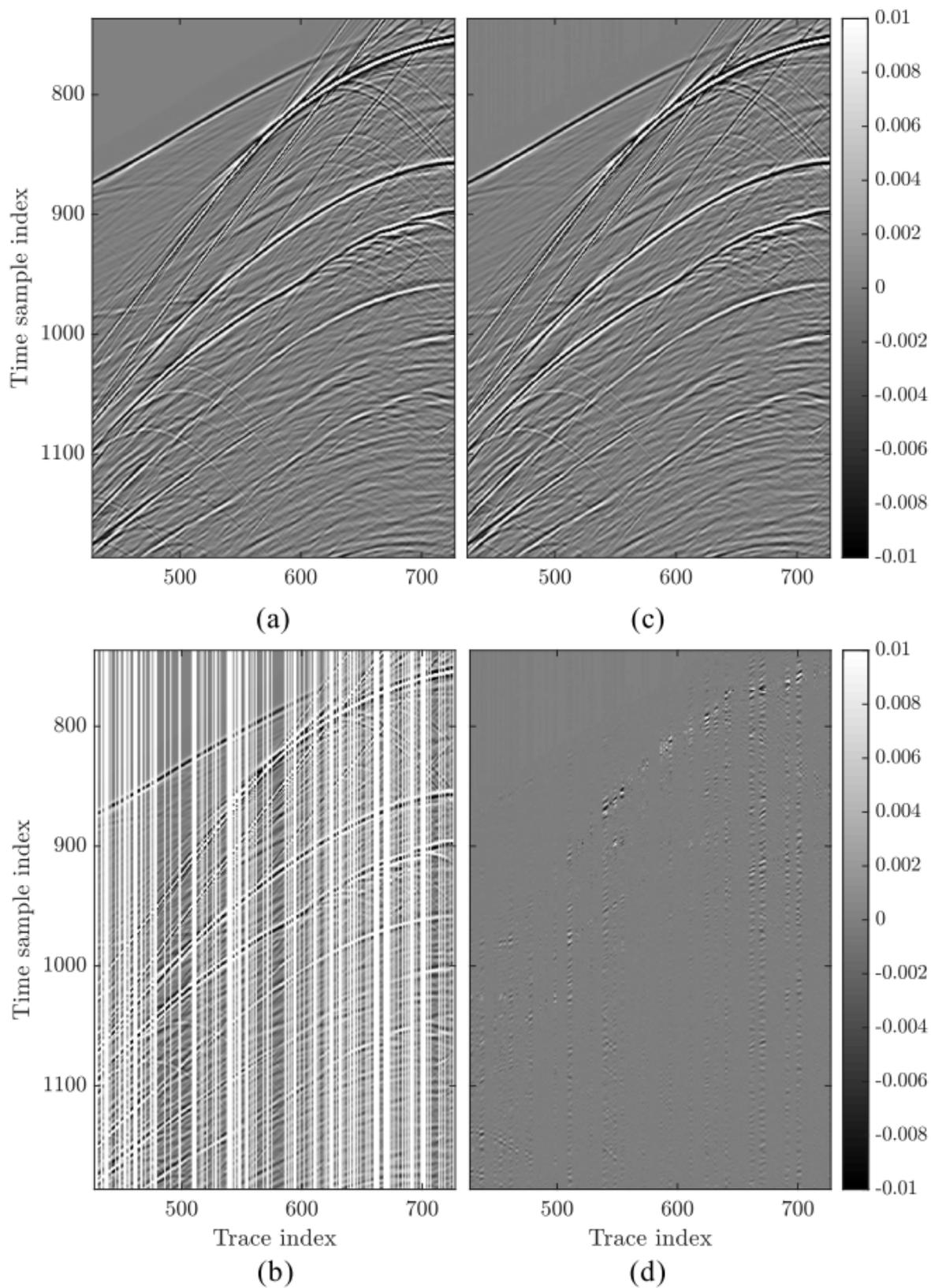


Figura 3.6: Resultados da interpolação dos traços sísmicos. (a) Imagem original, (b) Imagem com as lacunas inseridas aleatoriamente, (c) resultado da interpolação da rede neural e (d) diferença entre a imagem original e a reconstruída (Mandelli et al., 2019).

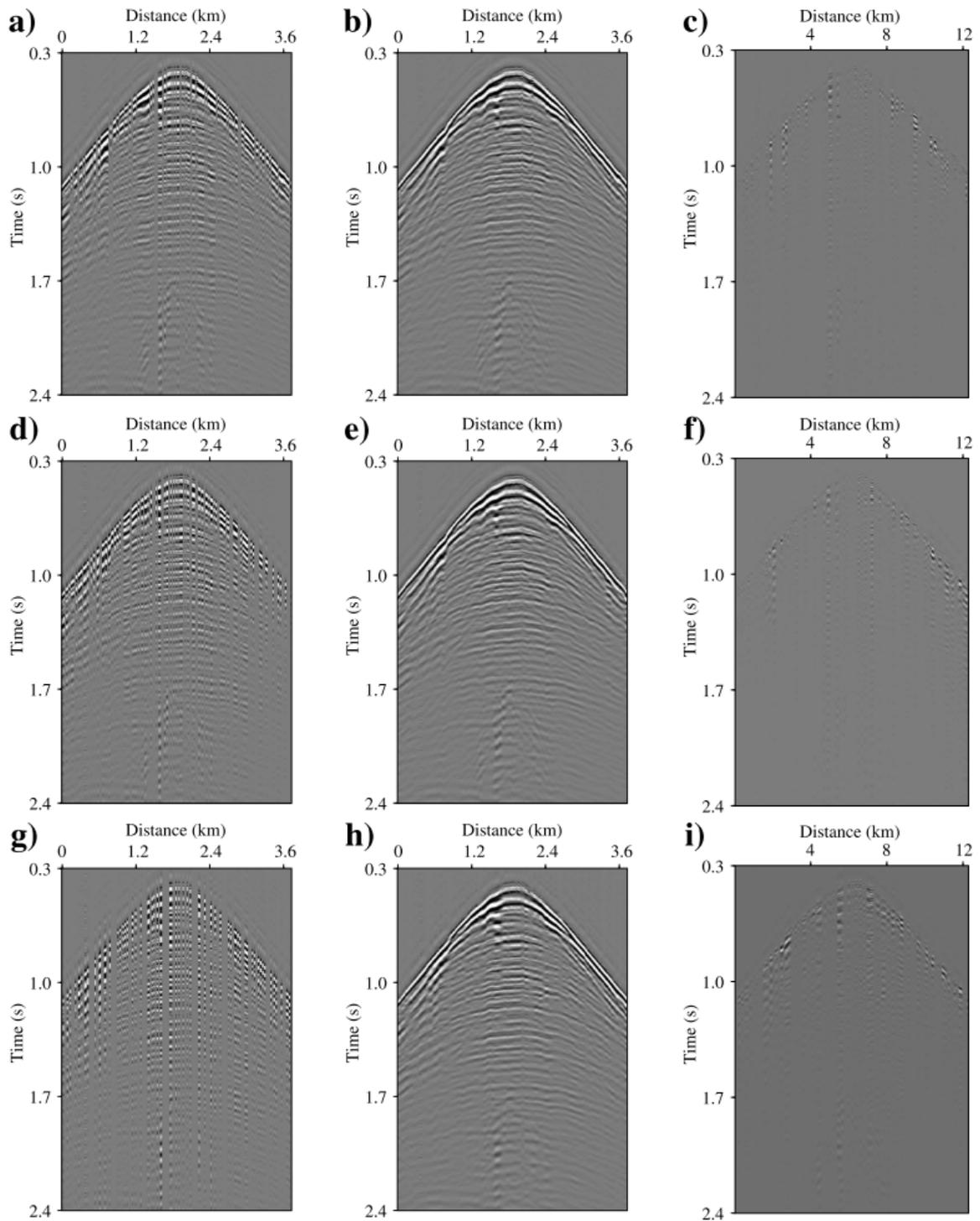


Figura 3.7: Resultados da interpolação dos traços sísmicos em dado real para diferentes taxas de ausência de dados. Lacunas de (a) 30%, (d) 50% e (g) 70%. (b), (e) e (h) São as respectivas interpolações e (c), (f) e (i) os resíduos (Wei et al., 2021).



## Base Experimental

### 4.1 Dados Utilizados

O conjunto de dados utilizados neste trabalho vem do litoral brasileiro, mais especificamente da bacia de Santos. Trata-se de um dado PSDM (migração em profundidade pós-empilhamento), *merge* de duas aquisições diferentes, com técnicas padrão aplicadas para mitigar o problema de obstrução, embora a lacuna ainda esteja presente (Figura 1.3). A localização exata dos dados é omitida, devido a confidencialidade exigida pela Petrobras para a liberação do mesmo.

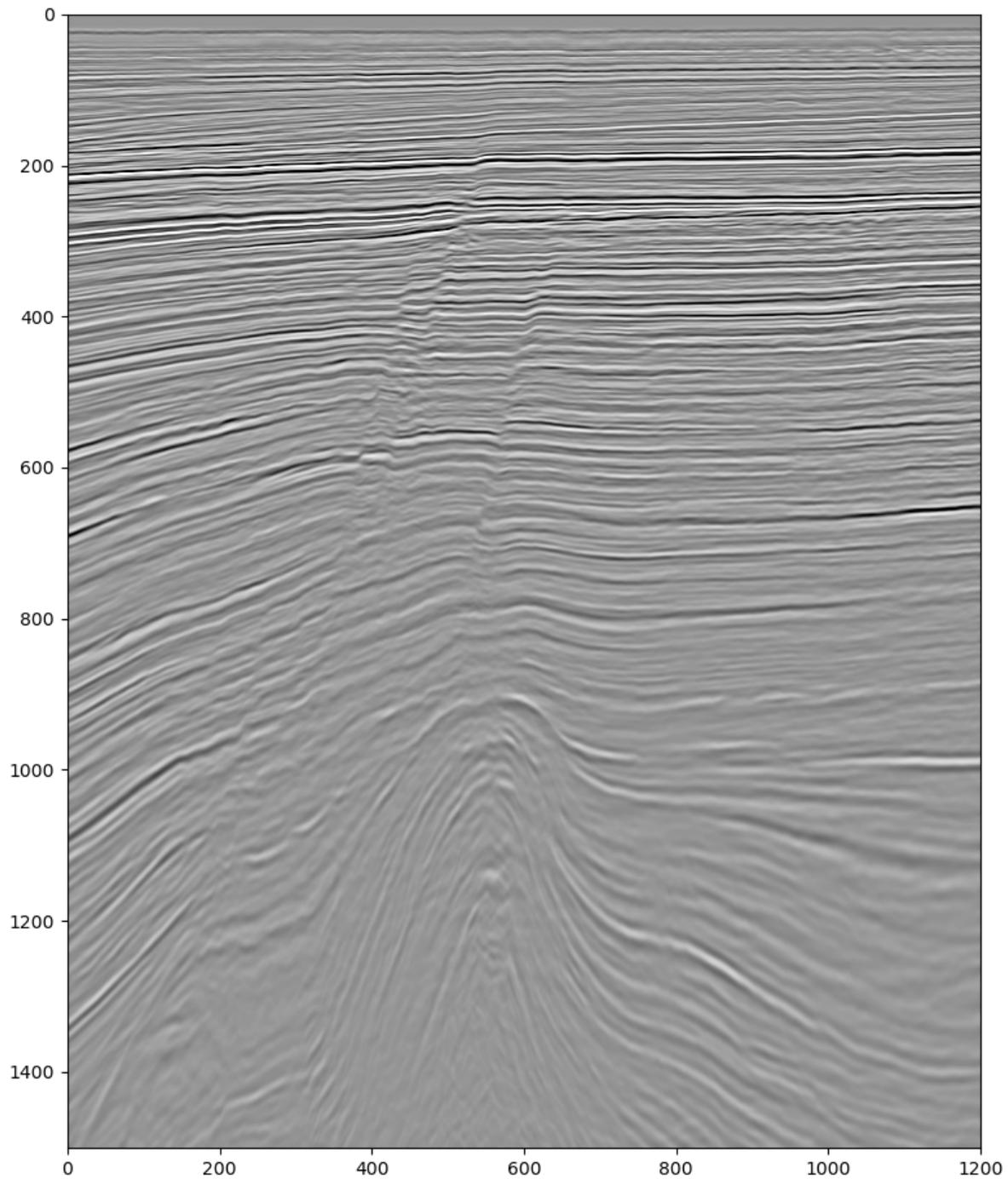
O dado sísmico utilizado é apenas um recorte do dado fornecido pela empresa e possui o tamanho de 800x1200x1500 amostras, das quais 800 são as linhas, 1200 são os traços (*xlines*) e 1500 são as amostras de profundidade (ver Figura 4.1). O intervalo entre linhas, de traços (*xlines*) e a taxa de amostragem vertical são omitidos, por questões contratuais.

O conjunto de treinamentos foi obtido a partir das primeiras 700 linhas, excluindo as linhas com lacuna de informação, totalizando 563 *inlines* utilizáveis. A partir desse volume reduzido, 15000 recortes (*patches*) de imagens 2D com 256x256 *pixels* foram escolhidos aleatoriamente (tanto da direção *inline* quanto da direção *xline*), deixando 12000 para treinamento e 3000 para validação.

As estatísticas dos valores médios e máximos do conjunto de treinamento são armazenadas e usadas posteriormente para normalizar as imagens. Usamos as últimas 100 linhas para testar o modelo, das quais 1000 *patches* foram gerados aleatoriamente.

Todas essas imagens de treino são perfeitas, sem nenhuma lacuna de informação. Para poder simular o problema das falhas nas imagens, geramos as lacunas artificialmente com uma máscara, assim temos a imagem com problema e o seu respectivo gabarito, que é a

imagem não corrompida. A Figura 4.2 mostra um exemplo desse procedimento.



*Figura 4.1:* Exemplo de uma linha completa, sem o problema de ausência de dados. O eixo vertical são as amostras em profundidade e o eixo horizontal são as *xlines*. A escala de cores vai de 200 (preto) a -200 (branco).

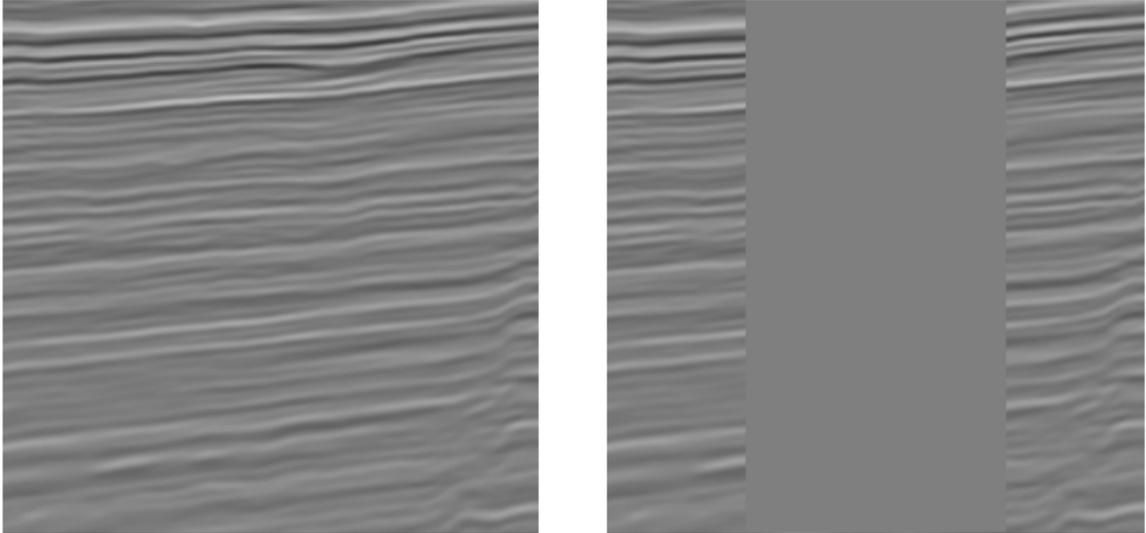


Figura 4.2: Recorte de uma linha sísmica (*patch*) de 256x256 para treinamento da rede. Imagem original e gabarito à esquerda e imagem com máscara retangular centralizada com largura constante de 124 *pixels* e 256 *pixels* de altura gerada artificialmente à direita.

## 4.2 Rede Geradora

A tarefa de interpolação de dados sísmicos utilizando-se de redes neurais já foi testada por alguns pesquisadores. Uma breve discussão sobre esses trabalhos foi realizada na seção 3.2.

A literatura com novos algoritmos de redes neurais é bastante extensa e com novas publicações sendo feitas a todo momento. Nesse cenário, a escolha do melhor modelo da literatura é um problema por si, dada a vastidão de opções e também à obsolescência célere, uma vez que modelos mais novos e melhores provavelmente serão publicados.

Sabendo dessas condições de contorno, a rede escolhida foi a ShiftNet (Yan et al., 2018). A motivação foi seu excelente desempenho na tarefa de completar imagens RGB tradicionais, comparado com *benchmarks* da literatura. Ela superou tanto a U-net simples, usada em Mandelli et al. (2019), quanto a pix2pix usada em Oliveira et al. (2018) e Wei et al. (2021), quando testada na coleção de dados Paris StreetView (Doersch et al., 2015).

A ShiftNet é uma U-net com uma camada adicional chamada *shift-layer*. A arquitetura U-net consiste em um esquema de codificador-decodificador com conexões de salto (*skip connections*) (He et al., 2015) da parte do codificador para o decodificador correspondente. A Figura 4.3 ilustra a ShiftNet, sendo a parte do codificador a primeira, que vai se afunilando (menor altura e largura) e se aprofundando (número de filtros), levando a imagem inicial para um espaço de representação mais compacto, que denominaremos espaço de características (*features*), enquanto que o decodificador vai revertendo a compactação (através de convoluções transpostas e das *skip connections*) até voltar a uma representação de mesmo tamanho da imagem inicial.

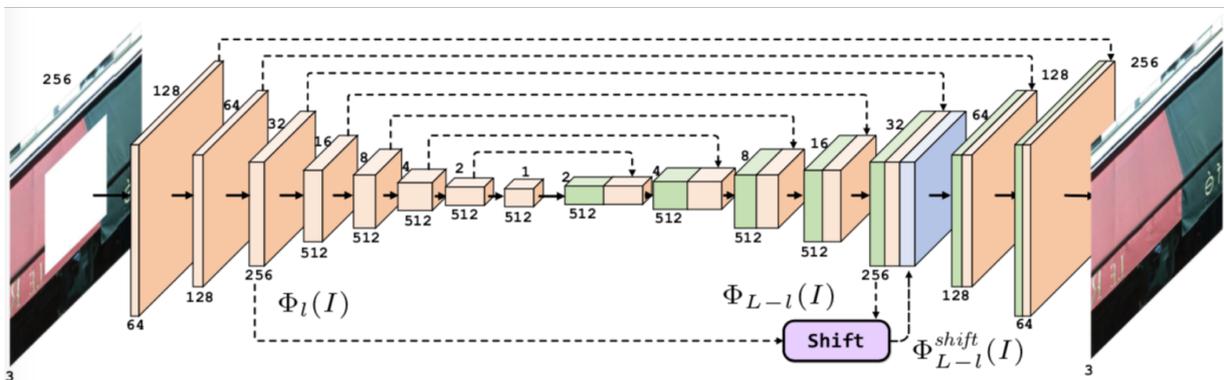


Figura 4.3: Arquitetura da ShiftNet, com a porção codificadora em laranja afunilando, as *skip connections* em pontilhado, as camadas convolucionais transpostas em verde e a aplicação do *shift* em azul. Os números acima dos blocos representam a quantidade de *pixels* e os números abaixo a quantidade de canais, que são 3 numa imagem RGB e são iguais ao número de filtros nas camadas convolucionais. Imagem obtida de [https://github.com/Zhaoyi-Yan/Shift-Net\\_pytorch](https://github.com/Zhaoyi-Yan/Shift-Net_pytorch).

A grande inovação dessa rede é a *shift-layer* (camada de deslocamento). Yan et al. (2018) foram inspirados pela técnica determinística classificada como método de completção baseado em exemplares. Como eles descrevem: “a reconstrução é conduzida do exterior para o interior da parte com a lacuna, procurando e copiando os melhores remendos correspondentes da região conhecida” (tradução nossa<sup>1</sup>). Em outras palavras, seria como copiar da parte que se conhece e colar na parte com ausência de informação. No entanto, ao invés de conduzir este “copia e cola” na imagem em si, isso é feito no espaço de características.

