

Thiago Pacheco Carneiro

Remoção de múltiplas do dado sísmico por modelos de aprendizado de máquina

Rio de Janeiro

2024

Thiago Pacheco Carneiro

Remoção de múltiplas do dado sísmico por modelos de aprendizado de máquina

Dissertação apresentada, como requisito parcial para obtenção do título de Mestre, ao Programa de Mestrado Profissional do Centro Brasileiro de Pesquisas Físicas (CBPF). Área de concentração: Aprendizado de Máquina

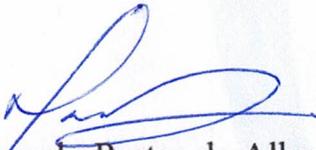
Centro Brasileiro de Pesquisas Físicas – CBPF
Programa de Mestrado Profissional

Rio de Janeiro
2024

"REMOÇÃO DE MÚLTIPLAS DO DADO SÍSMICO POR MODELOS DE
APRENDIZADO DE MÁQUINA"

THIAGO PACHECO CARNEIRO

Dissertação de Mestrado Profissional em Física
com ênfase em Instrumentação Científica,
apresentada no Centro Brasileiro de Pesquisas
Físicas do Ministério da Ciência Tecnologia e
Inovação. Fazendo parte da banca examinadora
os seguintes professores:



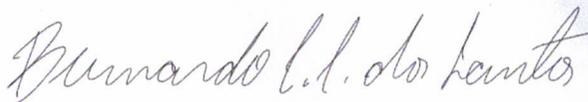
Marcelo Portes de Albuquerque – Orientador/CBPF



Elisangela Lopes de Faria – Coorientadora/ CBPF-FACC



Ana Paula Oliveira Muller - PETROBRAS



Bernardo Coutinho Camilo dos Santos - PETROBRAS

Rio de Janeiro, 17 de janeiro de 2024.

*Dedico este trabalho a Ana e Máira,
mulheres que me inspiram com sua ética
de trabalho e de vida.*

Agradecimentos

Agradeço primeiramente ao Professor Marcelo Portes de Albuquerque, que já me orientava há quase duas décadas atrás na Iniciação Científica, e que acreditou em mim neste retorno à academia.

Agradeço também à Elisangela Lopes Faria, que me guiou nos percalços do aprendizado de máquina.

Agradeço aos professores, todos, por sua dedicação e carinho.

Agradeço à Ana Paula Muller, cujas conversas foram imprescindíveis para este trabalho.

Agradeço ao Matheus Klatt pela ajuda com a simulação MDF e pelo treinamento da rede MobileNetV2-UNET.

Agradeço ao Clécio de Bom pelo incentivo e insistência para que eu fizesse este mestrado.

Por fim, agradeço ao CBPF, esta casa tão especial, por me receber novamente.

“The reasonable man adapts himself to the world; the unreasonable one persists in trying to adapt the world to himself. Therefore all progress depends on the unreasonable man.”
– George Bernard Shaw, *Man and Superman*

Resumo

Reflexões múltiplas são um ruído de tratamento complexo e computacionalmente caro no imageamento sísmico, o qual é ferramenta fundamental para diversas atividades econômicas, como a indústria de óleo e gás. No presente trabalho, investigamos a criação de modelos de aprendizado de máquina para remoção de múltiplas internas e de superfície livre de imagens sísmicas. Um simulador foi implementado para gerar imagens sísmicas com e sem múltiplas em diversos cenários geológicos. A seguir, redes neurais de diferentes tipos foram avaliadas e testadas em cenários geológicos não vistos durante o treinamento. Os resultados demonstram o treinamento adequado de diversos tipos de redes neurais, com modelos capazes de remover as múltiplas com erro absoluto médio de $7,91 \cdot 10^{-4}$ quando aplicados a imagens sísmicas simuladas.

Palavras-chaves: Aprendizado de Máquina. Geofísica. Múltiplas Sísmicas.

Abstract

Multiples are a kind of noise that is both complex and computationally expensive to deal with in seismic imaging, a fundamental tool for various economic activities, such as the oil & gas industry. Here we investigate the creation of machine learning models for internal and free surface multiple removal in seismic images. To that end, we developed a simulator capable of generating seismic images with and without multiples. The models are evaluated over a test set and a few selected geological scenarios, which were unseen during training. The results show that multiple kinds of neural networks were adequately trained, with models capable of removing multiples with a mean absolute error of $7.91 \cdot 10^{-4}$ when applied to simulated seismic images.

Key-words: Machine Learning. Geophysics. Seismic Multiples.

Sumário

Introdução	10
1 Aprendizado de Máquina	13
1.1 Redes Neurais	13
1.2 Redes Neurais Profundas	15
1.2.1 Redes Neurais Convolucionais	15
2 Imagem Sísmica	17
2.1 Simulação sísmica	19
3 Metodologia	21
4 Experimentos Computacionais e Resultados	23
4.1 Redes neurais	23
4.2 Imagens sísmicas sem convolução (Refletividade)	24
4.2.1 Geração de sistemas de subsuperfície	24
4.2.2 Conjunto de teste	24
4.2.2.1 Elemento do Conjunto de Teste	30
4.2.3 Generalização	36
4.2.3.1 Camadas Paralelas	37
4.2.3.2 Domo de Sal Homogêneo	40
4.2.3.3 Domo de Sal Heterogêneo sobre Reservatório	44
4.3 Imagens sísmicas com convolução	51
4.3.1 Redes neurais	51
4.3.2 Conjunto de teste	51
4.3.2.1 Elemento do Conjunto de Teste	55
4.3.3 Generalização	56
4.3.3.1 Camadas Paralelas	58
4.3.3.2 Domo de Sal Homogêneo	61
4.3.3.3 Domo de Sal Heterogêneo sobre Reservatório	64
4.3.4 Método de Diferenças Finitas	67
4.4 Análise dos Resultados	75
4.4.1 Imagens sísmicas sem convolução	75
4.4.2 Imagens sísmicas com convolução	77
Conclusão	79
Referências	82
Apêndices	85
APÊNDICE A Aprendizado de Máquina	86
A.1 IoU	86
A.2 Focal Loss	86
A.3 Dice Loss	86
A.4 Sensitividade	86
A.5 Precisão	87

A.6	Engenharia de Características	87
A.7	Curva Logística	87
APÊNDICE B	Redes Neurais	88
B.1	Treinamento da Rede Neural	88
B.2	Convolução	88
B.3	Max Pooling	89
B.4	Convolução Transposta	90
B.5	Vanishing Gradient	90
B.6	Conexão Residual	90
B.7	Batch Normalization	90
B.8	Squeeze and Excitation	91
B.9	Atrous Spatial Pyramidal Pooling (ASPP)	91
B.10	Camadas de Atenção	92
B.11	UNET	92
B.12	MobileNetV2-UNET	93
B.13	RESUNET	93
B.14	RESUNET++	94
APÊNDICE C	Geofísica	96
C.1	Reflexão da Frente de Onda	96
C.2	Função Recursiva	96
C.3	Modelos de Subsuperfície	97
C.4	Convolução de imagens sísmicas	98

Introdução

A exploração dos recursos naturais presentes no subsolo é de extrema importância para a sociedade contemporânea. Muitas das reservas de recursos naturais que nossa sociedade consome estão contidas nele: metais, carvão, petróleo, gás, dentre outros. Conhecer o subsolo também é importante para viabilizar o desenvolvimento de novas técnicas de combate ao aquecimento global, como por exemplo, o armazenamento de gás carbônico e a geração de energia geotermal.

Uma vez que a observação direta de cada trecho de grandes áreas de interesse do subsolo é inviável, especialmente em alto mar onde o solo está muito distante para o emprego de técnicas de observação direta (perfuração), são utilizadas técnicas geofísicas para sondar essas regiões à distância. Uma das principais técnicas empregadas é o imageamento sísmico (KEAREY; BROOKS; HILL, 2013), que se assemelha a uma ultrassonografia do subsolo: ondas mecânicas são emitidas e ecoam nas camadas em subsuperfície. Detectores espalhados pela superfície da água ou do solo registram a amplitude das ondas mecânicas que chegam às suas posições. O processamento cuidadoso dos dados obtidos possibilita a caracterização e mapeamento das estruturas geológicas em subsuperfície. No segundo capítulo deste trabalho, iremos detalhar melhor o método.

A técnica enfrenta diversos desafios, sendo um deles a presença de reflexões múltiplas. Essas reflexões múltiplas consistem em frentes de onda que sofreram mais de uma reflexão antes de atingir os detectores, havendo reverberado dentro das muitas camadas do subsolo e da lâmina d'água (se houver uma) (Fig. 1). Essas ondas se manifestam na imagem sísmica como repetições das reflexões primárias, que são as frentes de onda que nos fornecem as informações que sabemos interpretar corretamente com as técnicas atuais, dificultando assim o processamento do sinal. Desse modo, é essencial conseguir eliminar de forma cuidadosa as reflexões múltiplas dos dados adquiridos para extrair a melhor informação possível da subsuperfície.

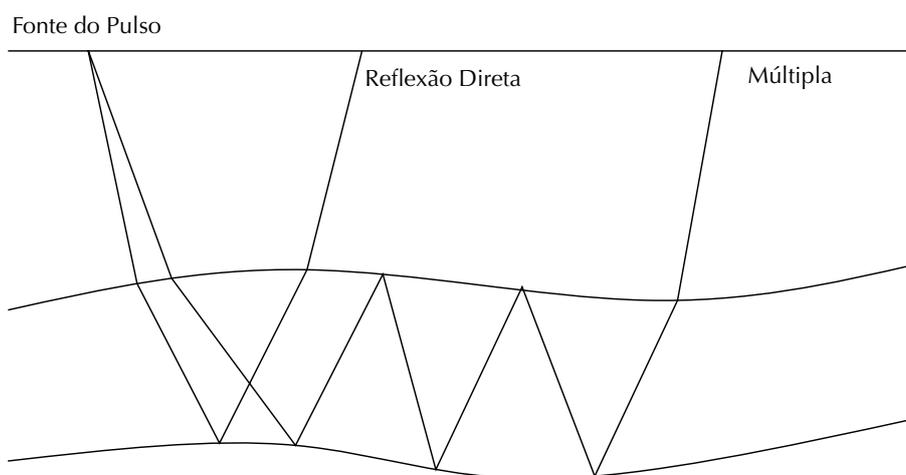


Figura 1 – Reflexão Direta e Reflexão Múltipla

Existem várias técnicas bem estabelecidas e de custo razoável para a remoção das múltiplas de superfície livre (aquelas nas quais a reverberação ocorre entre os refletores e a superfície livre da água ou do solo), tais como SRME (Surface-Related Multiple Elimination) (VERSCHUUR; BERKHOUT; WAPENAAR, 1992), EPSI (Estimating Primaries by Sparse Inversion) (GROENESTIJN; VERSCHUUR, 2009), ISS (Inverse Scattering Series) (CARVALHO; WEGLEIN; STOLT, 2005) e Transformada de Radon (HAMPSON, 2005). Mas a remoção de múltiplas internas (aquelas cuja reverberação ocorre nas camadas internas da subsuperfície) ainda é um grande desafio para a indústria (CGG, 2022). Técnicas como a de Marchenko (BROGGINI; SNIEDER; WAPENAAR, 2011), por exemplo, necessitam de um grande poder computacional, de uma ótima qualidade dos dados adquiridos e, ainda, de boas estimativas sobre as características das camadas presentes na subsuperfície. Recentemente, tem-se começado a investigar a remoção de múltiplas internas e de superfície livre por modelos de aprendizado de máquina. Os modelos são treinados a partir de exemplos reais ou sintéticos, dos quais foram removidas as múltiplas através do uso de técnicas tradicionais. Dentre as investigações recentes sobre o tema, destacamos três exemplos:

Wang et al. (WANG et al., 2022) treinaram uma RESUNET a partir de perfis reais, cujas múltiplas de superfície livre foram filtradas com técnicas tradicionais. Para isso, realizaram um processo de *data augmentation* utilizando um ruído de fundo e rotações dos dados. Em seu trabalho, porém, a rede treinada com dados reais de um local só classificou bem dados de locais próximos (geologia semelhante). Para aplicar a rede em locais com geologias diferentes seria preciso retreinar a rede.

Bugge et al. (BUGGE et al., 2021) tiveram uma outra proposta para o treinamento: criaram as reflexões primárias e múltiplas de superfície livre matematicamente, de forma aleatória. Tiveram bons resultados aplicando a rede assim treinada em dados reais, removendo as múltiplas de superfície livre com menos perda de sinal do que no método tradicional.

Bao et al. (BAO et al., 2022) treinaram uma UNET usando tanto dados simulados, como reais, os quais foram filtrados com técnicas tradicionais de remoção de múltiplas de superfície livre e internas. Assim como no trabalho de Wang et al., foi preciso retreinar a rede para cada geologia diferente.

Uma das limitações no treinamento de modelos de aprendizado de máquina usando como alvo dados obtidos pelas técnicas tradicionais de remoção de múltiplas é que esses modelos serão, no máximo, tão bons na remoção de múltiplas quanto as técnicas tradicionais. Se fosse possível obter dados perfeitamente limpos, por outro lado, poderíamos ambicionar obter modelos capazes de uma remoção de múltiplas melhor do que as técnicas tradicionais. Seria possível obter um conjunto de dados sintéticos simulados que já são gerados sem as múltiplas internas ou de superfície? É possível treinar um modelo com esses dados? Uma vez treinado, o quão bem o modelo se aplica a diferentes configurações geológicas?

Para responder a estas questões será necessário, primeiro, tentar criar um simulador capaz de gerar os dois tipos de imagem sísmica (com e sem reflexões múltiplas). O simulador deverá ser aplicado a diferentes modelos geológicos para produção de dados de treinamento. A primeira etapa deste trabalho, portanto, será gerar os dados de treinamento.

É preciso investigar se tal conjunto será suficiente para o treinamento, ou seja, se é possível obter um modelo capaz de distinguir múltiplas de primárias a partir do conjunto de informações que geramos. Esta será a segunda etapa desta investigação.

Para responder à terceira questão, será preciso verificar se o modelo assim treinado será capaz de generalizar seu aprendizado para situações distintas das que caracterizaram seu treinamento. Este será o último passo desta investigação.

Este trabalho está dividido em 4 partes. As duas primeiras fornecem uma breve exposição dos principais tópicos de Aprendizado de Máquina e de Imagem Sísmica relacionados a este trabalho. Em seguida, é detalhada a Metodologia adotada e os recursos computacionais utilizados e desenvolvidos. Por fim, são expostos e discutidos os Resultados dos experimentos. Nos apêndices podem ser encontradas explicações mais aprofundadas sobre alguns dos elementos abordados nesta dissertação.

1 Aprendizado de Máquina

Aprendizado de Máquina é um campo da Inteligência Artificial no qual não há um humano programando regras, ensinando ao computador uma lógica a ser seguida. Em vez disso, as regras para inferência são extraídas de um conjunto de dados de exemplo. Especificamente, no Aprendizado de Máquina Supervisionado, é usada uma técnica na qual se treina um modelo a partir de exemplos de entradas e suas respectivas respostas esperadas.

A intenção é que o modelo treinado com esse conjunto de dados seja capaz de gerar uma saída apropriada a partir de uma entrada inédita (RUSSEL; NORVIG, 2021). Por exemplo: usando um longo histórico de dados meteorológicos, pode-se tentar obter um modelo capaz de estimar, a partir dos dados meteorológicos das horas anteriores, as condições meteorológicas aproximadas nas horas seguintes.

Existem diversos tipos de modelos de Aprendizado de Máquina. Alguns são baseados nas chamadas Árvore de Decisão: a partir dos dados de treinamento o modelo aprende regras a serem aplicadas aos dados, de maneira a separá-los em diferentes classes. Outro exemplo é a Regressão Logística, em que uma curva logística (vide A.7) é ajustada aos dados, permitindo a separação de duas classes. Também existem modelos de Regressão Linear, em que um valor é estimado como função linear dos valores de entrada. Neste trabalho, empregaremos modelos de regressão e de classificação de pontos de imagens baseados em Redes Neurais Profundas.

1.1 Redes Neurais

Redes Neurais são bastante antigas, remontando à década de 1950. Elas se baseiam em complexas estruturas de conexões entre “neurônios artificiais” ligando uma entrada de dados a uma saída de dados, conforme apresentado na Figura 2. Esses neurônios artificiais nada mais são do que funções não-lineares, chamadas “funções de ativação”, aplicadas à soma X dos valores v_i fornecidos nas conexões de entrada de um neurônio (mais um valor constante chamado de “viés”¹) ponderados por seus respectivos pesos p_i . Uma das funções de ativação mais utilizadas é a ReLU:

$$ReLU(\sum v_i p_i) = ReLU(X) = \begin{cases} 0 & \text{se } X < 0, \\ X & \text{se } X \geq 0. \end{cases} \quad (1.1)$$

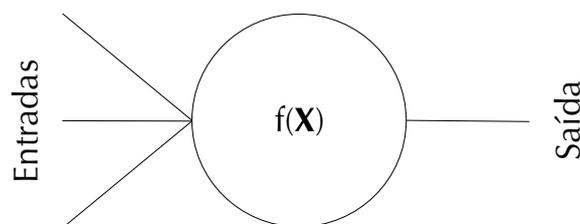


Figura 2 – Neurônio

¹ O viés pode ser visto como uma entrada formada de uma conexão com valor constante $v_0 = 1$ ponderada por um peso p_0 .

Normalmente os neurônios são organizados em sucessivas “camadas”: conjuntos de neurônios em paralelo, desconectados entre si, ligados aos neurônios de outras duas camadas (uma camada antecessora e uma camada sucessora) de maneira que os resultados de cada camada geram as entradas de cada neurônio da camada seguinte (Fig. 3). As funções de ativação escolhidas e as estruturas formadas pelas conexões entre os neurônios determinam uma “arquitetura”. A parametrização do modelo se dá no ajuste dos pesos que serão dados a cada uma das conexões que alimentam os neurônios.

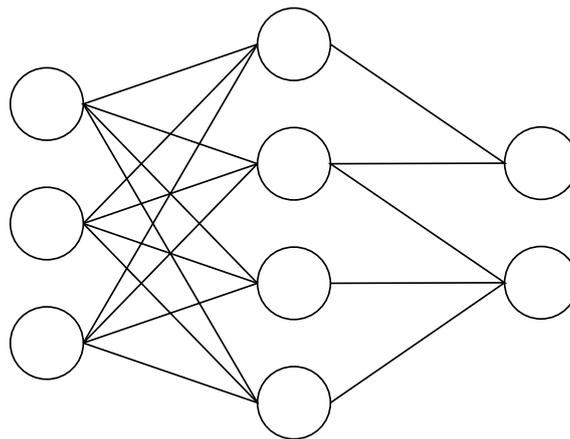


Figura 3 – Rede Neural

Um componente importante dos algoritmos de redes neurais é a medida de distância que vamos usar para quantificar o quão bem o modelo está funcionando. Ou seja: dada uma previsão y' , o quão distante ela está da resposta esperada y ? A essa medida chamamos “função de custo”². A função de custo é a pedra angular na construção de uma rede neural: durante o treinamento do modelo, o objetivo concreto do algoritmo será tentar minimizá-la. Ou seja, se queremos ter um bom resultado precisamos de uma função de custo que seja capaz de qualificar as saídas do modelo adequadamente.

Para minimizar a sua função custo (i.e. aprender), a Rede Neural passa por um processo iterativo. Inicialmente com pesos configurados em valores aleatórios, a rede é executada com algumas das entradas de treinamento. O resultado é comparado com os resultados esperados usando a função de custo. De acordo com a medida de erro obtida, os pesos são ajustados um pouco, num processo chamado de *back-propagation* (vide B.1). O ciclo se repete várias vezes para todo o conjunto de dados de treinamento até que, idealmente, obtenha-se um estado satisfatório dos pesos da rede neural. Os ciclos de treinamento são chamados de “épocas”.

Um ponto importante é que um estado satisfatório para os dados de treinamento não é, necessariamente, satisfatório para todos os conjuntos de dados possíveis. A depender da arquitetura empregada, da quantidade e da variedade de dados de treinamento disponível, é possível que o treinamento leve o modelo a um estado em que ele responde adequadamente aos dados de treinamento, mas não consegue “generalizar” este conhecimento para outros dados, um fenômeno chamado de *overfitting*. Uma das técnicas empregadas para monitorar esse risco é o uso de um conjunto de dados de “validação”: enquanto o conjunto de dados de

² Do termo em inglês “cost function”. Também chamada de “objective function” e de “loss function”.

treinamento é usado para ajustar continuamente o modelo, um outro conjunto, chamado de conjunto de validação, é usado para testar periodicamente a performance do modelo em dados não vistos no treinamento. Podemos esperar que no início do treinamento a performance melhore nos dois grupos (pois parte-se de um estado aleatório da rede). Se a partir de um certo ponto a performance continuar melhorando no conjunto de treinamento mas começar a piorar no conjunto de validação, interrompemos o treinamento e recuperamos o estado que melhor generaliza o aprendizado (vide Fig. 4).

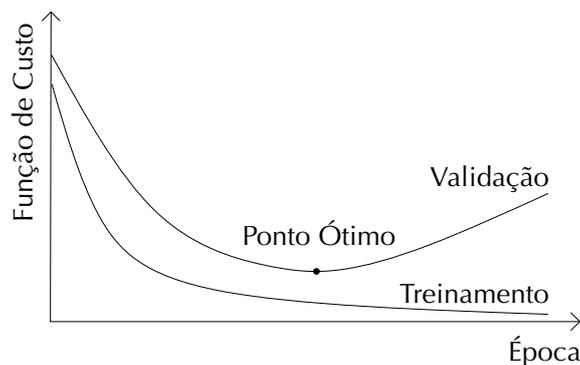


Figura 4 – Curvas de funções de erro dos conjuntos de treinamento e de validação.

1.2 Redes Neurais Profundas

Rede Neurais Profundas são aquelas em que há um grande número de camadas, nas quais a informação da entrada é abstraída em passos incrementais. A Rede Neural aprende quais as características mais importantes dos dados em diferentes níveis de abstração. (CHOLLET, 2021, pgs. 7 e 17) Isso traz um grande trunfo para essas Redes: não há necessidade de realizar a chamada “engenharia de características” (vide A.6) pois a própria Rede se encarrega disto. Conforme vimos na introdução, Redes Neurais Profundas tem sido empregadas com sucesso no tratamento de imagens sísmicas. (YORK et al., 2019) (WANG et al., 2022) (LI; WU; HU, 2022)

1.2.1 Redes Neurais Convolucionais

Neste trabalho iremos utilizar um tipo de Rede Neural Profunda chamado de Convolutional (em inglês, *Convolutional Neural Network* ou CNN), muito utilizado em imagens. Ele recebe esse nome porque pelo menos uma camada da rede realiza a convolução de matrizes, chamadas de filtros, pelos dados de entrada. Se o dado de entrada for uma imagem em escala de cinza, por exemplo, a Rede aprende a formar filtros para detectar características mais elementares (como bordas e texturas, por exemplo) nas primeiras camadas e para características mais complexas e informativas (como reconhecimento de formas complexas: rostos, objetos, expressões) nas camadas mais profundas. A função de ativação da última camada dependerá do tipo de saída desejado. Se a CNN estiver sendo usada para classificação de cada ponto da imagem entre duas classes, por exemplo, pode ser utilizada a Função Logística na última camada. Para a geração de uma imagem derivada da imagem de entrada, a última camada pode ter uma função de ativação linear.

Na última década tivemos uma rápida evolução das CNN, com arquiteturas cada vez mais sofisticadas sendo propostas sucessivamente em diferentes aplicações de classificação, onde uma imagem pode ser identificada por uma classe ou rótulo, ou em aplicações de segmentação semântica, onde cada pixel na imagem é identificada com uma classe ou rótulo. Dentre as diversas arquiteturas que surgiram é possível citar a rede convolucional UNET (B.11), que foi desenvolvida para segmentação de imagens médicas mas tem tido sua arquitetura modificada para ser aplicada em outras áreas do conhecimento, inclusive para dados sísmicos como em ((MULLER et al., 2022), (KLATT et al., 2022) e (JO et al., 2022)). Neste trabalho em específico, iremos usar UNET e algumas arquiteturas derivadas desta: a MobileNetV2-UNET (B.12), a RESUNET (B.13) e a RESUNET++ (B.14).

2 Imagem Sísmica

O método sísmico é utilizado para prospectar o subsolo a partir da análise da propagação de ondas mecânicas. Na aquisição sísmica de reflexão é usada uma série de detectores (geofones ou hidrofones, dependendo se o levantamento é realizado em terra ou sobre a água) alinhados, posicionados em intervalos regulares entre si, e uma fonte que irá gerar as ondas mecânicas. Parte da energia das ondas geradas pela fonte se propaga diretamente em direção aos detectores paralelamente à superfície livre. Na sísmica marinha, outra parte é parcialmente refletida na interface entre o mar e o assoalho marítimo. Uma terceira parte da energia das ondas penetra nas camadas inferiores do solo, onde são refletidas de volta à superfície ao passar por um contraste de impedância, no encontro entre diferentes tipos de rocha (C.1). A análise do tempo de viagem e da amplitude dessas componentes refletidas é que permite a inferência da geometria e do modelo de velocidades subsuperfície.

A partir de um modelo geológico (Fig. 5), podemos obter refletividades com e sem múltiplas como exemplo (Figs. 7 e 9).

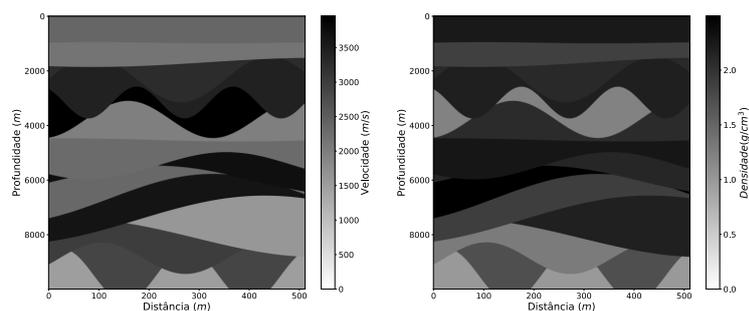


Figura 5 – Modelo geológico utilizado para gerar as imagens sísmicas de exemplo.

As reflexões primárias, ou diretas, são assim chamadas porque são o produto de apenas uma reflexão (Figs. 6, e 7):

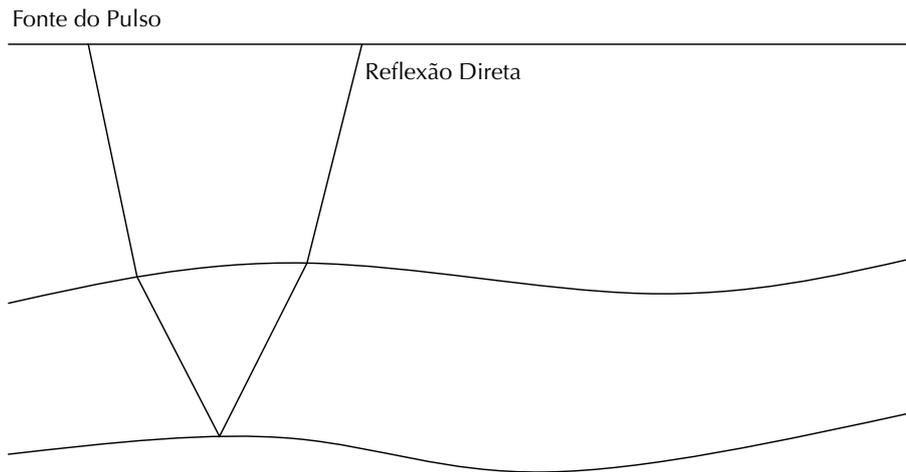


Figura 6 – Reflexão direta.

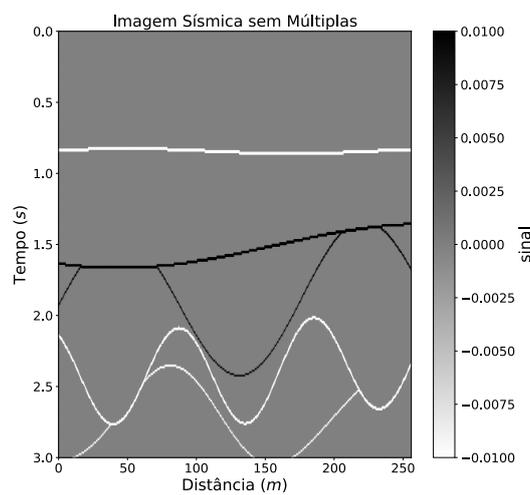


Figura 7 – Refletividade com reflexões diretas. Imagem gerada no simulador recursivo elaborado para este trabalho.

