Pontifícia Universidade Católica
do Rio de Janeiro

**Daniel Cesar Bosco de Miranda**

# Vision Transformers e Masked Autoenconders para segmentação de fácies sísmicas

**Dissertação de Mestrado**

Dissertação apresentada como requisito parcial para a obtenção do grau de Mestre pelo Programa de Pós–graduação em Informática da PUC-Rio.

Orientador: Prof. Marcelo Gattass

Rio de Janeiro
Abril de 2023

Pontifícia Universidade Católica
do Rio de Janeiro

**Daniel Cesar Bosco de Miranda**

# Vision Transformers e Masked Autoenconders para segmentação de fácies sísmicas

Dissertação apresentada como requisito parcial para a obtenção do grau de Mestre pelo Programa de Pós–graduação em Informática da PUC-Rio. Aprovada pela Comissão Examinadora abaixo:

**Prof. Marcelo Gattass**
Orientador
Departamento de Informática – PUC-Rio

**Prof. Alberto Barbosa Raposo**
Departamento de Informática – PUC-Rio

**Prof. Aristófanes Corrêa Silva**
UFMA

**Dr. Marcos de Carvalho Machado**
Petrobras

**Dr. Jan Jose Hurtado Jauregui**
Departamento de Informática – PUC-Rio

Rio de Janeiro, 28 de Abril de 2023

**Daniel Cesar Bosco de Miranda**

Graduado em Física pela Universidade Federal do Rio Grande do Norte - UFRN.

Ficha Catalográfica

Bosco de Miranda, Daniel Cesar

Vision Transformers e Masked Autoenconders para segmentação de fácies sísmicas / Daniel Cesar Bosco de Miranda; orientador: Marcelo Gattass. – 2023.

57 f: il. color. ; 30 cm

Dissertação (mestrado) - Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Informática, 2023.

Inclui bibliografia

1. Informática – Teses. 2. Sísmica. 3. Vision Transformers. 4. Aprendizado auto-supervisionado. 5. Masked autoencoders. I. Gattass, Marcelo. II. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Informática. III. Título.

CDD: 004

Ao meu pai, Fernando Cesar de Miranda.

## Agradecimentos

Agradeço aos meus pais, Fernando e Dirce, por todo o amor que me deram. Pelo exemplo e ensinamentos, que norteiam a minha vida.

A meus irmãos Lúcia, Fernanda e Felipe pela amizade e amor.

A Alan Albano pela ajuda e ideias ao longo de todo o trabalho. À minha gerente, Milena Frej, e ao meu coordenador Luis Henrique por terem me dado a oportunidade de fazer esse Mestrado.

Ao Professor Marcelo Gattass, meu orientador. A Luiz Fernando, meu coorientador, pelo incentivo e apoio constante. A Jônatas Wehrmann, pelos ensinamentos na etapa inicial deste trabalho.

Ao Professor Eduardo Miranda do IFGW/Unicamp, pelo exemplo e conselhos ao longo dos últimos anos.

À Petrobras por permitir a realização desse mestrado e aos colegas de empresa que me ensinaram muito ao longo dos anos. Em especial a Robson Lopes Prates.

Ao Centro de Supercomputação do Senai Simatec, pelo gerenciamento dos recursos computacionais que possibilitaram a realização desse trabalho.

À Tecgraf e a PUC-Rio, pelo aprendizado proporcionado ao longo dos últimos dois anos.

Por fim, agradeço ao amor e paciência da minha família, Analena, Ivan e Isadora, sem a qual esse trabalho não poderia ter sido realizado.

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

# Resumo

Bosco de Miranda, Daniel Cesar; Gattass, Marcelo. **Vision Transformers e Masked Autoenconders para segmentação de fácies sísmicas**. Rio de Janeiro, 2023. 57p. Dissertação de Mestrado – Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