<sup>1</sup> Original: “the completion is conducted from the exterior to the interior of the missing part by searching and copying best matching patches from the known region.”

Eles adicionaram uma camada (*shift-layer*), construída da comparação entre os vetores de características convolucionais transpostos da camada do decodificador, no campo receptivo da parte ausente da imagem (a camada verde na Figura 4.4), e as características da *skip connection* da camada correspondente, mas no campo complementar receptivo (a camada de salmão/laranja na Figura 4.4). Em seguida, eles copiam o vetor de característica mais semelhante da *skip connection* na *shift-layer*, no lugar correspondente ao vetor de características da camada transposta (a camada azul claro na Figura 4.4).

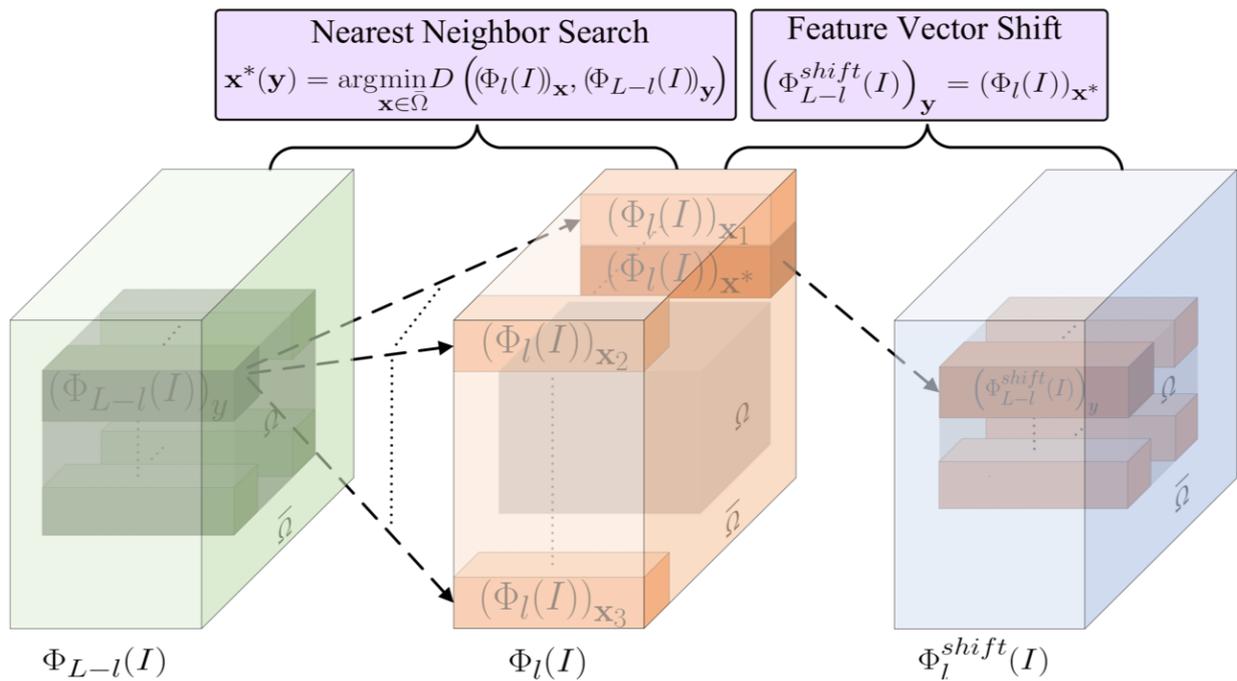


Figura 4.4: A Shift-layer (azul claro) é construída com os vetores de características da camada de skip connection (salmão/laranja) que são mais parecidos com os vetores de características da camada transposta (verde). Imagem obtida de [https://github.com/Zhaoyi-Yan/Shift-Net\\_pytorch](https://github.com/Zhaoyi-Yan/Shift-Net_pytorch).

A *shift-layer* (camada de deslocamento) recebe esse nome devido aos vetores de características da camada de *skip connection* serem deslocados para o lugar de dados faltantes. As camadas são concatenadas e alimentam a próxima camada convolucional. Yan et al. (2018) experimentaram a localização da *shift-layer* e descobriram que ela era mais eficaz (equilibrando tempo de computação e resultado final) na camada L-3, sendo L a última camada da rede geradora (ShiftNet).

### 4.3 Rede Discriminadora

A rede discriminadora  $D$ , cujo objetivo é apenas classificar as imagens que recebe como sendo verdadeiras ou falsas, possui uma arquitetura bem mais simples e é construída como uma rede neural convolucional sequencial, ou seja, é formada por sequências de camadas convolucionais. Uma representação da rede, quando a entrada tem  $256 \times 256$ , pode ser vista na Figura 4.5.

A rede possui cinco camadas convolucionais, todas com filtros de tamanho  $4 \times 4$ , *padding* de  $(1,1)$  e um *stride* de 2 nas três primeiras camadas e 1 nas duas últimas. As quatro primeiras camadas usam a função de ativação ReLU vazada (*leaky* ReLU). A normalização da instância também é aplicada após as camadas convolucionais 2, 3 e 4. A saída da rede  $D$  é obtida após uma função de ativação sigmoide e produz uma matriz de tamanho  $1 \times 30 \times 30$  em que cada elemento tem uma probabilidade da imagem ser real.

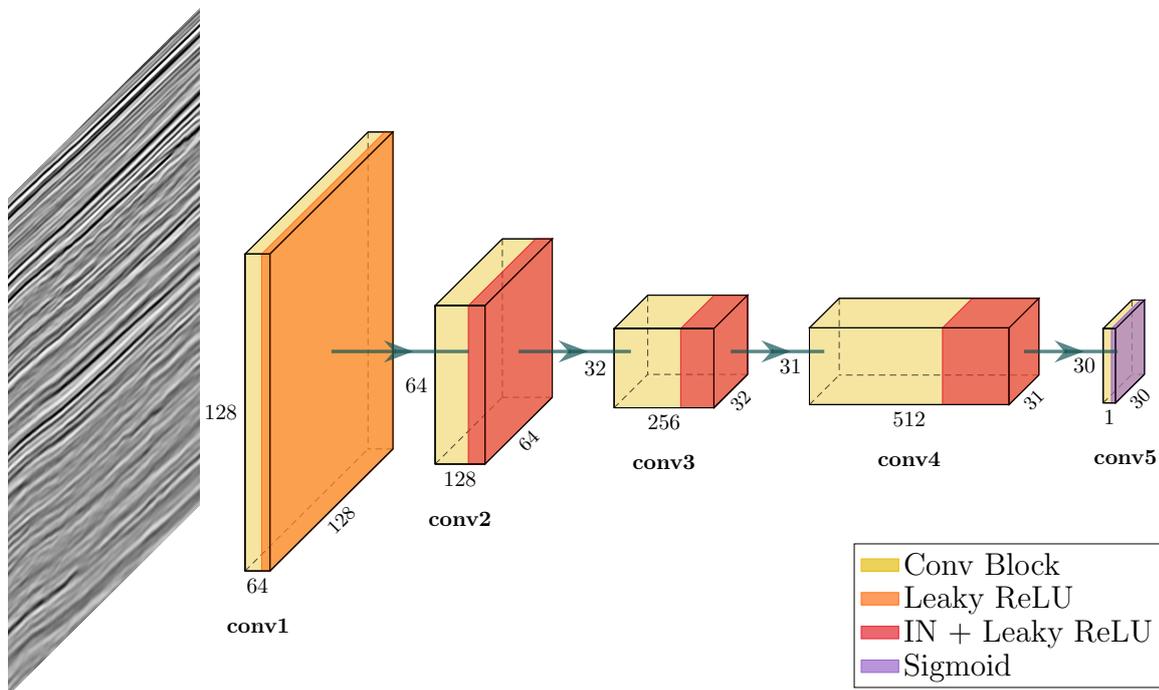


Figura 4.5: Cada bloco representa o tensor de saída da referida camada convolucional da rede Discriminadora  $D$ . O número de filtros fica abaixo, e na lateral está a altura do tensor e à frente a sua largura. As variadas cores ao final de cada bloco representam as operações realizadas na saída da convolução, que são a *Leaky ReLU* para o laranja, a *instance normalization + Leaky ReLU* para o vermelho e a ativação sigmoide para o roxo

#### 4.4 Funções Objetivo

Além da BCE, que é a função objetivo adversarial definida na seção 2.4, foi utilizada uma norma L1 para a rede  $G$  (equação 4.1) em toda imagem e uma função descontada. A norma L1 mede a diferença absoluta entre os *pixels* da imagem falsa gerada,  $F_n$ , e da verdadeira,  $R_n$ , e também é utilizada para acompanhamento da evolução do treinamento, sendo definida por:

$$L_1n(R_n, F_n) = |R_n - F_n|. \quad (4.1)$$

A função descontada é a L1, mas apenas na região com a lacuna e com peso diferente para cada pixel, buscando dar uma relevância maior às amostras mais próximas às bordas da lacuna e menor às amostras mais distantes da borda. O peso  $w$  de cada pixel, em que  $\gamma$  é um fator fixo (foi usado 0,9 neste trabalho seguindo a implementação de Yan et al. (2018)) e  $l$  é a distância do pixel avaliado ao pixel com informação mais próximo, é dado pela equação:

$$w = \gamma^l \quad (4.2)$$

#### 4.5 Treinamento Adversarial

Para realizar o treinamento adversarial, utilizamos uma amostra de imagem 256x256 e a normalizamos com os valores médios e máximos armazenados de todo o conjunto de treinamento. Além disso, fizemos uma reflexão horizontal aleatória, com probabilidade de 50%, para aumentar a variabilidade dos dados (*data augmentation*). Em seguida, adicionamos uma máscara para simular uma lacuna na imagem. A imagem mascarada foi alimentada no modelo generativo  $G$  que produziu uma imagem falsa preenchida. O discriminador  $D$  foi alimentado com a imagem falsa e a original, calculando probabilidades para cada uma. Em seguida,  $D$  foi atualizada com a *adversarial loss* para tentar melhorar a distinção entre imagem real e falsa. Posteriormente,  $G$  foi atualizada com a *adversarial loss*, a norma L1 e a *discounted loss*, e o ciclo então foi repetido.

## 4.6 Métricas de Desempenho

Para acompanhamento da evolução do treinamento, calculamos uma norma  $L1$  média (entre a imagem falsa e a original) para cada época para os conjuntos de treinamento e validação

Além da evolução da função objetivo, é conveniente utilizar outras métricas para avaliar o progresso e efetividade do modelo. Neste trabalho, foram utilizadas quatro métricas adicionais, sendo elas o coeficiente de correlação  $R$  de Pearson (Pearson, 1895), o *Normalized Root Mean Squared* (NRMS) (Kragh e Christie, 2002) e outras duas métricas bastante usadas em comparação de qualidade de imagens, o *Structural Similarity Index* (SSIM) (Wang et al., 2004) e o *Peak Signal to Noise Ratio* (PSNR) (Huynh-Thu e Ghanbari, 2008).

O coeficiente de correlação  $R$  de Pearson está onipresente na ciência e foi utilizado em Oliveira et al. (2018) para avaliar o resultado das interpolações de seu modelo convolucional. Com essa escolha, temos uma métrica robusta, universalmente conhecida e também uma base de comparação com trabalhos anteriores.

Além disso, também foi calculado o NRMS, que é uma métrica amplamente utilizada para a comparação entre levantamentos sísmicos 4D, em que é importantíssima a repetibilidade entre os mesmos, pois o que se deseja averiguar no dado é o efeito da variação na geologia causada pela produção de petróleo (i.e., alteração de saturação de fluidos com consequente aumento da impedância) e não as alterações na geometria de aquisição. Como as mudanças em levantamentos 4D são muito sutis, o NRMS é uma métrica bastante sensível, apesar de não ser muito intuitiva. O cálculo do NRMS pode ser feito a partir da equação:

$$NRMS(R_t, F_t) = \frac{200 \cdot RMS(R_t - F_t)}{RMS(R_t) + RMS(F_t)}, \quad (4.3)$$

em que  $t$  representa uma coluna da imagem (um traço sísmico),  $F$  a imagem gerada e  $R$  a imagem original. O fator 200 define a escala da métrica, que varia de 0 a 200% de modo que se as imagens são idênticas o valor do NRMS é 0 e se são completamente opostas o valor é 200%. RMS é a raiz quadrática média calculada no intervalo  $t_1 - t_2$ , dada por:

$$RMS(R_t) = \sqrt{\frac{\sum_{t_1}^{t_2} (R_t)^2}{N}}, \quad (4.4)$$

sendo que  $N$  representa o número de amostras no intervalo  $t_1 - t_2$ .

Tanto para o coeficiente  $R$  quanto para o NRMS o cálculo foi feito traço a traço. Os valores para cada imagem, portanto, são a média de cada métrica nos traços onde foi calculada. Para máscaras centralizadas de largura constante, as métricas são calculadas apenas na região da máscara de modo que as partes que são idênticas por construção não entram no cálculo, mas apenas aquelas em que a rede neural realmente atuou. Já para o acompanhamento do treinamento, é calculada a média sobre todas as imagens de cada época.

O SSIM é calculado a partir da comparação de três características principais da imagem, que são o brilho, o contraste e a estrutura. O brilho é a média de todos os *pixels*, o contraste é desvio padrão da média, e a estrutura é a imagem subtraída da média e em seguida dividida pelo desvio padrão, ficando com média zero e desvio padrão unitário. Uma função de comparação é definida para cada uma dessas características, e a função composta é o SSIM, como mostrado na equação abaixo:

$$SSIM(R, F) = \frac{(2\mu_R\mu_F + C_1)(2\sigma_{RF} + C_2)}{(\mu_R^2 + \mu_F^2 + C_1)(\sigma_R^2 + \sigma_F^2 + C_2)}, \quad (4.5)$$

sendo  $\mu_r$  a média da imagem real,  $\mu_s$  a média da imagem falsa,  $\sigma_{RF}$  é a covariância entre as imagens real e falsa,  $\sigma_R^2$  é a variância da imagem real,  $\sigma_F^2$  é a variância da imagem falsa,  $C_1$  e  $C_2$  são constantes para estabilização da divisão.

Por fim, o PSNR é uma medida da razão entre a máxima potência de um sinal pelo ruído da mesma. A comparação entre duas imagens é feita definindo previamente a melhor, a qual, em nosso caso, é a imagem sísmica original, extraída a sua máxima potência, que é o quadrado da intensidade do pixel mais intenso, enquanto que o ruído é definido como o erro quadrático médio entre os *pixels* da imagem original e a gerada pela rede neural. O PSNR é, comumente, definido em decibéis e pode ser calculado a partir da seguinte equação:

$$PSNR(R, F) = 10 \log_{10} \left( \frac{MAX_R}{MSE(R, F)} \right) \quad (4.6)$$

em que  $MAX_R$  é a máxima potência da imagem original e  $MSE$  é o *mean squared error* (erro quadrático médio) entre a imagem original e a falsa.

## 4.7 Método Comparativo

A fim de melhor avaliar o desempenho da rede neural, comparamos o resultado de sua aplicação ao resultado da aplicação de um método mais tradicional. Em seu trabalho, Wei et al. (2021) fazem uma breve discussão sobre diferentes métodos de interpolação para dados ausentes, tanto de forma irregular quanto regular. De todos eles, o método que escolhem para fazer a comparação é um método baseado nas PEFs (*Prediction Error Filters*)

Esse método também é o que utilizamos e pertence ao contexto de *machine learning*, pois PEF se trata de auto-regressão (Claerbout e Fomel (1994)). Além de ser um método robusto, possui implementação pronta no MADAGASCAR (um pacote de software aberto para análise de dados multidimensionais e experimentos computacionais reproduzíveis)<sup>2</sup>, de forma que bastou adaptá-lo ao nosso problema específico. Usamos o método descrito no capítulo 7 do livro *Geophysical Image Estimation by Example* (Claerbout e Fomel, 1994), adaptado da implementação feita em `book/gee/mda/hole`, dentro do pacote MADAGASCAR.

O método é baseado em dois estágios, no qual primeiro encontra-se a função de predição de erro (PEF) ótima e depois usa-se essa PEF para fazer a interpolação dos dados. Essa separação é importante pois, no primeiro estágio, o dado está fixo (a parte com dados faltantes é ignorada) e o que se busca otimizar são os coeficientes da PEF. No segundo estágio, a PEF está fixa (que é a encontrada no primeiro estágio) e o que buscamos otimizar é o dado na região dos dados faltantes.

## 4.8 Implementação

O código utilizado no processamento foi escrito em Python 3, usando a biblioteca para redes neurais PyTorch (Paszke et al., 2019) e foi modificado do repositório no Github da Shift-Net<sup>3</sup>. Por sua vez, as modificações realizadas também foram armazenadas no Github, no repositório: [https://github.com/mauriliosalg/Seis\\_Shift-Net\\_pytorch](https://github.com/mauriliosalg/Seis_Shift-Net_pytorch).

Os modelos foram treinados em computadores do Computer Vision Research Group

---

<sup>2</sup> Original: “Madagascar is an open-source software package for multidimensional data analysis and reproducible computational experiments.” - [https://www.reproducibility.org/wiki/Main\\_Page](https://www.reproducibility.org/wiki/Main_Page)

<sup>3</sup> Repositório: [https://github.com/Zhaoyi-Yan/Shift-Net\\_pytorch](https://github.com/Zhaoyi-Yan/Shift-Net_pytorch)

---

no Instituto de Matemática e Estatística da USP, equipados com placas de vídeo GeForce GTX 1080 ti com 6.1 cc (*computing capability*). Nesses equipamentos, com um conjunto de treinamento de 12000 imagens de 256x256 e com um conjunto de validação de 3000 imagens, o tempo de treinamento para 350 épocas com batch size de 1 ficou em torno de 96 horas.



## Resultados

### 5.1 Experimentos

Nosso objetivo neste trabalho é treinar a rede neural para completar o dado real na região com lacuna de informação. Uma discussão do problema foi apresentada na Seção 1.2 e um exemplo de linha com as lacunas se encontra na Figura 1.3, mas, para conforto do leitor, reproduzimo-la abaixo.

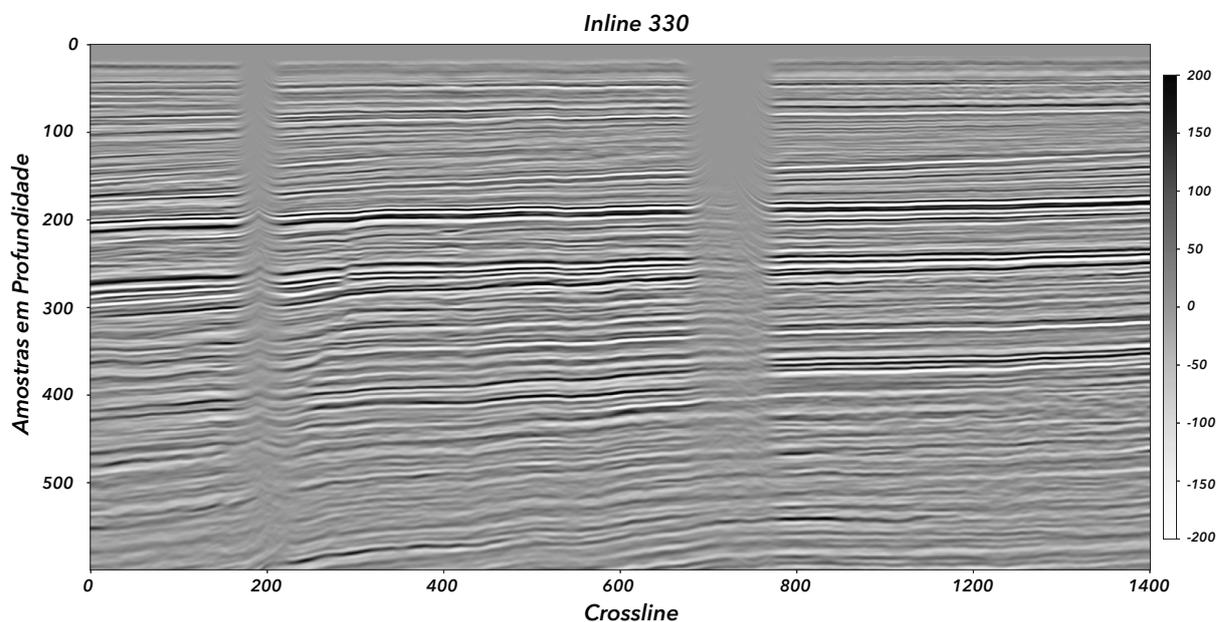


Figura 5.1: Seção sísmica mostrando lacunas na imagem, mais pronunciados na parte rasa. Na borda da região sem informação também se nota o efeito do operador de migração, “espalhando” o dado para cima. O eixo vertical é o número de amostras em profundidade e o eixo horizontal representa os traços (dimensão das *xlines*). Esta figura é a reprodução da Figura 1.3.