As reflexões múltiplas, por outro lado, são o resultado de reverberações entre as camadas (Figs. 8 e 9):

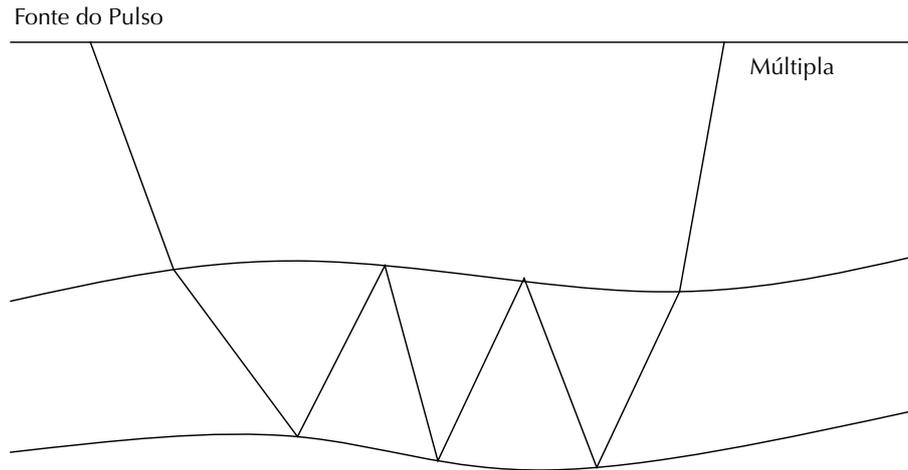


Figura 8 – Reflexão múltipla interna.

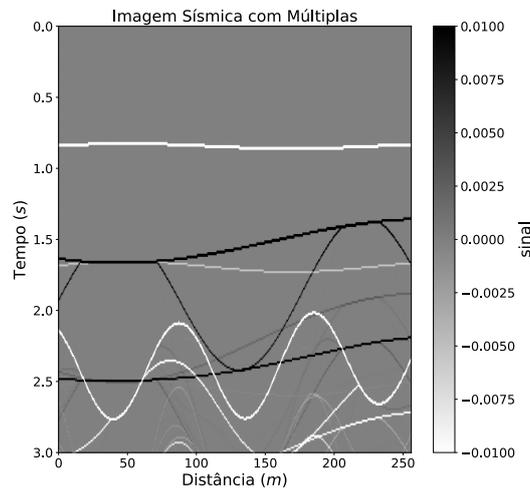


Figura 9 – Refletividade com reflexões diretas e múltiplas internas e de superfície livre. Imagem gerada no simulador recursivo elaborado para este artigo.

Note (Figs. 7 e 9) que o sinal das múltiplas se mistura com o das primárias na imagem sísmica.

Na maioria das técnicas de análise da imagem sísmica, as reflexões múltiplas não são aproveitadas para a investigação de subsuperfície: apenas o sinal das reflexões primárias é considerado (WEGLEIN, 2015). Desta forma, as múltiplas são consideradas um tipo de ruído e devem ser removidas para permitir análises corretas.

2.1 Simulação sísmica

A simulação de aquisição sísmica normalmente é feita pelo Método de Diferenças Finitas (MDF) (LANGTANGEN; LINGE, 2017), a partir da equação da onda. Entretanto,

nesse método não é possível separar automaticamente as reflexões primárias das múltiplas internas. Como desejamos obter imagens que já são geradas com e sem múltiplas internas e de superfície para o treinamento e avaliação quantitativa dos modelos de aprendizado de máquina, será preciso elaborar uma nova técnica capaz de gerar essas imagens.

Conseguimos fazer isso a partir de um método recursivo que simula a propagação unidimensional de um pulso (conforme apresentado em C.2). O pulso se desloca entre camadas do subsolo, as quais possuem velocidade de propagação de onda constante, e na interface entre elas é refletido e transmitido na direção vertical, gerando duas chamadas recursivas ao propagador de pulsos. Quando se deseja eliminar as múltiplas, basta programar o algoritmo para que não haja reflexão (para baixo) das frentes de onda que viajam para cima, apenas transmissão, eliminando as reverberações. Como só trata a propagação da frente de onda em uma dimensão, o método precisa ser aplicado uma vez para cada coluna do modelo. Para obter a refletividade de todo o modelo é preciso realizar uma simulação desta para cada posição horizontal, deslocando a fonte e o detector. O resultado desta técnica é semelhante ao que seria atingindo usando múltiplas aquisições de *offset* zero (ou seja, com a fonte do pulso sísmico e um único geofone tão próximos que podem ser considerados localizados no mesmo ponto) em cenários de camadas paralelas, a menos da convolução de uma *wavelet*. Aqui chamaremos esse algoritmo de “Simulador Recursivo”.



Figura 10 – Aquisição de Offset Zero

É importante notar também que em uma imagem sísmica real cada frente de onda possui uma assinatura contínua (*wavelet*) e não um pulso pontual, como o obtido no nosso simulador. Esta discrepância, no entanto, pode ser eliminada realizando a convolução das colunas da imagem gerada pelo simulador com uma assinatura de pulso (*wavelet*). Desta forma obtemos as imagens sísmicas a partir da refletividades.

3 Metodologia

O presente trabalho propõe investigar a remoção de múltiplas internas e de superfície por Redes Neurais Profundas (mais especificamente por redes do tipo UNET, MobileNetV2-UNET, RESUNET e RESUNET++) treinadas com dados gerados por um simulador capaz de gerar imagens sísmicas com e sem múltiplas, convoluídas com uma *wavelet* ou não.

Propositalmente, num primeiro momento não é empregada a convolução com uma *wavelet*: identificar as frentes de onda que chegam à superfície como pulsos pontuais nos permite tratar a questão como um problema de classificação para cada ponto da imagem sísmica (ou seja, as redes precisam identificar se uma reflexão primária está presente ou não no ponto). Essa abordagem também permite tratar todas as reflexões primárias igualmente, independentemente de sua amplitude, o que não é possível em um problema de regressão. Em um segundo momento, utilizaremos a convolução das imagens geradas pelo Simulador Recursivo com uma *wavelet*, obtendo um problema de regressão de cada ponto da imagem (ou seja, as redes precisam estimar o valor do ponto). Estes últimos modelos podem então ser aplicados, por exemplo, a imagens sísmicas obtidas por múltiplas simulações (uma para cada coluna da imagem) de aquisições de offset zero usando o MDF (Método de Diferenças Finitas).

Inicialmente, é necessário elaborar um simulador capaz de gerar dados sintéticos com e sem múltiplas. Como demonstrado na literatura (BUGGE et al., 2021), não é necessário que as imagens sejam resultados de uma simulação física para obter modelos capazes de tratar imagens sísmicas reais. Basta gerar dados com semelhança suficiente com uma imagem sísmica real para que tal modelo possa ser treinado. Desta forma, esperamos que, apesar de simular a propagação de onda em uma vertical, o nosso simulador recursivo possa ser útil para esse propósito.

Em seguida, é necessário elaborar um conjunto de configurações de subsuperfície, as quais devem ser bem diversas entre si. Para isso escolhemos uma parametrização utilizando senóides para separar as diferentes camadas geológicas¹. O período, fase, amplitude e profundidade de cada senóide são escolhidos aleatoriamente, assim como a velocidade e densidade de cada camada. A esses dados aplicamos o simulador recursivo, gerando dois conjuntos de imagens sísmicas de pulsos pontuais. O primeiro conjunto inclui tanto as reflexões primárias como as múltiplas internas e de superfície livre. O segundo conjunto contém apenas as reflexões primárias. Deste segundo conjunto obtemos um conjunto de máscaras binárias identificando, para cada ponto das imagens, se uma reflexão primária passa ou não por ele. As imagens com convolução são obtidas convoluindo o primeiro e o segundo conjuntos com uma *wavelet*.

Os conjuntos de imagens sísmicas são então divididos em subconjuntos de treino, validação e teste. Os diferentes modelos de Rede Neural são alimentados com os conjuntos de treino. Os conjuntos de validação são usados a cada época do treinamento para avaliar a capacidade de generalização do modelo em treinamento. Para cada modelo, a configuração de

¹ Escolhemos usar senóides pois elas podem parecer desde dobramentos até uma superfície quase plana, dependendo de seus parâmetros.

pesos que melhor atuar no conjunto de validação é escolhida como sua configuração de pesos ótima. Os dados de teste são usados para realizar comparações entre os diferentes modelos treinados.

Para as imagens sem convolução devem ser treinados diferentes modelos de classificação (ou segmentação semântica) baseados na arquitetura RESUNET++. Como a classificação é binária (ou seja, para cada ponto há ou não há uma reflexão primária presente), a última camada da rede é uma função sigmóide (A.7). Em particular, devem ser investigados os efeitos da adição de uma camada de Batch Normalization no início da Rede, do número de filtros usados, do emprego de diferentes métricas como função de custo, como MSE (*Mean Squared Error* – erro quadrático médio) (DEKKING et al., 2005), *Binary CrossEntropy* (MIRIYALA et al., 2021) e *Focal Loss* (LIN et al., 2017), e do tamanho de conjunto de treinamento. Além disso, também são treinadas redes baseadas nas arquiteturas UNET, MobileNetV2-UNET e RESUNET. Já para as imagens com convolução, a última camada da rede é uma função linear e devem ser treinados modelos de regressão baseados nas parametrizações mais bem sucedidas (para imagens sem convolução) das arquiteturas usadas nos modelos de classificação, sempre com a métrica MAE (*Mean Absolute Error* – erro absoluto médio) como função de custo.

A medida principal para os experimentos sem convolução será a Dice (DICE, 1945), sendo também calculadas a Interseção sobre União (IoU em inglês) (MURPHY, 1996), Sensibilidade (*recall*) e Precisão (POWERS, 2008), para fins de comparação. Nos experimentos com convolução, usaremos a métrica MAE.

Para avaliar a capacidade de generalização das Redes Neurais para dados com características diferentes daqueles usados em seu treinamento, serão realizadas avaliações de desempenho em 3 cenários de subsuperfície criados manualmente: camadas paralelas, domo de sal homogêneo e domo de sal heterogêneo sobre reservatório. Estes cenários também serão utilizados em simulações MDF de aquisições de offset zero, a fim de visualizar a aplicação dos modelos de regressão a dados mais próximos da realidade.

As simulações MDF serão realizadas com a biblioteca Devito, usando uma fonte que emite um único pulso Ricker de 25Hz em um grid de segunda ordem com 1001 pontos de altura e 1001 pontos de largura, com 5 metros de distância entre cada ponto. Será usada uma zona de amortecimento de 1000 pontos ao redor do grid, para diminuição dos efeitos de borda. As imagens de offset zero serão geradas repetindo a simulação do grid inteiro para cada coluna do grid, com a fonte e o receptor colocados naquela coluna, e depois colocando lado a lado o resultado de cada coluna.

Para avaliar o impacto de diferentes parâmetros de treinamento nos resultados, serão realizados 3 treinamentos rápidos² em cada possível combinação de um conjunto de parâmetros (Arquitetura, número de filtros na primeira camada de convolução, uso de camada de *Batch Normalization*, tamanho do conjunto de treinamento e função de custo). Para cada modelo assim treinado, mediremos a métrica Dice (para modelos treinados com dados sem convolução) ou a métrica MAE (para modelos treinados com dados com convolução) sobre o conjunto de teste, sobre um exemplo do conjunto de teste e sobre os 3 cenários.

² Nos treinamentos rápidos, limitados a 100 épocas, usamos parâmetros de treinamento mais agressivos do que nos treinamentos dos modelos listados anteriormente.

4 Experimentos Computacionais e Resultados

4.1 Redes neurais

Para avaliação da remoção de múltiplas sem convolução serão utilizadas 10 diferentes arquiteturas, baseadas na RESUNET++, RESUNET, UNET e MobileNetV2-UNET. Todas as redes, a menos que descrito em contrário, usam a *Focal Loss* como métrica de treinamento e tem a sigmoide como função de ativação da última camada:

RESUNET++ 32: A arquitetura RESUNET++ com 4 camadas de codificação e decodificação, sendo a primeira camada de codificação com 32 filtros.

RESUNET++ 32BN: A mesma arquitetura, mas com uma camada extra de *Batch Normalization* na entrada da rede.

RESUNET++ 32BN MSE: A mesma arquitetura da RESUNET++ 32BN, mas usando o erro médio quadrado (MSE em inglês) como métrica de treinamento.

RESUNET++ 32BN BCE: A mesma arquitetura da RESUNET++ 32BN, mas usando a entropia binária cruzada (BCE em inglês) como métrica de treinamento.

RESUNET++ 32BN 10K: A mesma arquitetura da RESUNET++ 32BN, mas usando 10.000 exemplos no treinamento, ao invés de 100.000.

RESUNET++ 16BN: A mesma arquitetura, mas com apenas a metade do número de filtros em suas camadas.

RESUNET 32: A arquitetura RESUNET com 6 camadas de codificação e decodificação, sendo a primeira camada de codificação com 32 filtros.

RESUNET 32BN: A mesma arquitetura, mas com uma camada extra de *Batch Normalization* na entrada da rede.

UNET 32: A arquitetura UNET com 6 camadas de codificação e decodificação, sendo a primeira camada de codificação com 32 filtros.

UNET 32BN: A mesma arquitetura, mas com uma camada extra de *Batch Normalization* na entrada da rede.

MobileNetV2-UNET: O codificador pré-treinado da arquitetura MobileNet v2 com o decodificador da UNET, implementados pela biblioteca *Segmentation Models*.

4.2 Imagens sísmicas sem convolução (Refletividade)

4.2.1 Geração de sistemas de subsuperfície

Conforme explicado anteriormente, precisamos elaborar um conjunto de modelos de subsuperfície. Para isso elaboramos um código (C.3) que gera modelos de subsuperfície a partir de parâmetros aleatórios, o que nos permite obter uma rica variedade de sistemas, bastante distintos entre si (Fig. 11). Durante a produção dos dados, são gerados os sistemas de subsuperfície, os quais posteriormente são separados em um conjunto de treinamento, teste e validação:

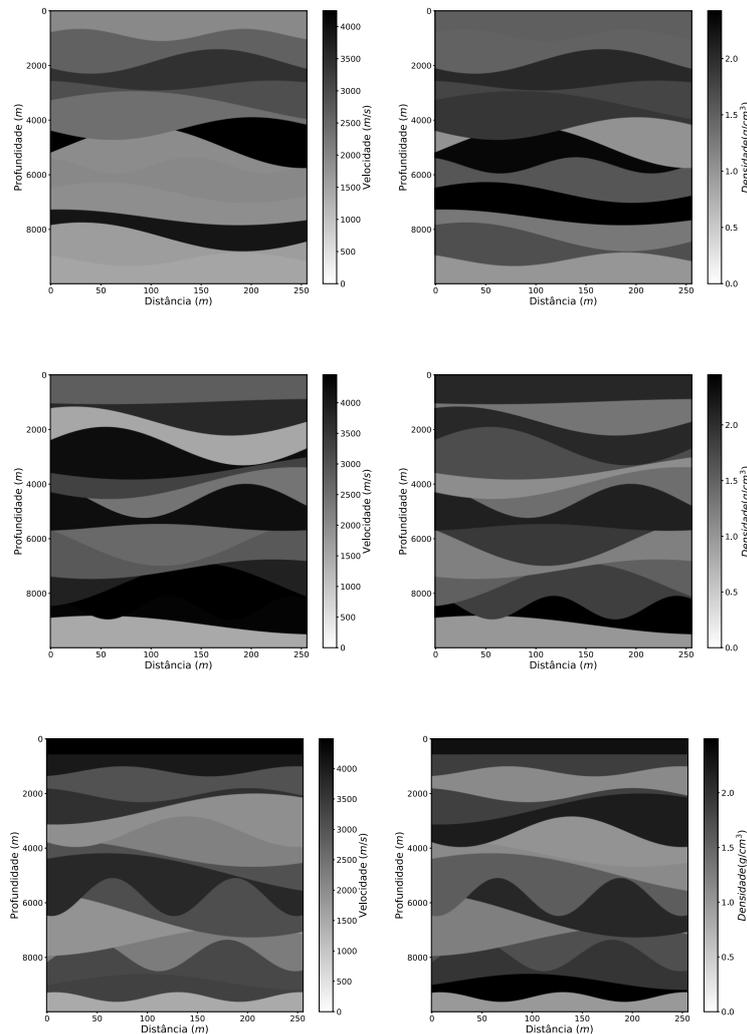


Figura 11 – Gráficos de velocidade e densidade de amostras de sistemas de subsuperfície de treinamento gerados aleatoriamente.

4.2.2 Conjunto de teste

Com o conjunto de teste podemos avaliar e comparar os modelos treinados (Fig. 12).

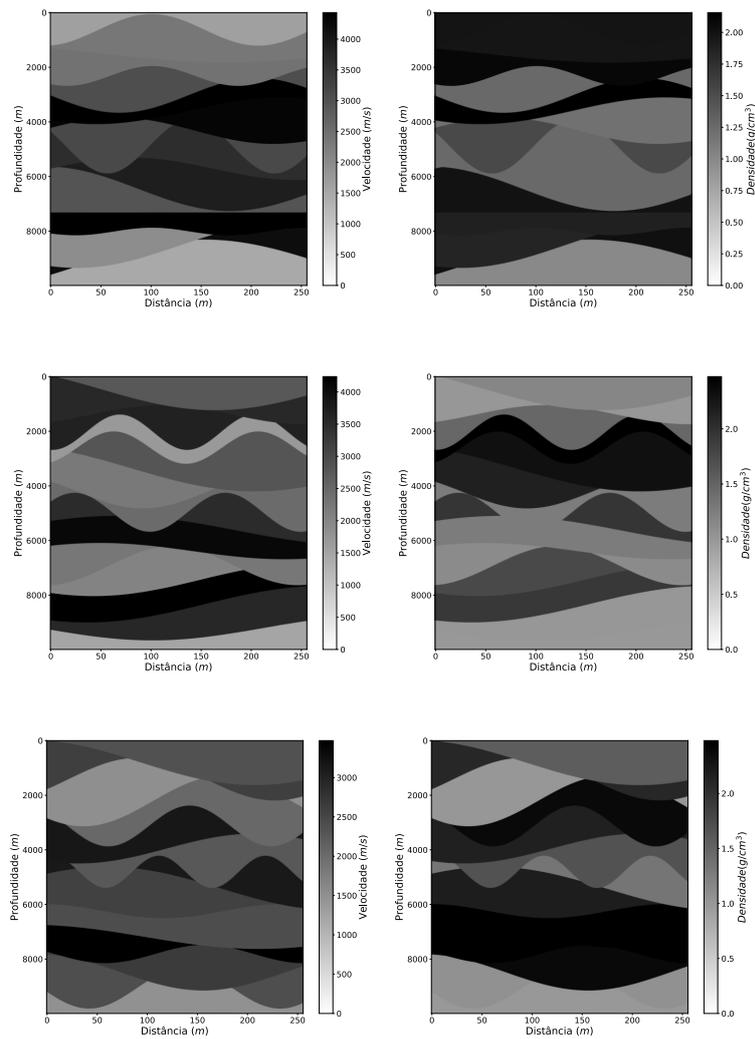


Figura 12 – Amostras dos sistemas subsuperfície geradores do conjunto de teste.

Na Tabela 1 podemos ver que os seis modelos melhor avaliados tem resultados muito próximos entre si e muito próximos a 1, o que indica uma forte capacidade de generalização para modelos de subsuperfície não vistos no treinamento, mas obtidos pelos mesmo código de geração aleatória de modelos de subsuperfície. Podemos observar ainda o forte impacto da diminuição do conjunto de treinamento no modelo de rede neural RESUNET++ 32BN 10K, com desempenho muito abaixo do modelos RESUNET++ 32BN. A camada de *Batch Normalization* também é claramente influente nos resultados, estando presente nos melhores modelos (com exceção do modelo MobileNetV2-UNET, que não possui este parâmetro). Por fim, observamos um impacto significativo no desempenho dos modelos que não usaram a *Focal Loss* como métrica de treinamento.

Modelo	DC
RESUNET++ 32BN	0,9879
MobileNetV2-UNET	0,9861
RESUNET++ 16BN	0,9854
UNET 32BN	0,9845
RESUNET 32BN	0,9832
RESUNET++ 32BN MSE	0,9517
UNET 32	0,9263
RESUNET 32	0,9087
RESUNET++ 32	0,8915
RESUNET++ 32BN BCE	0,8371
RESUNET++ 32BN 10K	0,7275

Tabela 1 – Tabela de resultados do conjunto de teste, ordenada pelo *Dice Coefficient*.

Para o conjunto de todos os modelos criados nos treinamentos rápidos, obtivemos as estatísticas para a métrica DC sobre o conjunto de teste (Fig. 13, Tab. 2 e Fig. 14). Em particular, nos interessa observar os valores máximos atingidos pelos modelos de cada partição das amostras, pois eles indicam o potencial de desempenho para cada parâmetro.

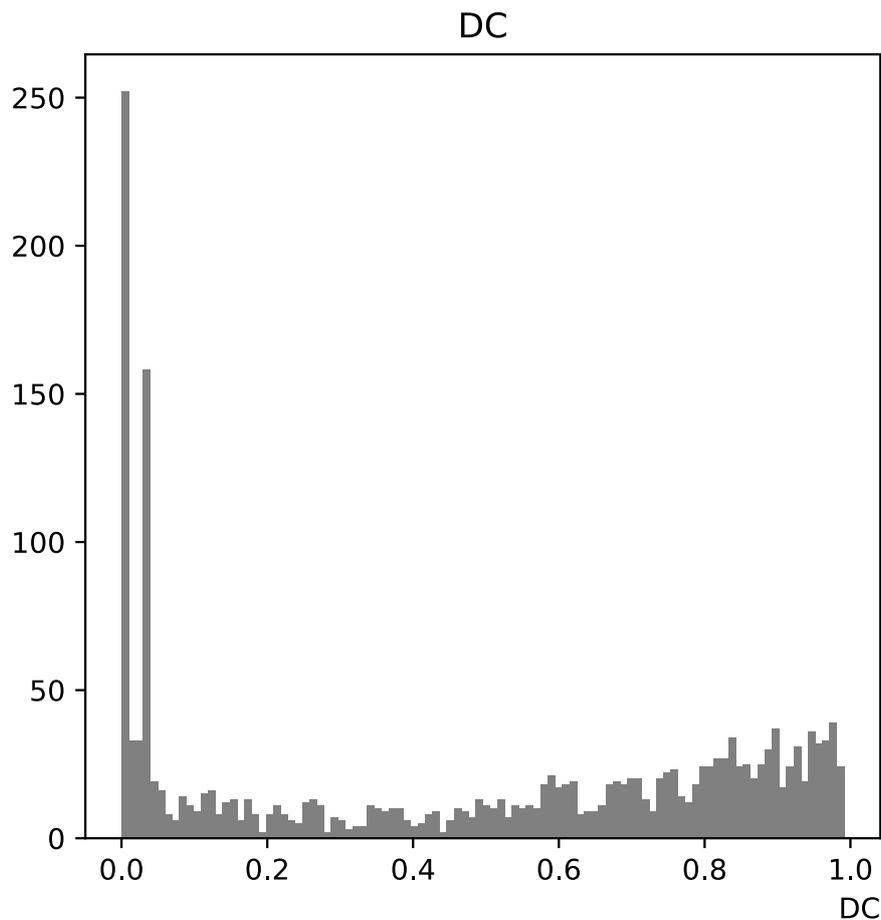


Figura 13 – Histograma da métrica DC dos modelos criados nos treinamentos rápidos aplicados no conjunto de teste.

Na Tabela 2, vemos uma diferença pequena no desempenho máximo entre os diversos parâmetros, com exceção das métricas e dos modelos treinados com 1.000 exemplos.

Dos histogramas de cada parâmetro (Fig. 14), podemos identificar mais alguns indícios, por exemplo, da dificuldade de obter bons resultados com as métricas BCE e MAE e da importância do tamanho dos conjuntos de treinamento. Também podemos verificar que o histograma dos modelos baseados na RESUNET++ tem uma concentração central, o que sugere que esta arquitetura tem uma forte sensibilidade a combinações subótimas de parâmetros. Note que, por não ser possível variar os parâmetros Base e *Batch Normalization* para a arquitetura MobileNetV2-UNET, há uma razão 1/4 do total de combinações possíveis de parâmetros para esta arquitetura em relação às demais, o que se reflete na área de seu histograma.

	Arquitetura	Média	Desvio Padrão	Mínimo	25%	Mediana	75%	Máximo
Arquitetura	MobileNetV2-UNET	0,4318	0,4162	0,0000	0,0000	0,5666	0,8364	0,9871
	RESUNET	0,5860	0,3556	0,0000	0,2046	0,7557	0,8710	0,9804
	RESUNET++	0,4454	0,3393	0,0000	0,0359	0,5422	0,7081	0,9916
	UNET	0,4043	0,3574	0,0000	0,0393	0,2945	0,7887	0,9884
Base	16	0,4710	0,3575	0,0000	0,0480	0,5218	0,8231	0,9867
	32	0,4861	0,3609	0,0000	0,0524	0,5745	0,8274	0,9916
Batch Normalization	Não	0,4770	0,3528	0,0000	0,0500	0,5390	0,8176	0,9914
	Sim	0,4801	0,3657	0,0000	0,0459	0,5555	0,8322	0,9916
Métrica	BCE	0,4008	0,3527	0,0000	0,0342	0,3884	0,7597	0,9314
	Focal Loss	0,7015	0,3090	0,0000	0,5548	0,8393	0,9451	0,9916
Tamanho Treino (K)	MAE	0,3226	0,3128	0,0000	0,0340	0,2354	0,6686	0,8860
	1	0,2823	0,2634	0,0000	0,0350	0,2054	0,5186	0,8734
	10	0,4906	0,3326	0,0000	0,1270	0,5893	0,7771	0,9689
	50	0,5612	0,3770	0,0000	0,0493	0,7522	0,8828	0,9876
Todos	100	0,5657	0,3944	0,0000	0,0355	0,7727	0,9030	0,9916
	Parâmetros	0,4750	0,3640	0,0000	0,0371	0,5502	0,8274	0,9916

Tabela 2 – Estatísticas da métrica DC para subconjuntos dos dos modelos criados nos treinamentos rápidos, agrupados pelos parâmetros de treinamento, quando aplicados ao conjunto de teste.

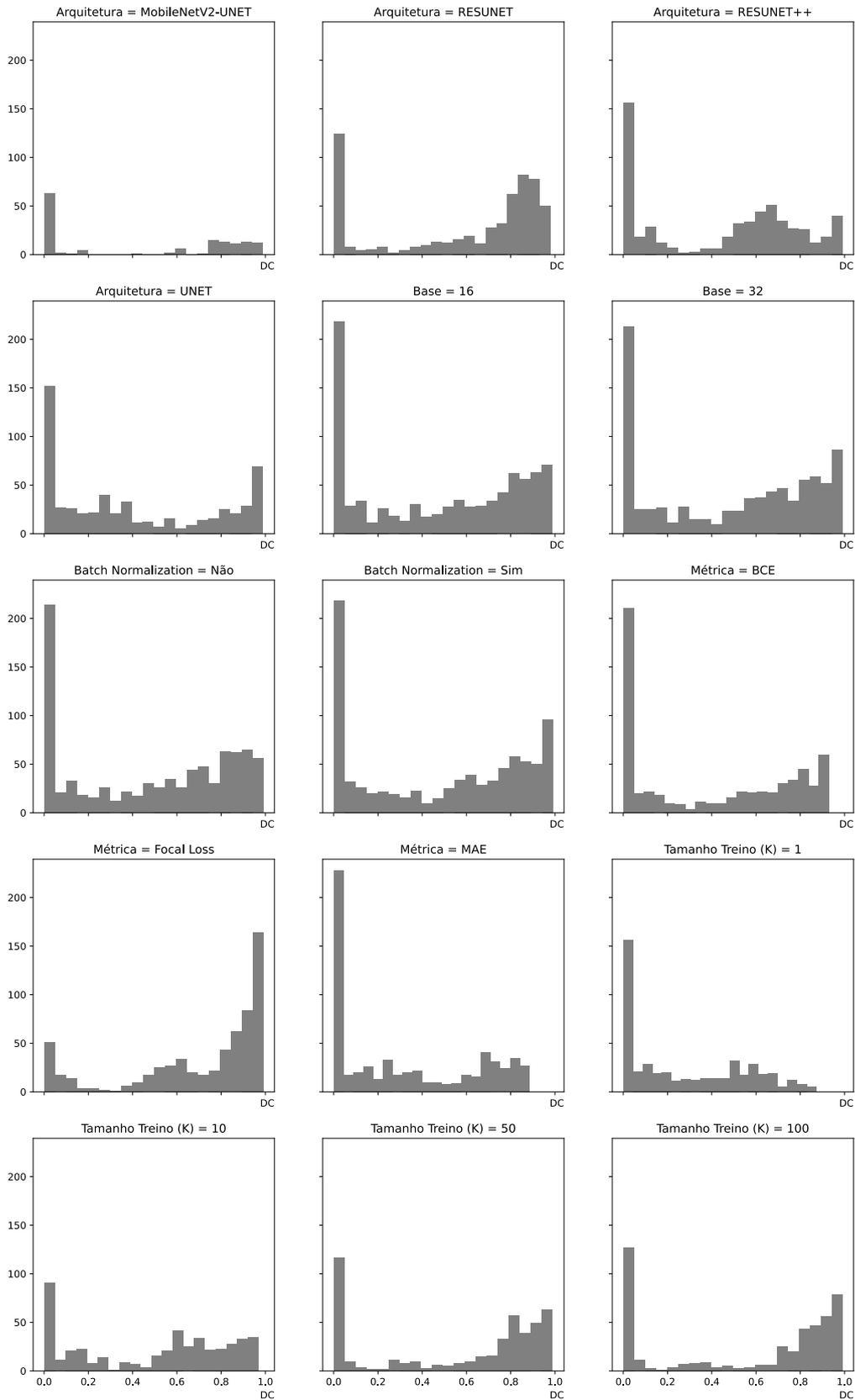


Figura 14 – Histogramas da métrica DC para subconjuntos dos modelos criados nos treinamentos rápidos, agrupados pelos parâmetros de treinamento, quando aplicados ao conjunto de teste.