Para esse fim, fizemos recortes de tamanho 256x256 na parte íntegra do dado, conforme descrito na Seção 4.1. Separamos os dados recortados em três conjuntos distintos: 12000 imagens de treinamento, 3000 de validação e 1380 para teste. O propósito de cada conjunto está descrito na Seção 2.2.

As lacunas são geradas artificialmente, de modo que a imagem original serve de referência para o aprendizado. Usamos máscaras retangulares, já que as lacunas no dado real são aproximadamente retangulares e como o problema começa com uma largura pequena e chega a um máximo de 120 *pixels* de ausência de dados, foram treinados três modelos, cada um para uma largura de máscara diferente (Figuras 5.2 a 5.4): um com 68 *pixels* (Modelo 68), outro com 96 *pixels* (Modelo 96) e o último com 124 *pixels* (Modelo 124).

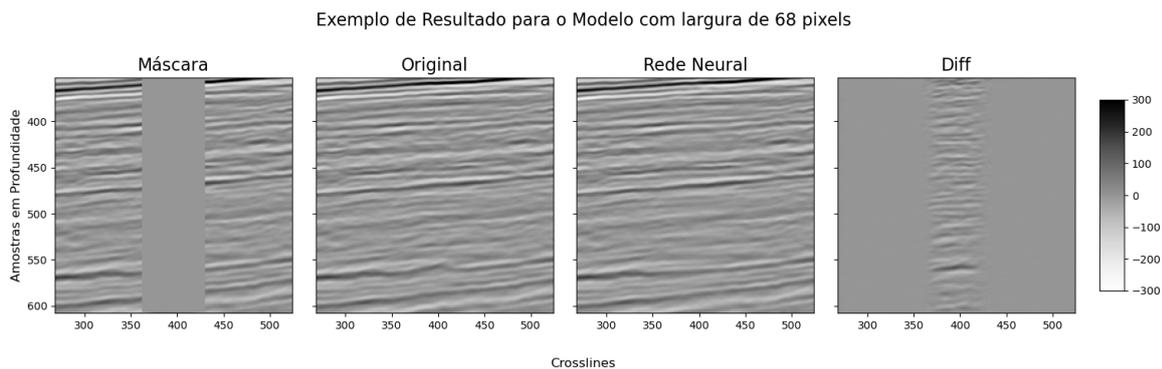


Figura 5.2: Exemplo de resultado para o modelo treinado com máscara retangular com 68 *pixels* de largura. À esquerda está a imagem real mascarada, no centro-esquerdo a imagem real e não corrompida, na centro-direita a falsificação gerada e, à direita, a diferença entre a imagem real e a interpolada.

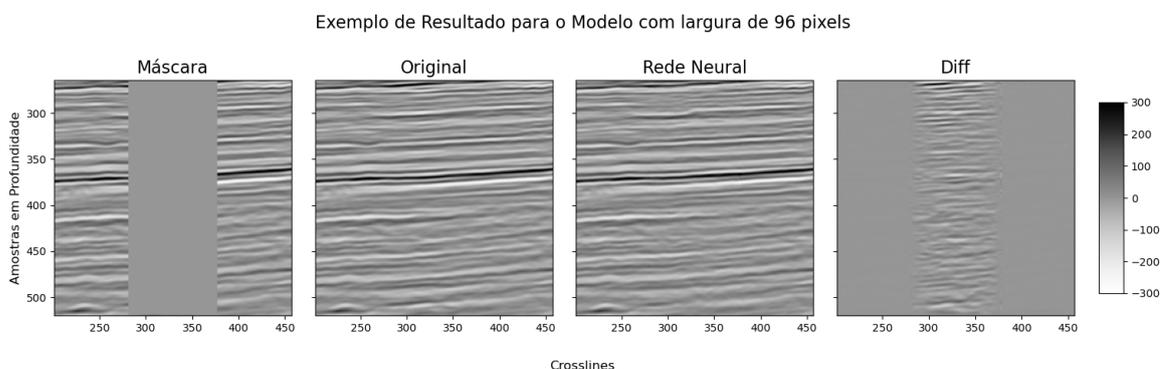


Figura 5.3: Exemplo de resultado para o modelo treinado com máscara retangular com 96 *pixels* de largura. À esquerda está a imagem real mascarada, no centro-esquerdo a imagem real e não corrompida, na centro-direita a falsificação gerada e, à direita, a diferença entre a imagem real e a interpolada.

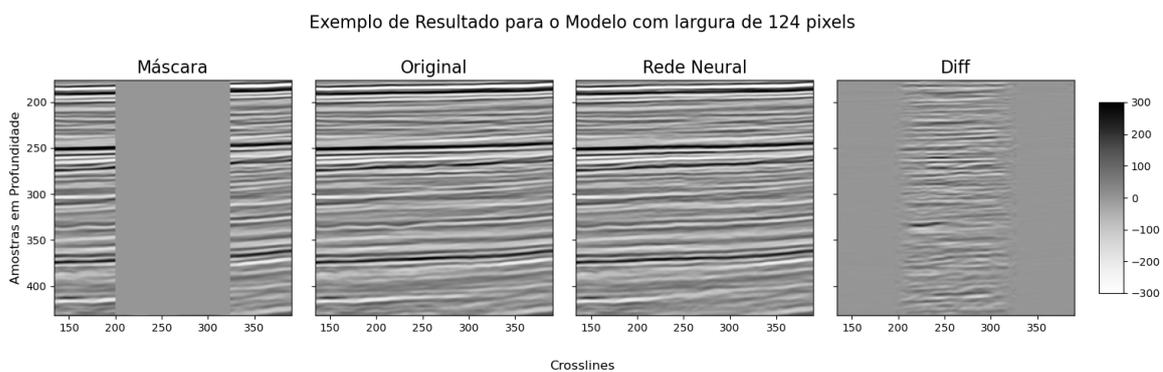


Figura 5.4: Exemplo de resultado para o modelo treinado com máscara retangular com 124 *pixels* de largura. À esquerda está a imagem real mascarada, no centro-esquerdo a imagem real e não corrompida, na centro-direita a falsificação gerada e, à direita, a diferença entre a imagem real e a interpolada.

O tamanho do lote (*batch size*) foi de uma única imagem (SGD – gradiente descendente estocástico), com o treinamento feito por 350 épocas. A evolução dos treinamentos é acompanhada com a média da norma  $L1$ , em cada época, e com a média das métricas de correlação, NRMS, SSIM e PSNR. As figuras 5.5, 5.6, 5.7, 5.8, 5.9, 5.10, 5.11, 5.12, 5.13, 5.14, 5.15, 5.16, 5.17, 5.18 e 5.19 mostram a evolução do treinamento para cada um dos modelos, com cinco imagens para cada uma das três larguras treinadas, iniciando pela largura de 68 *pixels* e terminando com a de 124 *pixels*.

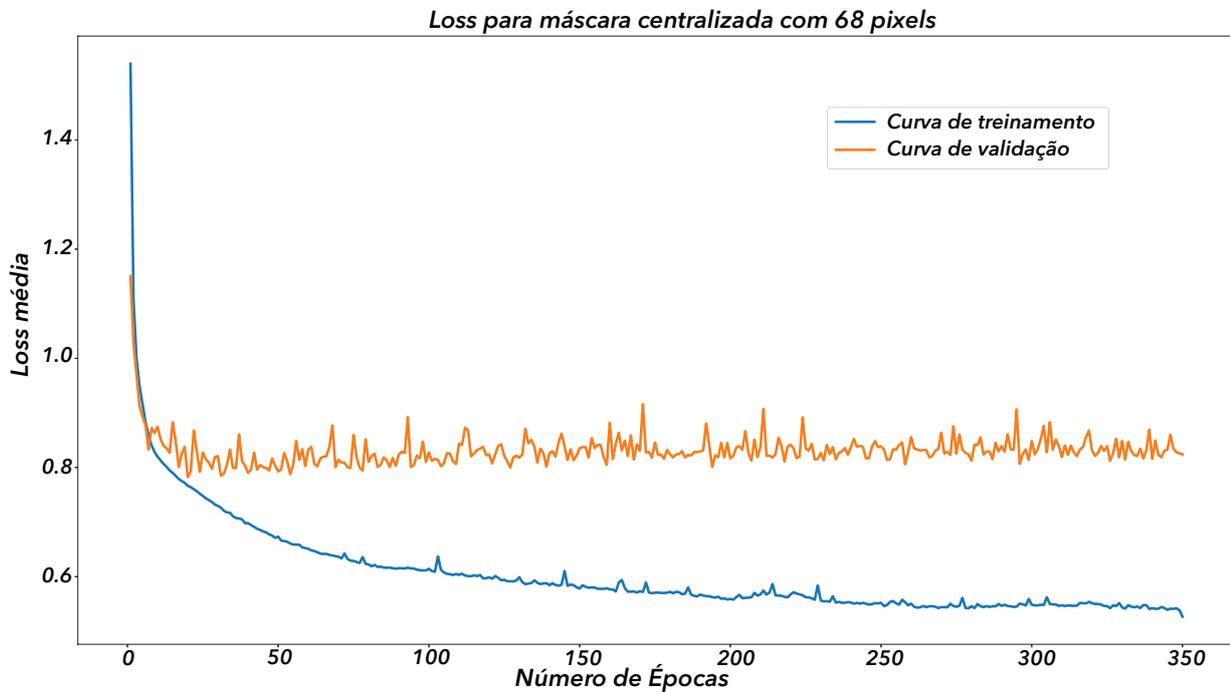


Figura 5.5: Loss do modelo gerador  $G$  treinado com com máscara retangular centralizada de 68 *pixels* de largura e 256 de altura.

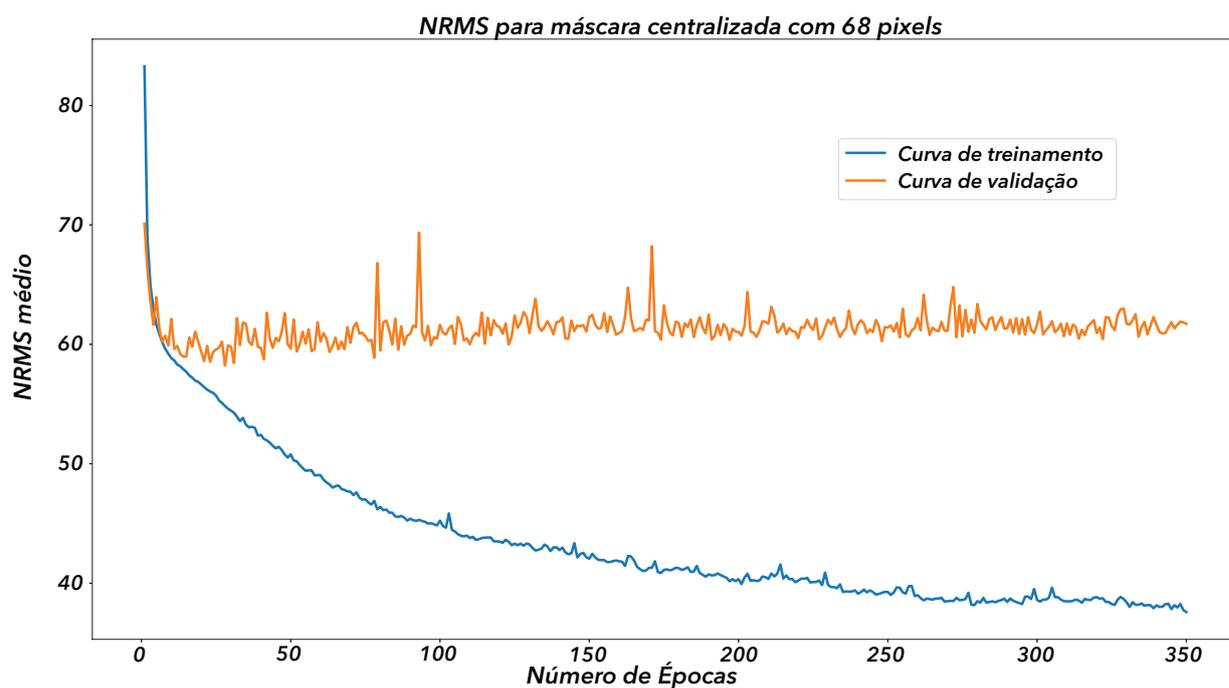


Figura 5.6: Loss do modelo gerador  $G$  treinado com com máscara retangular centralizada de 68 *pixels* de largura e 256 de altura.

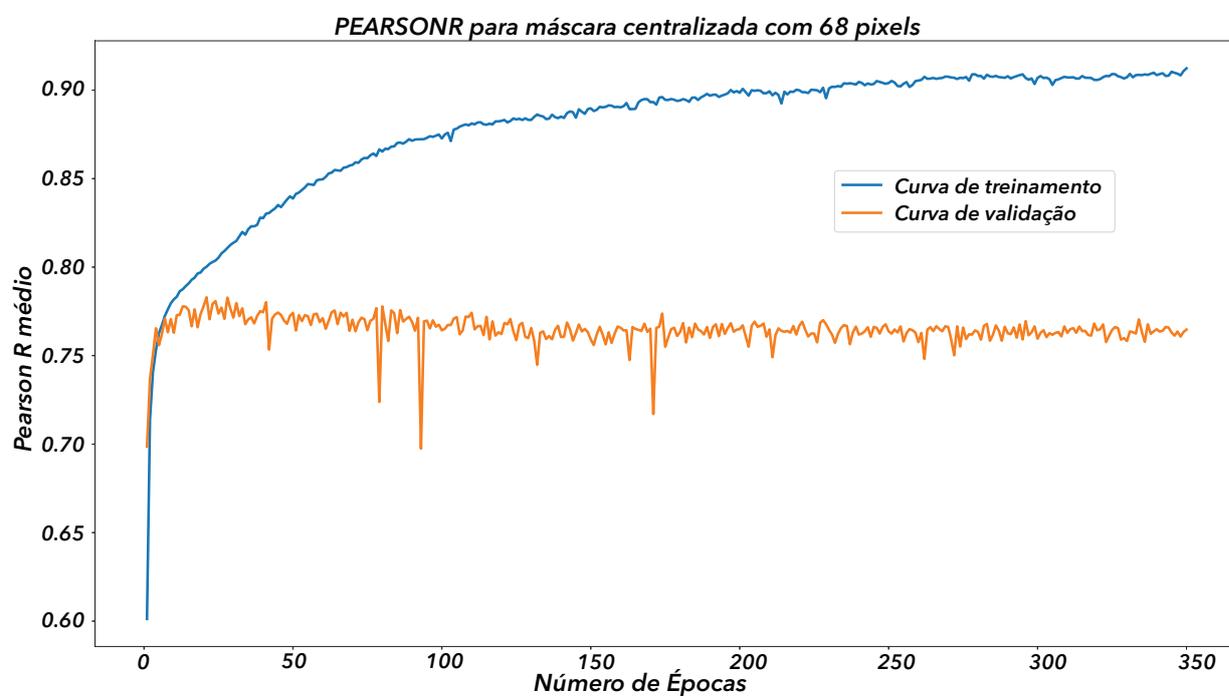


Figura 5.7: Pearson  $R$  do modelo gerador  $G$  treinado com com máscara retangular centralizada de 68 *pixels* de largura e 256 de altura.

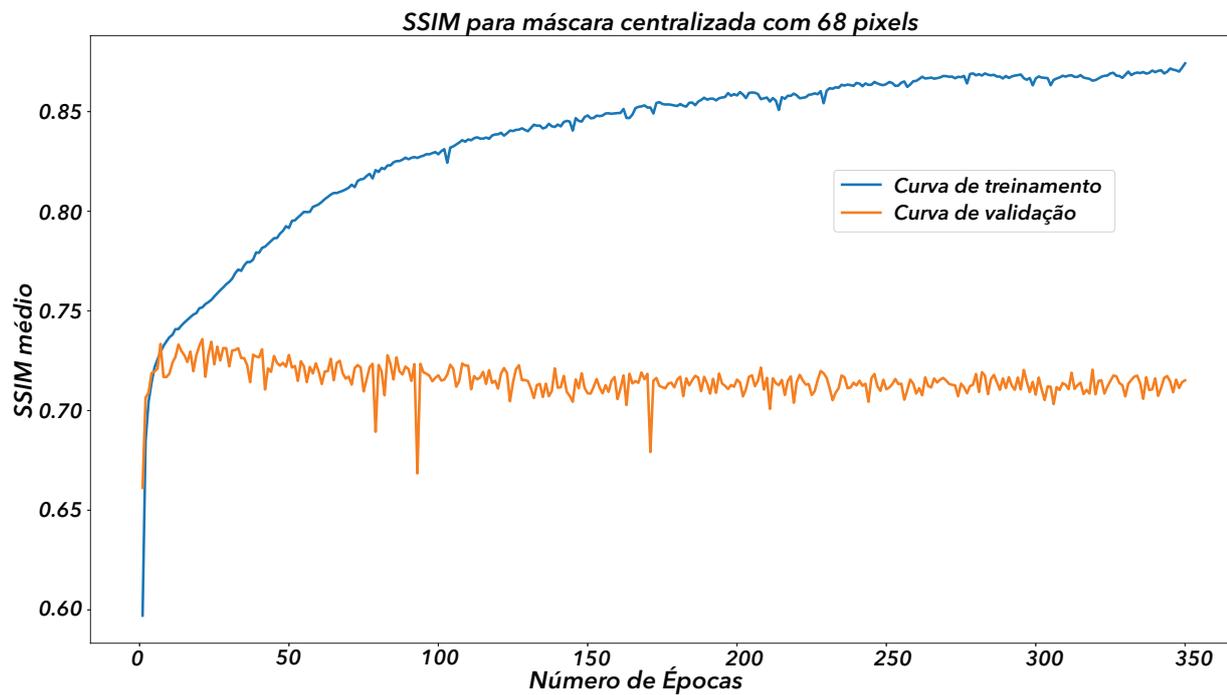


Figura 5.8: SSIM do modelo gerador  $G$  treinado com com máscara retangular centralizada de 68 *pixels* de largura e 256 de altura.

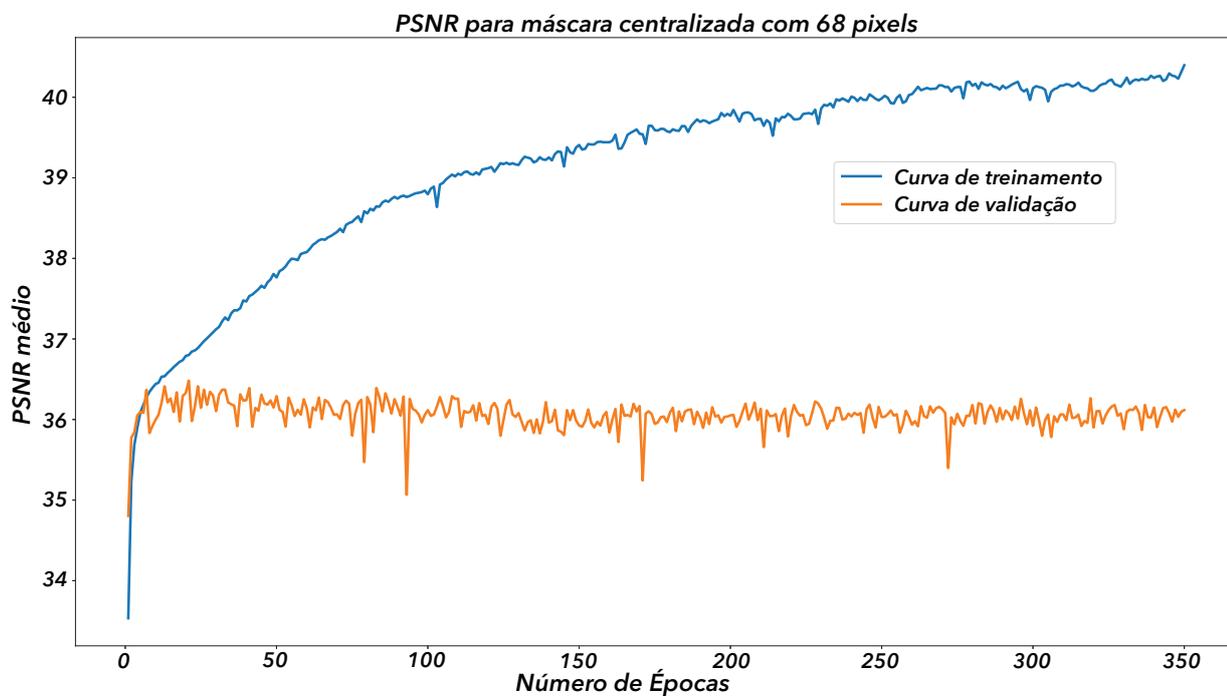


Figura 5.9: PSNR do modelo gerador  $G$  treinado com com máscara retangular centralizada de 68 *pixels* de largura e 256 de altura.

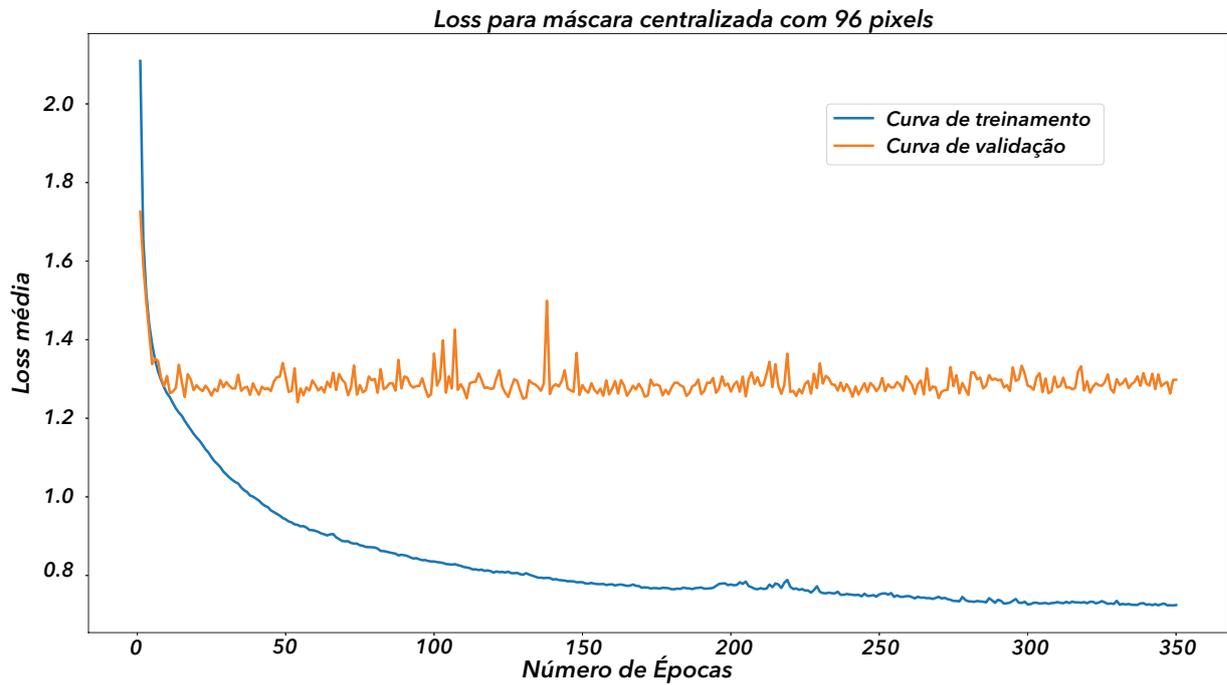


Figura 5.10: Loss do modelo gerador  $G$  treinado com com máscara retangular centralizada de 96 *pixels* de largura e 256 de altura.

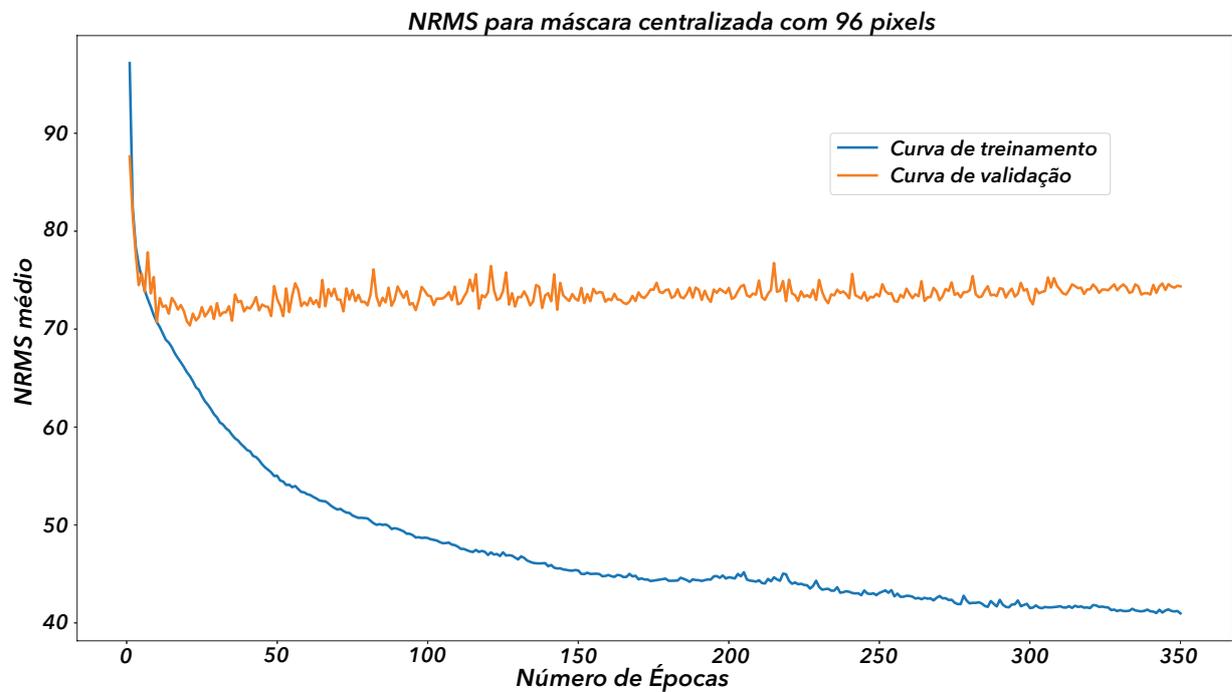


Figura 5.11: Loss do modelo gerador  $G$  treinado com com máscara retangular centralizada de 96 *pixels* de largura e 256 de altura.

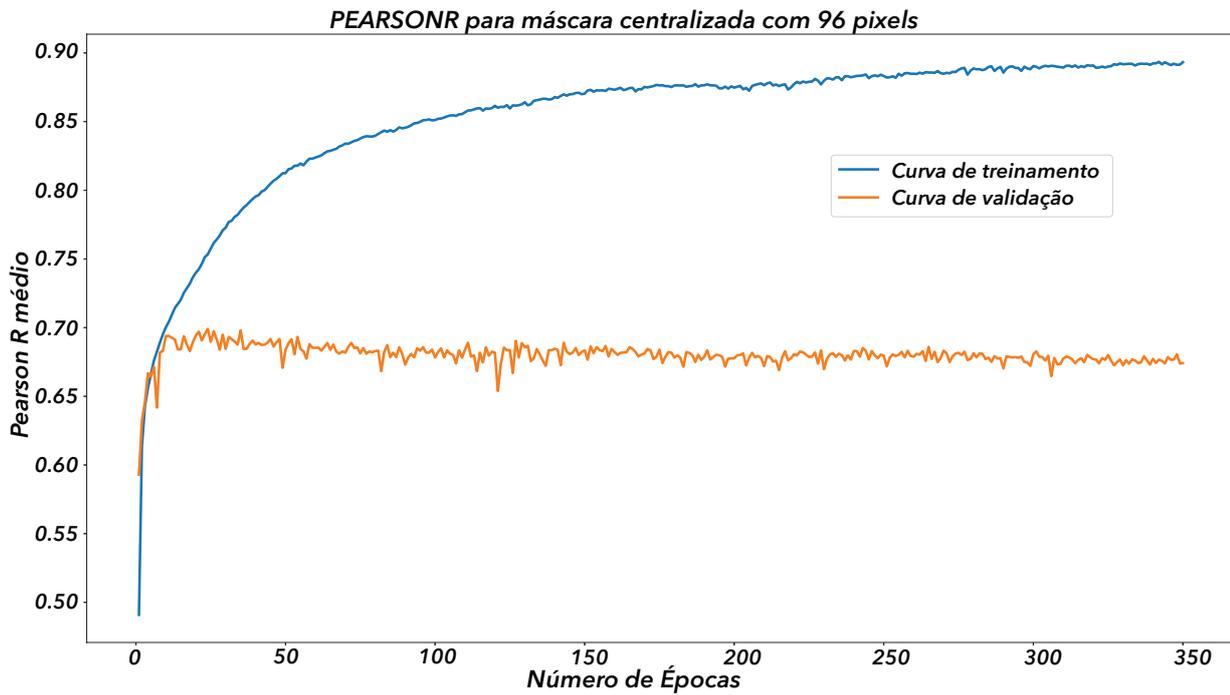


Figura 5.12: Pearson  $R$  do modelo gerador  $G$  treinado com com máscara retangular centralizada de 96 *pixels* de largura e 256 de altura.

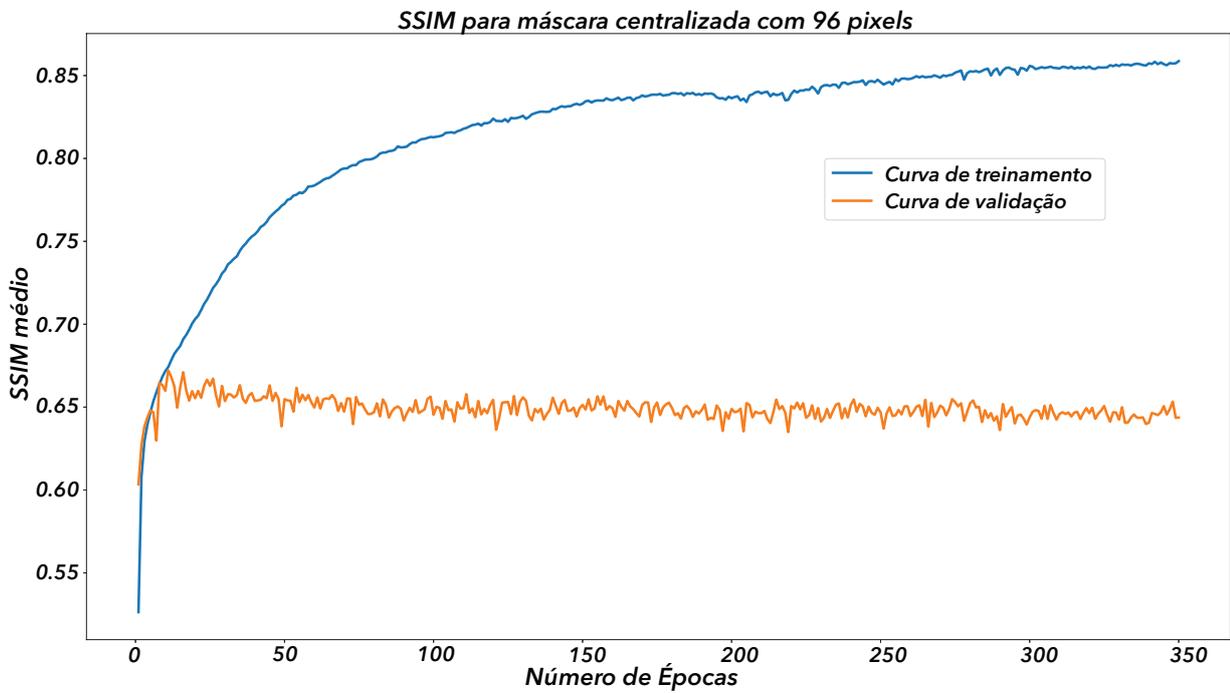


Figura 5.13: SSIM do modelo gerador  $G$  treinado com com máscara retangular centralizada de 96 *pixels* de largura e 256 de altura.

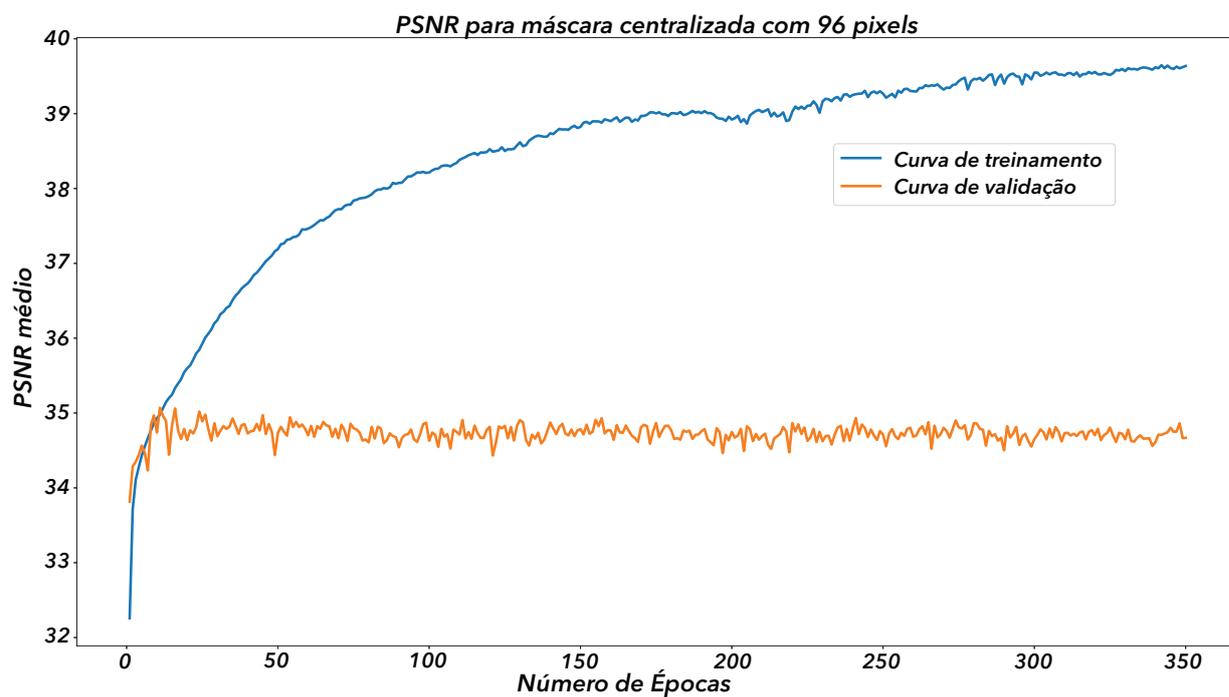


Figura 5.14: PSNR do modelo gerador  $G$  treinado com com máscara retangular centralizada de 96 *pixels* de largura e 256 de altura.

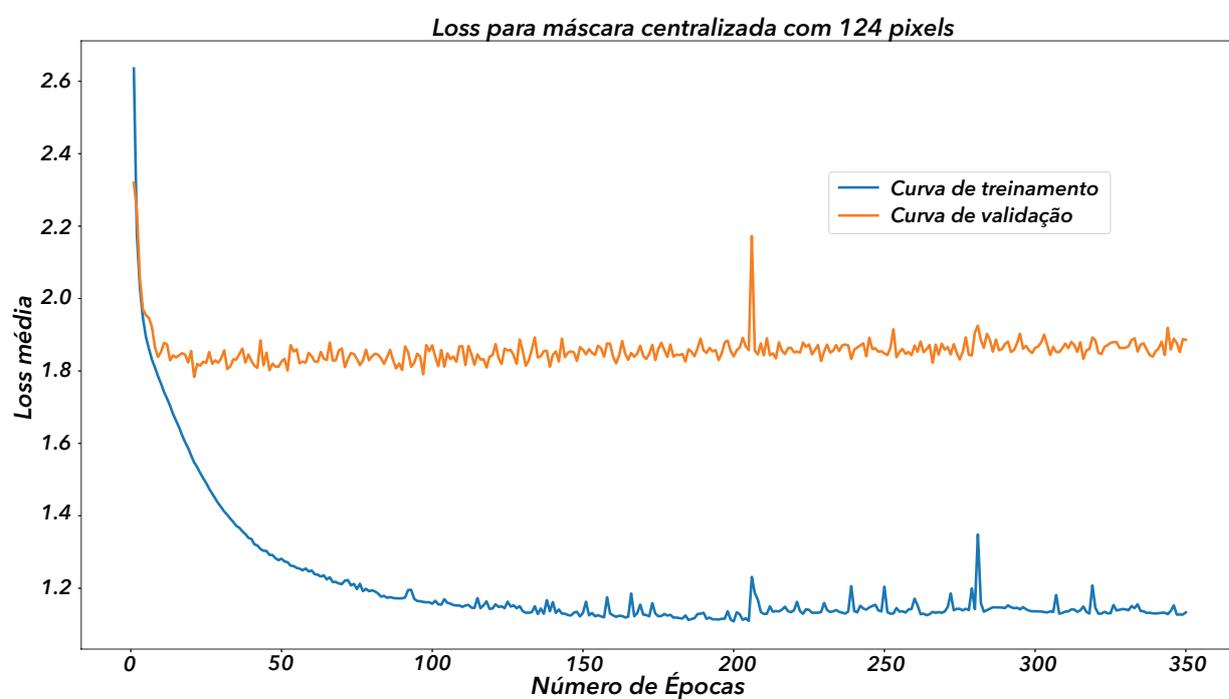


Figura 5.15: Loss do modelo gerador  $G$  treinado com com máscara retangular centralizada de 124 *pixels* de largura e 256 de altura.

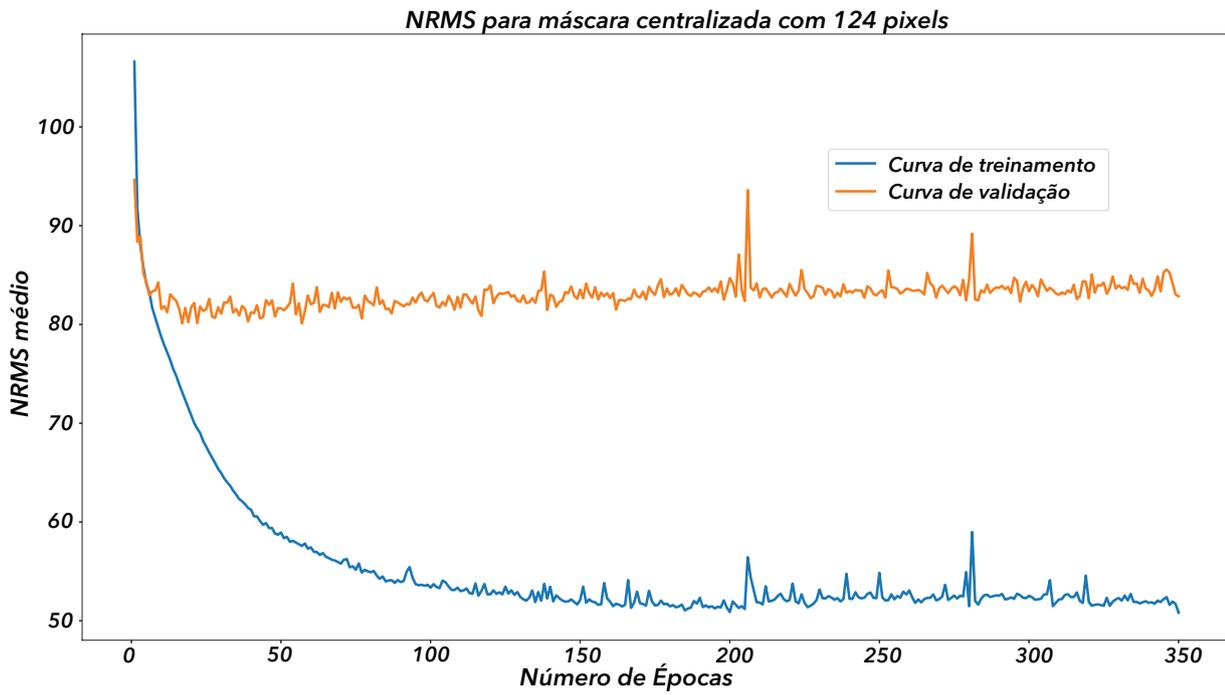


Figura 5.16: Loss do modelo gerador  $G$  treinado com com máscara retangular centralizada de 124 *pixels* de largura e 256 de altura.