4.2.2.1 Elemento do Conjunto de Teste

Como exemplo, apresentamos as configurações subsuperfície (densidade e velocidade) de um elemento retirado do conjunto de teste (Fig. 15) com sua respectivas imagens sísmicas com e sem a presença de múltiplas (Fig. 16). Nas imagens sísmicas podemos observar a presença das múltiplas, como esperado, e notar como um número muito reduzido de todos os sinais recebidos é oriundo de reflexões diretas. Em seguida, temos o resultado da aplicação de diferentes modelos de aprendizado de máquina sobre este exemplo (Fig. 17 a Fig. 25), onde podemos verificar visualmente a saída prevista pelo modelo, o tratamento da saída prevista pelo modelo para formar uma máscara (a partir da seleção apenas dos pontos com valor maior que 0,5) e a diferença entre esta máscara e a máscara-alvo ¹. Embora os dados apresentados em 4.2.2 sejam mais significativos, uma vez que tratam de todo o conjunto de teste, é importante poder visualizar a aplicação dos diferentes modelos a uma amostra, bem como saber como estes resultados são avaliados pela métrica DC (Tab. 3). Observe que o desempenho dos modelos na amostra não repetem exatamente a ordem observada nos resultados para todo o conjunto de teste (Tab. 1), mas que a distinção entre o grupo dos modelos com melhor performance (RESUNET++ 32BN, MobileNetV2-UNET, RESUNET++ 16BN, UNET 32BN, RESUNET 32BN e RESUNET++ 32BN MSE) se manteve preservada:

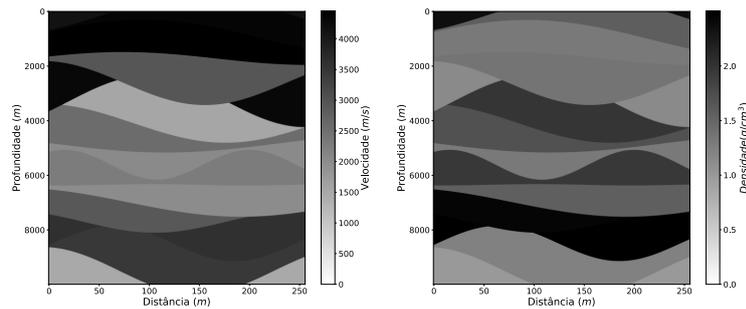


Figura 15 – Amostra de sistema subsuperfície do conjunto de teste.

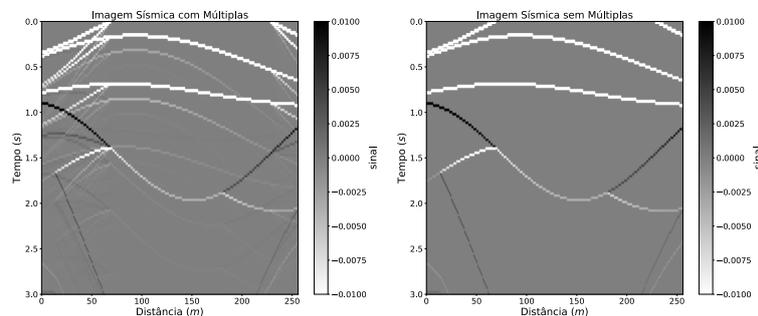


Figura 16 – Imagens sísmicas com e sem múltiplas de elemento do conjunto de teste.

¹ A máscara-alvo é obtida a partir da imagem sísmica sem múltiplas internas, sendo selecionados todos os pontos com valor diferente de 0, ou seja, todos os pontos em que ocorre uma reflexão primária, independentemente de seu valor absoluto.

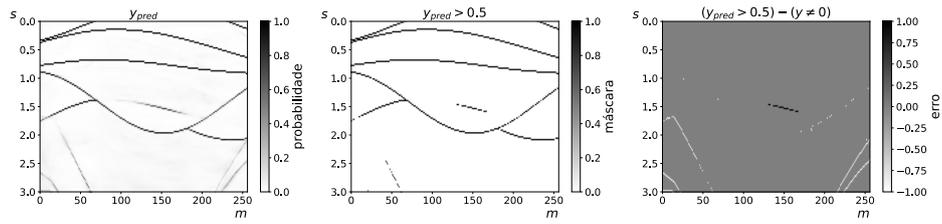


Figura 17 – RESUNET++ 32 aplicada em elemento do conjunto de teste.

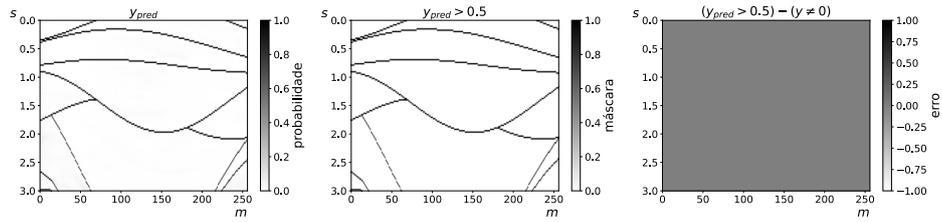


Figura 18 – RESUNET++ 32BN aplicada em elemento do conjunto de teste.

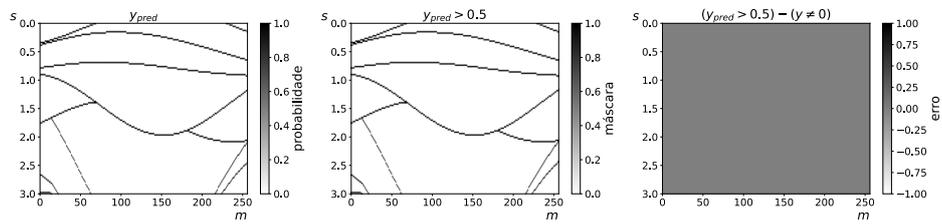


Figura 19 – RESUNET++ 32BN MSE aplicada em elemento do conjunto de teste.

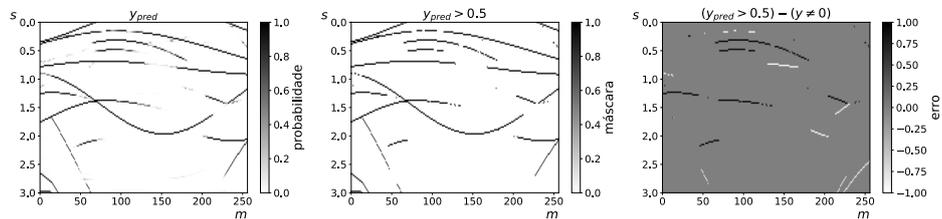


Figura 20 – RESUNET++ 32BN BCE aplicada em elemento do conjunto de teste.

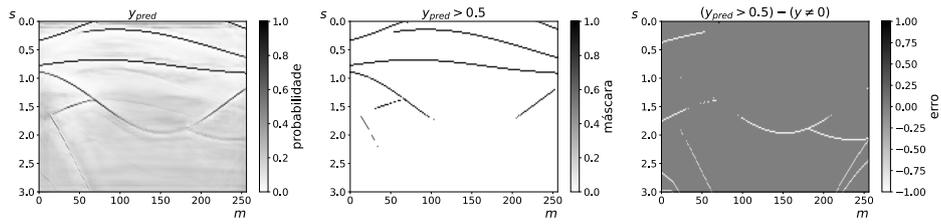


Figura 21 – RESUNET++ 32BN 10K aplicada em elemento do conjunto de teste.

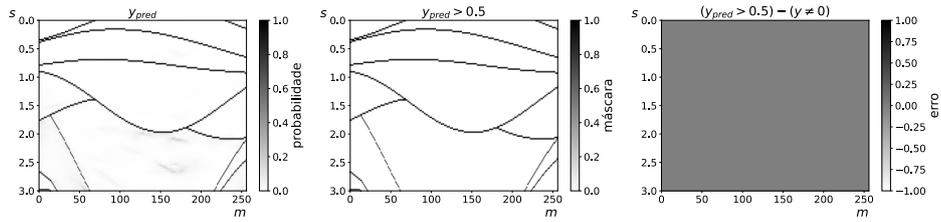


Figura 22 – RESUNET++ 16BN aplicada em elemento do conjunto de teste.

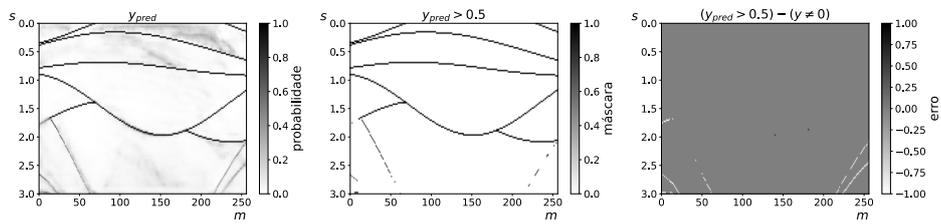


Figura 23 – RESUNET 32 aplicada em elemento do conjunto de teste.

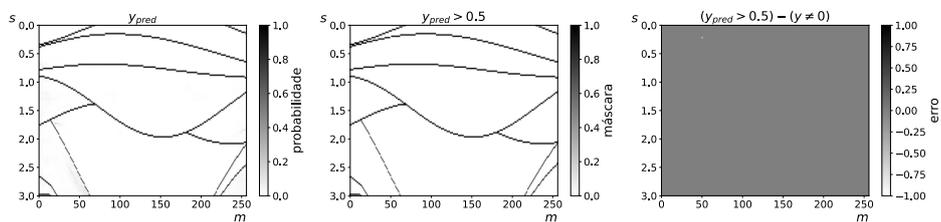


Figura 24 – RESUNET 32BN aplicada em elemento do conjunto de teste.

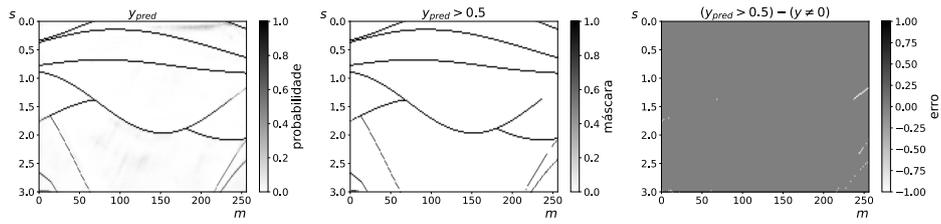


Figura 25 – UNET 32 aplicada em elemento do conjunto de teste.

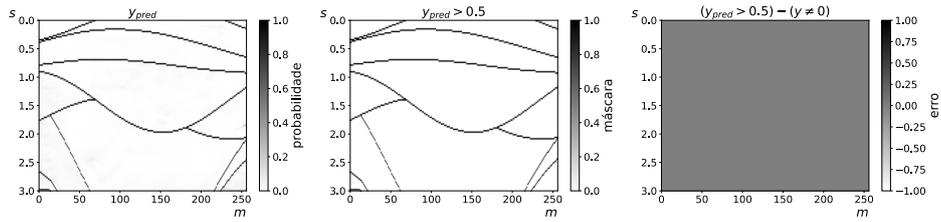


Figura 26 – UNET 32BN aplicada em elemento do conjunto de teste.

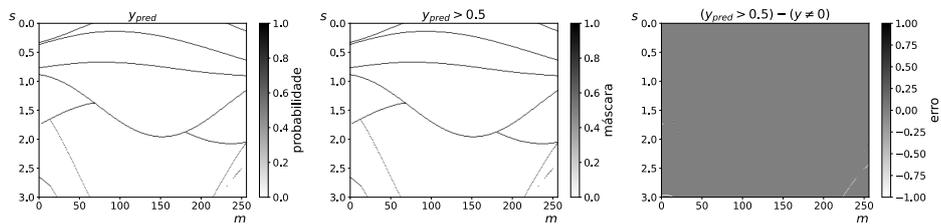


Figura 27 – MobileNetV2-UNET aplicada em elemento do conjunto de teste.

Modelo	DC
RESUNET++ 32BN	0,9944
UNET 32BN	0,9933
RESUNET++ 32BN MSE	0,9907
RESUNET 32BN	0,9832
RESUNET++ 16BN	0,9816
MobileNetV2-UNET	0,9688
RESUNET 16BN	0,9666
UNET 32	0,8685
RESUNET++ 32	0,8372
RESUNET++ 32BN BCE	0,8621
RESUNET++ 32BN 10K	0,8014
RESUNET 32	0,7978

Tabela 3 – Medida numérica da aplicação dos diferentes modelos a um exemplo do conjunto de teste.

A Tabela 4 ilustra, para essa amostra, nossas observações anteriores sobre a escolha de métricas, bem como sobre o uso de conjuntos de treinamento menores.

Arquitetura	Média	Desvio Padrão	Mínimo	25%	Mediana	75%	Máximo
MobileNetV2-UNET	0,4610	0,4438	0,0000	0,0000	0,6200	0,8981	0,9995
RESUNET	0,6248	0,3772	0,0000	0,2384	0,8428	0,9094	0,9995
RESUNET++	0,5080	0,3853	0,0000	0,0314	0,6500	0,8604	1,0000
UNET	0,4208	0,3680	0,0000	0,0320	0,3167	0,8135	0,9995
Base	0,5088	0,3845	0,0000	0,0341	0,6104	0,8793	1,0000
Batch Normalization	0,5269	0,3872	0,0000	0,0342	0,6527	0,8877	0,9998
	0,5122	0,3776	0,0000	0,0340	0,6118	0,8692	0,9998
Sim	0,5234	0,3940	0,0000	0,0351	0,6526	0,8911	1,0000
BCE	0,4378	0,3856	0,0000	0,0234	0,4630	0,8450	0,9623
Focal Loss	0,7364	0,3178	0,0000	0,6117	0,8744	0,9632	1,0000
MAE	0,3661	0,3627	0,0000	0,0280	0,2563	0,8256	0,9362
Tamanho Treino (K)	0,3159	0,3060	0,0000	0,0305	0,2296	0,5913	0,9281
1	0,5471	0,3645	0,0000	0,1478	0,7248	0,8581	0,9936
10	0,5976	0,3995	0,0000	0,0319	0,8445	0,9178	1,0000
50	0,5932	0,4146	0,0000	0,0308	0,8668	0,9442	0,9998
100	0,5135	0,3907	0,0000	0,0315	0,6271	0,8853	1,0000
Parâmetros							

Tabela 4 – Estatísticas da métrica DC para subconjuntos dos modelos criados nos treinamentos rápidos, agrupados pelos parâmetros de treinamento, quando aplicados a um exemplo do conjunto de teste.

4.2.3 Generalização

A fim de avaliar a capacidade de generalização dos diferentes modelos treinados, além do conjunto de teste serão utilizados 3 cenários geológicos diferentes: camadas paralelas (Fig. 28), domo de sal homogêneo (Fig. 29) e domo de sal heterogêneo com um reservatório abaixo do mesmo (Fig. 30). Estes cenários serão preparados no nosso Simulador Recursivo para avaliar nossos modelos de remoção de múltiplas.

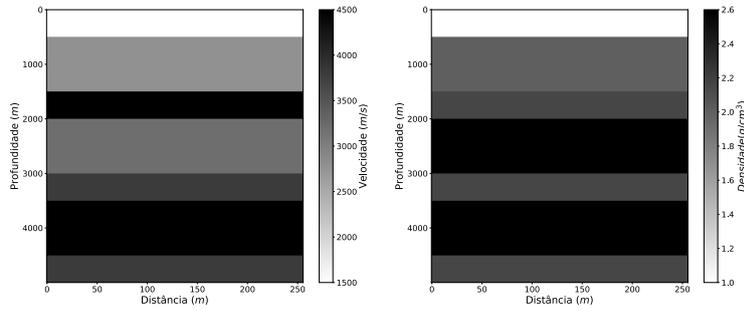


Figura 28 – Sistema de camadas paralelas

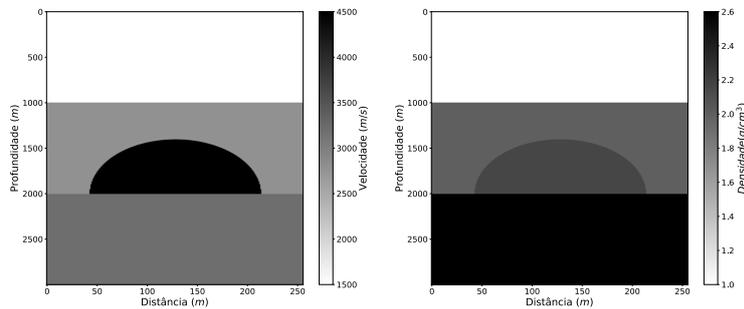


Figura 29 – Sistema de domo de sal homogêneo

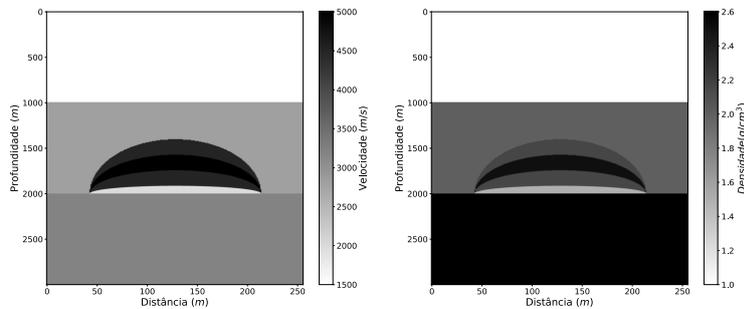


Figura 30 – Sistema de domo de sal heterogêneo sobre reservatório

Para as diferentes camadas de subsuperfície foram consideradas velocidades entre 2.800 e 4.500m/s e densidades entre 2 e 2.6g/cm³. Para o sal, foram consideradas velocidades de 4.500 a 5.000m/s e densidades de 2.16 a 2.5g/cm³. Para o reservatório, usamos a velocidade de 2.400m/s e densidade de 1.6g/cm³.

As imagens sísmicas vistas nas figuras 31, 43 e 55 mostram a simulação recursiva aplicada a estes sistemas geológicos.

4.2.3.1 Camadas Paralelas

Nas Figuras 32 a 42 podemos ver a aplicação dos modelos ao cenário de placas paralelas. Pode-se observar o ótimo resultado dos modelos RESUNET++ 32BN e RESUNET 32.

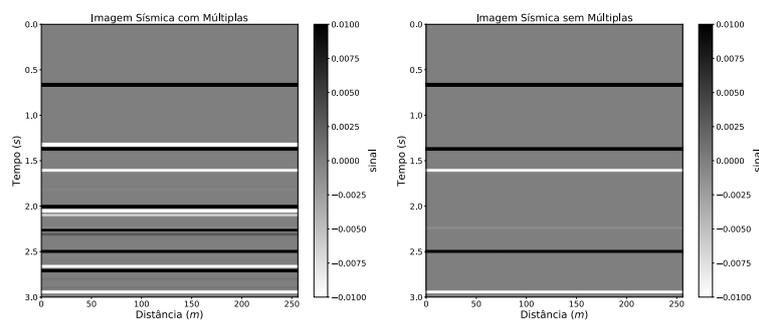


Figura 31 – Imagens sísmicas com e sem múltiplas do sistema de camadas paralelas

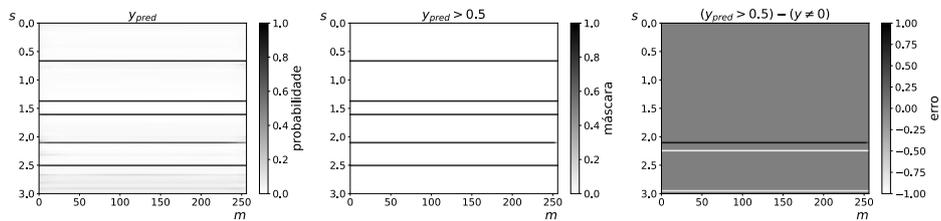


Figura 32 – RESUNET++ 32 aplicada no cenário de camadas paralelas

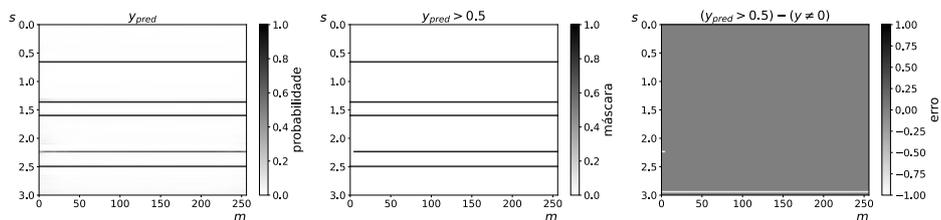


Figura 33 – RESUNET++ 32BN aplicada no cenário de camadas paralelas

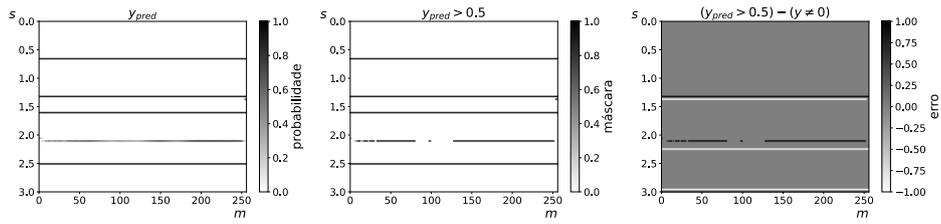


Figura 34 – RESUNET++ 32BN MSE aplicada no cenário de camadas paralelas

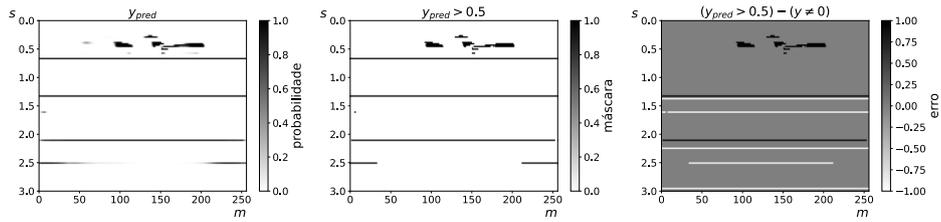


Figura 35 – RESUNET++ 32BN BCE aplicada no cenário de camadas paralelas

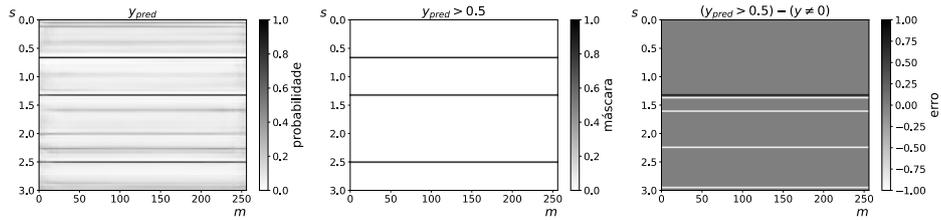


Figura 36 – RESUNET++ 32BN 10K aplicada no cenário de camadas paralelas

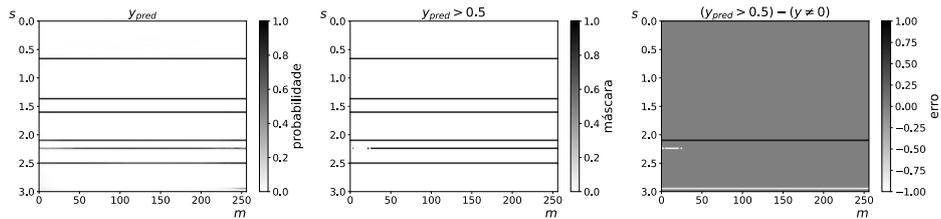


Figura 37 – RESUNET++ 16BN aplicada no cenário de camadas paralelas

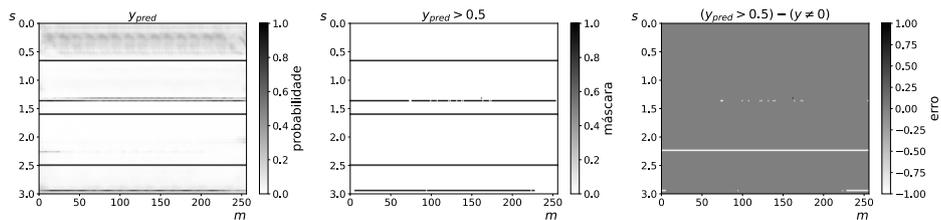


Figura 38 – RESUNET 32 aplicada no cenário de camadas paralelas

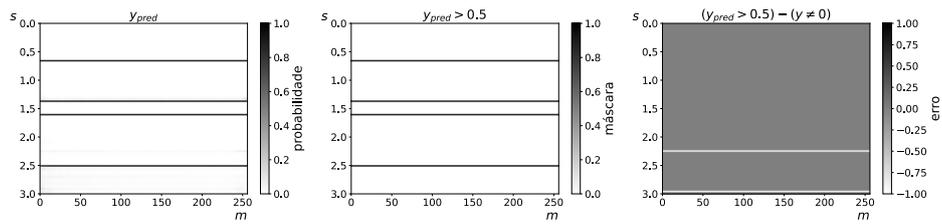


Figura 39 – RESUNET 32BN aplicada no cenário de camadas paralelas

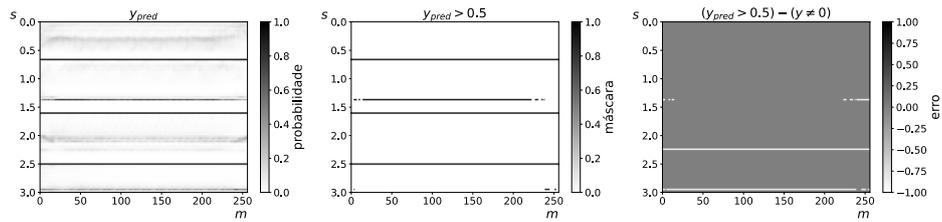


Figura 40 – UNET 32 aplicada no cenário de camadas paralelas

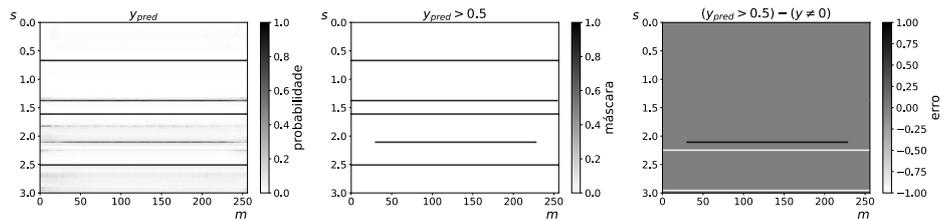


Figura 41 – UNET 32BN aplicada no cenário de camadas paralelas

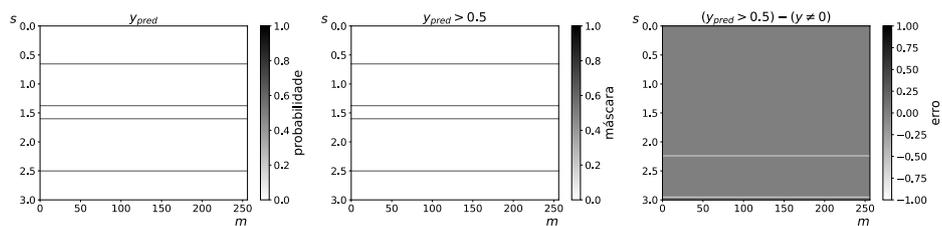


Figura 42 – MobileNetV2-UNET aplicada no cenário de camadas paralelas

A métrica DC para este cenário (Tab. 5) nos permite fazer comparações com o desempenho dos modelos no conjunto de teste (Tab. 1). Primeiramente é preciso observar que os resultados, para todos os modelos, são substancialmente piores neste cenário do que no conjunto de teste. Destaca-se a constatação de que o modelo RESUNET 32 teve um desempenho (relativo aos demais modelos) substancialmente melhor, enquanto o modelo RESUNET++ 32BN MSE teve um desempenho pior.

Modelo	DC
RESUNET++ 32BN	0,9072
RESUNET 32	0,8878
RESUNET++ 16BN	0,8245
RESUNET 32BN	0,8000
MobileNetV2-UNET	0,8000
UNET 32	0,7872
UNET 32BN	0,7417
RESUNET++ 32	0,7280
RESUNET++ 32BN MSE	0,5589
RESUNET++ 32BN 10K	0,4444
RESUNET++ 32BN BCE	0,2569

Tabela 5 – Paralelas - Tabela de resultados

Também nos treinamentos rápidos (Tab. 6) observamos um desempenho máximo substancialmente inferior aos resultados da Tabela 2. A diferença de resultados entre a RESUNET++ e as demais arquiteturas é muito maior, porém, bem como para conjuntos de treinamento muito pequenos (com 1K ou 10K amostras).

4.2.3.2 Domo de Sal Homogêneo

Nas Figuras 44 a 54 podemos ver a aplicação dos modelos ao cenário de domo de sal homogêneo. Observa-se um excelente desempenho de todos os modelos, com exceção do modelo RESUNET++ 32BN 10K.