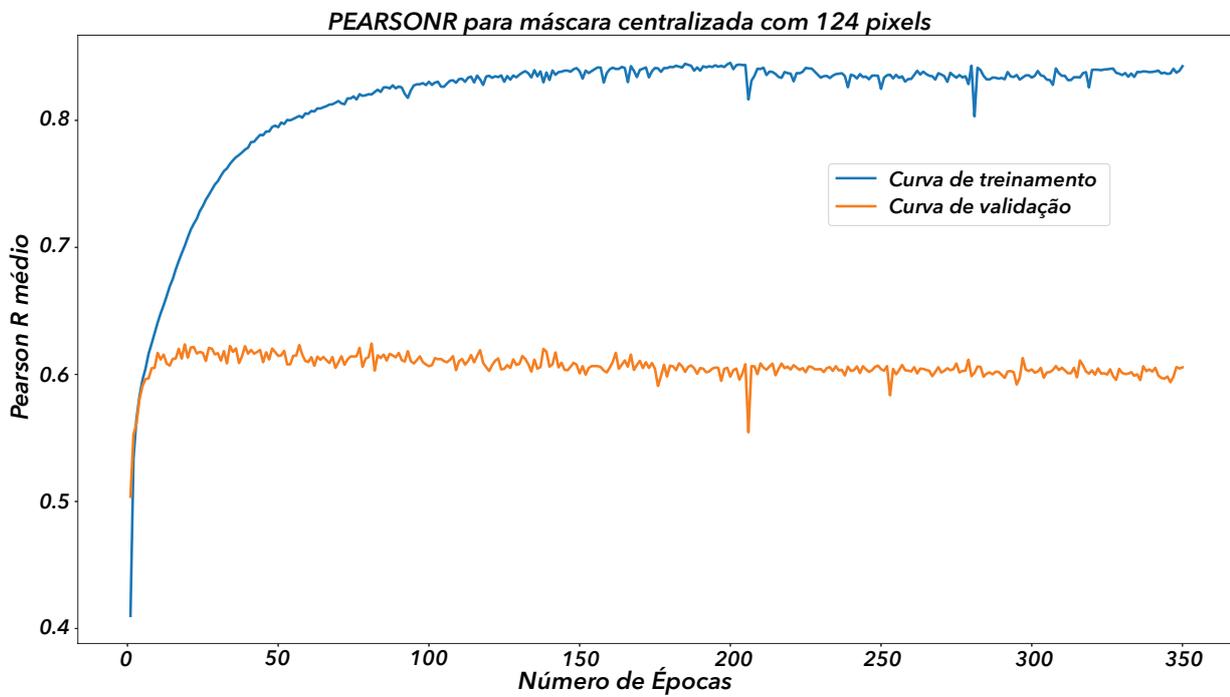


Figura 5.17: Pearson  $R$  do modelo gerador  $G$  treinado com com máscara retangular centralizada de 124 *pixels* de largura e 256 de altura.

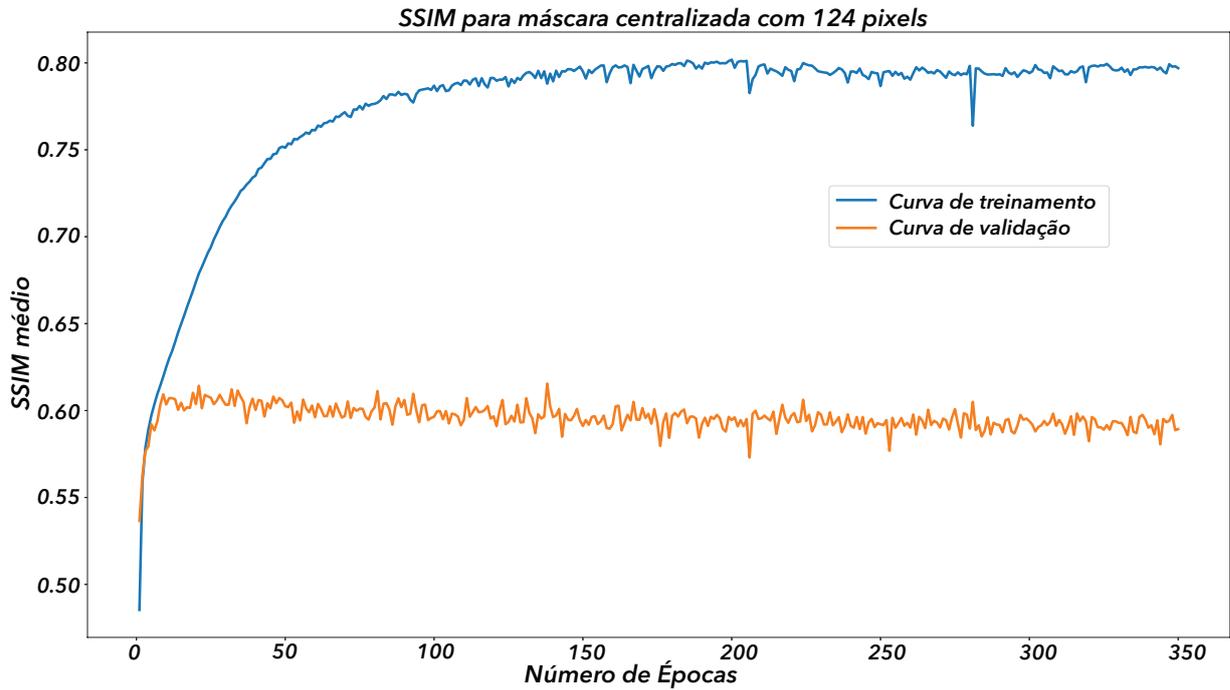


Figura 5.18: SSIM do modelo gerador  $G$  treinado com com máscara retangular centralizada de 124 *pixels* de largura e 256 de altura.

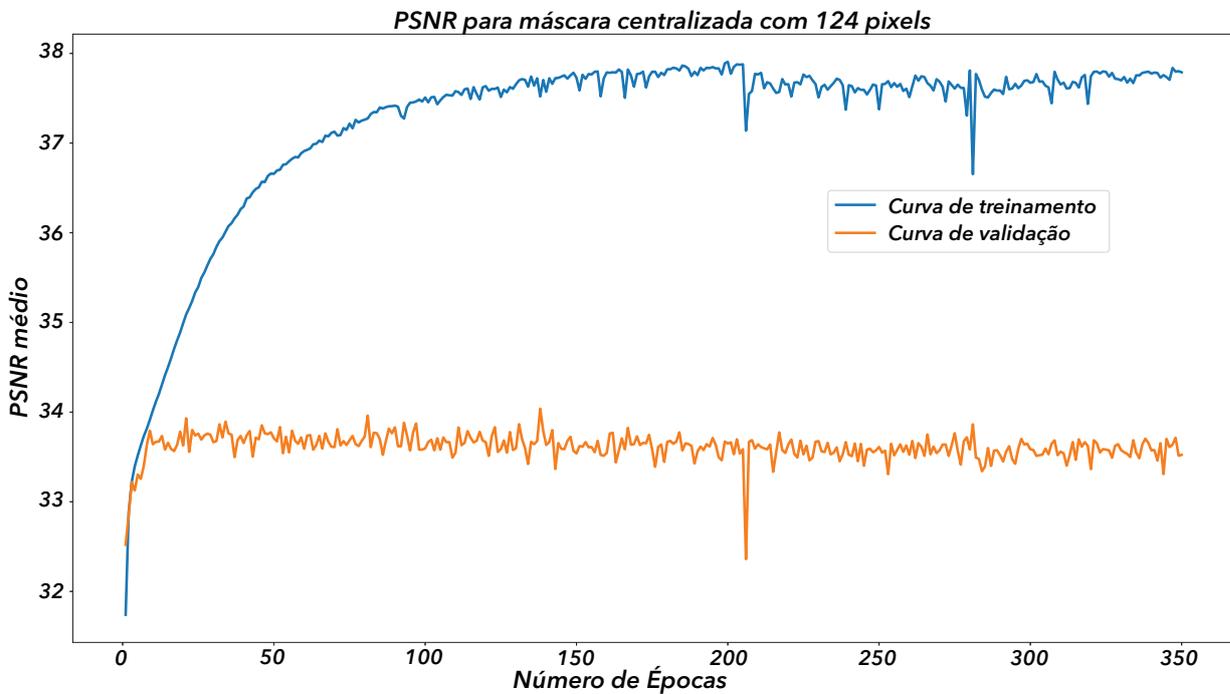


Figura 5.19: PSNR do modelo gerador  $G$  treinado com com máscara retangular centralizada de 124 *pixels* de largura e 256 de altura.

Das Figuras 5.5 a 5.19, notamos que para os conjuntos de treinamento (curvas azuis), as curvas estão sempre melhorando nas suas respectivas métricas, enquanto que para o conjunto de validação, um platô é atingido rapidamente. A literatura reporta que quando a validação começa a piorar, devemos parar o treinamento, pois o modelo está perdendo capacidade de generalização, fato conhecido como *overfitting*.

Em nossa avaliação, houve uma estabilização em torno de um certo valor, com uma leve tendência de piora, mas mantivemos o treinamento, pois se fosse o caso de *overfitting*, o conjunto de validação teria suas métricas muito prejudicadas. Ainda assim, embora o modelo da última época não tenha um *overfitting* clássico caracterizado, ele possui um erro de generalização maior do que épocas anteriores. Como critério de parada do treinamento, escolhemos o início aproximado do platô da validação, uma vez que o SGD acaba gerando uma curva ruidosa. Embora tal escolha seja arbitrária, o início aproximado do platô da validação foi observado próximo à vigésima época de treinamento.

Neste estudo, o treinamento não foi interrompido na vigésima época, mas realizado até a tricentésima quinquagésima época, como plotado nos gráficos das figuras de 5.5 a 5.19. No entanto, como modelo final para as análises que serão apresentadas nas seções e capítulos seguintes, utilizamos os obtidos até a vigésima época.

Outro ponto bastante notável é que para um mesmo parâmetro, correlação de Pearson por exemplo, o resultado é pior quanto maior é o GAP de informação. Analisando a última época, para a validação, a correlação do modelo de 68 pixels ficou por volta de 76% (Figura 5.7), a do modelo 96 pixels em torno de 68% (Figura 5.12) e a do modelo de 124 pixels em aproximadamente 59% (Figura 5.17). Tal resultado era esperado, uma vez que com lacunas maiores existe maior variedade de soluções possíveis e, quanto mais longe das bordas, menor é a confiabilidade da informação gerada. Tal deterioração na qualidade da interpolação para lacunas maiores também foi observada por Oliveira et al. (2018).

## 5.2 Fase de Testes

Na fase de testes, aplicamos os modelos treinados a imagens que a rede não teve contato durante o treinamento. Cada conjunto de testes possuía 1380 imagens de 256x256 extraídas aleatoriamente do volume de testes de forma que a média de cada métrica foi avaliada para cada um dos três modelos. Os resultados obtidos podem ser vistos na Tabela 5.1, para os

modelos da época 20 e, para comparação, na Tabela 5.2, para os modelos da época 350.

Tabela 5.1 - Resultados do teste dos modelos para a época 20

Modelo	NRMS	Pearson R	SSIM	PSNR
68	53	0,82	0,77	37,3
96	63	0,75	0,70	35,8
124	73	0,68	0,65	34,8

Tabela 5.2 - Resultados do teste dos modelos para a última época

Modelo	NRMS	Pearson R	SSIM	PSNR
68	56	0,81	0,76	37,4
96	68	0,72	0,68	35,7
124	78	0,65	0,63	34,6

Aqui, novamente, fica nítida a queda de performance dos modelos em relação ao aumento do tamanho da lacuna a ser interpolada. Também, é possível ver que as métricas de NRMS e Pearson  $R$  são mais sensíveis do que as métricas SSIM e PSNR. Como era de se esperar, a parada antecipada na época 20 possui melhores métricas do que a parada na época 350, com os ganhos sendo mais relevantes nos modelos com lacunas maiores. Em termos de tempo de treinamento, uma parada na época 20 significa uma redução de 17,5 vezes no tempo necessário para o treinamento do modelo, caindo de 96 horas (350 épocas) para 5 horas e 30 minutos (20 épocas).

### 5.3 Estudo de Caso

Nesta seção, apresentamos um exemplo de aplicação a um dado real, isto é, onde existem lacunas devido a obstrução do sinal. Assim, aplicamos os modelos treinados com a largura de máscara mais adequada a cada linha apresentada.

Em cada uma dessas linhas, extraímos imagens de 256x256 centralizadas na lacuna a cada 256 amostras em profundidade, portanto, sem interseção na vertical entre as imagens. Como as lacunas não atingem uma profundidade grande, uma sequência de três imagens por buraco, ou seja, um total de 768 *pixels* em profundidade, por linha, foi suficiente para

o tratamento do problema. Em outras palavras, utilizamos três recortes de 256x256, sem intersecção, sequencialmente em profundidade para cada lacuna, em cada linha do dado. O procedimento adotado é ilustrado a seguir na Figura 5.20.

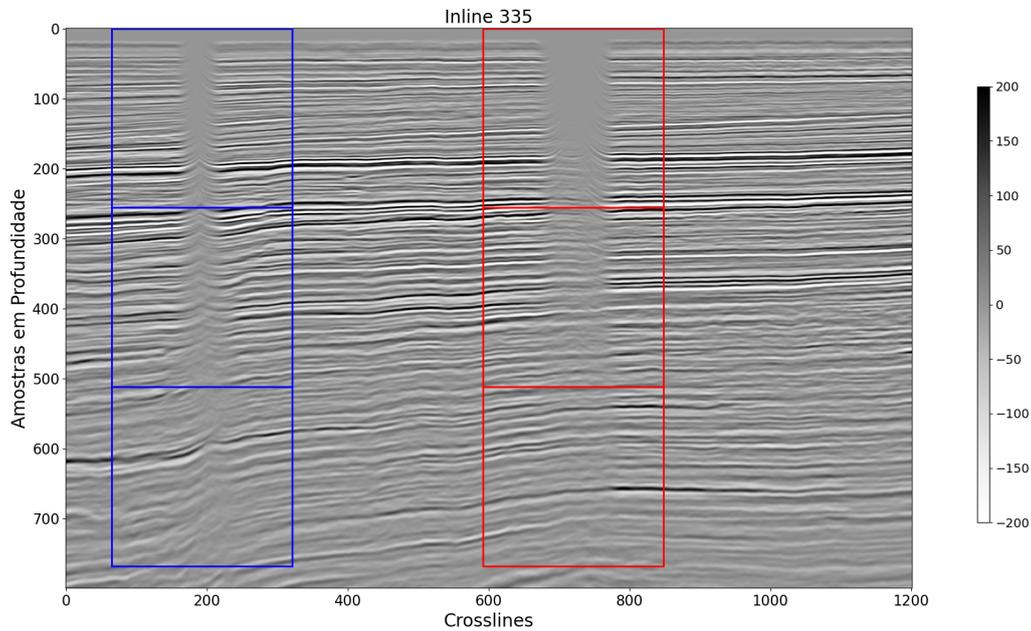


Figura 5.20: Exemplo dos recortes feitos na linha 335, centralizados na lacuna maior (em vermelho) e na lacuna menor (em azul). São três *patches* de tamanho 256x256 em sequência na vertical, sem intersecção.

Aplicamos o modelo correspondente a essas imagens e por fim substituímos as imagens geradas pelos modelos nas linhas sísmicas. É importante notar que o modelo foi treinado com o dado normalizado, então antes de fazer a substituição, é preciso fazer a operação inversa da normalização a fim de recuperar as amplitudes na escala sísmica. Um exemplo de uma linha recuperada pode ser encontrado na Figura 5.21. Em (a), mostramos uma linha antes da interpolação e, em (b), como a mesma fica após a interpolação. Observa-se que a interpolação realiza uma boa reconstrução na maior parte da seção, com alguns problemas na região do fundo do mar, em que a falta de informação logo acima prejudica a reconstrução da imagem.

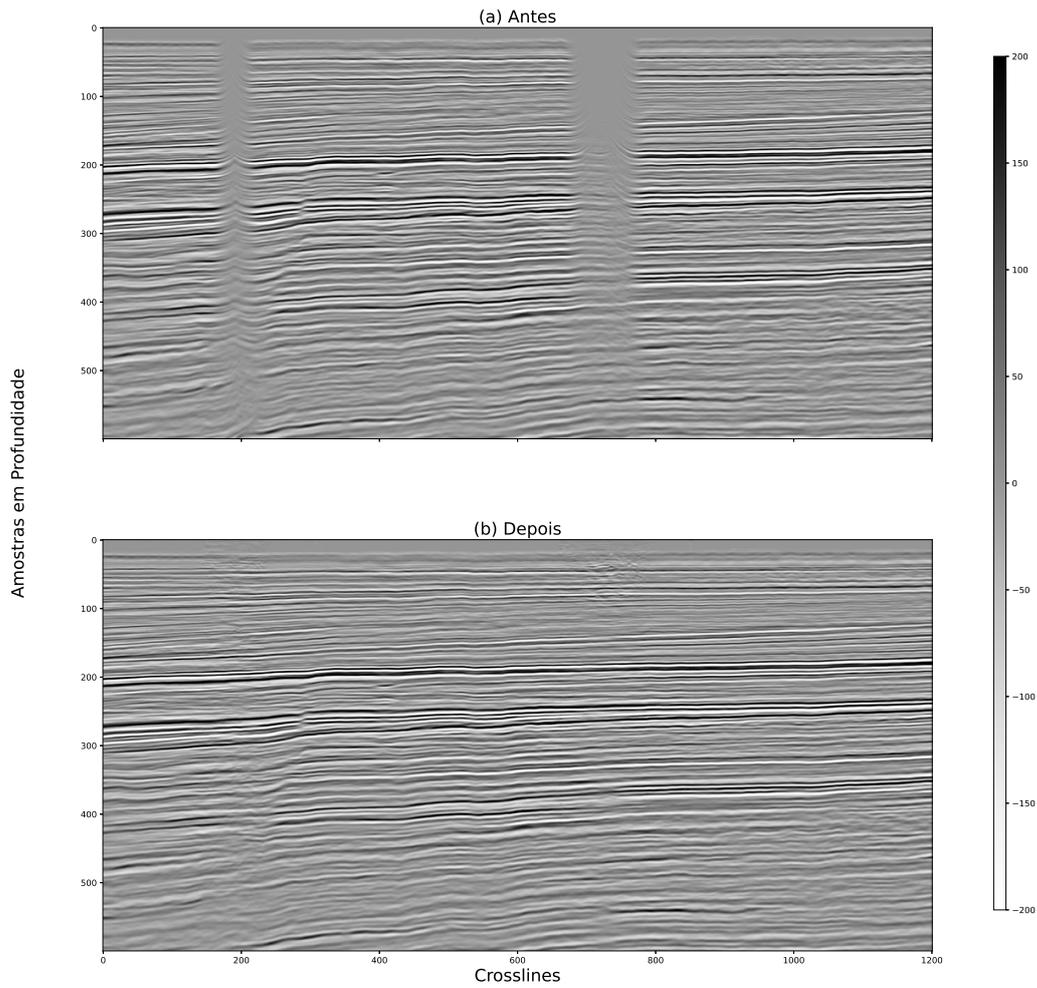


Figura 5.21: Linha 331 antes da interpolação usando o modelo treinado (superior) e depois da interpolação (inferior). Notar que as duas lacunas na porção superior da seção foram interpoladas satisfatoriamente, exceto no fundo do mar.



# Análises

### 6.1 Avaliação Qualitativa

Nesta seção, realizamos uma avaliação qualitativa sobre as regiões da seção em que o modelo foi melhor reconstruído e nas regiões em que foi pior. Aqui, também buscamos apontar algumas armadilhas que podem surgir quando algum intérprete (profissional responsável por fazer a correspondência da imagem sísmica para a geologia) for examinar os dados interpolados.

Os resultados são exibidos em três profundidades diferentes para mostrar o efeito de conteúdo de frequência. Mostramos quatro imagens para cada profundidade selecionada: (1) a imagem mais à esquerda é o *patch* verdadeiro com a máscara artificial; (2) a imagem seguinte é o gabarito; (3) a próxima é a imagem falsa gerada pelo modelo Shift-Net treinado; (4) a imagem mais à direita é a diferença entre o gabarito e a imagem interpolada. Como essa análise não é influenciada pelo tamanho da lacuna, exibiremos apenas as imagens para a máscara de largura de 124 *pixels*.

A imagem rasa analisada inicia-se na amostra 176 em profundidade e é exibida na Figura 6.1. Observa-se que a maioria dos refletores (interfaces entre camadas geológicas com impedâncias acústicas diferentes) estão bem recuperados e informações detalhadas também estão presentes. Se a imagem falsa não for apresentada como falsa, o intérprete poderia assumi-la como a verdadeira.

A segunda profundidade inicia-se na amostra 528 e pode ser vista na Figura 6.2. A perda de conteúdo de frequência é clara. O desempenho é um pouco inferior em comparação com a contrapartida mais rasa, mas ainda muito realista.

A terceira profundidade inicia-se na amostra 880 e possui ainda menos conteúdo de

frequência, com refletores um pouco mais inclinados. As imagens podem ser vistas na Figura 6.3. O desempenho é próximo da profundidade média, com uma pequena piora devido à inclinação dos refletores.

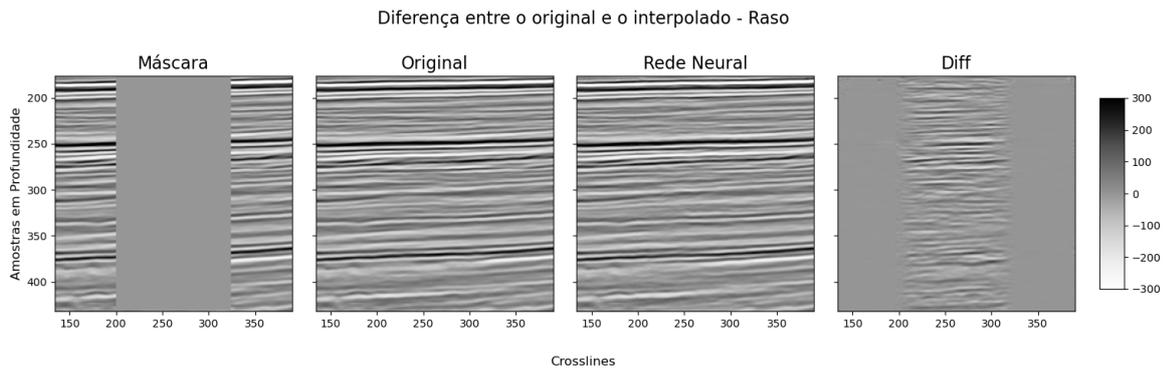


Figura 6.1: Imagens de teste na parte rasa. À esquerda está a imagem real mascarada, no centro-esquerdo a imagem real e não corrompida, na centro-direita a falsificação gerada e, à direita, a diferença entre a imagem real e a interpolada.

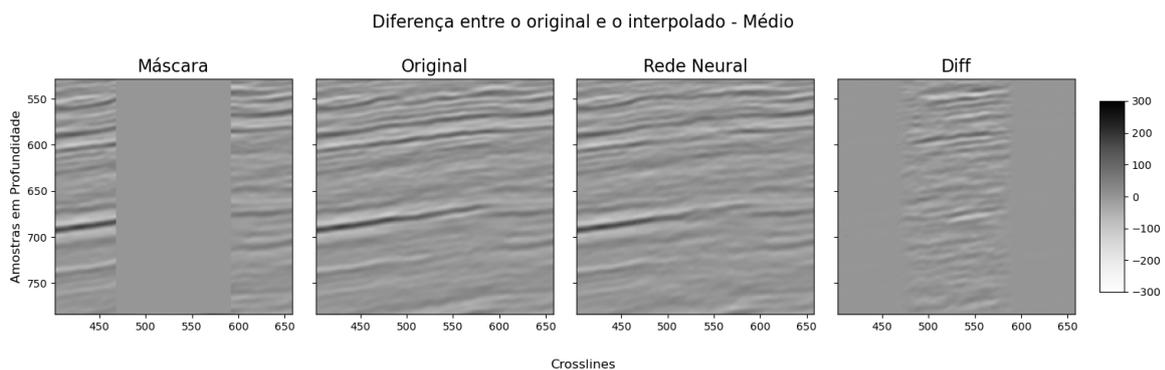
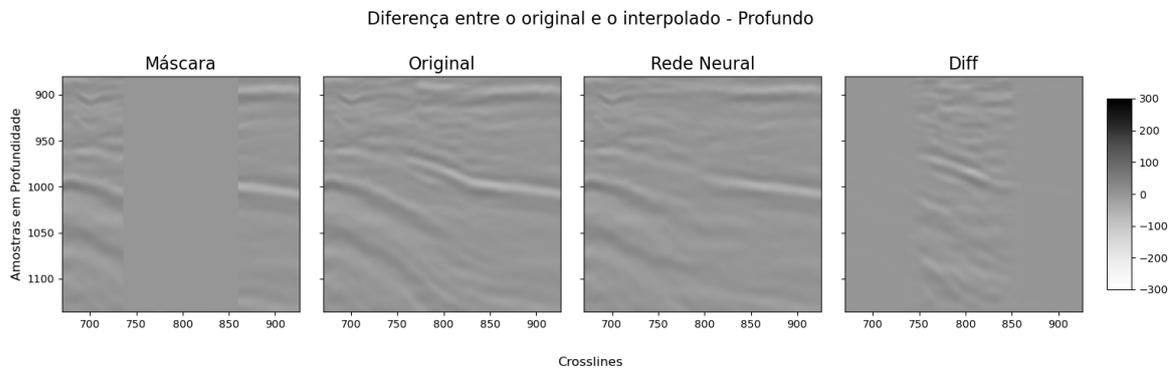
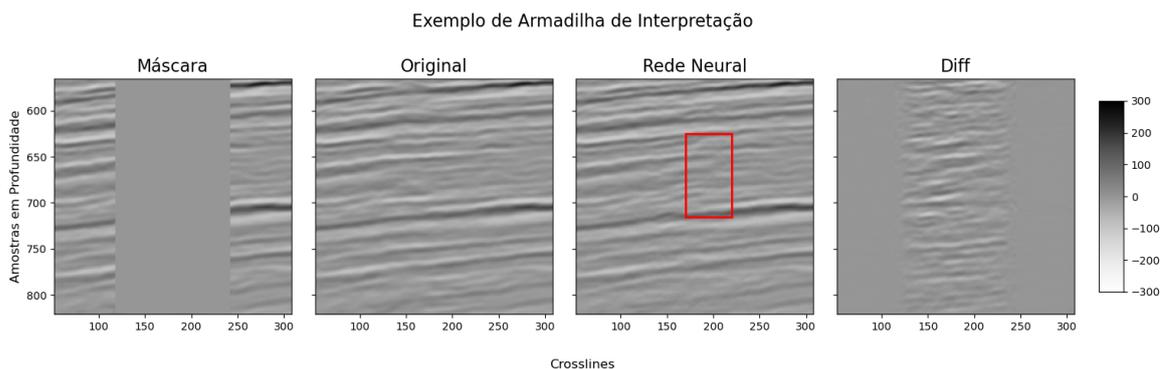


Figura 6.2: Imagens de teste na porção média. À esquerda está a imagem real mascarada, no centro-esquerdo a imagem real e não corrompida, na centro-direita a falsificação gerada e, à direita, a diferença entre a imagem real e a interpolada.



*Figura 6.3:* Imagens de teste na porção profunda. À esquerda está a imagem real mascarada, no centro-esquerdo a imagem real e não corrompida, na centro-direita a falsificação gerada e, à direita, a diferença entre a imagem real e a interpolada.

Cabe também a ressalva que o processo de interpolação pode gerar alguns artefatos, os quais podem ser confundidos com feições geológicas genuínas, de modo que o intérprete deve estar sempre consciente de tais possibilidades. Na Figura 6.4 mostramos um exemplo de artefato que pode ser interpretado como uma falha geológica.



*Figura 6.4:* Imagens de teste mostrando uma armadilha de interpretação. A região destacada pelo retângulo vermelho poderia ser interpretada como uma falha geológica genuína, quando é apenas um artefato de interpolação.

## 6.2 Coerência entre as Linhas

Nesta seção, realizamos uma análise qualitativa da coerência entre linhas consecutivas, após a reconstrução do dado. É comum que modelos geradores produzam resultados bastante diferentes, mudando muito pouco o dado de entrada e, no caso de um dado sísmico, que tem correlação espacial entre as diferentes linhas, é importante que as linhas reconstruídas, a partir de *patches* 2D, também sejam coerentes entre si. Isso permite um melhor uso do dado reconstruído, principalmente para a interpretação do mesmo.

Em seguida, mostramos duas imagens, a Figura 6.5 com a linha 330 (superior) e a linha 331 (inferior), e a Figura 6.6 com a linha 331 e a linha 332. Nota-se a semelhança entre as mesmas na região da interpolação (relembrar a Figura 5.21).

Atribuímos essa coerência entre as linhas ao fato que a camada *Shift* funciona como um vínculo, mantendo os resultados gerados numa região limitada do espaço de soluções possíveis.

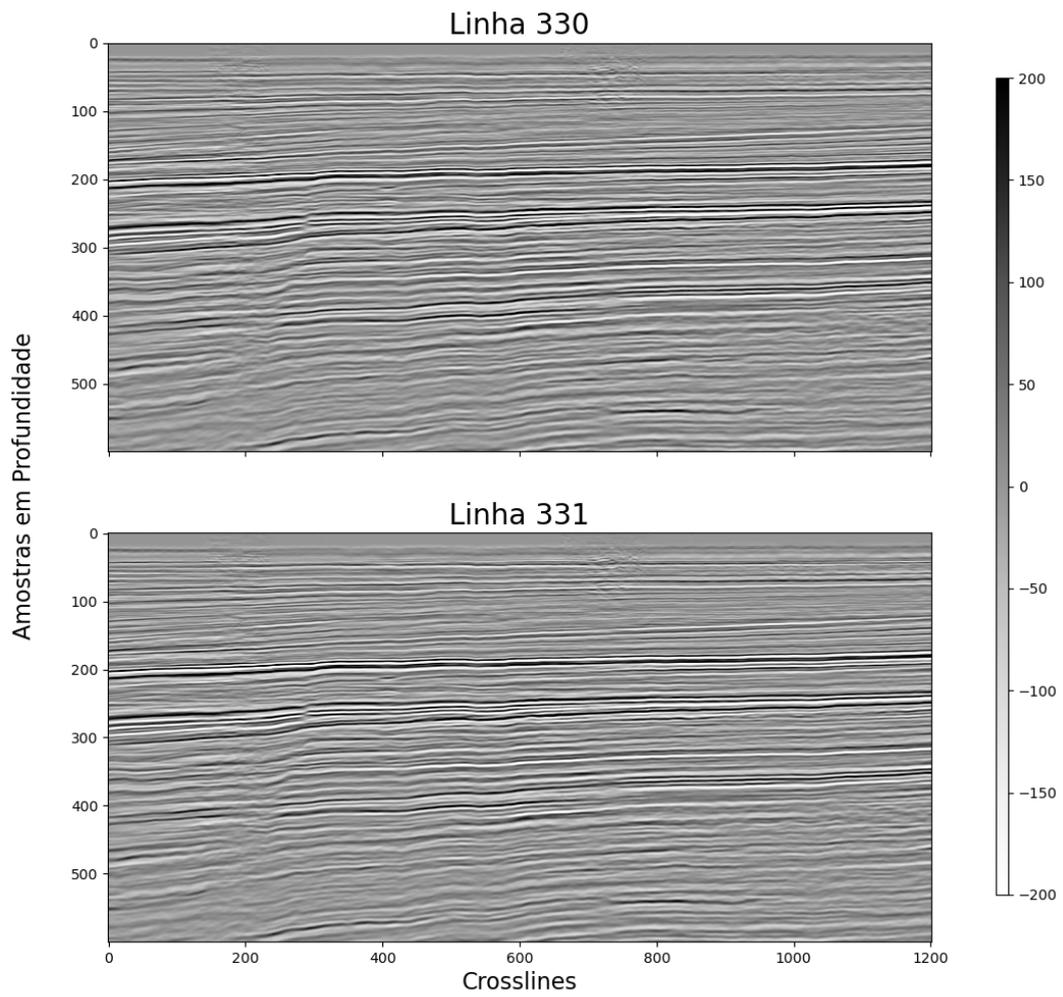


Figura 6.5: Comparação entre as linhas 330 e 331 após a reconstrução. Notar a semelhança entre elas na região da interpolação.

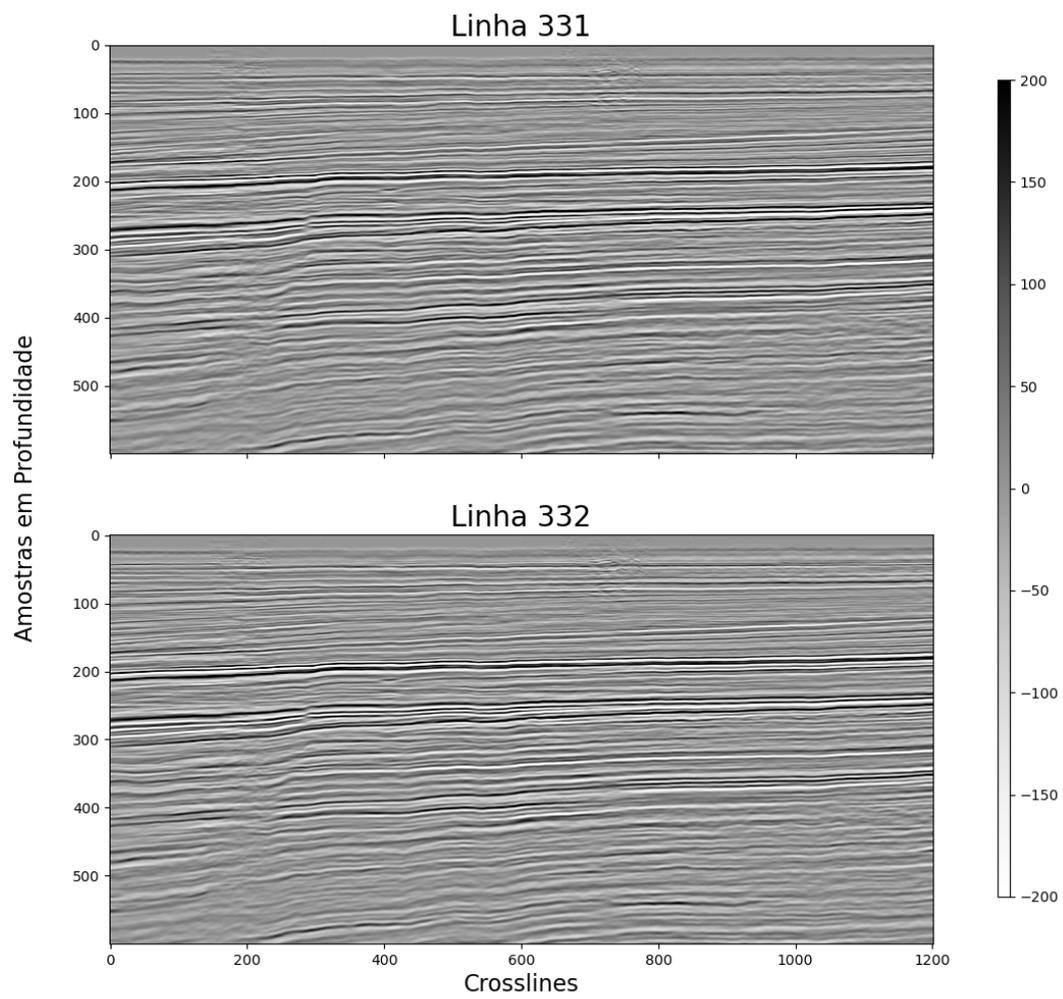


Figura 6.6: Comparação entre as linhas 331 e 332 após a reconstrução. Notar a semelhança entre elas na região da interpolação.

### 6.3 Avaliação Quantitativa

Nesta seção, comparamos os resultados obtidos com um método tradicional de interpolação baseado em PEF (Seção 4.7) e também com um resultado apresenta na literatura (Seção 3.2).

Uma primeira diferença entre o método de GANs e o de PEFs é que a rede neural precisa ser treinada, e isso leva tempo. Mas, uma vez que o treinamento seja realizado, a sua aplicação é rápida quando comparada com as PEFs. Os métodos não foram executados na mesma máquina, de modo que uma comparação quantitativa dos tempos de execução não é muito acurada. No entanto, apresentamos os tempos para fins de referência. As interpolações foram realizadas em 2min com o modelo convolucional já treinado e em 36min com as PEFs, para 1380 imagens de teste.

Outra diferença é que apenas um modelo convolucional é treinado e aplicado nas imagens (para uma lacuna de tamanho fixo), enquanto no método tradicional é necessário calcular uma PEF para cada *patch*.

Na fase de testes, os recortes são retirados aleatoriamente, mas para essa comparação os recortes foram extraídos de forma ordenada, buscando amostrar todas as regiões do dado, sendo que o mesmo conjunto de 1380 imagens foi usado, com a lacuna fixa de 124 *pixels*. As métricas de comparação estão resumidas na tabela 6.1. Notamos que a performance da GAN é superior em todos os quesitos.

Tabela 6.1 - Resultados da comparação PEF-GAN

Método	NRMS	Pearson R	SSIM	PSNR
PEF	122	0,35	0,15	21,2
Shift-Net	77	0,65	0,64	34,5

A seguir, mostramos três figuras que comparam o desempenho das PEFs e da GAN em três profundidades diferentes. Observamos que na Figura 6.7 ambos os métodos possuem um resultado verossímil, embora a PEF produza uma imagem mais linear e com uma perda de energia central mais acentuada. No entanto, nas Figuras 6.8 e 6.9, que possuem um menor conteúdo de frequências, vemos que o método de redes neurais acaba tendo um resultado bastante superior, destacado pelas imagens de diferença entre o original e a

interpolada, em que vemos mais energia não representada nas imagens geradas via PEF.

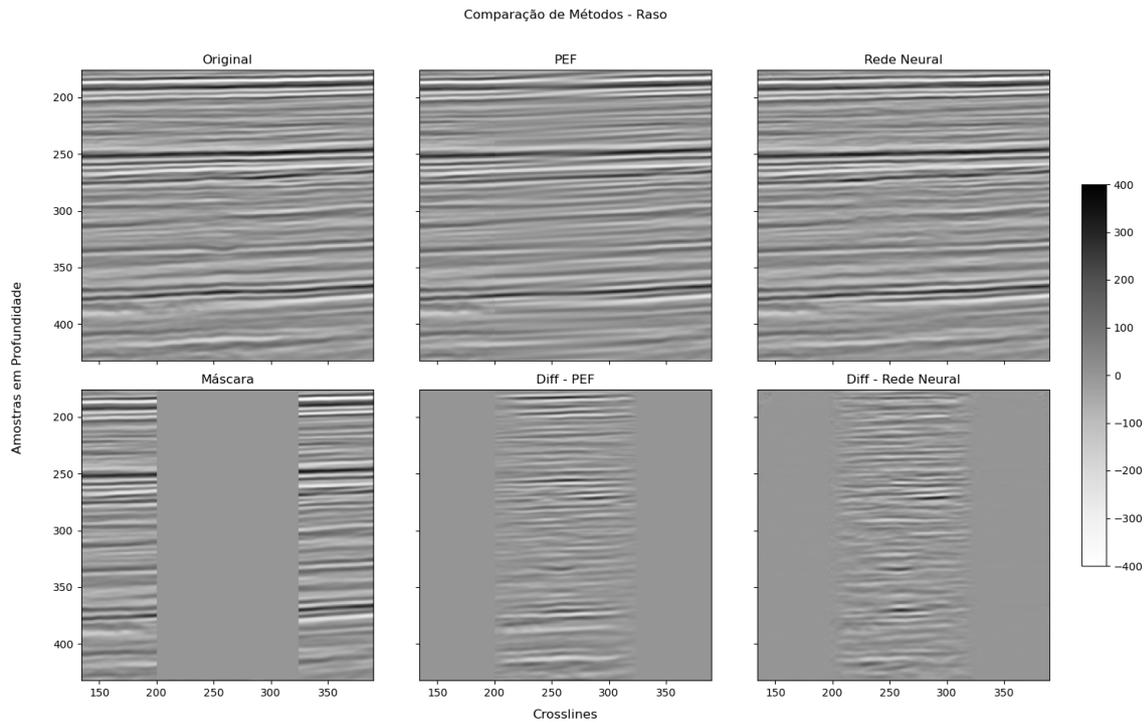


Figura 6.7: Comparação entre as interpolações por PEF e pela GAN para o raso. Na parte superior temos a imagem original à esquerda, a interpolação da PEF ao centro e a da rede neural à direita. Na parte inferior temos a imagem com máscara à esquerda e a diferença do original à PEF no centro e do original à GAN à direita.