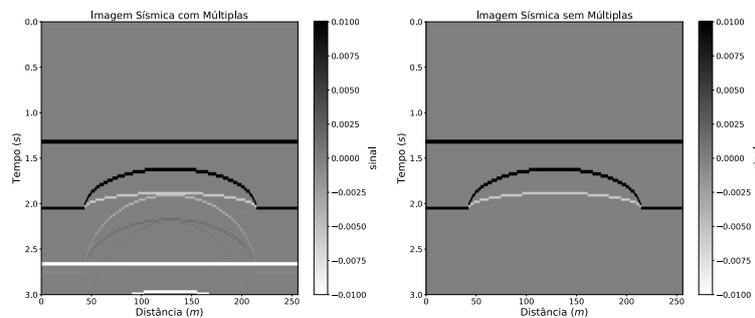


Figura 43 – Imagens sísmicas com e sem múltiplas do sistema de domo de sal homogêneo

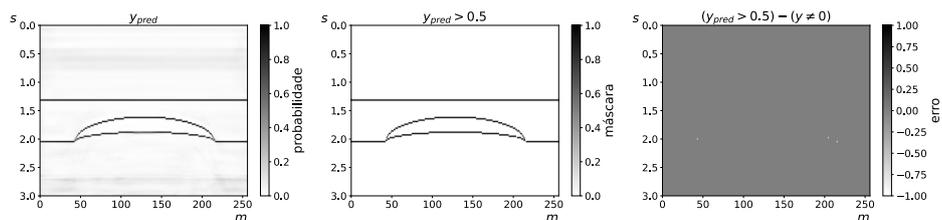


Figura 44 – RESUNET++ 32 aplicada no cenário de domo de sal homogêneo

	Média	Desvio Padrão	Mínimo	25%	Mediana	75%	Máximo
Arquitetura							
	MobileNetV2-UNET	0,2600	0,2869	0,0000	0,0000	0,2232	0,4898
	RESUNET	0,3061	0,2350	0,0000	0,1287	0,2500	0,4950
	RESUNET++	0,2357	0,2354	0,0000	0,0458	0,2228	0,2856
	UNET	0,2825	0,2504	0,0000	0,0473	0,2310	0,4647
Base	16	0,2687	0,2380	0,0000	0,0465	0,2346	0,4074
	32	0,2809	0,2459	0,0000	0,0465	0,2469	0,4540
Batch Normalization	Não	0,2730	0,2367	0,0000	0,0467	0,2407	0,4199
	Sim	0,2766	0,2473	0,0000	0,0463	0,2447	0,4413
Métrica	BCE	0,2300	0,2231	0,0000	0,0405	0,2020	0,3641
	Focal Loss	0,4391	0,2624	0,0000	0,2471	0,4567	0,6672
	MAE	0,1519	0,1360	0,0000	0,0296	0,1262	0,2526
Tamanho Treino (K)	1	0,1405	0,1208	0,0000	0,0374	0,1172	0,2499
	10	0,2675	0,2184	0,0000	0,0700	0,2313	0,3985
	50	0,3309	0,2656	0,0000	0,0470	0,2820	0,5816
	100	0,3558	0,2849	0,0000	0,0458	0,2851	0,6515
Todos	Parâmetros	0,2737	0,2457	0,0000	0,0460	0,2429	0,4373

Tabela 6 – Estatísticas da métrica DC para subconjuntos de modelos criados nos treinamentos rápidos, agrupados pelos parâmetros de treinamento, quando aplicados no cenário de camadas paralelas.

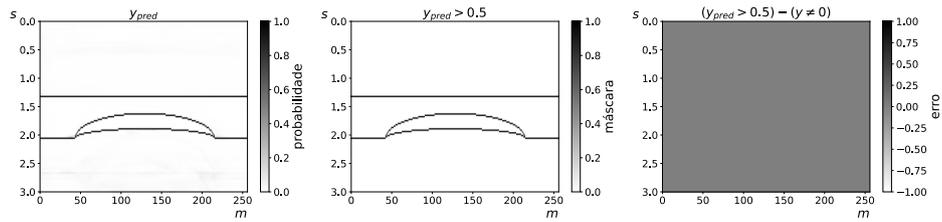


Figura 45 – RESUNET++ 32BN aplicada no cenário de domo de sal homogêneo

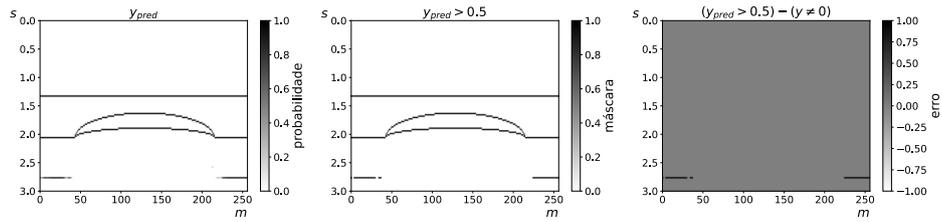


Figura 46 – RESUNET++ 32BN MSE aplicada no cenário de domo de sal homogêneo

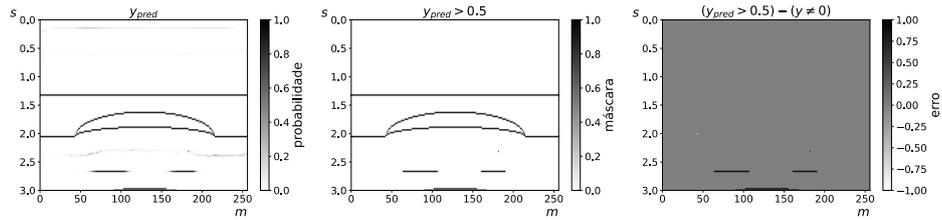


Figura 47 – RESUNET++ 32BN BCE aplicada no cenário de domo de sal homogêneo

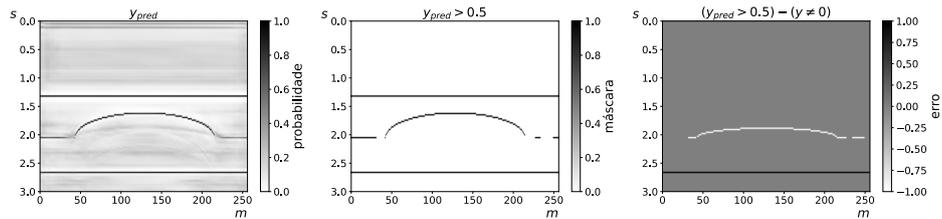


Figura 48 – RESUNET++ 32BN 10K aplicada no cenário de domo de sal homogêneo

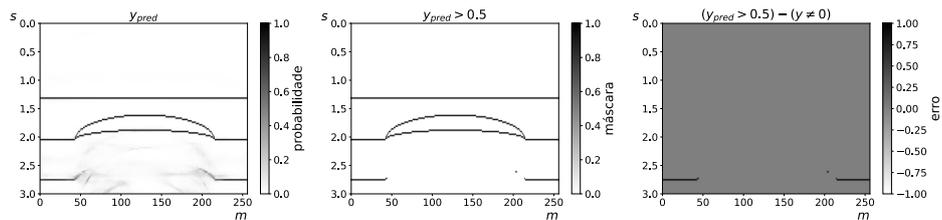


Figura 49 – RESUNET++ 16BN aplicada no cenário de domo de sal homogêneo

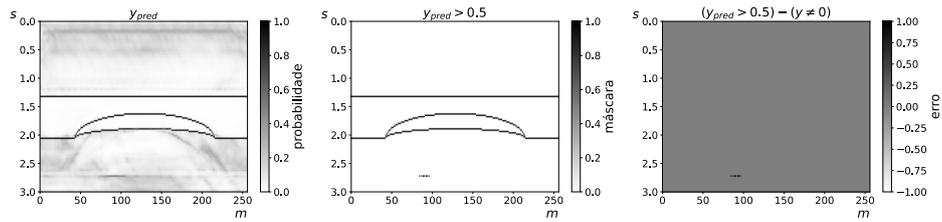


Figura 50 – RESUNET 32 aplicada no cenário de domo de sal homogêneo

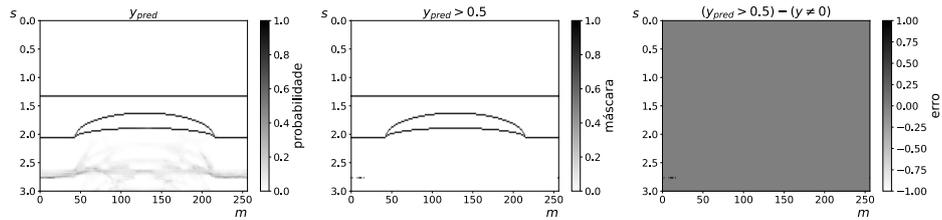


Figura 51 – RESUNET 32BN aplicada no cenário de domo de sal homogêneo

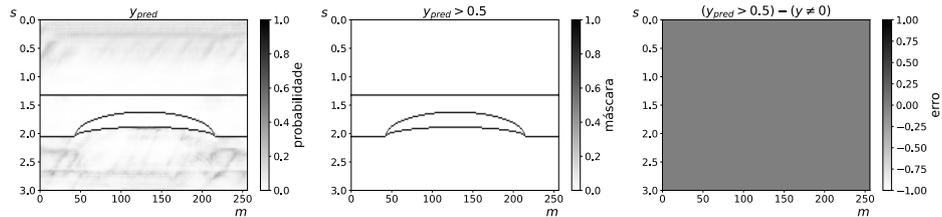


Figura 52 – UNET 32 aplicada no cenário de domo de sal homogêneo

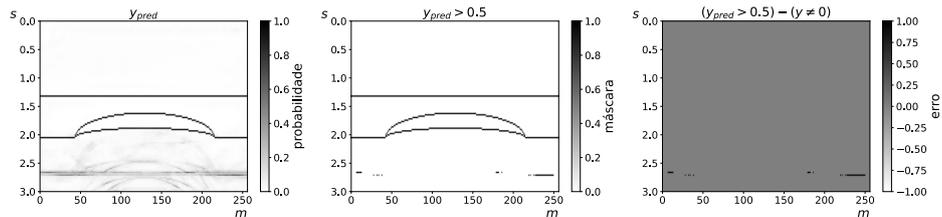


Figura 53 – UNET 32BN aplicada no cenário de domo de sal homogêneo

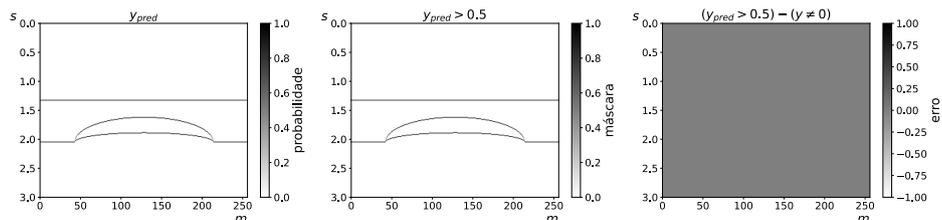


Figura 54 – MobileNetV2-UNET aplicada no cenário de domo de sal homogêneo

Na Tabela 7 vemos como o desempenho dos modelos se traduz para a métrica DC.

Modelo	DC
RESUNET++ 32BN	1,0000
RESUNET++ 32BN BCE	1,0000
RESUNET 32BN	1,0000
UNET 32	1,0000
MobileNetV2-UNET	1,0000
RESUNET++ 32	0,9993
RESUNET 32	0,9956
UNET 32BN	0,9579
RESUNET++ 32BN MSE	0,9573
RESUNET++ 16BN	0,9401
RESUNET++ 32BN 10K	0,7195

Tabela 7 – Domo de sal homogêneo - Tabela de resultados

Na Tabela 8 a maioria dos parâmetros apresenta resultados máximos melhores do que os vistos na Tabela 2. Apenas a RESUNET e conjuntos de treinamento com 1.000 amostras apresentaram resultados piores.

4.2.3.3 Domo de Sal Heterogêneo sobre Reservatório

Nas Figuras 56 a 66 podemos ver a aplicação dos modelos ao cenário de domo de sal heterogêneo sobre reservatório. Destaca-se o desempenho do modelo MobileNetV2-UNET.

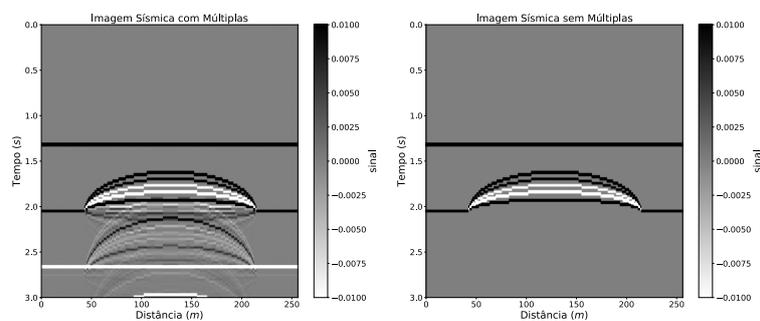


Figura 55 – Imagens sísmicas com e sem múltiplas do sistema de domo de sal heterogêneo sobre reservatório

Arquitetura	Média	Desvio Padrão	Mínimo	25%	Mediana	75%	Máximo
MobileNetV2-UNET	0,4490	0,4268	0,0000	0,0000	0,5307	0,8893	1,0000
RESUNET	0,4892	0,2963	0,0000	0,2168	0,5655	0,7227	0,9241
RESUNET++	0,4343	0,3452	0,0000	0,0213	0,5421	0,6695	1,0000
UNET	0,4092	0,3684	0,0000	0,0221	0,3416	0,7948	1,0000
Base	0,4403	0,3411	0,0000	0,0236	0,5261	0,7185	1,0000
Batch Normalization	0,4482	0,3379	0,0000	0,0235	0,5278	0,7218	1,0000
	0,4377	0,3327	0,0000	0,0238	0,5168	0,7064	1,0000
Sim	0,4508	0,3461	0,0000	0,0230	0,5319	0,7394	1,0000
BCE	0,3698	0,3360	0,0000	0,0136	0,3942	0,6706	0,9989
Focal Loss	0,6678	0,3004	0,0000	0,5456	0,7411	0,9179	1,0000
MAE	0,2963	0,2837	0,0000	0,0107	0,2630	0,5493	0,9624
Tamanho Treino (K)	0,2767	0,2530	0,0000	0,0202	0,2250	0,5440	0,8058
1	0,4526	0,3142	0,0000	0,1150	0,5456	0,6910	1,0000
10	0,5236	0,3633	0,0000	0,0238	0,6225	0,8370	1,0000
50	0,5255	0,3822	0,0000	0,0206	0,6316	0,8649	1,0000
100	0,4446	0,3468	0,0000	0,0215	0,5266	0,7350	1,0000
Parâmetros							

Tabela 8 – Estatísticas da métrica DC para subconjuntos dos modelos criados nos treinamentos rápidos, agrupados pelos parâmetros de treinamento, quando aplicados no cenário de domo de sal homogêneo.

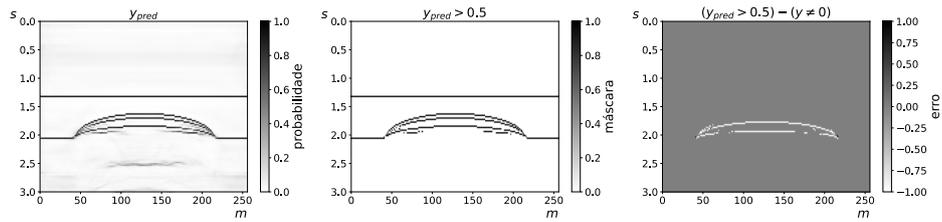


Figura 56 – RESUNET++ 32 aplicada no cenário de domo de sal heterogêneo sobre reservatório

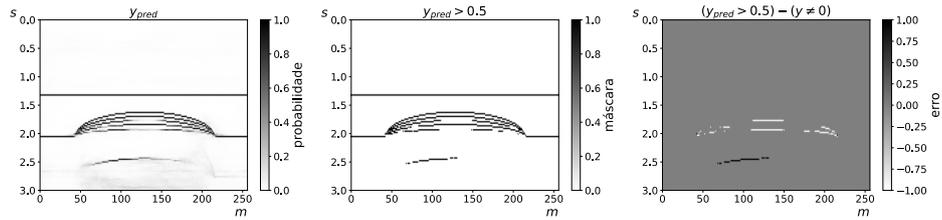


Figura 57 – RESUNET++ 32BN aplicada no cenário de domo de sal heterogêneo sobre reservatório

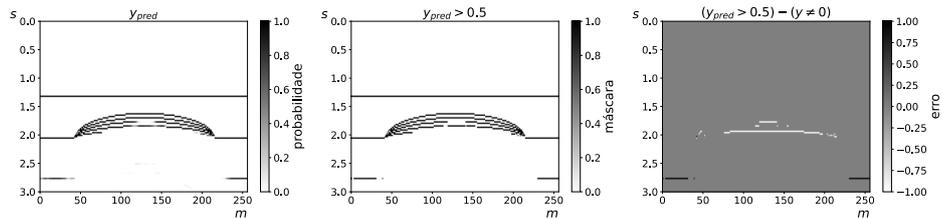


Figura 58 – RESUNET++ 32BN MSE aplicada no cenário de domo de sal heterogêneo sobre reservatório

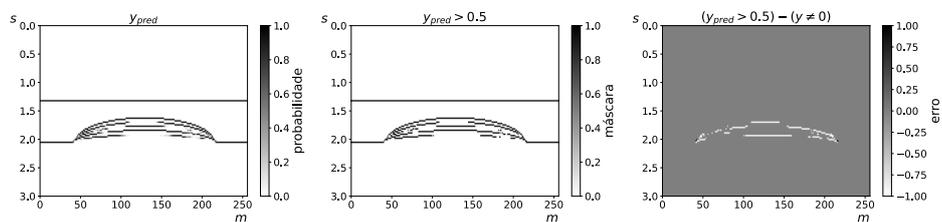


Figura 59 – RESUNET++ 32BN BCE aplicada no cenário de domo de sal heterogêneo sobre reservatório

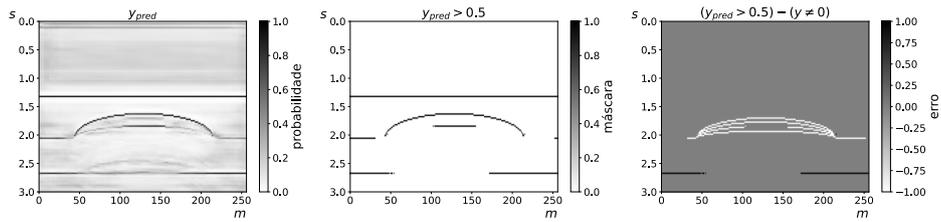


Figura 60 – RESUNET++ 32BN 10K aplicada no cenário de domo de sal heterogêneo sobre reservatório

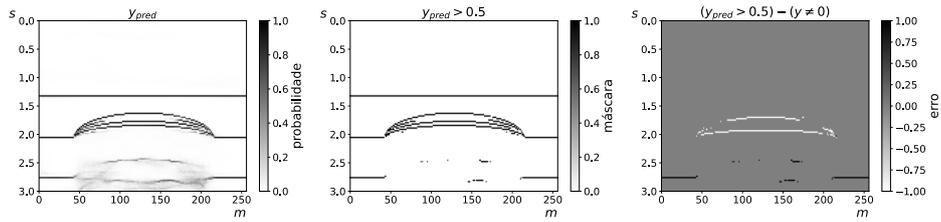


Figura 61 – RESUNET++ 16BN aplicada no cenário de domo de sal heterogêneo sobre reservatório

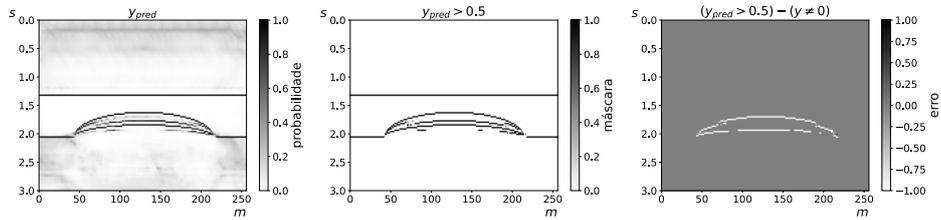


Figura 62 – RESUNET 32 aplicada no cenário de domo de sal heterogêneo sobre reservatório

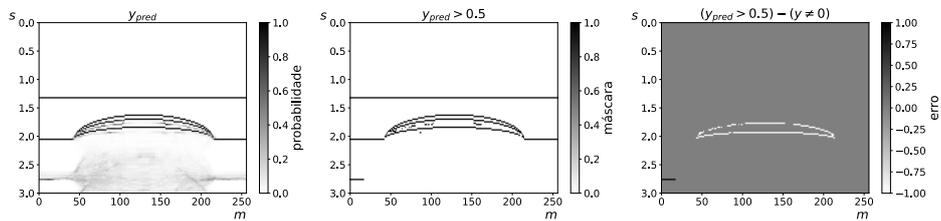


Figura 63 – RESUNET 32BN aplicada no cenário de domo de sal heterogêneo sobre reservatório

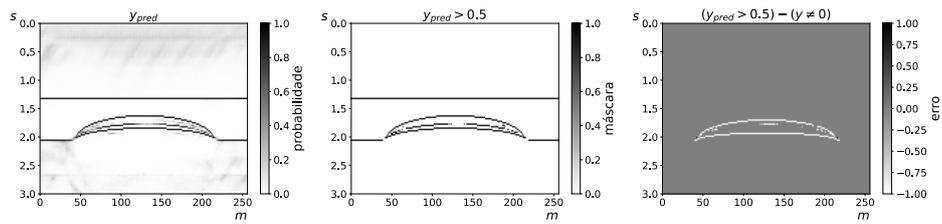


Figura 64 – UNET 32 aplicada no cenário de domo de sal heterogêneo sobre reservatório

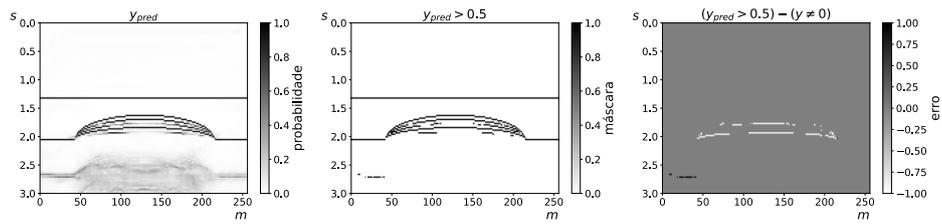


Figura 65 – UNET 32BN aplicada no cenário de domo de sal heterogêneo sobre reservatório

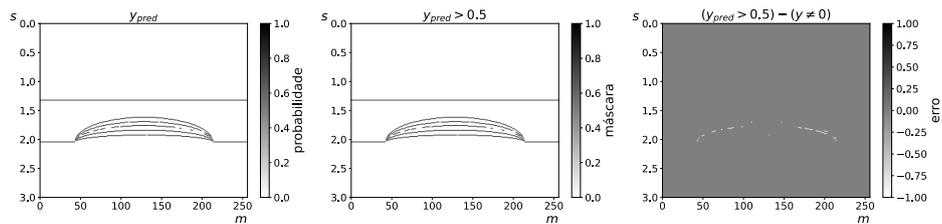


Figura 66 – MobileNetV2-UNET aplicada no cenário de domo de sal heterogêneo sobre reservatório

Na Tabela 9 vemos com clareza os diferentes resultados da aplicação dos modelos ao cenário de domo de sal heterogêneo sobre reservatório.

Modelo	DC
MobileNetV2-UNET	0,9722
UNET 32BN	0,8832
RESUNET++ 32BN	0,8763
RESUNET 32BN	0,8580
RESUNET++ 32BN MSE	0,8504
RESUNET 32	0,8277
RESUNET++ 32BN BCE	0,8260
RESUNET++ 32	0,8226
UNET 32	0,8175
RESUNET++ 16BN	0,7939
RESUNET++ 32BN 10K	0,6458

Tabela 9 – Domo de sal heterogêneo sobre reservatório - Tabela de resultados

Na Tabela 10 verificamos uma grande homogeneidade nos resultados máximos entre os diversos parâmetros, com exceção da arquitetura RESUNET, da métrica MAE e de conjuntos de treinamento com 1.000 amostras.

	Arquitetura	Média	Desvio Padrão	Mínimo	25%	Mediana	75%	Máximo
	MobileNetV2-UNET	0,4122	0,4025	0,0000	0,0000	0,3565	0,8093	0,9774
	RESUNET	0,4674	0,2924	0,0000	0,2416	0,5649	0,6898	0,9181
	RESUNET++	0,4127	0,3286	0,0000	0,0368	0,4132	0,6945	0,9808
	UNET	0,4170	0,3509	0,0000	0,0387	0,3612	0,7864	0,9907
Base	16	0,4282	0,3243	0,0000	0,0440	0,4442	0,7147	0,9907
	32	0,4366	0,3271	0,0000	0,0398	0,4453	0,7175	0,9836
Batch Normalization	Não	0,4283	0,3220	0,0000	0,0417	0,4278	0,7068	0,9907
	Sim	0,4365	0,3294	0,0000	0,0433	0,4655	0,7321	0,9887
Métrica	BCE	0,3800	0,3381	0,0000	0,0136	0,3538	0,7121	0,9907
	Focal Loss	0,6197	0,2958	0,0000	0,3807	0,7062	0,8704	0,9887
	MAE	0,2928	0,2683	0,0000	0,0192	0,3003	0,5321	0,8907
Tamanho Treino (K)	1	0,2577	0,2287	0,0000	0,0356	0,2609	0,4008	0,8290
	10	0,4400	0,2970	0,0000	0,1799	0,4697	0,6723	0,9907
	50	0,5133	0,3515	0,0000	0,0414	0,6303	0,8155	0,9817
	100	0,5124	0,3664	0,0000	0,0361	0,6241	0,8382	0,9836
Todos	Parâmetros	0,4309	0,3321	0,0000	0,0372	0,4442	0,7324	0,9907

Tabela 10 – Estatísticas da métrica DC para subconjuntos dos modelos criados nos treinamentos rápidos, agrupados pelos parâmetros de treinamento, quando aplicados no cenário de domo de sal heterogêneo sobre reservatório.

4.3 Imagens sísmicas com convolução

4.3.1 Redes neurais

Para avaliação da remoção de múltiplas com convolução (vide C.4) serão utilizadas 4 diferentes arquiteturas, baseadas na RESUNET++, RESUNET, UNET e MobileNetV2-UNET. Todas as redes usam o erro absoluto médio (MAE em inglês) como métrica de treinamento e tem a função linear como função de ativação da última camada:

RESUNET++: A arquitetura RESUNET++ com 4 camadas de codificação e decodificação, sendo a primeira camada de codificação com 32 filtros, mas com uma camada extra de *Batch Normalization* na entrada da rede.

RESUNET: A arquitetura RESUNET com 6 camadas de codificação e decodificação, sendo a primeira camada de codificação com 32 filtros, mas com uma camada extra de *Batch Normalization* na entrada da rede.

UNET: A arquitetura UNET com 6 camadas de codificação e decodificação, sendo a primeira camada de codificação com 32 filtros, mas com uma camada extra de *Batch Normalization* na entrada da rede.

MobileNetV2-UNET: O encoder pré-treinado da arquitetura MobileNet v2 com o decoder da UNET, implementados pela biblioteca Segmentation Models.

4.3.2 Conjunto de teste

Na Tabela 11 temos resultados próximos para todos os modelos, exceto para o modelo baseado na UNET.

Modelo	MAE
RESUNET	$2,3887.10^{-3}$
MobileNetV2-UNET	$2,9824.10^{-3}$
RESUNET++	$4,3277.10^{-3}$
UNET	$1,7203.10^{-2}$

Tabela 11 – Tabela de resultados do conjunto de teste. Devido ao longo tempo necessário para o treinamento de cada modelo, não foi possível realizar múltiplos treinamentos e assim estimar a variância das medidas.

Realizando múltiplos treinamentos rápidos, obtivemos as seguintes estatísticas para a MAE no conjunto de teste (Fig. 67, Tab. 12 e Fig. 68):

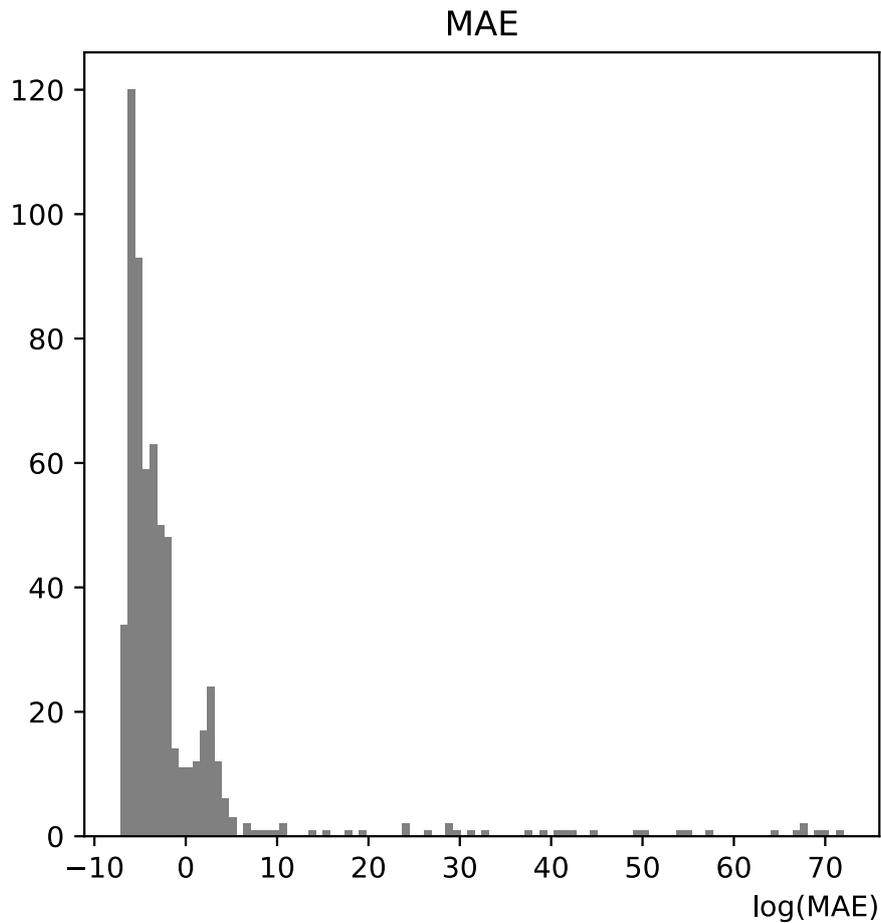


Figura 67 – Histograma \log da métrica MAE dos modelos criados nos treinamentos rápidos aplicados no conjunto de teste.

Podemos observar na Tabela 12 que a maior variação nos valores mínimos de MAE para os treinamentos rápidos ocorreu entre as diferentes arquiteturas e para os modelos com treinados com conjuntos de treinamento com 1.000 amostras.

Para analisarmos os diferentes parâmetros, os histogramas do \log da métrica MAE consideram apenas os modelos para os quais $\log(\text{MAE}) \leq 10$ (Fig. 68), pois isso nos permite visualizar adequadamente a distribuição dos resultados relevantes. Nestes histogramas, podemos notar que as diferentes arquiteturas tem distribuições bem diferentes de $\log(\text{MAE})$, assim como os diferentes tamanhos de conjunto de treino. Já o número de filtros na primeira camada de codificação e a presença ou não de uma camada extra de *Batch Normalization* na entrada da rede não apresentam comportamentos tão diferentes entre si.

Arquitetura	Média	Desvio Padrão	Mínimo	25%	Mediana	75%	Máximo
MobileNetV2-UNET	$3,54.10^{-2}$	$5,07.10^{-2}$	$1,57.10^{-3}$	$3,02.10^{-3}$	$4,29.10^{-3}$	$5,14.10^{-2}$	$1,70.10^{-1}$
RESUNET	$1,30.10^{-1}$	$4,63.10^{-1}$	$7,91.10^{-4}$	$1,99.10^{-3}$	$6,86.10^{-3}$	$7,65.10^{-2}$	$4,30.10^{+0}$
RESUNET++	$1,19.10^{+29}$	$1,35.10^{+30}$	$2,27.10^{-3}$	$1,79.10^{-2}$	$3,58.10^{+0}$	$4,28.10^{+1}$	$1,81.10^{+31}$
UNET	$2,55.10^{+0}$	$1,27.10^{+1}$	$9,60.10^{-4}$	$4,66.10^{-3}$	$1,02.10^{-2}$	$4,27.10^{-2}$	$1,17.10^{+2}$
Base	$6,69.10^{+28}$	$1,07.10^{+30}$	$7,91.10^{-4}$	$4,16.10^{-3}$	$2,55.10^{-2}$	$3,60.10^{-1}$	$1,81.10^{+31}$
	$9,85.10^{+27}$	$1,48.10^{+29}$	$9,60.10^{-4}$	$4,19.10^{-3}$	$1,58.10^{-2}$	$1,94.10^{-1}$	$2,48.10^{+30}$
Batch Normalization	$6,53.10^{+28}$	$1,07.10^{+30}$	$9,60.10^{-4}$	$4,02.10^{-3}$	$2,25.10^{-2}$	$2,19.10^{-1}$	$1,81.10^{+31}$
	$1,17.10^{+28}$	$1,55.10^{+29}$	$7,91.10^{-4}$	$4,27.10^{-3}$	$2,02.10^{-2}$	$3,15.10^{-1}$	$2,48.10^{+30}$
Tamanho Treino (K)	$1,35.10^{+8}$	$1,66.10^{+9}$	$1,45.10^{-3}$	$6,96.10^{-3}$	$2,10.10^{-2}$	$5,26.10^{-2}$	$2,06.10^{+10}$
	$1,85.10^{+28}$	$2,02.10^{+29}$	$9,60.10^{-4}$	$9,62.10^{-3}$	$7,59.10^{-2}$	$3,25.10^{+0}$	$2,48.10^{+30}$
	$2,25.10^{+19}$	$2,79.10^{+20}$	$7,91.10^{-4}$	$3,02.10^{-3}$	$7,23.10^{-3}$	$3,60.10^{-1}$	$3,48.10^{+21}$
	$1,24.10^{+29}$	$1,46.10^{+30}$	$8,89.10^{-4}$	$3,02.10^{-3}$	$4,86.10^{-3}$	$9,73.10^{-1}$	$1,81.10^{+31}$
Todos	$3,55.10^{+28}$	$7,37.10^{+29}$	$7,91.10^{-4}$	$3,82.10^{-3}$	$1,90.10^{-2}$	$1,70.10^{-1}$	$1,81.10^{+31}$

Tabela 12 – Estatísticas da métrica MAE para subconjuntos dos modelos criados nos treinamentos rápidos, agrupados pelos parâmetros de treinamento, quando aplicados ao conjunto de teste.

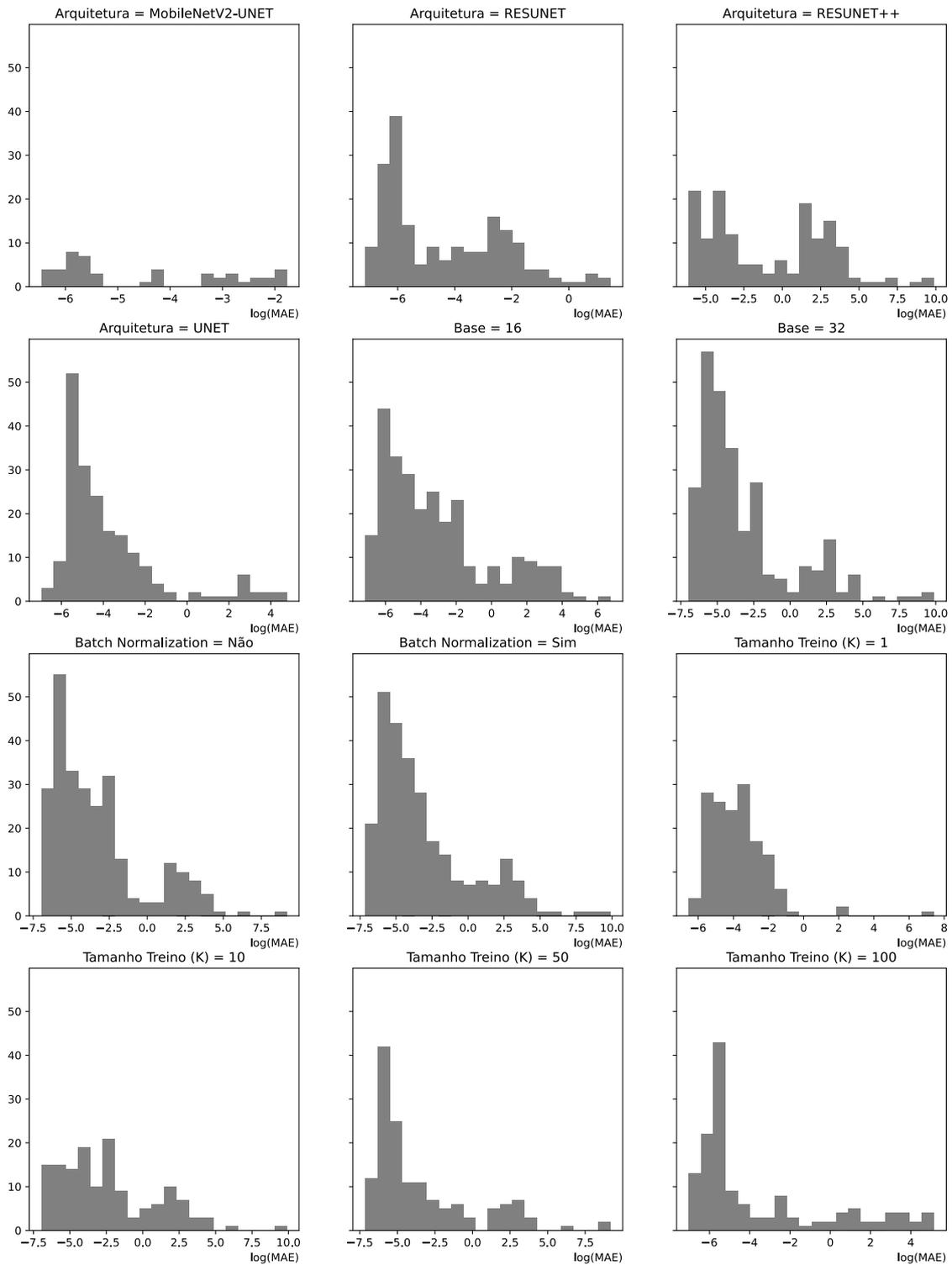


Figura 68 – Histograma do \log da métrica MAE para subconjuntos dos modelos selecionados pelos parâmetros de treinamento, filtrados pelos modelos para os quais $\log(\text{MAE}) \leq 10$.

4.3.2.1 Elemento do Conjunto de Teste

Para facilitar a compreensão, usamos aqui a mesma geologia de teste utilizada anteriormente.:

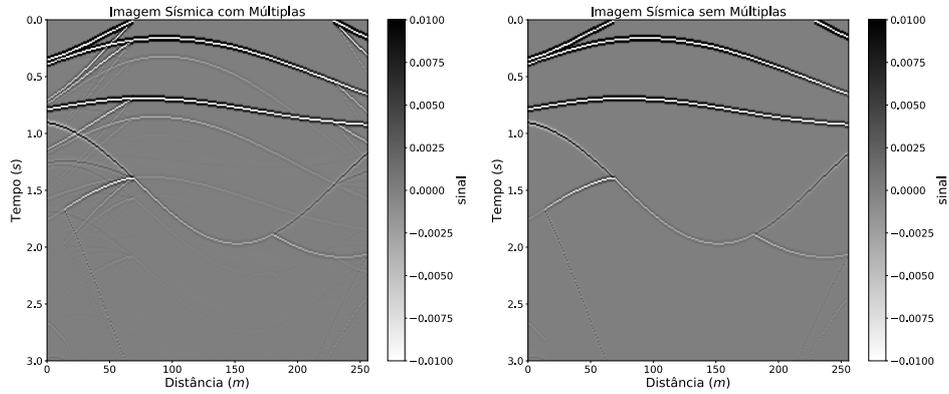


Figura 69 – Imagens sísmicas com e sem múltiplas de elemento do conjunto de teste.

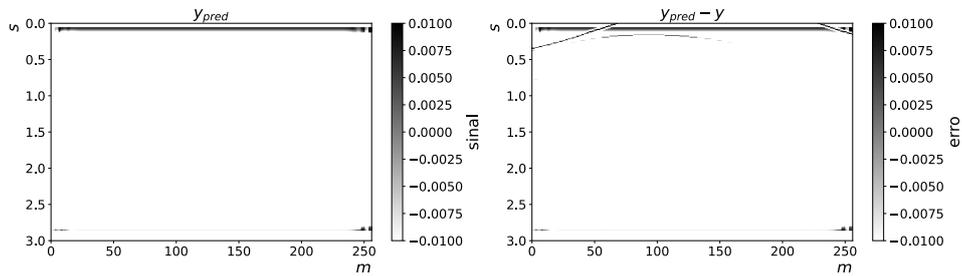


Figura 70 – RESUNET++ aplicada em elemento do conjunto de teste.

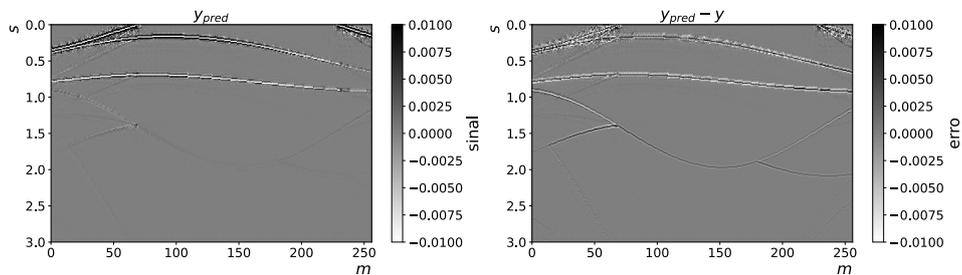


Figura 71 – RESUNET aplicada em elemento do conjunto de teste.

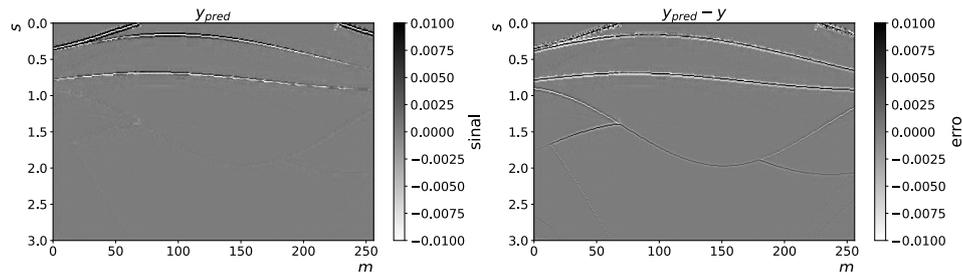


Figura 72 – UNET aplicada em elemento do conjunto de teste.

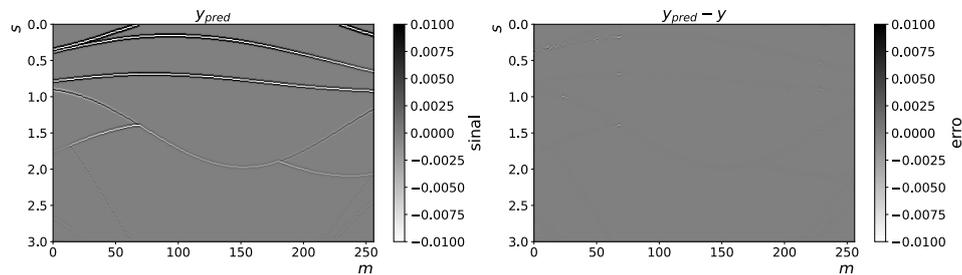


Figura 73 – MobileNetV2-UNET aplicada em elemento do conjunto de teste.

Na Tabela 13 vemos um resultado bastante distinto da Tabela 11. Observando a Figura 70, fica claro que o modelo RESUNET++ não está produzindo uma imagem sísmica adequada. Também podemos apreciar como a grande diferença qualitativa entre os resultados da RESUNET (Fig. 71) e UNET (Fig. 72) e o resultado da MobileNetV2-UNET (Fig. 73) se traduz em suas métricas MAE.

Modelo	MAE
MobileNetV2-UNET	$1,2487.10^{-5}$
RESUNET	$6,2036.10^{-4}$
UNET	$7,5746.10^{-4}$
RESUNET++	$6,1442.10^{-2}$

Tabela 13 – Resultados de um exemplo do conjunto de teste.

Na Tabela 14 os melhores resultados para os treinamentos rápidos apresentaram uma variação menor do que na Tabela 12, sendo que todos esses resultados foram melhores para a amostra do conjunto de teste do que para o conjunto de teste como um todo.

4.3.3 Generalização

A convolução da simulação recursiva destes sistemas geológicos gera as imagens sísmicas apresentadas nas Figuras 74, 79 e 84.

Arquitetura	Média	Desvio Padrão	Mínimo	25%	Mediana	75%	Máximo
MobileNetV2-UNET	$3,45.10^{-2}$	$5,11.10^{-2}$	$7,02.10^{-4}$	$1,92.10^{-3}$	$2,61.10^{-3}$	$5,31.10^{-2}$	$1,70.10^{-1}$
RESUNET	$1,25.10^{-1}$	$4,67.10^{-1}$	$2,44.10^{-4}$	$8,77.10^{-4}$	$4,08.10^{-3}$	$6,40.10^{-2}$	$4,42.10^{+0}$
RESUNET++	$1,38.10^{+29}$	$1,63.10^{+30}$	$5,79.10^{-4}$	$2,51.10^{-3}$	$8,34.10^{-3}$	$5,15.10^{-2}$	$2,21.10^{+31}$
UNET	$2,54.10^{+0}$	$1,27.10^{+1}$	$3,90.10^{-4}$	$9,71.10^{-4}$	$2,72.10^{-3}$	$3,67.10^{-2}$	$1,17.10^{+2}$
Base	$7,94.10^{+28}$	$1,31.10^{+30}$	$2,44.10^{-4}$	$1,07.10^{-3}$	$5,48.10^{-3}$	$5,15.10^{-2}$	$2,21.10^{+31}$
	$1,07.10^{+28}$	$1,69.10^{+29}$	$4,66.10^{-4}$	$1,34.10^{-3}$	$4,36.10^{-3}$	$4,03.10^{-2}$	$2,85.10^{+30}$
Batch Normalization	$7,85.10^{+28}$	$1,31.10^{+30}$	$3,32.10^{-4}$	$1,29.10^{-3}$	$5,12.10^{-3}$	$5,68.10^{-2}$	$2,21.10^{+31}$
	$1,15.10^{+28}$	$1,71.10^{+29}$	$2,44.10^{-4}$	$1,12.10^{-3}$	$4,41.10^{-3}$	$4,00.10^{-2}$	$2,85.10^{+30}$
Tamanho Treino (K)	$2,09.10^{+13}$	$2,61.10^{+14}$	$7,95.10^{-4}$	$3,80.10^{-3}$	$1,43.10^{-2}$	$4,60.10^{-2}$	$3,26.10^{+15}$
	$2,10.10^{+28}$	$2,32.10^{+29}$	$4,66.10^{-4}$	$1,99.10^{-3}$	$6,58.10^{-3}$	$5,68.10^{-2}$	$2,85.10^{+30}$
	$1,23.10^{+19}$	$1,53.10^{+20}$	$3,48.10^{-4}$	$9,35.10^{-4}$	$1,92.10^{-3}$	$2,39.10^{-2}$	$1,91.10^{+21}$
	$1,46.10^{+29}$	$1,78.10^{+30}$	$2,44.10^{-4}$	$8,31.10^{-4}$	$1,93.10^{-3}$	$3,37.10^{-2}$	$2,21.10^{+31}$
Todos	$4,16.10^{+28}$	$8,97.10^{+29}$	$2,44.10^{-4}$	$1,26.10^{-3}$	$4,50.10^{-3}$	$4,77.10^{-2}$	$2,21.10^{+31}$

Tabela 14 – Estatísticas da métrica MAE para subconjuntos dos modelos criados nos treinamentos rápidos, agrupados pelos parâmetros de treinamento, quando aplicados a um exemplo do conjunto de teste.

Temos, para cada cenário e para cada modelo de Rede Neural Profunda: a imagem sísmica sem múltiplas prevista pelos nossos modelos e a diferença entre as duas (demais Figs. 75 a 88).

Para todos os cenários, observamos que o resultado geral dos diferentes modelos é consistente com o que observamos para o exemplo do conjunto de teste, mas os resultados gerados pela UNET e RESUNET apresentam artefatos visíveis que não ocorreram nas Figuras 71 e 72.

Da mesma forma, a MAE de cada cenário (Tabelas 15, 17 e 19) e os resultados dos treinamentos rápidos (Tabelas 16, 18 e 20) corroboram os resultados das Tabelas 13 e 14, respectivamente.

4.3.3.1 Camadas Paralelas

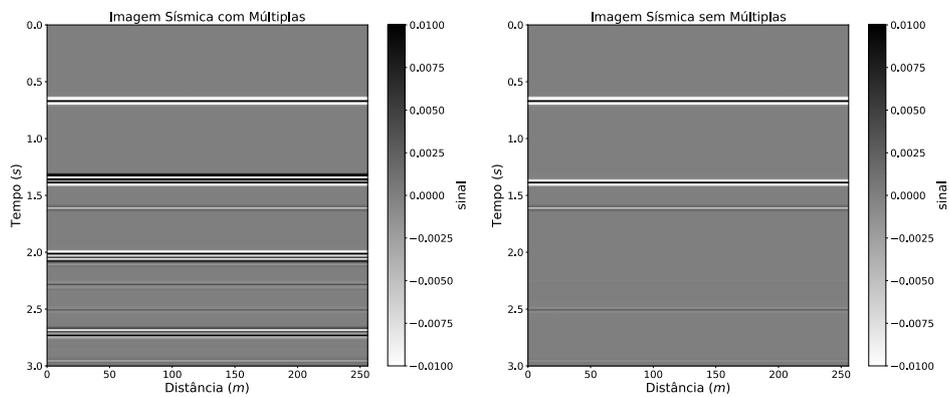


Figura 74 – Imagens sísmicas com e sem múltiplas do sistema de camadas paralelas

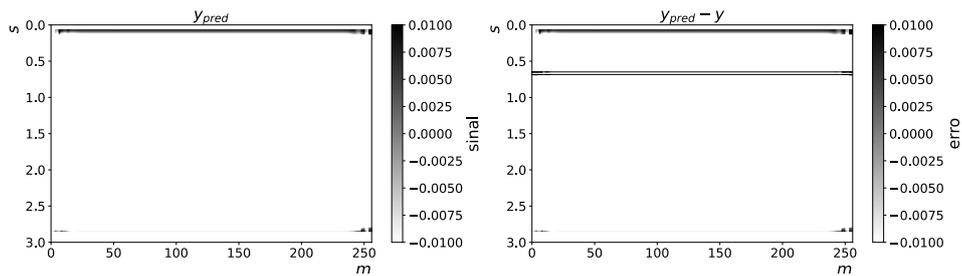


Figura 75 – RESUNET++ aplicada no cenário de camadas paralelas

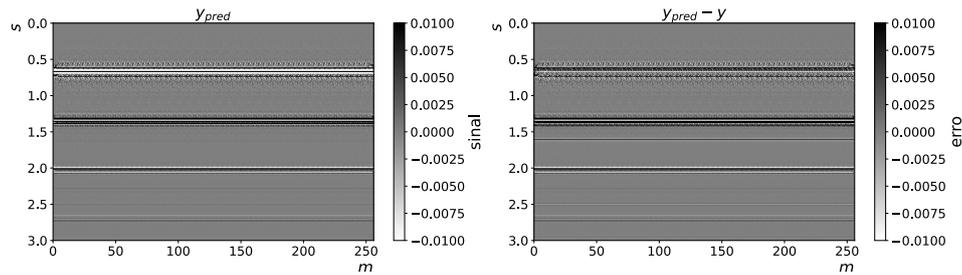


Figura 76 – RESUNET aplicada no cenário de camadas paralelas

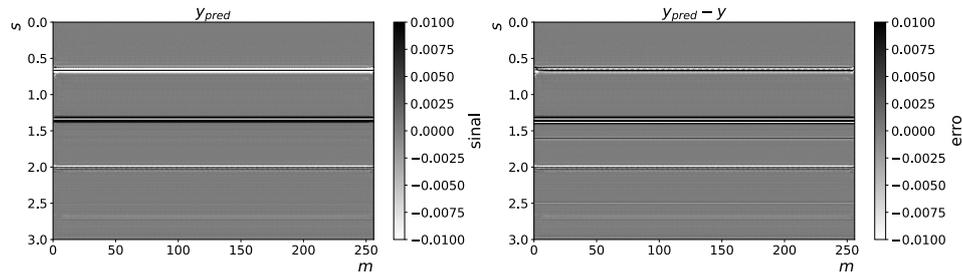


Figura 77 – UNET aplicada no cenário de camadas paralelas

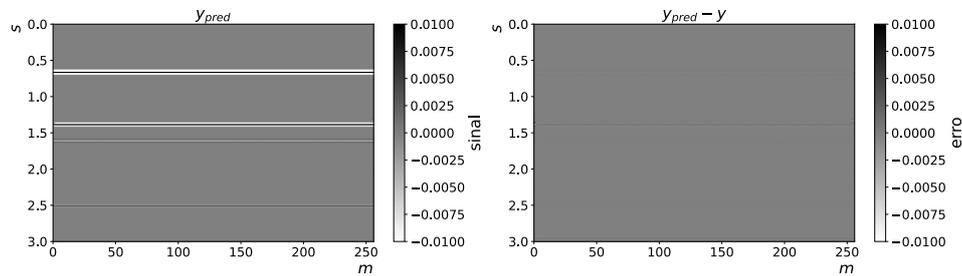


Figura 78 – MobileNetV2-UNET aplicada no cenário de camadas paralelas

Modelo	MAE
MobileNetV2-UNET	$2,8990.10^{-5}$
UNET	$2,2137.10^{-3}$
RESUNET	$2,4359.10^{-3}$
RESUNET++	$6,1850.10^{-2}$

Tabela 15 – Paralelas - Tabela de resultados

Na Tabela 16, os resultados para a RESUNET++ foram substancialmente piores que seus pares, assim como para os modelos treinados com até 10.000 amostras.

Arquitetura	Média	Desvio Padrão	Mínimo	25%	Mediana	75%	Máximo
MobileNetV2-UNET	$3,60.10^{-2}$	$5,11.10^{-2}$	$8,23.10^{-4}$	$3,20.10^{-3}$	$3,97.10^{-3}$	$4,89.10^{-2}$	$1,71.10^{-1}$
RESUNET	$1,28.10^{-1}$	$4,67.10^{-1}$	$8,88.10^{-4}$	$1,84.10^{-3}$	$5,15.10^{-3}$	$6,86.10^{-2}$	$4,42.10^{+0}$
RESUNET++	$1,35.10^{+29}$	$1,45.10^{+30}$	$1,87.10^{-3}$	$5,23.10^{-3}$	$1,94.10^{-2}$	$4,36.10^{-1}$	$1,94.10^{+31}$
UNET	$2,54.10^{+0}$	$1,27.10^{+1}$	$8,10.10^{-4}$	$1,82.10^{-3}$	$3,58.10^{-3}$	$4,28.10^{-2}$	$1,17.10^{+2}$
Base	$7,72.10^{+28}$	$1,16.10^{+30}$	$8,10.10^{-4}$	$1,93.10^{-3}$	$8,91.10^{-3}$	$8,95.10^{-2}$	$1,94.10^{+31}$
	$1,07.10^{+28}$	$1,69.10^{+29}$	$1,03.10^{-3}$	$2,31.10^{-3}$	$6,61.10^{-3}$	$6,55.10^{-2}$	$2,85.10^{+30}$
Batch Normalization	$6,90.10^{+28}$	$1,15.10^{+30}$	$8,10.10^{-4}$	$2,13.10^{-3}$	$8,55.10^{-3}$	$8,89.10^{-2}$	$1,94.10^{+31}$
	$1,90.10^{+28}$	$2,26.10^{+29}$	$8,79.10^{-4}$	$2,12.10^{-3}$	$7,16.10^{-3}$	$5,94.10^{-2}$	$2,85.10^{+30}$
Tamanho Treino (K)	$2,07.10^{+13}$	$2,58.10^{+14}$	$1,48.10^{-3}$	$5,29.10^{-3}$	$1,71.10^{-2}$	$4,83.10^{-2}$	$3,22.10^{+15}$
	$2,09.10^{+28}$	$2,32.10^{+29}$	$1,31.10^{-3}$	$3,09.10^{-3}$	$1,14.10^{-2}$	$1,03.10^{-1}$	$2,85.10^{+30}$
	$3,35.10^{+19}$	$4,19.10^{+20}$	$9,69.10^{-4}$	$1,89.10^{-3}$	$3,61.10^{-3}$	$4,60.10^{-2}$	$5,23.10^{+21}$
	$1,42.10^{+29}$	$1,57.10^{+30}$	$8,10.10^{-4}$	$1,87.10^{-3}$	$3,19.10^{-3}$	$6,96.10^{-2}$	$1,94.10^{+31}$
Todos	$4,06.10^{+28}$	$7,95.10^{+29}$	$8,10.10^{-4}$	$2,21.10^{-3}$	$7,67.10^{-3}$	$6,96.10^{-2}$	$1,94.10^{+31}$

Tabela 16 – Estatísticas da métrica MAE para subconjuntos dos modelos criados nos treinamentos rápidos, agrupados pelos parâmetros de treinamento, quando aplicados no cenário de camadas paralelas.