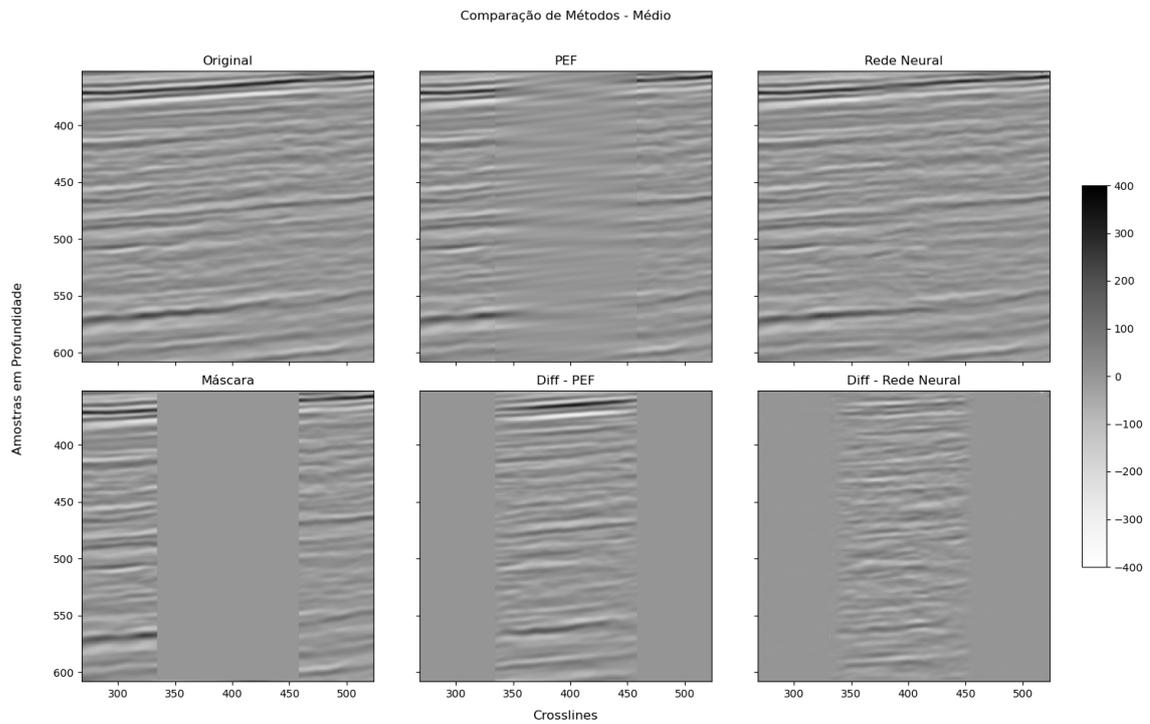


Figura 6.8: Comparação entre as interpolações por PEF e pela GAN para a profundidade média. Na parte superior temos a imagem original à esquerda, a interpolação da PEF ao centro e a da rede neural à direita. Na parte inferior temos a imagem com máscara à esquerda e a diferença do original à PEF no centro e do original à GAN à direita.

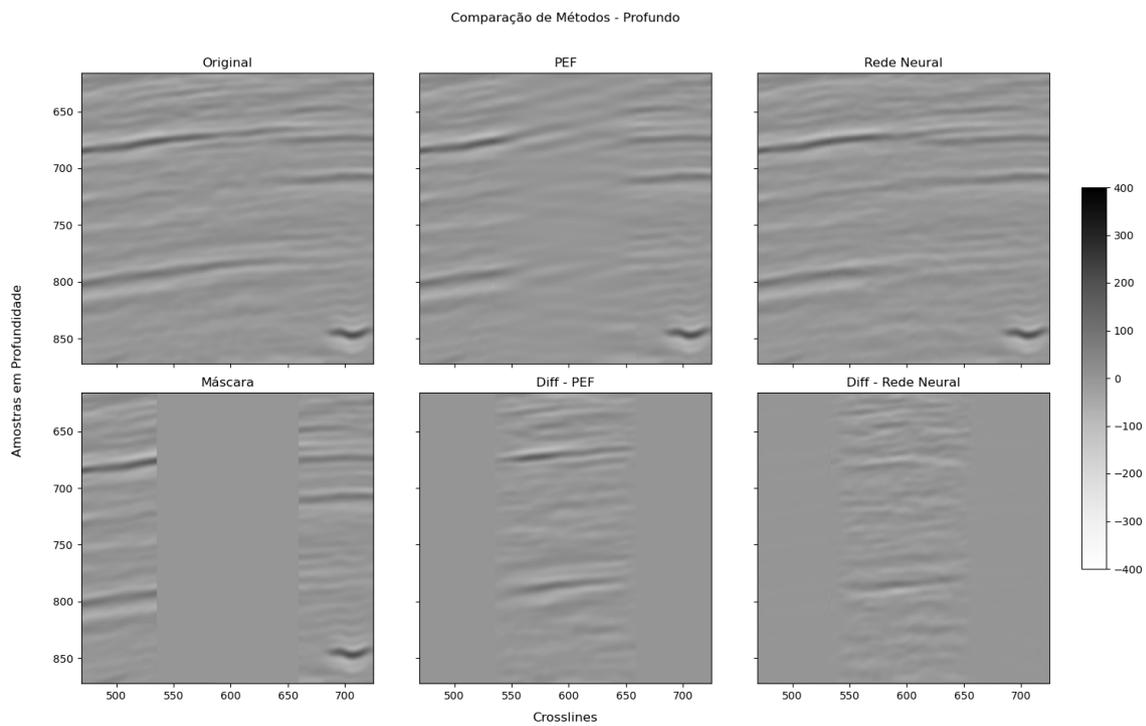


Figura 6.9: Comparação entre as interpolações por PEF e pela GAN para o profundo. Na parte superior temos a imagem original à esquerda, a interpolação da PEF ao centro e a da rede neural à direita. Na parte inferior temos a imagem com máscara à esquerda e a diferença do original à PEF no centro e do original à GAN à direita.

A comparação entre os resultados obtidos nesta dissertação e aqueles apresentados na literatura é menos trivial, uma vez que ainda não há uma padronização nas métricas de avaliação e muitas vezes a formulação dos problemas é feita de forma diferente. Na seção 3.2, mostramos três artigos que trabalharam com o problema de interpolação de dados usando redes neurais, mas dois deles (Mandelli et al. (2019) e Wei et al. (2021)) trabalharam apenas com pequenas lacunas aleatórias nos dados, simulando traços faltantes em dados abertos, no domínio do tiro ou receptor. O trabalho de Oliveira et al. (2018) já possui maior similaridade com o nosso, interpolando em lacunas de tamanho fixo e com mais *pixels* em dados migrados.

Assim, nossa comparação se limitou ao trabalho de Oliveira et al. (2018), que usou apenas a métrica do coeficiente de correlação  $R$  de Pearson. Como não reproduzimos o referido trabalho, avaliamos a correlação média obtida para a maior lacuna interpolada em cada um dos trabalhos. Os resultados são apresentados na tabela 6.2.

Tabela 6.2 - Resultados da comparação com a literatura

Método	Pearson $R$ da maior lacuna	Tamanho da lacuna (% da imagem)
Pix2Pix <sup>1</sup>	0,49	30
Shift-Net	0,68	48,4

<sup>1</sup> Oliveira et al. (2018).

Podemos notar que, apesar dos dados usados serem diferentes, o resultado do nosso método é efetivo e pode ser usado como alternativa ao previamente publicado na literatura.



## Conclusões

Nesta dissertação, exploramos um problema recorrente em aquisições de dados sísmicos de reflexão do tipo *towed streamer*, em que uma obstrução à navegação do navio gera uma lacuna no dado final. Para mitigar os problemas de imageamento gerados por esse tipo de levantamento, fizemos uso de redes neurais. No Capítulo 2, mostramos os fundamentos das redes neurais e exemplos de aplicação na geofísica foram apresentadas no Capítulo 3, em especial, na interpolação de dados (Seção 3.2). Neste capítulo, resumimos os resultados e as principais conclusões obtidas e discutimos melhorias e possíveis desenvolvimentos futuros.

### 7.1 Dos Resultados

Neste trabalho, realizamos a interpolação de dados sísmicos utilizando uma rede neural convolucional treinada de forma competitiva (GAN) em um dado real. Avaliamos o desempenho da rede sob a ótica de quatro métricas de avaliação diferentes (Seção 4.6) e comparamos o resultado com uma técnica tradicional de interpolação (Seção 4.7), obtendo melhor pontuação em todas as métricas (Tabela 6.1). Também, comparamos quantitativamente o desempenho do nosso método com um resultado publicado na literatura e obtivemos uma melhor correlação do dado interpolado com o dado real para uma lacuna percentual da imagem maior que a lacuna máxima da literatura (Tabela 6.2). Confirmamos a observação feita por Oliveira et al. (2018) de que, para um mesmo modelo, a correlação é pior quanto maior for a lacuna a ser interpolada.

Do ponto de vista qualitativo, observamos que o desempenho da rede é melhor nas partes rasas do dado, em que a geologia é mais plano-paralela e o conteúdo de frequência

da imagem sísmica é maior, deteriorando a qualidade da interpolação com a inclinação das camadas e com a perda de energia e frequência da parte mais profunda. Também analisamos a coerência das linhas sísmicas reconstruídas, mostrando que a rede não gera imagens completamente diferentes com pequenas variações da imagem de entrada. Essa análise em particular não foi encontrada na literatura e é importante para a interpretação do dado final.

## 7.2 Do Trabalho Futuro

Todo o trabalho desenvolvido foi aplicado a um dado migrado e empilhado, de tal forma que um passo adiante seria verificar o desempenho da rede para interpolações num dado aberto. Na rede discriminadora, identificamos duas melhorias possíveis, através da arquitetura *patch* GAN desenvolvida no trabalho Isola et al. (2017) e da inclusão de uma camada de ruído Gaussiano como feito em Wei et al. (2021).

Sabe-se que quanto maior a variabilidade dos dados de treinamento, melhor é a capacidade de generalização da rede neural, de modo que o aumento da quantidade de dados de treinamento poderia ainda contribuir para uma rede mais robusta, podendo evoluir principalmente nas deficiências já identificadas, como camadas inclinadas e conteúdo de frequências.

Outro tópico que pode ser futuramente abordado é o uso de máscaras com posicionamento e largura variáveis, pois um modelo único, com resultado satisfatório para esse tipo de configuração de máscaras, pode substituir o uso de três modelos com largura fixa.

## Referências Bibliográficas

- Carpenter C., Artificial Intelligence Improves Seismic-Image Reconstruction, *Journal of Petroleum Technology*, 2019, vol. 71, p. 65
- Claerbout J., Fomel S., GEOPHYSICAL IMAGE ESTIMATION BY EXAMPLE. [lulu.com](http://lulu.com), 1994, 252
- Doersch C., Singh S., Gupta A., Sivic J., Efros A. A., What makes Paris look like Paris?, *Communications of the ACM*, 2015, vol. 58, p. 103
- Dutta P., Power B., Halpert A., Ezequiel C., Subramanian A., Chatterjee C., Hari S., Prindle K., Vaddina V., Leach A., Domala R., Bandura L., Mascaro M., 3D Conditional Generative Adversarial Networks to enable large-scale seismic image enhancement, *arXiv preprint arXiv:1911.06932*, 2019, pp 1–8
- Goodfellow I., Bengio Y., Courville A., *Deep Learning*. MIT Press, 2016
- Goodfellow I. J., Pouget-Abadie J., Mirza M., Xu B., Warde-Farley D., Ozair S., Courville A., Bengio Y., Generative adversarial networks, *Advances in Neural Information Processing Systems*, 2014, vol. 3, p. 2672
- Haavik K. E., Landrø M., Variable source depth acquisition for improved marine broadband seismic data, *Geophysics*, 2015, vol. 80
- Halpert A. D., Deep learning-enabled seismic image enhancement, 2018 SEG International Exposition and Annual Meeting, SEG 2018, 2019, pp 2081–2085
- He K., Gkioxari G., Dollar P., Girshick R., Mask R-CNN, *Proceedings of the IEEE International Conference on Computer Vision*, 2017, vol. 2017-Octob, p. 2980

- He K., Zhang X., Ren S., Sun J., Deep residual learning for image recognition, Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2015, pp 770–778
- Hill A. W., Two-ship undershooting of a production platform: The magnus high-resolution seismic project, Oceanology: Proceedings of an International Conference, SUT 1986, 1986, pp 191–202
- Huynh-Thu Q., Ghanbari M., Scope of validity of PSNR in image/video quality assessment, Electronics Letters, 2008, vol. 44, p. 800
- Ioffe S., Szegedy C., Batch normalization: Accelerating deep network training by reducing internal covariate shift, 32nd International Conference on Machine Learning, ICML 2015, 2015, vol. 1, p. 448
- Isola P., Zhu J. Y., Zhou T., Efros A. A., Image-to-image translation with conditional adversarial networks. In Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, (CVPR) , IEEE, Honolulu, HI, USA, 2017, p. 5967
- Janocha K., Czarnecki W. M., On loss functions for deep neural networks in classification, arXiv preprint arXiv:1702.05659, 2017
- Johann P. R. S., Thedy E. A., Gomes F. A., Schinelli M. C., , 2006 4D Seismic in Brazil: Experiences in Reservoir Monitoring
- Jouno F., Martinez A., Ferreira D., Donno D., Khalil A., Illuminating Santos Basin’s pre-salt with OBN data: Potential and challenges of FWI. In SEG International Exposition and Annual Meeting 2019 , 2020
- Kingma D. P., Ba J. L., Adam: A method for stochastic optimization, 3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings, 2015, pp 1–15
- Kragh E., Christie P., Seismic repeatability, normalized rms, and predictability, The Leading Edge, 2002, vol. 21, p. 640
- Krizhevsky A., Sutskever I., Hinton G. E., ImageNet Classification with Deep Convolutional Neural Networks, NIPS, 2012, pp 1–9

- Krueger J., Donno D., Pereira R., Mondini D., Souza A., Espinoza J., Khalil A., Internal multiple attenuation for four pre-salt fields in the Santos Basin, Brazil, 2018 SEG International Exposition and Annual Meeting, SEG 2018, 2019, pp 4523–4527
- Lecun Y., Bengio Y., Hinton G., Deep learning, *Nature*, 2015, vol. 521, p. 436
- Liu M., Li W., Jervis M., Nivlet P., 3D Seismic facies classification using convolutional neural network and semi-supervised generative adversarial network, SEG International Exposition and Annual Meeting 2019, 2019, pp 4995–4999
- Maas A. L., Hannun A. Y., Ng A. Y., Rectifier nonlinearities improve neural network acoustic models, in *ICML Workshop on Deep Learning for Audio, Speech and Language Processing*, 2013, vol. 28
- McCulloch W. S., Pitts W., A logical calculus of the ideas immanent in nervous activity, *The bulletin of mathematical biophysics* 1943 5:4, 1943, vol. 5, p. 115
- Mandelli S., Lipari V., Bestagini P., Tubaro S., Member S., Interpolation and Denoising of Seismic Data using Convolutional Neural Networks, arXiv preprint arXiv:1901.07927, 2019, pp 1–17
- Nath A., Verschuur D. J., Imaging with surface-related multiples to overcome large acquisition gaps, *Journal of Geophysics and Engineering*, 2020, pp 742–758
- Oliveira D. A. B., Ferreira R. S., Silva R., Brazil E. V., Interpolating Seismic Data With Conditional Generative Adversarial Networks, *IEEE GEOSCIENCE AND REMOTE SENSING LETTERS*, 2018, vol. 15, p. 1952
- Paszke A., Gross S., Massa F., Lerer A., Bradbury J., Chanan G., Killeen T., Lin Z., Gimelshein N., Antiga L., Desmaison A., Köpf A., Yang E., DeVito Z., Raison M., Tejani A., Chilamkurthy S., Steiner B., Fang L., Bai J., Chintala S., PyTorch: An imperative style, high-performance deep learning library, *Advances in Neural Information Processing Systems*, 2019, vol. 32
- Pearson K., VII. Note on regression and inheritance in the case of two parents, *Proceedings of the Royal Society of London*, 1895, vol. 58, p. 240

- Penna R., Araújo S., Geisslinger A., Sansonowski R., Oliveira L., Rosseto J., Matos M., Carbonate and igneous rock characterization through reprocessing, FWI imaging, and elastic inversion of a legacy seismic data set in Brazilian presalt province, *Leading Edge*, 2019, vol. 38, p. 11
- Redmon J., Divvala S., Girshick R., Farhadi A., You only look once: Unified, real-time object detection, *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2016, vol. 2016-December, p. 779
- Ronneberger O., Fischer P., Brox T., U-Net: Convolutional Networks for Biomedical Image Segmentation. In *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, Springer International Publishing, Cham, 2015, p. 234
- Rosenblatt F., The perceptron: A probabilistic model for information storage and organization in the brain, *Psychological Review*, 1958, vol. 65, p. 386
- Rumelhart D. E., Hinton G. E., Williams R. J., Learning representations by back-propagating errors, *Nature*, 1986, vol. 323, p. 533
- Sano S., Tan T. Q., Jo G., High resolution 3D seismic undershooting acquisition over platforms and seismic processing challenges in a gas producing field, *Journal of the Japanese Association for Petroleum Technology*, 2020, vol. 85, p. 44
- Siahkoobi A., Kumar R., Herrmann F. J., Deep-learning based ocean bottom seismic wavefield recovery, *SEG International Exposition and Annual Meeting 2019*, 2019, pp 2232–2237
- Simonyan K., Zisserman A., Very Deep Convolutional Networks for Large-Scale Image Recognition, conference paper at *ICLR*, 2015, pp 1–14
- Szegedy C., Liu W., Jia Y., Sermanet P., Reed S., Anguelov D., Erhan D., Vanhoucke V., Rabinovich A., Going deeper with convolutions, *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2014, vol. 07-12-June, p. 1
- Telford W. M., Geldart L. P., Sheriff R. E., *Applied Geophysics*, Applied Geophysics, 1990

- Ulyanov D., Vedaldi A., Lempitsky V., Instance Normalization: The Missing Ingredient for Fast Stylization, arXiv preprint arXiv:1607.08022, 2016
- Waldeland A. U., Jensen A. C., Gelius L.-J., Solberg A. H. S., Convolutional neural networks for automated seismic interpretation, *The Leading Edge*, 2018, vol. 37, p. 529
- Wang Z., Bovik A. C., Sheikh H. R., Simoncelli E. P., Image quality assessment: from error visibility to structural similarity, *IEEE Transactions on Image Processing*, 2004, vol. 13, p. 600
- Wei Q., Xiangyang L., Song M., Reconstruction of irregular missing seismic data using conditional generative adversarial networks, *Geophysics*, 2021, vol. 86
- Wrona T., Fossen H., Lecomte I., Eide C., Gawthorpe R., Seismic expression of shear zones: Insights from 2-D point-spread-function-based convolution modelling, *Journal of Structural Geology*, 2020, p. 104121
- Wu X., Geng Z., Shi Y., Pham N., Fomel S., Caumon G., Building realistic structure models to train convolutional neural networks for seismic structural interpretation, *Geophysics*, 2020, vol. 85, p. WA27
- Yan Z., Li X., Li M., Zuo W., Shan S., Shift-net: Image inpainting via deep feature rearrangement, *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2018, vol. 11218 LNCS, p. 3
- Zhang H., Wang W., Wang X., Chen W., Zhou Y., Wang C., Zhao Z., An implementation of the seismic resolution enhancing network based on GAN, *SEG International Exposition and Annual Meeting 2019*, 2019, pp 2478–2482
- Zhao T., Seismic facies classification using different deep convolutional neural networks, *2018 SEG International Exposition and Annual Meeting*, SEG 2018, 2019, pp 2046–2050



# Apêndice



Apêndice A

---

Resumo Expandido Publicado no 17<sup>o</sup> Congresso  
Internacional da Sociedade Brasileira de Geofísica

(Essa página é deixada em branco)



## Convolutional Neural Networks as a Tool for Seismic Data Interpolation

Maurílio Ferreira Salgado<sup>\*1,2</sup>, Carlos Alberto Moreno Chaves<sup>2</sup> and Roberto Hirata Júnior<sup>2, 1</sup> Petrobras, <sup>2</sup> USP

Copyright 2021, SBGf - Sociedade Brasileira de Geofísica

This paper was prepared for presentation during the 17<sup>th</sup> International Congress of the Brazilian Geophysical Society held in Rio de Janeiro, Brazil, 16-19 August 2021.