4.3.3.2 Domo de Sal Homogêneo

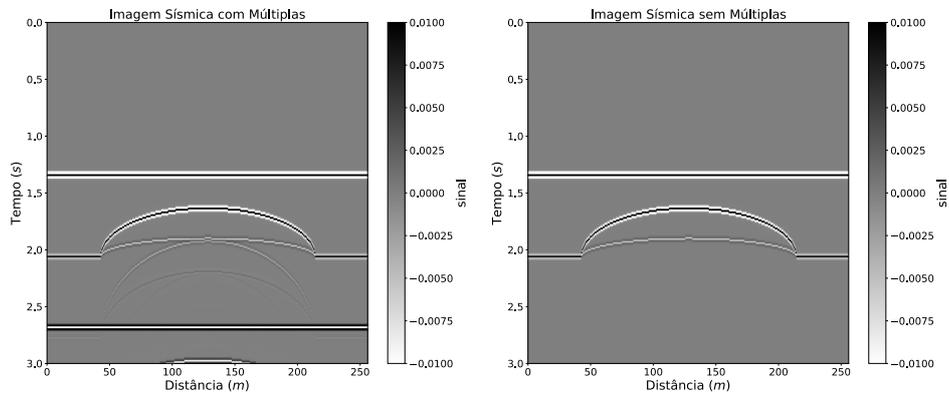


Figura 79 – Imagens sísmicas com e sem múltiplas do sistema de domo de sal homogêneo

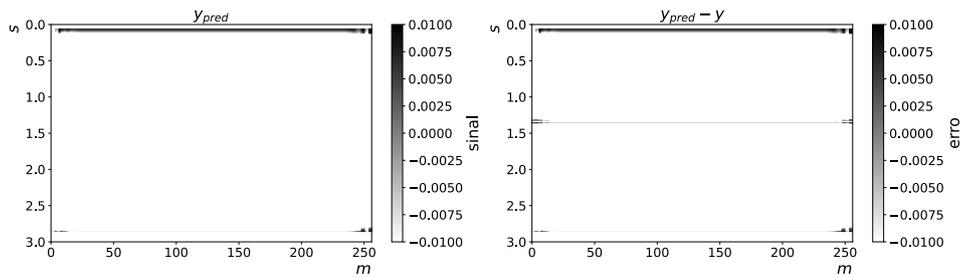


Figura 80 – RESUNET++ aplicada no cenário de domo de sal homogêneo

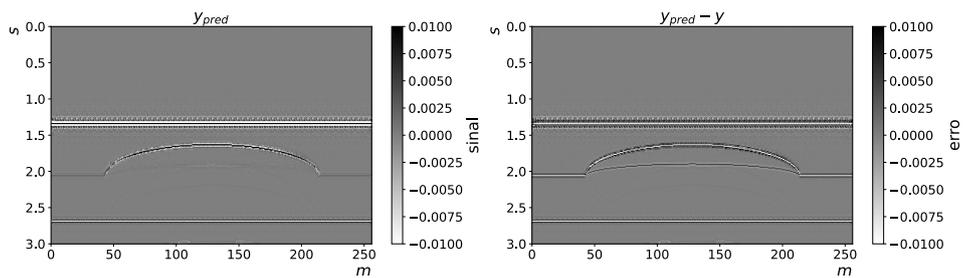


Figura 81 – RESUNET aplicada no cenário de domo de sal homogêneo

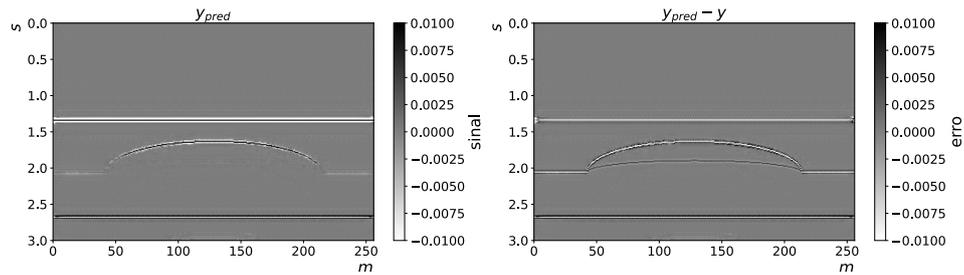


Figura 82 – UNET aplicada no cenário de domo de sal homogêneo

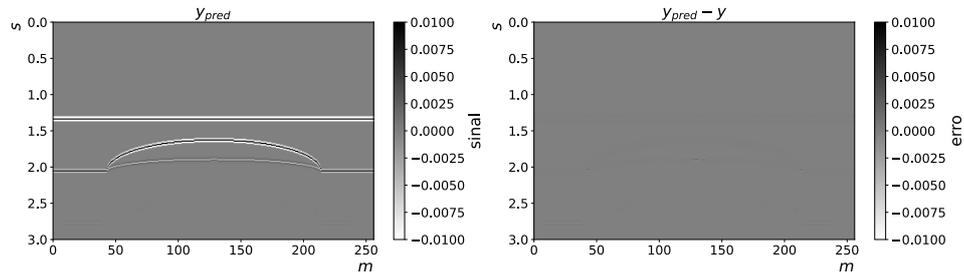


Figura 83 – MobileNetV2-UNET aplicada no cenário de domo de sal homogêneo

Na Tabela 17, observamos que o cenário de domo de sal homogêneo teve resultados consistentes com o de camadas paralelas e com o exemplo do conjunto de treinamento.

Modelo	MAE
MobileNetV2-UNET	$5,7895.10^{-6}$
UNET	$7,1674.10^{-4}$
RESUNET	$9,3617.10^{-4}$
RESUNET++	$6,1165.10^{-2}$

Tabela 17 – Domo de sal homogêneo - Tabela de resultados

Na Tabela 18, vemos que a MobileNetV2-UNET e a RESUNET++ tiveram desempenhos ligeiramente piores do que seus pares.

Arquitetura	Média	Desvio Padrão	Mínimo	25%	Mediana	75%	Máximo
MobileNetV2-UNET	$3,50.10^{-2}$	$5,11.10^{-2}$	$6,49.10^{-4}$	$1,94.10^{-3}$	$6,52.10^{-3}$	$5,11.10^{-2}$	$1,70.10^{-1}$
RESUNET	$1,22.10^{-1}$	$4,52.10^{-1}$	$3,67.10^{-4}$	$6,01.10^{-4}$	$3,18.10^{-3}$	$5,25.10^{-2}$	$4,07.10^{+0}$
RESUNET++	$1,44.10^{+29}$	$1,70.10^{+30}$	$8,57.10^{-4}$	$2,46.10^{-3}$	$7,96.10^{-3}$	$7,27.10^{-2}$	$2,31.10^{+31}$
UNET	$2,54.10^{+0}$	$1,27.10^{+1}$	$2,35.10^{-4}$	$7,40.10^{-4}$	$2,35.10^{-3}$	$3,52.10^{-2}$	$1,17.10^{+2}$
Base	$8,35.10^{+28}$	$1,36.10^{+30}$	$2,35.10^{-4}$	$7,72.10^{-4}$	$4,61.10^{-3}$	$5,31.10^{-2}$	$2,31.10^{+31}$
	$1,00.10^{+28}$	$1,47.10^{+29}$	$4,37.10^{-4}$	$1,08.10^{-3}$	$4,16.10^{-3}$	$3,67.10^{-2}$	$2,46.10^{+30}$
Batch Normalization	$8,26.10^{+28}$	$1,36.10^{+30}$	$3,67.10^{-4}$	$9,41.10^{-4}$	$4,39.10^{-3}$	$5,76.10^{-2}$	$2,31.10^{+31}$
	$1,09.10^{+28}$	$1,51.10^{+29}$	$2,35.10^{-4}$	$9,33.10^{-4}$	$4,44.10^{-3}$	$4,00.10^{-2}$	$2,46.10^{+30}$
Tamanho Treino (K)	$1,41.10^{+13}$	$1,76.10^{+14}$	$3,78.10^{-4}$	$3,46.10^{-3}$	$1,43.10^{-2}$	$4,59.10^{-2}$	$2,20.10^{+15}$
	$1,85.10^{+28}$	$2,00.10^{+29}$	$3,67.10^{-4}$	$1,28.10^{-3}$	$6,37.10^{-3}$	$5,36.10^{-2}$	$2,46.10^{+30}$
	$2,05.10^{+19}$	$2,55.10^{+20}$	$2,35.10^{-4}$	$7,21.10^{-4}$	$1,94.10^{-3}$	$4,06.10^{-2}$	$3,19.10^{+21}$
	$1,55.10^{+29}$	$1,86.10^{+30}$	$4,37.10^{-4}$	$7,65.10^{-4}$	$1,94.10^{-3}$	$3,33.10^{-2}$	$2,31.10^{+31}$
Todos	$4,33.10^{+28}$	$9,34.10^{+29}$	$2,35.10^{-4}$	$1,04.10^{-3}$	$4,44.10^{-3}$	$4,86.10^{-2}$	$2,31.10^{+31}$

Tabela 18 – Estatísticas da métrica MAE para subconjuntos dos modelos criados nos treinamentos rápidos, agrupados pelos parâmetros de treinamento, quando aplicados no cenário de domo de sal homogêneo.

4.3.3.3 Domo de Sal Heterogêneo sobre Reservatório

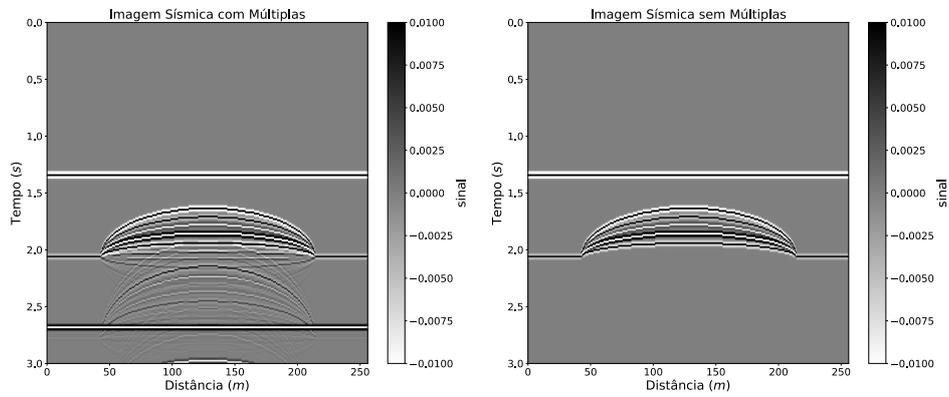


Figura 84 – Imagens sísmicas com e sem múltiplas do sistema de domo de sal heterogêneo sobre reservatório

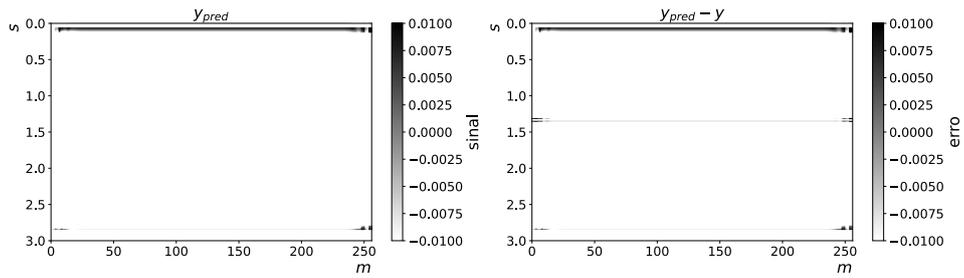


Figura 85 – RESUNET++ aplicada no cenário de domo de sal heterogêneo sobre reservatório

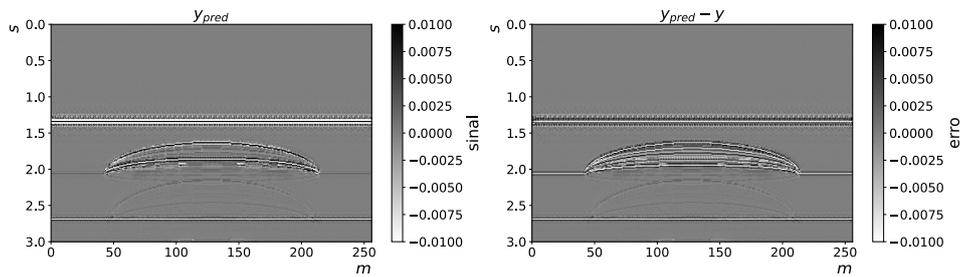


Figura 86 – RESUNET aplicada no cenário de domo de sal heterogêneo sobre reservatório

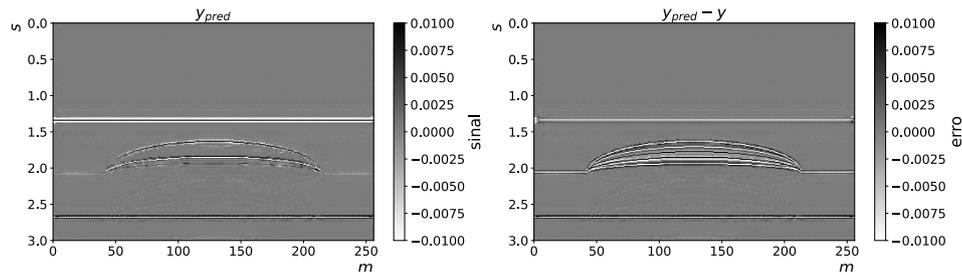


Figura 87 – UNET aplicada no cenário de domo de sal heterogêneo sobre reservatório

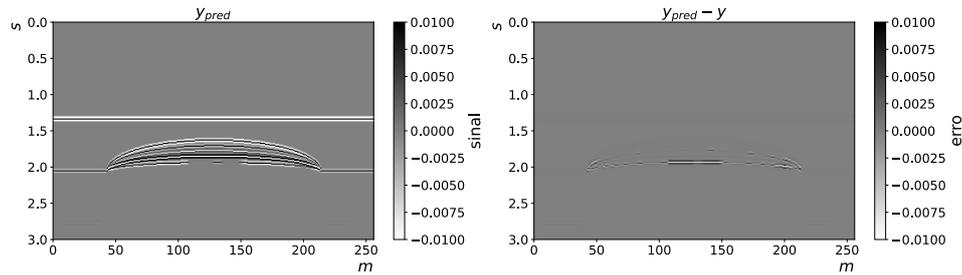


Figura 88 – MobileNetV2-UNET aplicada no cenário de domo de sal heterogêneo sobre reservatório

Na Tabela 19, novamente temos resultados consistentes com os dos exemplos anteriores.

Modelo	MAE
MobileNetV2-UNET	$7,0832 \cdot 10^{-5}$
UNET	$1,1030 \cdot 10^{-3}$
RESUNET	$1,2620 \cdot 10^{-3}$
RESUNET++	$6,1165 \cdot 10^{-2}$

Tabela 19 – Domo de sal heterogêneo sobre reservatório - Tabela de resultados

Assim, como na Tabela 18, na Tabela 20 temos um pior desempenho para as arquiteturas MobileNetV2-UNET e a RESUNET++, mas ainda próximos a seus pares.

Arquitetura	Média	Desvio Padrão	Mínimo	25%	Mediana	75%	Máximo
MobileNetV2-UNET	$3,51.10^{-2}$	$5,08.10^{-2}$	$1,10.10^{-3}$	$2,40.10^{-3}$	$7,15.10^{-3}$	$4,99.10^{-2}$	$1,71.10^{-1}$
RESUNET	$1,22.10^{-1}$	$4,43.10^{-1}$	$5,04.10^{-4}$	$9,92.10^{-4}$	$4,14.10^{-3}$	$5,83.10^{-2}$	$3,88.10^{+0}$
RESUNET++	$1,05.10^{+29}$	$1,22.10^{+30}$	$1,13.10^{-3}$	$2,93.10^{-3}$	$8,58.10^{-3}$	$5,98.10^{-2}$	$1,66.10^{+31}$
UNET	$2,54.10^{+0}$	$1,27.10^{+1}$	$5,73.10^{-4}$	$1,14.10^{-3}$	$3,01.10^{-3}$	$3,52.10^{-2}$	$1,17.10^{+2}$
Base	$6,03.10^{+28}$	$9,80.10^{+29}$	$5,04.10^{-4}$	$1,19.10^{-3}$	$5,32.10^{-3}$	$5,31.10^{-2}$	$1,66.10^{+31}$
	$7,98.10^{+27}$	$1,16.10^{+29}$	$8,21.10^{-4}$	$1,42.10^{-3}$	$4,23.10^{-3}$	$4,07.10^{-2}$	$1,95.10^{+30}$
Batch Normalization	$5,96.10^{+28}$	$9,80.10^{+29}$	$7,24.10^{-4}$	$1,31.10^{-3}$	$5,12.10^{-3}$	$5,98.10^{-2}$	$1,66.10^{+31}$
	$8,70.10^{+27}$	$1,20.10^{+29}$	$5,04.10^{-4}$	$1,23.10^{-3}$	$4,55.10^{-3}$	$4,10.10^{-2}$	$1,95.10^{+30}$
Tamanho Treino (K)	$1,72.10^{+13}$	$2,15.10^{+14}$	$7,83.10^{-4}$	$3,86.10^{-3}$	$1,45.10^{-2}$	$4,57.10^{-2}$	$2,69.10^{+15}$
	$1,48.10^{+28}$	$1,59.10^{+29}$	$7,34.10^{-4}$	$1,75.10^{-3}$	$6,48.10^{-3}$	$5,91.10^{-2}$	$1,95.10^{+30}$
	$2,83.10^{+19}$	$3,53.10^{+20}$	$5,73.10^{-4}$	$1,10.10^{-3}$	$2,40.10^{-3}$	$3,52.10^{-2}$	$4,41.10^{+21}$
	$1,12.10^{+29}$	$1,34.10^{+30}$	$5,04.10^{-4}$	$1,16.10^{-3}$	$2,40.10^{-3}$	$3,34.10^{-2}$	$1,66.10^{+31}$
Todos	$3,16.10^{+28}$	$6,72.10^{+29}$	$5,04.10^{-4}$	$1,39.10^{-3}$	$4,87.10^{-3}$	$4,99.10^{-2}$	$1,66.10^{+31}$

Tabela 20 – Estatísticas da métrica MAE para subconjuntos dos modelos criados nos treinamentos rápidos, agrupados pelos parâmetros de treinamento, quando aplicados no cenário de domo de sal heterogêneo sobre reservatório.

4.3.4 Método de Diferenças Finitas

As imagens sísmicas para os 3 cenários simuladas em MDF, assim como a imagem sísmica sem múltiplas prevista por cada modelo são apresentadas nas figuras 89 a 103.

Embora neste trabalho não tenhamos obtido as imagens sísmicas sem múltiplas simuladas em MDF, o que nos impede de obter métricas e de ilustrar os pontos de erro, é possível apreciar qualitativamente que há uma boa correlação de qualidade com os resultados anteriormente apresentados na seção 4.3.

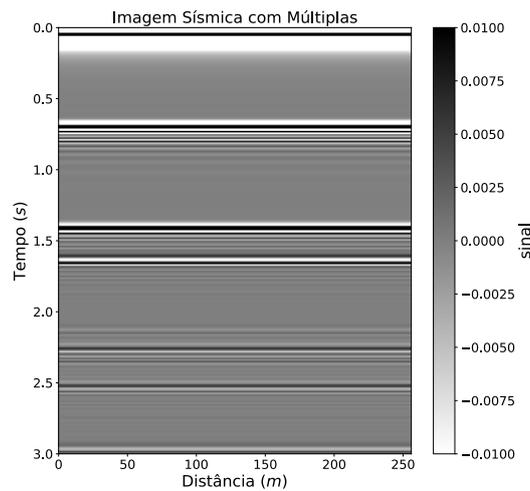


Figura 89 – Camadas paralelas

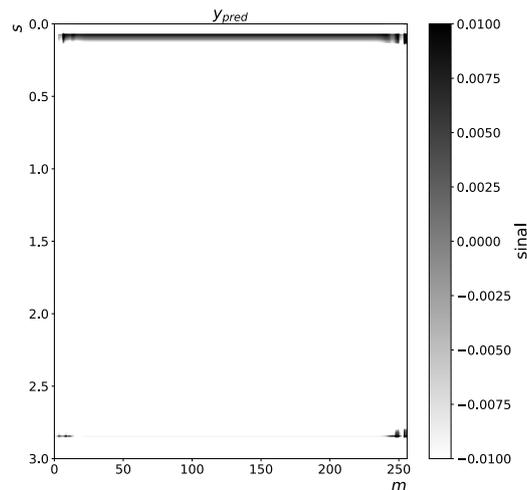


Figura 90 – RESUNET++ aplicada no cenário de camadas paralelas

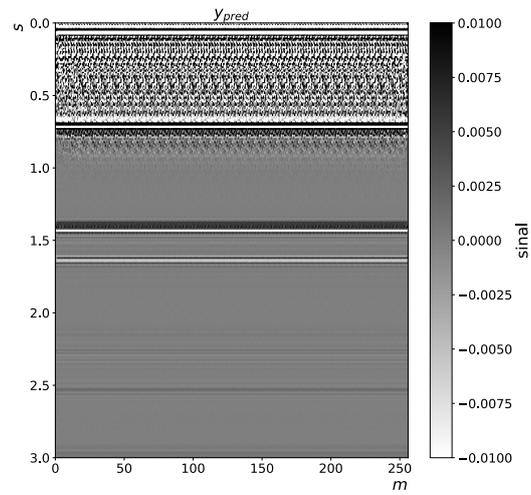


Figura 91 – RESUNET aplicada no cenário de camadas paralelas

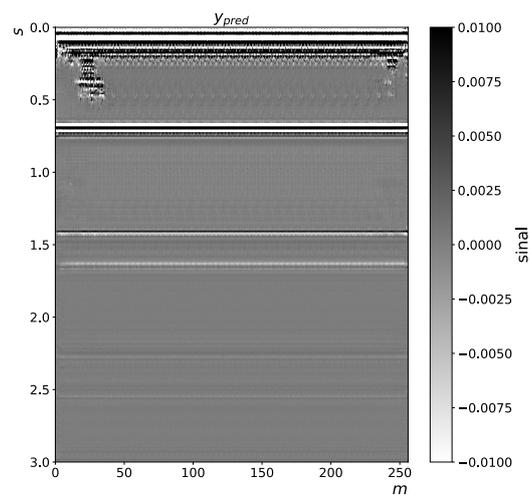


Figura 92 – UNET aplicada no cenário de camadas paralelas

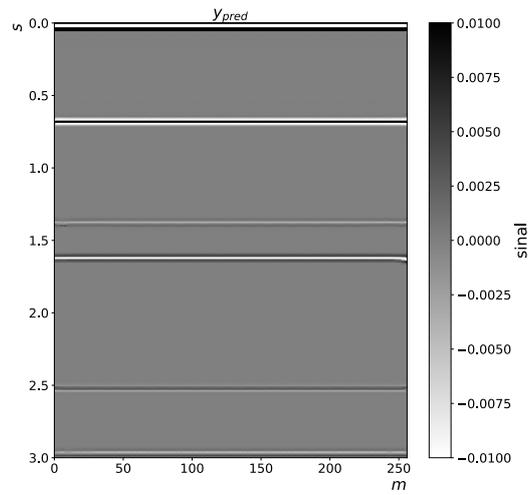


Figura 93 – MobileNetV2-UNET aplicada no cenário de camadas paralelas

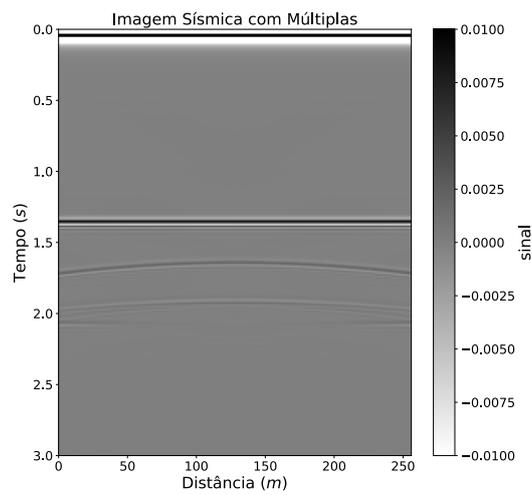


Figura 94 – Domo de sal homogêneo

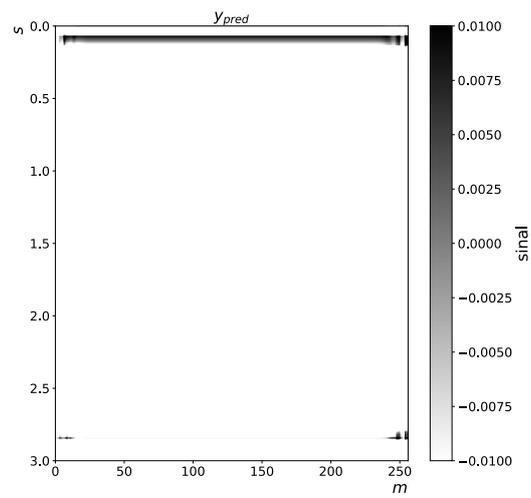


Figura 95 – RESUNET++ aplicada no cenário de domo de sal homogêneo

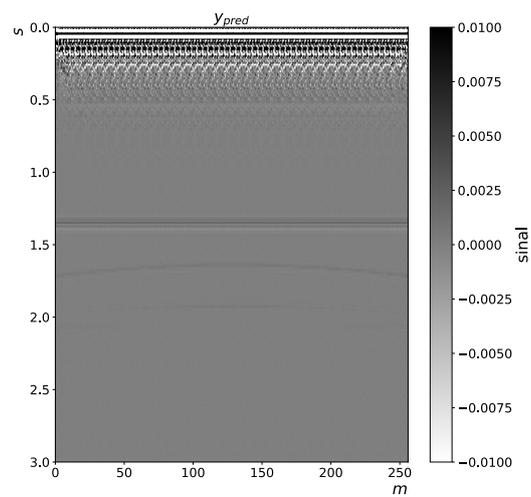


Figura 96 – RESUNET aplicada no cenário de domo de sal homogêneo

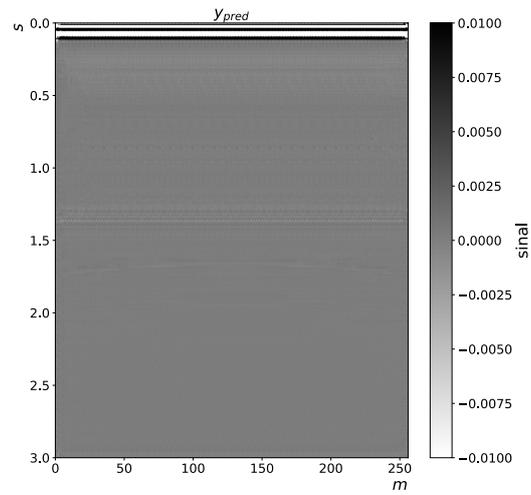


Figura 97 – UNET aplicada no cenário de domo de sal homogêneo

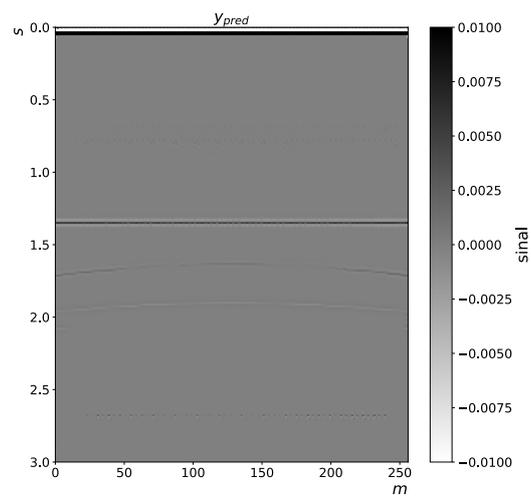


Figura 98 – MobileNetV2-UNET aplicada no cenário de domo de sal homogêneo

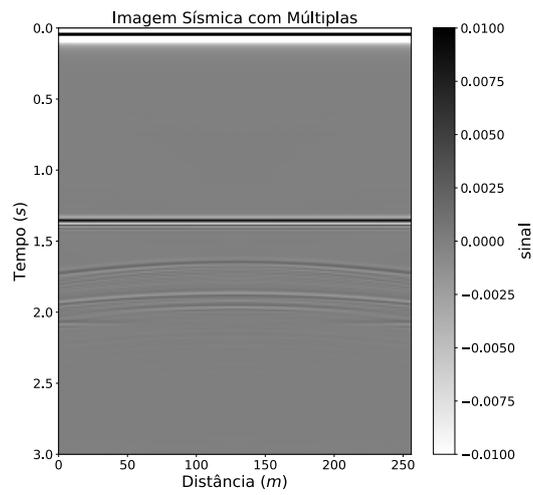


Figura 99 – Domo de sal heterogêneo com reservatório

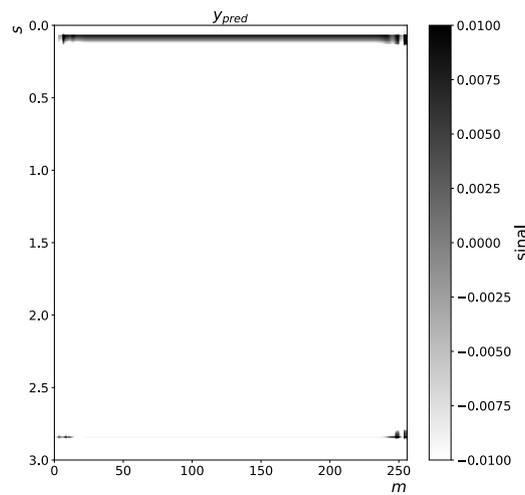


Figura 100 – RESUNET++ aplicada no cenário de domo de sal homogêneo sobre reservatório

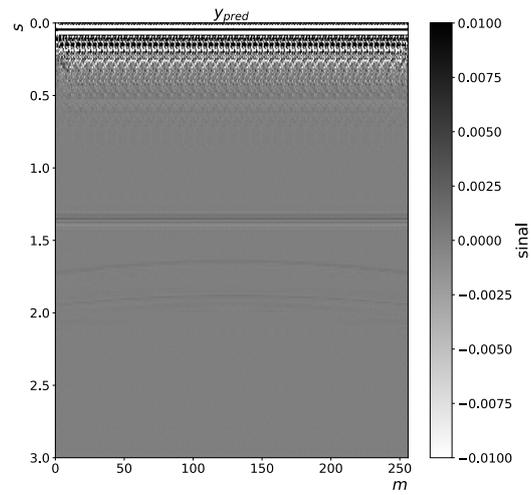


Figura 101 – RESUNET aplicada no cenário de domo de sal homogêneo sobre reservatório

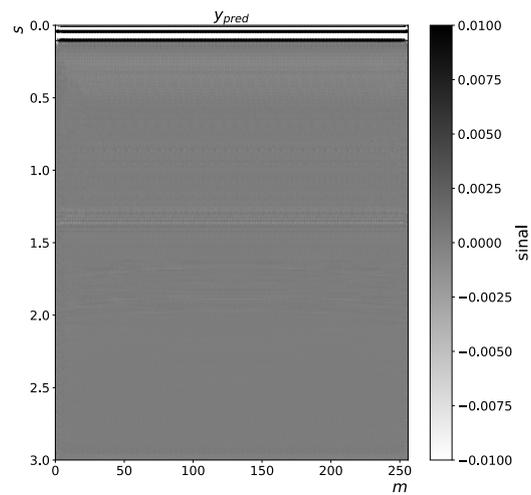


Figura 102 – UNET aplicada no cenário de domo de sal homogêneo sobre reservatório

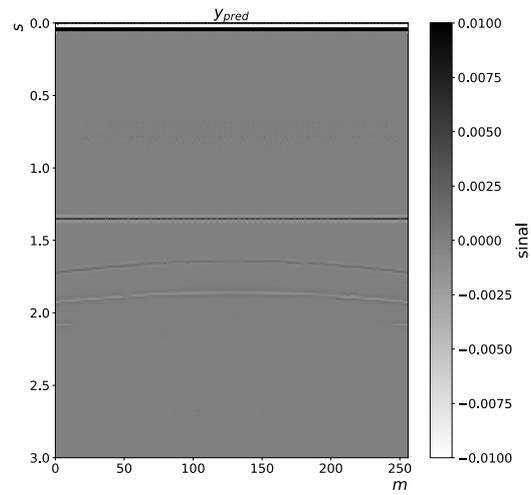


Figura 103 – MobileNetV2-UNET aplicada no cenário de domo de sal homogêneo sobre reservatório

4.4 Análise dos Resultados

4.4.1 Imagens sísmicas sem convolução

Para as imagens sísmicas não convoluídas, a performance dos modelos no conjunto de teste (Tab. 1) deixa clara a importância de usarmos uma camada de *Batch Normalization* na entrada da rede para os dados da simulação. As vantagens da RESUNET++ 32BN (*Dice Coefficient* 0,9879) nas métricas do modelo, embora numericamente pequenas, demonstram a superioridade da arquitetura. Em comparação com a RESUNET++16BN (*Dice Coefficient* 0,9854), por exemplo, vemos como a presença de um maior número de filtros na rede teve impacto positivo apreciável, embora pequeno.

O conjunto dos resultados sugere que a RESUNET não possui superioridade frente à UNET na maioria dos cenários, com exceção com o de camadas paralelas. Também verificamos que a métrica de treinamento *Focal Loss* é a mais adequada para esta tarefa, com grandes perdas ao adotar outras métricas. Também vemos que, em comparação com os modelo treinados com 100.000 exemplos, o modelo treinado com 10.000 exemplos tem pior desempenho, o que deixa claro a importância de uma grande massa de dados de treinamento.

Analisando os resultados dos 2 primeiros cenários sem convolução da verificação de generalização (Figs. 31 a 40 e 43 a 52, respectivamente), podemos observar que o modelo RESUNET++ 32BN generalizou muito bem para esses cenários inéditos. No terceiro cenário (Figs. 55 a 64), mais complexo e de difícil tratamento, o modelo não teve uma performance tão boa, tendo a MobileNetV2-UNET exibido um resultado distintamente melhor do que as demais arquiteturas.

O percentual de modelos de cada classe presente na faixa superior (5^o percentil) de valores (Tab. 21) corrobora a análise anterior sobre a *Focal Loss* e a *Batch Normalization*. Fica mais evidente a importância do tamanho do conjunto de treinamento. Embora as diferenças não sejam muito grandes, vemos que a UNET é a arquitetura com maior representatividade neste conjunto, o que sugere uma menor sensibilidade desta arquitetura a configurações subótimas. Também observa-se uma predominância no uso de 32 filtros na primeira camada convolucional.

		Contagem	Total	Percentual
Arquitetura	MobileNetV2-UNET	6	144	4,17 %
	RESUNET	21	576	3,65 %
	RESUNET++	28	576	4,86 %
	UNET	40	576	6,94 %
Base	16	35	864	4,05 %
	32	54	864	6,25 %
Batch Normalization	Não	27	864	3,12 %
	Sim	62	864	7,18 %
Métrica	BCE	0	624	0,00 %
	Focal Loss	95	624	15,22 %
	MAE	0	624	0,00 %
Tamanho Treino (K)	1	0	468	0,00 %
	10	2	468	0,43 %
	50	37	468	7,91 %
	100	56	468	11,97 %
Todos	Parâmetros	95	1872	5,07 %

Tabela 21 – Estatísticas sobre a ocorrência dos modelos na faixa superior (5%) de valores da métrica DC.

Conforme podemos observar (Tab. 22), os múltiplos treinamentos de cada combinação de parâmetros resultaram em valores de DC com boa correlação, para cada parâmetro fixado. Ou seja, não foram muito diferentes entre si.

		Média	Desvio Padrão
Arquitetura	MobileNetV2-UNET	0,91	0,03
	RESUNET	0,87	0,03
	RESUNET++	0,86	0,02
	UNET	0,86	0,03
Base	16	0,85	0,01
	32	0,88	0,01
Batch Normalization	Não	0,83	0,00
	Sim	0,90	0,01
Métrica	BCE	0,83	0,02
	Focal Loss	0,84	0,01
	MAE	0,87	0,02
Tamanho Treino (K)	1	0,82	0,02
	10	0,83	0,01
	50	0,86	0,03
	100	0,90	0,02
Todos	Parâmetros	0,87	0,01

Tabela 22 – Estatísticas da correlação entre os resultados de múltiplos treinamentos de cada combinação de parâmetros com dados não convoluídos.

4.4.2 Imagens sísmicas com convolução

Para as imagens sísmicas convoluídas, podemos constatar que não foi possível obter um modelo da RESUNET++ capaz de gerar um resultado minimamente aceitável. Para as redes RESUNET e UNET obtivemos resultados intermediários. Conseguimos, no entanto, obter ótimos resultados com o modelo MobileNetV2-UNET. É importante notar que os valores da MAE no conjunto de teste (Tab. 11) não se refletiram perfeitamente nos diferentes cenários (Tabs. 13, 15, 17 e 19)

Nas imagens sísmicas obtidas por simulação de MDF, observamos que o desempenho de cada modelo foi bastante parecido com os exemplos de imagens sísmicas convoluídas, o que sugere que o treinamento com dados obtidos pelo simulador recursivo pode ser adequado para dados mais realistas, entretanto não possuímos as imagens sem múltiplas para quantificar este desempenho.

Analisando o Percentual de modelos de cada classe presente no 5º percentil inferior (Tab. 23), observamos uma ocorrência distintamente maior de modelos com a Arquitetura RESUNET, 16 filtros na Base e 100.000 elementos no conjunto de treinamento.

		Contagem	Total	Percentual
Arquitetura	MobileNetV2-UNET	2	48	4,17
	RESUNET	26	191	13,61
	RESUNET++	0	184	0,00
	UNET	3	192	1,56
Base	16	20	285	7,02
	32	9	282	3,19
Batch Normalization	Não	16	284	5,63
	Sim	13	283	4,59
Tamanho Treino (K)	1	2	155	1,29
	10	5	152	3,29
	50	10	155	6,45
	100	14	153	9,15
Todos	Parâmetros	31	615	5,04

Tabela 23 – Estatísticas sobre a ocorrência dos modelos na faixa inferior (5%) de valores da MAE.

Conforme podemos observar (Tab. 24), os múltiplos treinamentos de cada combinação de parâmetros com dados convoluídos resultaram, em diversos casos, em valores com baixa correlação. Ou seja, o uso de uma mesma combinação de parâmetros gerou resultados pouco semelhantes entre si.

		Média	Desvio Padrão
Arquitetura	MobileNetV2-UNET	0,76	0,11
	RESUNET	0,45	0,12
	RESUNET++	0,32	0,48
	UNET	0,85	0,03
Base	16	0,33	0,48
	32	0,33	0,48
Batch Normalization	Não	0,33	0,48
	Sim	-0,01	0,00
Tamanho Treino (K)	1	0,34	0,47
	10	0,99	0,01
	50	0,32	0,48
	100	0,32	0,48
Todos	Parâmetros	0,33	0,47

Tabela 24 – Estatísticas da correlação entre os resultados de múltiplos treinamentos de cada combinação de parâmetros com dados convoluídos.

Conclusão

Nesta investigação, dedicamos nossos esforços ao desenvolvimento do Simulador Recursivo, uma ferramenta essencial para o avanço do estudo sobre a distinção entre múltiplas e reflexões primárias utilizando Redes Neurais. Ao longo do processo, tínhamos importantes hipóteses a confirmar, e constatamos que os produtos gerados pelo Simulador Recursivo fornecem elementos suficientes para que uma Rede Neural possa de fato aprender a fazer essa distinção tão crucial. Ao validar nossas hipóteses, comprovamos que várias Redes Neurais treinadas utilizando os cenários geológicos modelados com senoidais obtiveram um bom desempenho. Esse resultado positivo não se limitou apenas aos cenários gerados pelo mesmo algoritmo usado no treinamento, mas se estendeu a cenários geológicos diversos e distintos. Essa descoberta é especialmente relevante, pois indica que a Rede Neural desses modelos adquiriu a capacidade de generalização, tornando-se versátil o suficiente para lidar com desafios além daqueles com os quais foi treinada. A possibilidade de as Redes Neurais demonstrarem um bom desempenho em cenários diferentes dos encontrados durante o treinamento é um resultado altamente promissor. Isso sugere que nossa abordagem de utilizar os cenários geológicos modelados por senoidais como base de treinamento proporcionou uma representação adequada e abrangente de elementos essenciais para a tarefa de distinguir múltiplas de reflexões primárias. A capacidade de generalização da rede mostra que o uso de um conjunto de treinamento parametrizado por ondas senoidais promove uma boa independência dos modelos treinados em relação a diferentes cenários geológicos, que é uma limitação frequentemente presente em outros trabalhos.

Como principal resultado positivo, enfatiza-se o notável sucesso alcançado pelo modelo MobileNetV2-UNET na remoção de múltiplas em imagens sísmicas que passaram pelo processo de convolução e em imagens obtidas por simulação em MDF. Essa conquista é importante para o avanço da tecnologia de processamento sísmico, uma vez que a remoção eficiente das múltiplas é fundamental para obter uma melhor interpretação das reflexões primárias, contribuindo para uma análise mais precisa das camadas geológicas.

Além disso, ao adotar a arquitetura RESUNET++ nas imagens sísmicas sem convolução, constatamos um ganho consistente nos resultados, o que sinaliza possibilidades ainda maiores de aprimoramento à medida que novas e avançadas arquiteturas de Redes Neurais forem empregadas.

Outro ponto relevante de nossas descobertas foi o benefício proporcionado pelo uso da camada de *Batch Normalization* no início da rede para imagens sem convolução. Essa técnica contribuiu de forma notável para a estabilização do processo de treinamento da Rede Neural, otimizando sua capacidade de aprender com maior eficiência e precisão. A aplicação bem-sucedida dessa camada mostra como a escolha cuidadosa dos componentes da arquitetura de uma Rede Neural pode resultar em avanços significativos em sua performance, e isso nos abre novas perspectivas para futuras pesquisas no campo do processamento de imagens sísmicas. Também destacamos a clara superioridade da métrica de *Focal Loss* nos treinamentos com imagens sem convolução.

Talvez o aspecto mais promissor de nossos estudos seja a constatação de que as dife-

rentes arquiteturas podem ser treinadas com sucesso para a remoção de múltiplas, inclusive as internas, desde que seja fornecido um volume adequado e representativo de dados de treinamento. Esse resultado é extremamente animador, pois evidencia o potencial da Rede Neural em aprender com diversidade de informações e cenários, proporcionando uma habilidade notável de generalização. A capacidade de operar de forma independente e precisa em diferentes cenários geológicos representa um ganho significativo de uma limitação comum em outros estudos, o que abre novas perspectivas para sua aplicação em um amplo espectro de aplicações práticas. Essa conquista é um estímulo para investigações futuras que almejem ainda mais aprimoramentos e avanços na área.

Para as imagens sísmicas com convolução, o desempenho dos modelos RESUNET++ foi pior do que o das demais arquiteturas, em especial o modelo treinado convencionalmente. Entendemos que isto indica um maior desafio para o treinamento de modelos baseados nesta arquitetura.

Quanto ao tamanho do conjunto de treinamento, identificamos que há uma grande diferença para modelos treinados com menos de 50.000 exemplos, mas uma diferença menor acima disso. Isto indica que em treinamentos futuros pode ser mais interessante buscar criar conjuntos de treinamento ainda mais abrangentes, ao invés de simplesmente aumentar o seu tamanho.

Diferentemente do observado para as imagens sem convolução, a baixa correlação na maioria dos casos entre os resultados de múltiplos treinamentos rápidos para imagens com convolução sugere haver uma forte influência de fatores não-controlados (tais como a inicialização dos pesos da rede ou a separação dos conjuntos de treinamento e de validação) na qualidade final dos modelos assim parametrizados. Uma hipótese é que a estratégia de treinamentos rápidos tenha sido muito agressiva para a convergência dos parâmetros internos das redes mais complexas, que apresentaram menor correlação.

O excelente desempenho exemplificado pela rede MobileNetV2-UNET nas imagens com convolução representa um ponto de partida promissor para novos desenvolvimentos a partir de nosso trabalho. Reconhecemos que ainda há espaço para aprimoramentos e refinamentos, mas, ao avaliar os resultados obtidos até o momento, percebemos que estamos construindo uma base sólida para futuras pesquisas. A abordagem adotada de construir um conjunto de treinamento diversificado através de senoidais parametrizadas, com baixo custo computacional, permitiu uma cobertura eficiente de diferentes modelos geológicos. Essa conquista é particularmente significativa, pois supera uma das principais dificuldades observadas em outros trabalhos de processamento sísmico, onde a generalização para além dos modelos geológicos presentes no treinamento tem sido um desafio recorrente. O resultado do trabalho é animador, pois estimula a busca por melhorias na aplicação de novas Redes Neurais, visando aprimorar seu desempenho e uso em problemas sísmicos desafiadores.

Como investigação futura, uma possibilidade seria a criação de uma versão aprimorada do Simulador Recursivo utilizando o método de ray-tracing. Essa abordagem proporcionaria simulações mais precisas da propagação das frentes de onda no plano, no lugar de repetidas simulações de offset zero. Com a aplicação do ray-tracing, poderíamos obter resultados mais realistas e detalhados, permitindo uma análise mais refinada e minuciosa das características sísmicas. Essa evolução no simulador permitiria a inclusão de exemplos de interações mais complexas entre ondas sísmicas e estruturas geológicas, aprimorando o treinamento das Redes

Neurais com cenários mais realistas.

Outro aspecto a ser explorado em futuras investigações é a aplicação de diferentes tratamentos aos produtos gerados pelo Simulador Recursivo. Seria relevante investigar os efeitos da convolução com diferentes *wavelets* e da adição de ruídos diversos, por exemplo. Essas abordagens poderiam fornecer informações valiosas sobre como a variação das *wavelets* ou a presença de ruídos afetam o desempenho e a capacidade de generalização das Redes Neurais treinadas com esses dados.

Também entendemos que aprimoramentos e novos desenvolvimentos na metodologia para a geração dos modelos geológicos que originam os dados de treinamento podem ser um elemento chave para o treinamento de modelos com capacidade de generalização ainda maior.

Além disso, explorar novas arquiteturas de Redes Neurais para a tarefa de remoção de múltiplas internas e de superfície, inclusive incorporando as arquiteturas aqui estudadas, também é um caminho promissor para pesquisas futuras. O campo de redes neurais profundas está em constante evolução, com novas arquiteturas e técnicas surgindo regularmente. Investigar o desempenho de diferentes arquiteturas pode oferecer entendimentos valiosos para melhorar ainda mais a precisão e a eficácia das Redes Neurais na tarefa de remoção de múltiplas de imagens sísmicas.

Com essas possíveis direções de pesquisa, novos trabalhos podem contribuir significativamente para o avanço da área de processamento sísmico e da aplicação de técnicas de aprendizado de máquina em problemas geofísicos complexos.

Referências

BAO, P. et al. Surface-related and internal multiple elimination using deep learning. *Energies*, MDPI AG, v. 15, p. 3883, 5 2022. Citado na página 11.

BROGGINI, F.; SNIEDER, R. K.; WAPENAAR, K. Connection of scattering principles: Focusing the wavefield without source or receiver. *SEG Technical Program Expanded Abstracts*, 2011. Disponível em: <<https://api.semanticscholar.org/CorpusID:14229427>>. Citado na página 11.

BUGGE, A. J. et al. Demonstrating multiple attenuation with model-driven processing using neural networks. *Leading Edge*, v. 40, p. 831–836, 2021. ISSN 19383789. Citado 2 vezes nas páginas 11 e 21.

CARVALHO, F. M.; WEGLEIN, A. B.; STOLT, R. H. *Examples of a nonlinear inversion method based on the T matrix of scattering theory: Application to multiple suppression*. 2005. 1319–1322 p. Disponível em: <<https://library.seg.org/doi/abs/10.1190/1.1889114>>. Citado na página 11.

CGG. *CGG: Seismic Internal Multiple Attenuation (IMA)*. 2022. Disponível em: <<https://www.cgg.com/geoscience/subsurface-imaging/internal-multiple-attenuation>>. Citado na página 11.

CHEN, L.-C. et al. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. 6 2016. Disponível em: <<http://arxiv.org/abs/1606.00915>>. Citado na página 91.

CHOLLET, F. *Deep Learning with Python*. 2nd. ed. [S.l.]: Manning, 2021. ISBN 9781617296864. Citado na página 15.

DEKKING, F. M. et al. *A Modern Introduction to Probability and Statistics: Understanding Why and How*. Springer, 2005. ISBN 9781852338961. Disponível em: <https://books.google.com.br/books?id=odn7_aUSAnEC>. Citado na página 22.

DICE, L. R. Measures of the amount of ecologic association between species. *Ecology*, v. 26, p. 297–302, 1945. Disponível em: <<https://esajournals.onlinelibrary.wiley.com/doi/abs/10.2307/1932409>>. Citado na página 22.

GROENESTIJN, G. J. van; VERSCHUUR, D. J. Estimating primaries by sparse inversion and application to near-offset data reconstruction. *GEOPHYSICS*, v. 74, p. A23–A28, 2009. Disponível em: <<https://doi.org/10.1190/1.3111115>>. Citado na página 11.

HAMPSON, D. Inverse velocity stacking for multiple elimination. In: _____. *SEG Technical Program Expanded Abstracts 1986*. [s.n.], 2005. p. 422–424. Disponível em: <<https://library.seg.org/doi/abs/10.1190/1.1893060>>. Citado na página 11.

HE, K. et al. Deep residual learning for image recognition. 12 2015. Disponível em: <<http://arxiv.org/abs/1512.03385>>. Citado na página 93.

- HU, J. et al. Squeeze-and-excitation networks. 9 2017. Disponível em: <<http://arxiv.org/abs/1709.01507>>. Citado na página 91.
- JHA, D. et al. Resunet++: An advanced architecture for medical image segmentation. 11 2019. Disponível em: <<http://arxiv.org/abs/1911.07067>>. Citado 2 vezes nas páginas 94 e 95.
- JO, H. et al. Machine-learning-based porosity estimation from multifrequency poststack seismic data. *GEOPHYSICS*, v. 87, p. M217–M233, 9 2022. ISSN 0016-8033. Citado na página 16.
- KEAREY, P.; BROOKS, M.; HILL, I. *An Introduction to Geophysical Exploration*. 3. ed. [S.l.]: Wiley-Blackwell, 2013. Citado 3 vezes nas páginas 10, 96 e 97.
- KERAS. *BatchNormalization layer*. 2022. Disponível em: <https://keras.io/api/layers/normalization_layers/batch_normalization/>. Citado na página 91.
- KLATT, M. et al. Deep learning strategy for salt model building. *GEOPHYSICS*, p. 1–51, 9 2022. ISSN 0016-8033. Citado na página 16.
- LANGTANGEN, H. P.; LINGE, S. *Finite Difference Computing with PDEs*. [S.l.]: Springer, 2017. Citado na página 19.
- LI, H. et al. Pyramid attention network for semantic segmentation. 5 2018. Disponível em: <<http://arxiv.org/abs/1805.10180>>. Citado na página 92.
- LI, J.; WU, X.; HU, Z. Deep learning for simultaneous seismic image super-resolution and denoising. *IEEE Transactions on Geoscience and Remote Sensing*, v. 60, p. 1–11, 2022. Citado na página 15.
- LIN, T.-Y. et al. Focal loss for dense object detection. 8 2017. Disponível em: <<http://arxiv.org/abs/1708.02002>>. Citado 2 vezes nas páginas 22 e 86.
- MILLETARI, F.; NAVAB, N.; AHMADI, S.-A. V-net: Fully convolutional neural networks for volumetric medical image segmentation. 6 2016. Disponível em: <<http://arxiv.org/abs/1606.04797>>. Citado na página 86.
- MIRIYALA, G. P. et al. A review on recent machine learning algorithms used in cad diagnosis. In: . [S.l.: s.n.], 2021. p. 269–274. Citado na página 22.
- MULLER, A. P. O. et al. Complete identification of complex salt-geometries from inaccurate migrated subsurface offset gathers using deep learning. *GEOPHYSICS*, p. 1–50, 9 2022. ISSN 0016-8033. Citado na página 16.
- MURPHY, A. H. The finley affair: A signal event in the history of forecast verification. *Weather and Forecasting*, v. 11, p. 3–20, 3 1996. Citado na página 22.
- POWERS, D. Evaluation: From precision, recall and f-factor to roc, informedness, markedness correlation. *Mach. Learn. Technol.*, v. 2, 1 2008. Citado na página 22.
- QIAN, H. et al. Research on noise suppression technology of marine optical fiber towed streamer seismic data based on resunet. *Energies*, v. 15, 2022. ISSN 1996-1073. Disponível em: <<https://www.mdpi.com/1996-1073/15/9/3362>>. Citado na página 93.

RONNEBERGER, O.; FISCHER, P.; BROX, T. U-net: Convolutional networks for biomedical image segmentation. 5 2015. Disponível em: <<http://arxiv.org/abs/1505.04597>>. Citado 2 vezes nas páginas 92 e 93.

RUSSEL, S.; NORVIG, P. *Artificial Intelligence: A Modern Approach*. [S.l.: s.n.], 2021. Citado na página 13.

VASWANI, A. et al. Attention is all you need. *CoRR*, abs/1706.03762, 2017. Disponível em: <<http://arxiv.org/abs/1706.03762>>. Citado na página 92.

VERSCHUUR, D. J.; BERKHOUT, A. J.; WAPENAAR, C. P. A. Adaptive surface-related multiple elimination. *GEOPHYSICS*, v. 57, p. 1166–1177, 1992. Disponível em: <<https://doi.org/10.1190/1.1443330>>. Citado na página 11.

WANG, W.; YANG, F.; MA, J. Automatic salt detection with machine learning. *European Association of Geoscientists and Engineers*, v. 2018, p. 1–5, 2018. ISSN 2214-4609. Disponível em: <<https://www.earthdoc.org/content/papers/10.3997/2214-4609.201800917>>. Citado na página 92.

WANG, X. et al. Random noise attenuation by self-supervised learning from single seismic data. *Mathematical Geosciences*, 2022. ISSN 1874-8953. Disponível em: <<https://doi.org/10.1007/s11004-022-10032-y>>. Citado 2 vezes nas páginas 11 e 15.

WEGLEIN, A. B. Primaries — the only events that can be migrated and for which migration has meaning. *The Leading Edge*, v. 34, p. 808–813, 2015. Disponível em: <<https://doi.org/10.1190/tle34070808.1>>. Citado na página 19.

YORK, Z. et al. Applications of supervised deep learning for seismic interpretation and inversion. *Leading Edge (Tulsa, OK)*, v. 38, p. 526 – 533, 2019. ISSN 1070-485X. Disponível em: <<https://search.ebscohost.com/login.aspx?direct=true&AuthType=ip,sso&db=edsgsw&AN=edsgsw.2020.007098&lang=pt-br&site=eds-live&authtype=ip,sso&custid=s7173344&groupid=main&profile=eds>>. Citado na página 15.

ZHANG, Z.; LIU, Q.; WANG, Y. Road extraction by deep residual u-net. 11 2017. Disponível em: <<http://arxiv.org/abs/1711.10684><http://dx.doi.org/10.1109/LGRS.2018.2802944>>. Citado 2 vezes nas páginas 93 e 94.

Apêndices

APÊNDICE A – Aprendizado de Máquina

A.1 IoU

No nosso trabalho, a métrica *Intersection over Union* (também conhecida como Índice Jaccard) nos dá a razão entre o número de pontos que foram corretamente classificados como reflexões primárias, em relação à união dos pontos que são de fato reflexões primárias com todos os pontos que foram estimados como sendo reflexões primárias.

$$IoU = \frac{y \cap y_{pred}}{y \cup y_{pred}} = \frac{TP}{TP + FP + FN} \quad (A.1)$$

A.2 Focal Loss

A *Focal Loss* (LIN et al., 2017) é uma versão da Entropia Cruzada Binária em que um peso é atribuído a cada classe, de maneira que as classes com maior dificuldade de classificação tem um peso relativo maior na função de custo. Isso permite que o modelo aprenda rapidamente a melhorar essas classes, acelerando o treinamento da rede.

$$FL(p_t) = -(1 - p_t)^\gamma \log(p_t) \quad (A.2)$$

Quando $\gamma > 0$, o coeficiente diminui rapidamente para classes cuja probabilidade é muito alta, mas não tão rapidamente para classes cuja probabilidade é baixa.

A.3 Dice Loss

A *Dice Loss* (MILLETARI; NAVAB; AHMADI, 2016) é capaz de comparar a semelhança entre duas máscaras mesmo que possuam classes muito desbalanceadas e é muito usada em segmentação de imagens. Ela é dada por:

$$\frac{2TP}{2TP + FP + FN} \quad (A.3)$$

A.4 Sensitividade

A Sensitividade nos diz, dentre todos os pontos que contém reflexões primárias, qual proporção foi corretamente identificada.

$$Sensitividade = \frac{TP}{TP + FN} \quad (A.4)$$

A.5 Precisão

A Precisão nos diz, dentre todos os pontos identificados como contendo reflexões primárias, qual proporção foi corretamente identificada.

$$Precisao = \frac{TP}{TP + FP} \quad (A.5)$$

A.6 Engenharia de Características

A engenharia de características (*feature engineering*) é a criação de novas características a partir do dado original. Nas primeiras redes neurais que usavam convoluções para identificar padrões em imagens, por exemplo, os filtros tinham valores fixos, definidos manualmente, ao invés de serem aprendidos a partir dos dados de treinamento.

A.7 Curva Logística

A Curva Logística, também conhecida como sigmoide) é definida a partir dos parâmetros x_0 (ponto médio) e k (declividade da curva):

$$f(x) = \frac{1}{1 + e^{-k(x-x_0)}} \quad (A.6)$$

APÊNDICE B – Redes Neurais

B.1 Treinamento da Rede Neural

A etapa de *backpropagation* é onde acontece o aprendizado da Rede Neural. Após uma etapa de avaliação da performance da Rede (chamada de *forward pass*), na qual o sinal de entrada é propagado até a saída, obtém-se a medida do quanto a Rede errou em relação à saída esperada. Na *backpropagation* o gradiente do erro é calculado em relação a todos os pesos da rede e então os pesos são todos ajustados na direção de minimização do erro.

Dado um erro E na saída da última camada, seu gradiente será, para cada neurônio $n_{i,-1}$ na última camada, $\partial E / \partial n_{i,-1}$. Para os neurônios $n_{j,-2}$ da penúltima camada, serão $\sum_i (\partial E / \partial n_{i,-1}) (\partial n_{i,-1} / \partial n_{j,-2})$. E assim por diante. Como o cálculo do gradiente é feito “de trás pra frente” (da camada de saída para a camada de entrada) seguindo a regra da cadeia, deu-se o nome de *backpropagation* a esta etapa.

B.2 Convolução

A camada de convolução realiza a aplicação de um filtro em todos os pontos da representação da imagem na sua entrada, por canal, como vemos na Figura 104.