Contents of this paper were reviewed by the Technical Committee of the 17<sup>th</sup> International Congress of the Brazilian Geophysical Society and do not necessarily represent any position of the SBGf, its officers or members. Electronic reproduction or storage of any part of this paper for commercial purposes without the written consent of the Brazilian Geophysical Society is prohibited.

### Abstract

Convolutional Neural Networks (CNNs) algorithms have been very successful in solving some image problems, such as some classification and object detection ones. Applying CNNs on seismic images is straight forward and academic research on this field is gaining momentum. Uses range from facies classification, semantic segmentation, and automatic faults and horizons interpretation to the improvement of seismic resolution, noise filtering, data reconstruction and others. In this work, we aim to solve a common seismic acquisition problem, which is to image the subsurface beneath an obstruction. Undershooting procedures and processing techniques are normally tried and usually can achieve good results in mitigating this problem, but on near offsets images are generally blank. We use GANs (Generative Adversarial Networks) to train a CNN model that interpolates seismic data, and thus, can be used to generate data in cases of platform obstruction with high likelihood and coherence.

### Introduction

Seismic imaging of producing oil and gas reservoirs faces many challenges, one of them is to overcome the lack of subsurface sampling beneath platforms. Towed streamer vessels have restrictions to navigate the production structures vicinity due to safety and maneuverability reasons. The fold of these areas can reach zero value, with no information at all.

Undershooting (Hill, 1986) is a crucial technique to mitigate the probing of an area of difficult access, as reported by Johann et al. (2006), Sano et al. (2020), among others. It consists of using two separate vessels, one for the source only (air gun i.e.), that navigates on one side of the obstruction and the other vessel carries a reduced number of receptor cables (or cable) that navigates on the other side, getting closer to the platform. This arrangement is a very efficient technique, because it places the midpoint between source and receptor on the gap area recovering much data, albeit, for the near offsets, most information is completely lost.

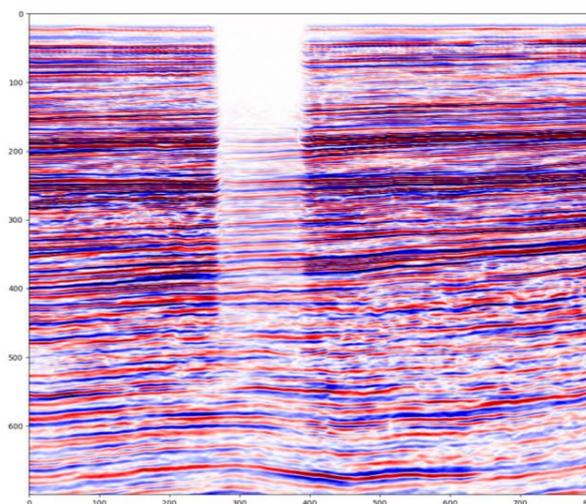
In addition to this acquisition solution, processing procedures further mitigate the problem by using regularization, migration, seismic merging and matching, but as pointed by Nath and Vershuur (2020): "If we use

conventional seismic imaging algorithms that rely on primary reflection data only, migration artefacts become unavoidable". They propose to use the surface-related multiples to get around cases like this one, with large gaps of information.

From the current level of technology development, it is clear that no complete solution exists and non-conventional techniques must address these near offsets gaps.

Our approach uses recent CNNs architectures that successfully solve image problems and apply them to seismic data interpolation. Mandelli et al. (2019), for example, have performed great work using the U-net (Ronneberger et al., 2015) architecture, but only on synthetic data and have not addressed a large coherent gap, while Oliveira et al. (2018) have used the pix2pix (Isola et al., 2017) network, applying in seismic patches of 80x80 samples, further reshaped to 256x256, increasing the possible number of training images, but introducing an up-sampled training image. They showed that the larger the size of the gap, the lower the correlation of the interpolated image with the real image.

This work uses the ShiftNet (Yan et al. 2018), trained on a GAN scheme (Goodfellow et al., 2014). This network outperforms both the plain vanilla U-net, used in Mandelli et al. (2019), and the pix2pix network, used in Oliveira et al. (2018), in the inpainting task on the Paris StreetView dataset (Doersh et al. 2012). The network is trained and tested on real data from Brazil's offshore, kindly handed over by Petrobras.



**Figure 1-** Migrated and stacked seismic session showing the information gap due to platform obstruction.

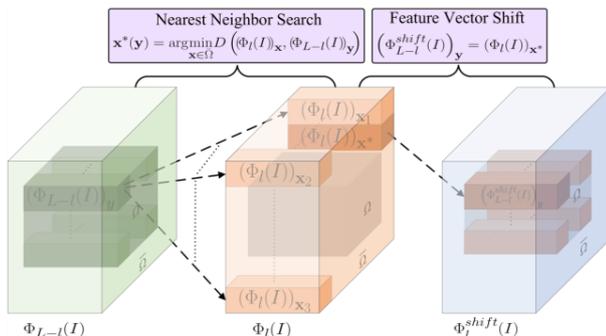
**Method**

*Model Description*

Literature on Neural Networks architectures is extensive and exponentially growing. We chose the Shiftnet model because it outperforms other benchmark models in the task of inpainting RGB images.

Shiftnet (Yan et al. 2018) is a U-net with an additional layer named shift-layer. The U-net architecture consists of an encoder-decoder scheme with skip-connections (He, et al., 2015) from the encoder part to the correspondent decoder one. We do not discuss its details in this paper, and we invite the readers to refer to Shiftnet and U-net papers. We will focus on the Shiftnet innovation, which is the shift-layer. Yan et al. (2018) were inspired by the deterministic technics classified as exemplar-based inpainting methods. As they describe: “the completion is conducted from the exterior to the interior of the missing part by searching and copying best matching patches from the known region.”

Their insight was to make a similar approach, but in the convolutional feature space. They added a layer constructed by comparing the transpose convolutional feature vectors of the decoder layer on the receptive field of the missing image part (the green layer in Fig. 2) with the skip-connection features of the corresponding layer, but on the complementary receptive field (the salmon layer in Fig. 2). Then they copied the most similar feature vector of the skip-connection on the shift-layer, in the same place as the transpose connection (the light blue layer in Fig. 2).



**Figure 2** – The shift-layer (light blue) is constructed with feature vectors of the skip-connection layer (light orange) that are most alike to the transpose convolutional layer feature vectors (light green). Image obtained from [https://github.com/Zhaoyi-Yan/Shift-Net\\_pytorch](https://github.com/Zhaoyi-Yan/Shift-Net_pytorch).

The name shift-layer is due to its feature vectors being the ones of the skip-connection layer shifted to the missing data slot. The layers are stacked and feed the next convolutional layer. Yan et al. (2018) experimented with the shift-layer location and found that it was most effective (balancing time of computation and final result) in the L-3 layer, L being the last layer of the U-net.

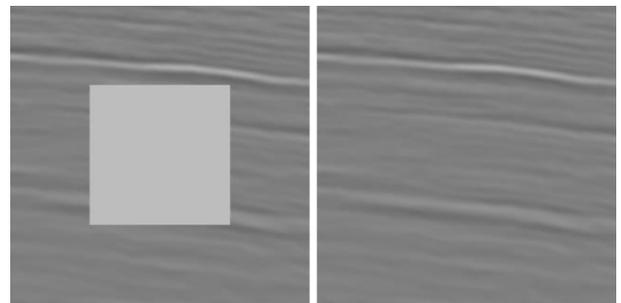
*Data description*

The dataset comes from Brazil’s offshore in Santos basin. It is a PSDM (post-stack depth migration) merge of two different acquisitions, with standard techniques applied to mitigate the obstruction problem, but the gap is still

present (Fig. 1). The exact location of the data is omitted, due to a previous agreement with Petrobras.

Seismic data has a size of 800x1200x1500 samples, of which 800 are the inlines, 1200 are the xlines and 1500 are the depth samples. The training set was obtained from the first 700 inlines, excluding the inlines with an information gap, totalizing 563 usable inlines. From these inlines, 15000 2D images patches with 256x256 pixels were randomly picked, leaving 12000 for training and 3000 for validation. Statistics of the mean and maximum values of the training set is stored, and used later in training to normalize the images. We use the last 100 inlines to test the model, from which 1000 patches were randomly generated.

All those images are perfect, with no information gap. For training purposes, we introduce the missing samples artificially as a mask in order to have the ground truth from the uncorrupted image. At the present stage, we use only a central mask that covers 25% of the image. Figure 3 shows an example of artificial deleting.



**Figure 3** – Training patches of size 256x256 with central mask on the left and with the ground truth on the right.

*Training Phase*

The model was trained in an adversarial manner (GAN). Shiftnet is the generative network G and the discriminator network D is constructed as a sequential convolutional neural network. It has five convolutional layers with leaky ReLU activation. Instance normalization is also applied after convolutional layers 2, 3 and 4. The discriminator output is obtained after a sigmoid function and yields the probability of the image be a real image (ground truth).

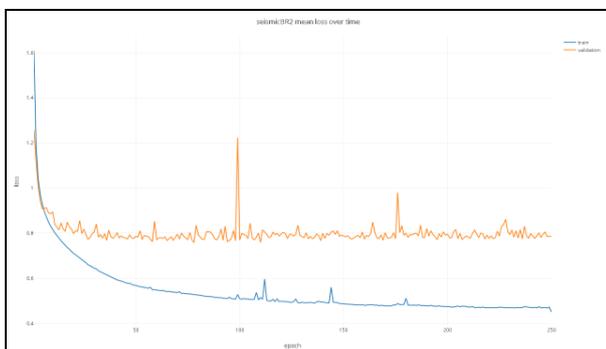
Adversarial training is a competition between the Generative network and the Discriminative network. While D is trained to uncover if an image is authentic or fake (generated by G), G generates increasingly better images to be tested by the discriminator. G fills the image gap generating fake images with detailed near realistic seismic image. Besides the adversarial loss, an L1 norm is used to ensure that the final fake image should be an approximation of the ground truth image.

In summary, we get a 256x256 image sample and normalize it with the mean and maximum values stored from the whole training set. Also, we do a random horizontal flip with a 50% probability in order to increase data variability (data augmentation). Then, a mask is added in order to simulate an image gap. This masked image is fed to the generative model, which outputs a filled fake image. The discriminator is fed with both fake and ground truth, outputting probabilities for each one. D

is updated (with adversarial loss), G is updated (with adversarial loss and L1 norm), and the cycle is repeated.

We trained the network for 250 epochs with a batch size of one (SGD – stochastic gradient descent), and within each epoch, the model sees all training set. A mean L1 norm is computed for each epoch for both the training and validation sets and evolution can be observed in Fig. 4. We see that the validation L1 norm reached a plateau roughly after epoch 50, oscillating around 0.79, but the network was still learning. The plateau can be explained because the network does not recover the same data. Therefore, the L1 norm cannot improve, but on the training set, the model is still learning to generate realistic data. Overfitting is discarded based on validation performance.

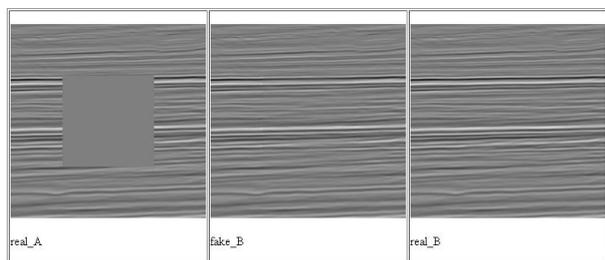
We run the model training on GPU, a GeForce GTX 1080 ti with 6.1 computing capability. The code was written in Python, modified from the Shift-Net PyTorch repository. The total time of training was 93h.



**Figure 4** – Mean L1 loss evolution for the training set (blue) and the validation set (orange).

## Results

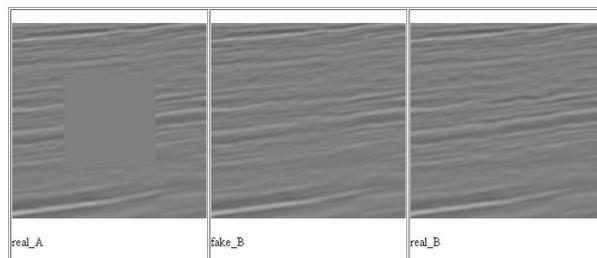
After training, the resulting model was applied to the test set with 1000 patch images. Results are displayed for three different depths to show the effect of varying frequency content. We show three images for each selected depth: (1) the leftmost image is the true patch with the artificial mask (*real\_A*); (2) the center image is the fake one generated by the trained Shift-Net model (*fake\_B*); (3) rightmost image is the ground truth (*real\_B*).



**Figure 5** – Test images sampled from 550m. On the left is the real masked image (*real\_A*), on the center the fake generated (*fake\_B*) and, on the right, the real uncorrupted image (*real\_B*).

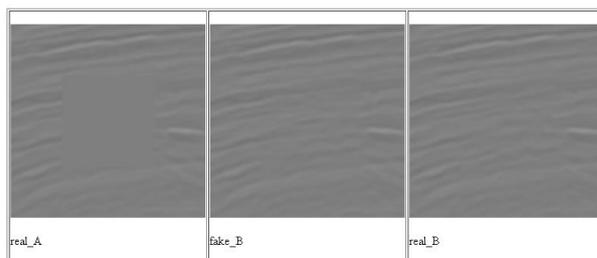
The first sampled depth is around 550m and it is shown in Fig. 5. One can observe that most energetic reflectors are well recovered and finely detailed information is also

present. If the fake image is not presented as a fake one, the interpreter could be fooled by it. Yet, with a closer look, it is possible to see a slight difference of energy in the center of the fake image, and some border artifacts, but nothing too flashy.



**Figure 6** – Test images sampled from 2250m. On the left is the real masked image (*real\_A*), on the center the fake generated (*fake\_B*) and, on the right, the real uncorrupted image (*real\_B*).

The second depth sample is close to 2250m and can be seen in Fig. 6. The loss of frequency content is clear and the reflectors are inclined. Performance is a little inferior as compared to the shallower counterpart, but still very realistic.



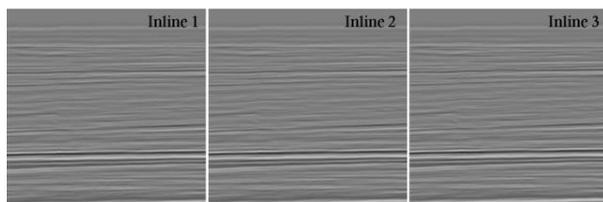
**Figure 7** - Test images sampled from 4000m. On the left is the real masked image (*real\_A*), on the center the fake generated (*fake\_B*) and, on the right, the real uncorrupted image (*real\_B*).

The third depth sample is from near 4000m with even less frequency content, and can be seen in Fig. 7. Performance is close to the depth of 2250m.

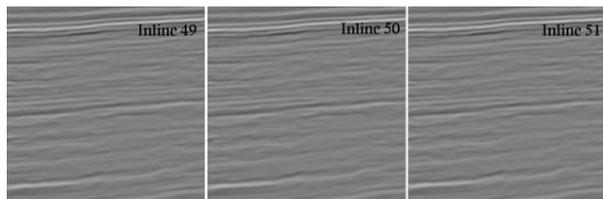
We also analyze the coherence of the generated images between adjacent inlines. This is important to reconstruct seismic data because if one inline is completely different from the next one, it is impossible to interpret a horizon. Generative models can produce images completely different from inputs slightly distinct. In our case, the Shift-layer serves as a contour condition, because it uses the information from the uncorrupted image, producing coherent images for sequential inlines. This kind of analysis was not done in previous related work and is the first to our knowledge.

Below is possible to apprise the coherence between three generated patches from sequential inlines. We show images for three different depths, the shallower is shown in Fig. 8, the middle depth is shown in Fig. 9 and the deeper one is shown in Fig. 10. The high similarity between these sequential inline makes this trained network suitable for reconstructing large gap information seismic data. We reason that this depends on the data itself, if the inline distance is too high so that images from

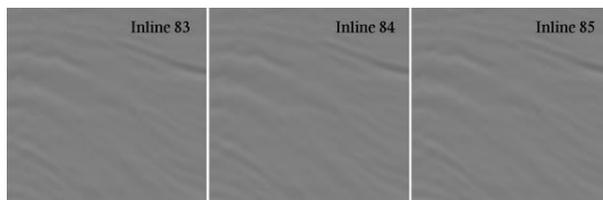
one inline to another are too different, this level of coherence will not be achieved.



**Figure 8** – Fake (generated) images from sequential inlines for shallow depth.



**Figure 9** - Fake (generated) images from sequential inlines for middle range depth.



**Figure 10** - Fake (generated) images from sequential inlines for deeper depth.

## Conclusions

Generative Adversarial Networks are applied in this work to fill large gaps of information in real seismic data. The sources of these gaps vary and they are not rare in real images. The proposed method achieved good performance by filling the images with realistic and fine-detailed information using a real dataset and artificially generated gaps. We showed that this novel neural network provides a contour condition for generating coherent images across different adjacent inlines. Future work will address comparison metrics and the reconstruction of the data itself.

## Acknowledgments

The authors would like to thank Petrobras for handing over the data and for permission to publish it. We thank FAPESP, project #2018/10767-0. This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Finance Code 001.

## References

DOERSCH, C. et al. What makes Paris look like Paris? Communications of the ACM, v. 58, n. 12, p. 103–110, 2015.

GOODFELLOW, I. J. et al. Generative adversarial networks. Advances in Neural Information Processing Systems, v. 3, n. January, p. 2672–2680, 2014.

HE, K. et al. Deep residual learning for image recognition. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, p. 770–778, 2015.

HILL, A. W. Two-ship undershooting of a production platform: The magnus high-resolution seismic project. Oceanology: Proceedings of an International Conference, p. 191–202, 1986.

ISOLA, P. et al. Image-to-image translation with conditional adversarial networks. Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, (CVPR). Honolulu, HI, USA: IEEE, 2017

JOHANN, P. R. S. et al. 4D Seismic in Brazil: Experiences in Reservoir Monitoring. Paper presented at the Offshore Technology Conference, Houston, Texas, USA, 2006.

MANDELLI, S. et al. Seismic data interpolation through convolutional autoencoder. SEG Technical Program Expanded Abstracts. Society of Exploration Geophysicists, 2018

NATH, A.; VERSCHUUR, D. J. Imaging with surface-related multiples to overcome large acquisition gaps. Journal of Geophysics and Engineering, p. 742–758, 2020.

OLIVEIRA, D. A. B. et al. Interpolating Seismic Data With Conditional Generative Adversarial Networks. IEEE GEOSCIENCE AND REMOTE SENSING LETTERS, v. 15, n. 12, p. 1952–1956, 2018.

RONNEBERGER, O.; FISCHER, P.; BROX, T. U-Net: Convolutional Networks for Biomedical Image Segmentation. (N. Navab et al., Eds.) Medical Image Computing and Computer-Assisted Intervention -- MICCAI 2015. Lecture Notes in Computer Science, vol 9351. Springer, Cham. 2015

SANO, S.; TAN, T. Q.; JO, G. High resolution 3D seismic undershooting acquisition over platforms and seismic processing challenges in a gas producing eld. Journal of the Japanese Association for Petroleum Technology, v. 85, n. 1, p. 44–53, 2020.

YAN, Z. et al. Shift-net: Image inpainting via deep feature rearrangement. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), v. 11218 LNCS, p. 3–19, 2018.