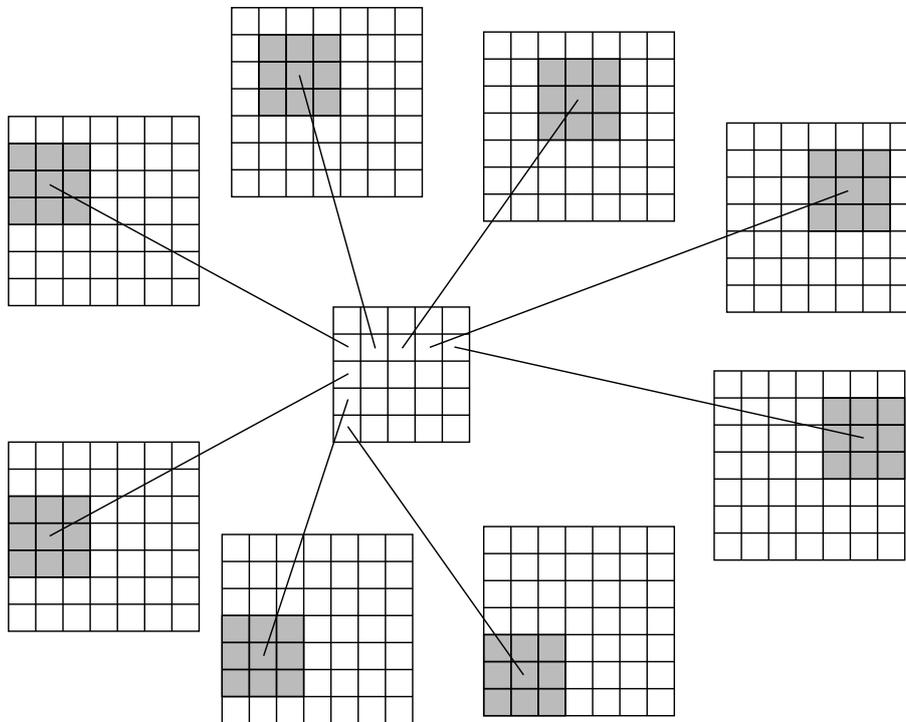


Figura 104 – Convolução: o filtro é aplicado em diferentes áreas (cinza), tendo o resultado de seu produto escalar armazenado em uma célula referente a seu ponto central.

A aplicação de um filtro gera uma nova representação da imagem. Assim, múltiplos filtros geram múltiplas representações. A essas representações denominamos “canais”. Em cada camada da rede temos a passagem simultânea de múltiplos canais da imagem.

O resultado da convolução nos diz, para cada região da entrada, a similaridade com o padrão presente no filtro. Então, conforme os exemplos citados acima, na primeira camada convolucional poderia haver um filtro que nos informa em quais pontos da imagem há bordas horizontais, outro que nos informa onde está presente uma determinada textura, outro informa onde existem quinas, etc. Se aplicarmos uma outra camada convolucional na saída desta primeira camada, teremos filtros que detectam características mais abstratas, tais como o desenho traçado por uma borda, o formato de uma área preenchida por uma textura, etc. Assim, Redes Neurais Convolucionais (CNN em inglês) são capazes de identificar padrões nas imagens com alto grau de sofisticação.

B.3 Max Pooling

A operação de *Max Pooling* extrai de uma área da representação da imagem (normalmente um quadrado 2x2) o maior valor. Assim, a representação da imagem é reduzida à metade de seu tamanho original, retendo os sinais mais fortes, como pode ser visto na Figura 105. Dessa forma, a próxima camada da rede terá, numa determinada área, informações referentes a uma área 4 vezes maior na imagem original. Assim, características presentes em distâncias maiores, como a repetição de uma textura ou o contorno de um rosto, podem ser identificadas nas camadas subsequentes.

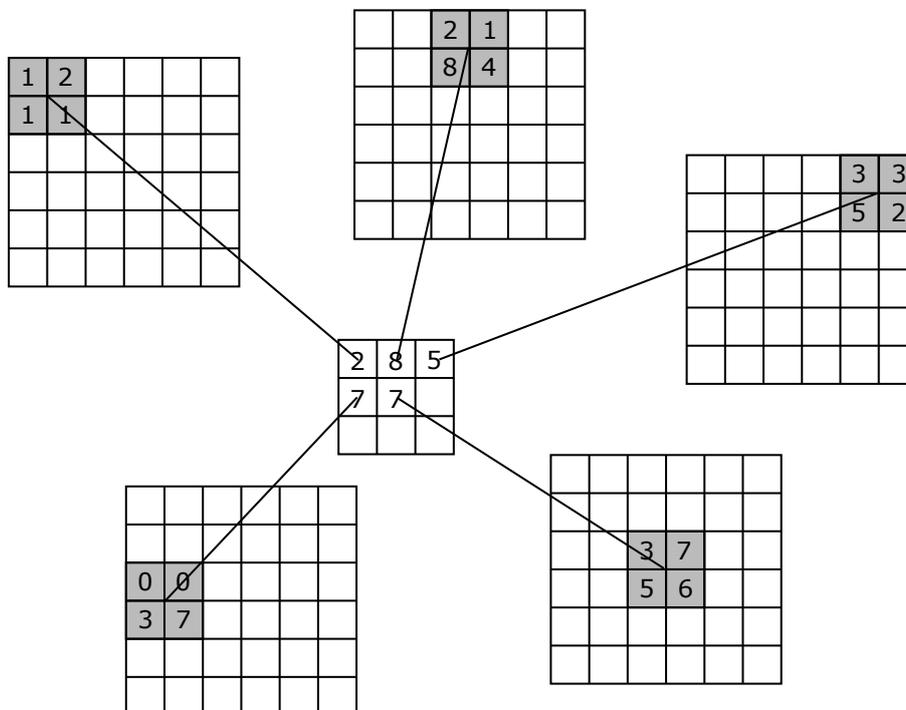


Figura 105 – Max Pooling: o valor máximo de cada área (cinza) é armazenado em sua posição relativa às demais áreas.

B.4 Convolução Transposta

Convolução transposta é um tipo de convolução que resulta numa imagem com o dobro do tamanho original. Num primeiro passo, novos pontos são inseridos entre cada ponto da imagem original, com valor zero, como visto na Figura 106. Então é aplicada uma convolução, a qual irá gerar uma representação da imagem em maior escala.

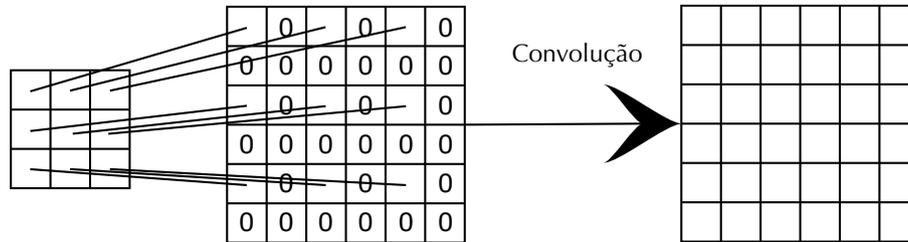


Figura 106 – Convolução Transposta: duplicação do tamanho da imagem usando zeros, seguida de convolução.

B.5 Vanishing Gradient

Pela regra da cadeia, conforme as redes se tornam muito profundas o gradiente pode se tornar muito pequeno para poder ser efetivo no ajuste dos pesos da rede. Ou seja, a rede pode ficar incapaz de aprender. Várias técnicas foram desenvolvidas para solucionar este problema (vide B.6 e B.7), o que permitiu o treinamento de redes profundas.

B.6 Conexão Residual

A conexão residual é uma reentrada de informação na rede. Isso permite que a rede “lembre”, após uma operação, do dado sobre o qual ela operou. Essa é uma das técnicas utilizadas para resolver o problema do *Vanishing Gradient*.

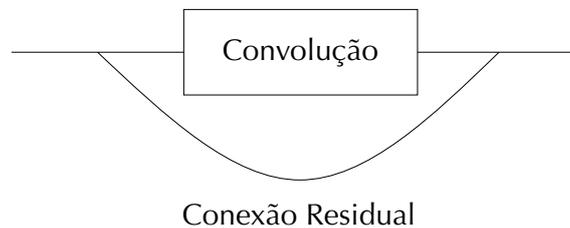


Figura 107 – Conexão residual sobre uma convolução.

B.7 Batch Normalization

A camada de *Batch Normalization* realiza uma normalização do sinal propagado pela rede, o que ajuda a evitar o problema de *Vanishing Gradient*. Ao passar pela camada de *Batch Normalization*, o sinal terá valor médio próximo de 0 e desvio padrão próximo de 1. Os valores

não são exatos, pois os coeficientes de ajuste são calculados para um lote de entradas durante o treinamento, e são fixados pelos valores de treinamento durante a inferência (KERAS, 2022).

B.8 Squeeze and Excitation

As camadas de *Squeeze and Excitation* (HU et al., 2017) identificam e amplificam os canais mais importantes na representação da imagem. Para cada canal é atribuído um peso, e os canais são multiplicados por esses pesos. A rede aprende a fazer essa distribuição de pesos de maneira que os canais mais importantes para o aprendizado são reforçados, enquanto que canais menos importantes tem seu sinal enfraquecido. De certa forma, a ação das camadas de *Squeeze and Excitation* (Fig. 108) é análoga à das camadas de atenção: ambas ajudam a rede a reforçar informação relevante e a ignorar informação irrelevante, porém atuam em níveis diferentes da informação.

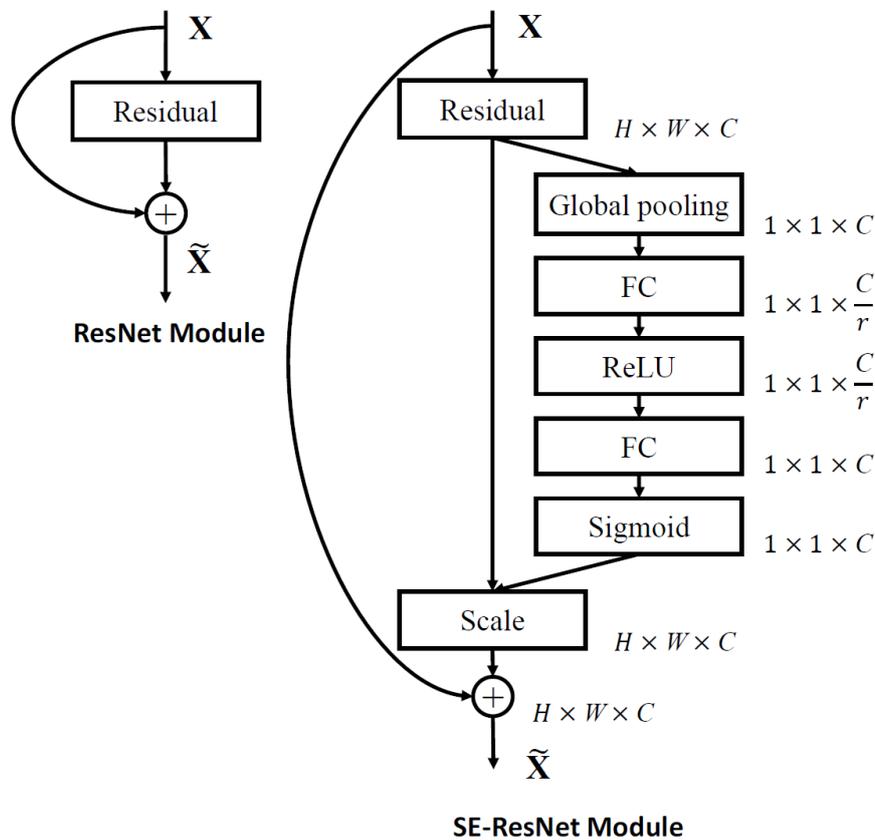


Figura 108 – Bloco residual com *Squeeze and Excitation*. Imagem de (HU et al., 2017).

B.9 Atrous Spatial Pyramidal Pooling (ASPP)

As camadas de *Atrous Spatial Pyramidal Pooling* (ASPP) (CHEN et al., 2016), são capazes de extrair informações em várias escalas. Em uma *Atrous Convolution*, também conhecida como “Convolução Dilatada”, a convolução é feita sobre pontos espaçados, “enxergando” uma região maior da imagem sem incorrer em maior consumo de processamento ou memória do que uma

convolução tradicional. A ASPP faz várias Convoluções Dilatadas (Fig. 109) em paralelo, cada uma com um espaçamento diferente entre os pontos observados, e concatena os resultados. Assim, pode detectar padrões relevantes em diferentes escalas.

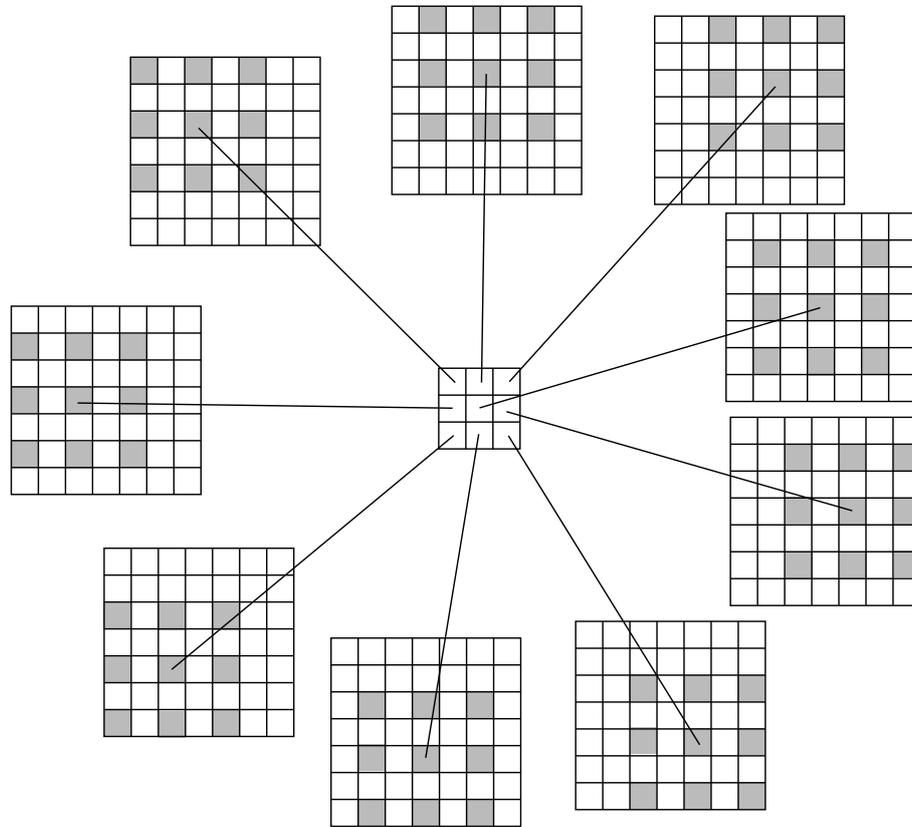


Figura 109 – Convolução Dilatada: pontos espaçados (cinza) ao invés de área contínua.

B.10 Camadas de Atenção

Os mecanismos de Atenção (VASWANI et al., 2017), (LI et al., 2018) podem ser entendidas por uma analogia com a visão: assim como nossos olhos veem uma imagem focalizada no centro de nossa visão e borrada ao redor, os mecanismos de Atenção aprendem a identificar quais partes da imagem precisam de mais destaque, aumentando seu peso, enquanto diminuem o peso das partes menos importantes. Assim, melhoram a qualidade das características a serem tratadas pela Rede. São utilizados nos blocos de decodificação na RESUNET++, antes das Convoluções Transpostas, para identificar as regiões mais importantes nas representações da imagem.

B.11 UNET

Uma rede convolucional muito eficiente é a UNET (RONNEBERGER; FISCHER; BROX, 2015) (WANG; YANG; MA, 2018), a qual podemos dividir em três partes: uma “descida” formada por convoluções (codificador) e *Max Poolings*, e depois uma subida (decodificador)

formada por convoluções transpostas, formando um “U” (o que dá origem ao nome deste modelo), conforme vemos na Figura 110.

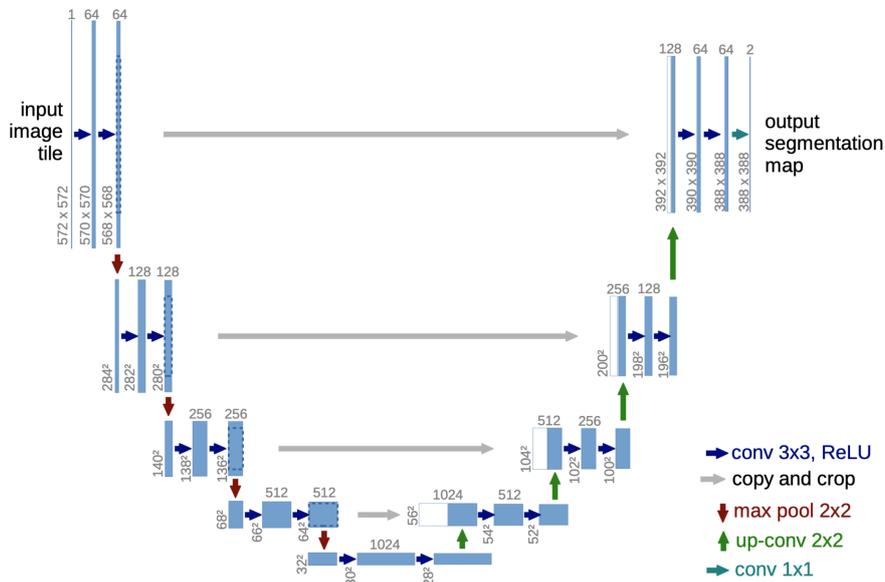


Figura 110 – UNET. Imagem de (RONNEBERGER; FISCHER; BROX, 2015)

B.12 MobileNetV2-UNET

A biblioteca *Segmentation Models*¹ oferece a possibilidade de usar codificadores pré-treinados no conjunto de dados “2012 ILSVRC ImageNet” em conjunto com diversos decodificadores. A MobileNetV2-UNET, como o nome diz, é formada pelo codificador MobileNet V2 e pelo decodificador da UNET.

B.13 RESUNET

A rede convolucional RESUNET (ZHANG; LIU; WANG, 2017) (QIAN et al., 2022) (Fig. 111) é derivada da UNET, inserindo conexões residuais dentro dos blocos convolucionais. Um dos problemas em redes com muitas camadas é o risco do gradiente da rede se tornar tão pequeno que a rede não consegue ajustar seus pesos. Para resolver esse problema, He et al. (HE et al., 2015) propuseram usar ligações que “pulam” uma ou mais camadas da rede, permitindo que o sinal de camadas anteriores seja “lembrado” pela rede. A essas conexões chamamos de “Conexões Residuais”.

¹ https://github.com/qubvel/segmentation_models

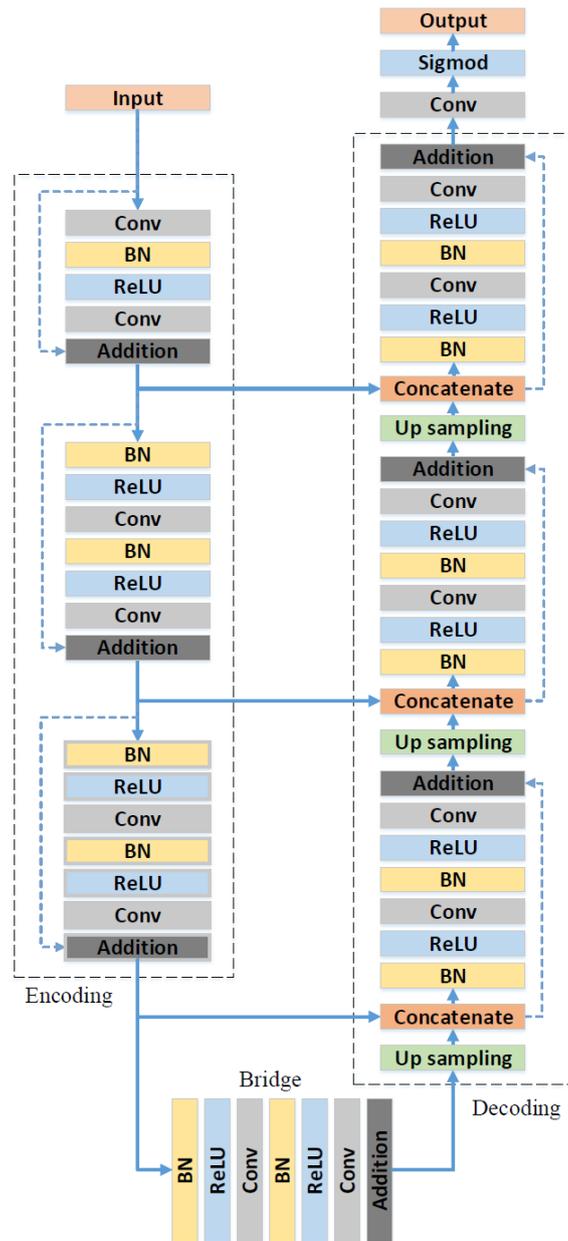


Figura 111 – RESUNET. Imagem de (ZHANG; LIU; WANG, 2017)

B.14 RESUNET++

A RESUNET++ (JHA et al., 2019) é uma rede baseada na RESUNET, com a inclusão de camadas de *Squeeze and Excitation*, *Atrous Spatial Pyramidal Pooling* e de “Atenção”. Como toda arquitetura descendente da UNET, possui três partes: uma “descida” na qual a imagem é codificada, uma base na qual temos a representação mais abstrata da imagem, e uma “subida” na qual a imagem é decodificada, usando convoluções transpostas, de volta ao tamanho original.

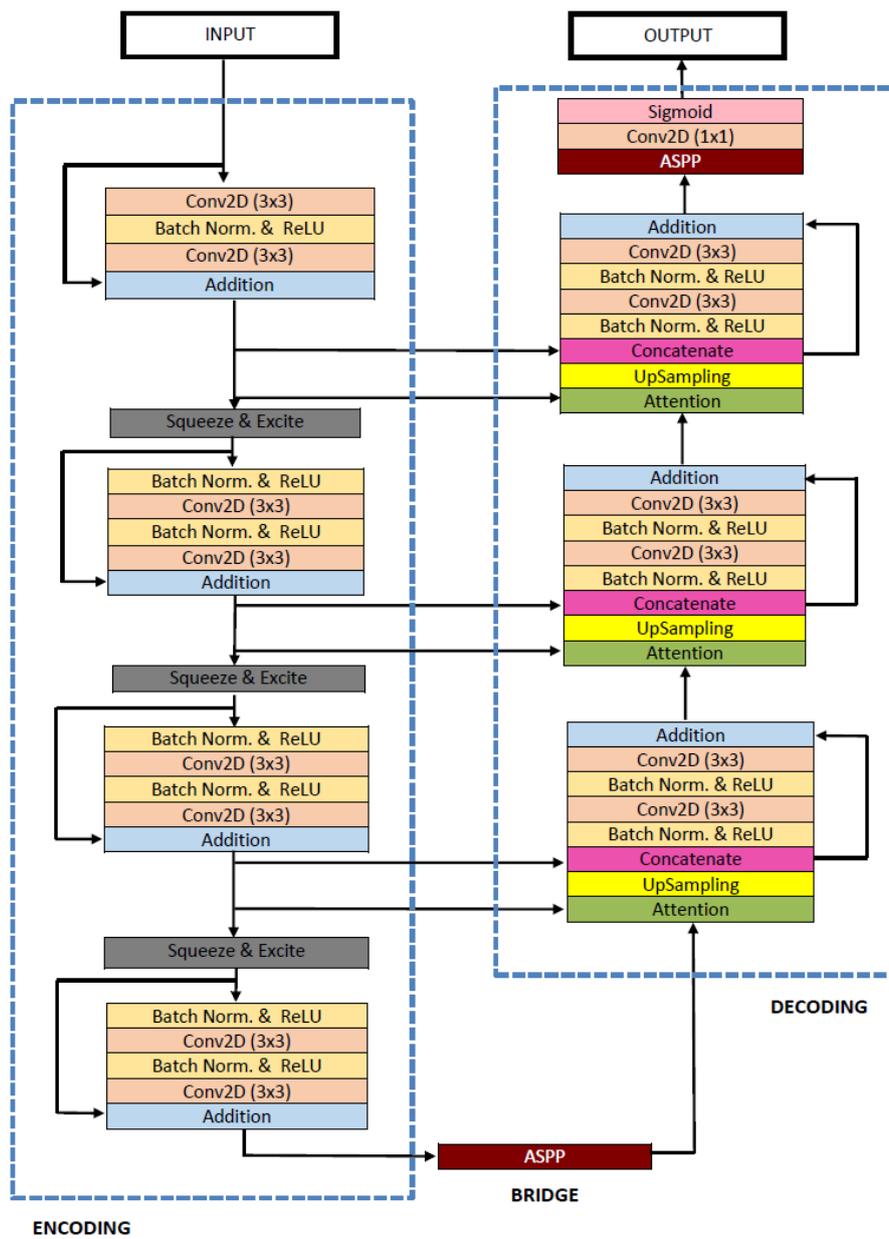


Figura 112 – RESUNET++. Imagem de (JHA et al., 2019)

APÊNDICE C – Geofísica

C.1 Reflexão da Frente de Onda

Em um meio com densidade ρ em que ondas acústicas (paralelas à direção de propagação) viajam com velocidade v , a impedância Z é dada por:

$$Z = \rho v \quad (\text{C.1})$$

Quando uma frente onda acústica atinge perpendicularmente a interface entre dois meios de impedâncias diferentes, Z_A e Z_B , a amplitude A_r da onda refletida de uma onda incidente de amplitude A_i é dada por:

$$A_r = A_i \frac{Z_B - Z_A}{Z_B + Z_A} \quad (\text{C.2})$$

Ou seja, quanto maior a diferença entre as impedâncias, maior a parcela da onda que será refletida (se $Z_1 > Z_2$, a onda é refletida com uma diferença de fase π) (KEAREY; BROOKS; HILL, 2013).

C.2 Função Recursiva

A função recursiva utilizada no nosso simulador segue a seguinte lógica:

```

FUNÇÃO PULSAR(camada, tempo, pulso, sentido, com_múltiplas):
  SE sentido == descida:
    ida = PULSAR(PROXIMA_CAMADA(camada),
      tempo+TEMPO_PERCURSO(camada),
      pulso*ATENUAÇÃO(camada)*
      COEFICIENTE_TRANSMISSÃO_DESCENDO(camada),
      descida,
      com_múltiplas)
    vinda = PULSAR(camada,
      tempo+TEMPO_PERCURSO(camada),
      pulso*ATENUAÇÃO(camada)*
      COEFICIENTE_REFLEXÃO_DESCENDO(camada),
      subida,
      com_múltiplas)
  SE sentido == subida:
    SE com_múltiplas == verdadeiro:
      ida = PULSAR(camada,
        tempo+TEMPO_PERCURSO(camada),
        pulso*ATENUAÇÃO(camada)*

```

```

        COEFICIENTE_REFLEXAO_SUBINDO(camada),
        descida,
        com_múltiplas)
SE camada == primeira:
    RETORNAR (tempo+TEMPO_PERCURSO(camada),
             pulso*ATENUAÇÃO(camada))
SENÃO:
    vinda = PULSAR(CAMADA_ANTERIOR(camada),
                  tempo+TEMPO_PERCURSO(camada),
                  pulso*ATENUAÇÃO(camada)*
                  COEFICIENTE_TRANSMISSÃO_SUBINDO(camada),
                  subida,
                  com_múltiplas)
RETORNAR ida + vinda

```

Sendo a atenuação da amplitude A_0 do pulso ao atravessar uma camada dada por¹:

$$A_{profundidade} = A_0 e^{-0,625 \frac{profundidade_{camada}}{velocidade_{camada}}} \quad (C.3)$$

O coeficiente de reflexão de uma camada A para uma camada B é dado por (KEAREY; BROOKS; HILL, 2013, p. 29):

$$A_r = A_i \frac{\rho_B v_B - \rho_A v_A}{\rho_B v_B + \rho_A v_A} \quad (C.4)$$

E o coeficiente de transmissão por:

$$A_t = A_i \frac{2\rho_A v_A}{\rho_B v_B + \rho_A v_A} \quad (C.5)$$

Quando o pulso está descendo, A é a camada superior e B é a camada inferior. Na subida, é o contrário.

A variável booleana `com_múltiplas` indica se queremos obter a imagem sísmica com ou sem múltiplas.

C.3 Modelos de Subsuperfície

Os modelos de subsuperfícies foram criados com o código

```

while z < nz-prof_max:
    d = np.random.randint(prof_min,prof_max)
    a = np.random.randint(0,prof_max)
    k = np.pi/np.random.randint(50,nx)

```

¹ Consideramos um pulso de $25Hz$ e uma atenuação de energia de $0,5dB\lambda^{-1}$. (KEAREY; BROOKS; HILL, 2013, p. 27)

```

fase = np.random.random()*np.pi*2
velocidade = np.random.randint(v_min,v_max)
rho = np.random.random()*(rho_max-rho_min)+rho_min
z = z + d
for x in range(nx):
    profundidade = int(nz-z-a*np.sin(k*x+fase))
    if profundidade < 0:
        profundidade = 0
    rast[x,:profundidade,0] = velocidade
    rast[x,:profundidade,1] = rho

```

Onde nz é a profundidade do fundo do modelo (configurada em 10.000), $prof_min$ e $prof_max$ são os tamanhos mínimo e máximo de uma interface entre camadas (configurados em 500 e 1.000, respectivamente), nx é a largura do modelo (512), v_min e v_max são as velocidades mínima e máxima que uma camada pode apresentar, rho_min e rho_max são as densidades mínima e máxima.

C.4 Convolução de imagens sísmicas

Para geração das imagens sísmicas convoluídas usamos usamos uma função² para gerar a *wavelet* Ricker centrada em $25Hz$ e a função `convolve` da biblioteca `scipy.signal`:

```

def ricker(f, length, dt):
    t = np.arange(-length/2, (length-dt)/2, dt)
    y = (1.0 - 2.0*(np.pi**2)*(f**2)*(t**2)) \
        * np.exp(-(np.pi**2)*(f**2)*(t**2))
    return y

def convolve_img(img):
    convolved = np.zeros(img.shape)
    wavelet = ricker(25, 3, 3/256)
    for i in range(convolved.shape[0]):
        convolved[i] = convolve(img[i], wavelet, mode='same')
    return convolved

```

² adaptada de https://subsurfwiki.org/wiki/Ricker_wavelet