O desenvolvimento de técnicas de aprendizado auto-supervisionado vem ganhando muita visibilidade na área de Visão Computacional pois possibilita o pré-treinamento de redes neurais profundas sem a necessidade de dados anotados. Em alguns domínios, as anotações são custosas, pois demandam muito trabalho especializado para a rotulação dos dados. Esse problema é muito comum no setor de Óleo e Gás, onde existe um vasto volume de dados não interpretados. O presente trabalho visa aplicar a técnica de aprendizado auto-supervisionado denominada *Masked Autoencoders* para pré-treinar modelos *Vision Transformers* com dados sísmicos. Para avaliar o pré-treino, foi aplicada a técnica de transfer learning para o problema de segmentação de fácies sísmicas. Na fase de pré-treinamento foram empregados quatro volumes sísmicos distintos. Já para a segmentação foi utilizado o dataset *Facies-Mark* e escolhido o modelo da literatura *Segmentation Transformers*. Para avaliação e comparação da performance da metodologia foram empregadas as métricas de segmentação utilizadas pelo trabalho de benchmarking de ALAUDAH (2019). As métricas obtidas no presente trabalho mostraram um resultado superior. Para a métrica *frequency weighted intersection over union*, por exemplo, obtivemos um ganho de 7.45% em relação ao trabalho de referência. Os resultados indicam que a metodologia é promissora para melhorias de problemas de visão computacional em dados sísmicos.

## Palavras-chave

Sísmica; Vision Transformers; Aprendizado auto-supervisionado; Masked autoencoders.

## Abstract

Bosco de Miranda, Daniel Cesar; Gattass, Marcelo (Advisor). **Vision Transformers and Masked Autoenconders for seismic faceis segmentation**. Rio de Janeiro, 2023. 57p. Dissertação de Mestrado – Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

The development of self-supervised learning techniques has gained a lot of visibility in the field of Computer Vision as it allows the pre-training of deep neural networks without the need for annotated data. In some domains, annotations are costly, as they require a lot of specialized work to label the data. This problem is very common in the Oil and Gas sector, where there is a vast amount of uninterpreted data. The present work aims to apply the self-supervised learning technique called Masked Autoencoders to pre-train Vision Transformers models with seismic data. To evaluate the pre-training, transfer learning was applied to the seismic facies segmentation problem. In the pre-training phase, four different seismic volumes were used. For the segmentation, the Facies-Mark dataset was used and the Segmentation Transformers model was chosen from the literature. To evaluate and compare the performance of the methodology, the segmentation metrics used by the benchmarking work of ALAUDAH (2019) were used. The metrics obtained in the present work showed a superior result. For the frequency weighted intersection over union (FWIU) metric, for example, we obtained a gain of 7.45% in relation to the reference work. The results indicate that the methodology is promising for improving computer vision problems in seismic data.

## Keywords

Seismic; Vision Transformers; Self-supervised learning; Masked autoencoders.

# Sumário

# Lista de figuras

# Lista de tabelas

## Lista de Abreviaturas

AC – Acurácia por classe

AP – Acurácia por pixel

CNN – Convolutional Neural Network

FWIU – Frequency Weighted Intersection over Union

IoU – Intersection over Union

MAC – Média da acurácia por classe

MAE – Masked Autoencoders

MLP – Multilayer Perceptron

ReLU – Rectified Linear Unit

SETR – Segmentation Transformers

ViT – Vision Transformer

*"The best way out is always through."*

**Robert Frost**, *A Servant To Servants.*

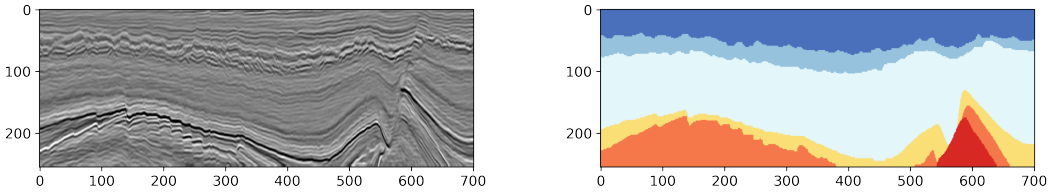
# 1
# Introdução

A interpretação de volumes sísmicos é um trabalho fundamental para toda a cadeia de exploração e produção de campos de petróleo e gás. Essa atividade demanda um esforço considerável de geocientistas experientes, sendo uma tarefa cara e repetitiva. Os volumes muitas vezes abrangem áreas com milhares de quilômetros quadrados, e o trabalho de interpretação de horizontes, falhas e fácies sísmicas pode demorar meses (DELLINGER et al., 2017).

Isso tem levado a uma demanda cada vez maior por soluções automatizadas, que otimizem a descoberta de oportunidades e aproveitem a enorme quantidade de dados que as empresas do setor possuem. Dentro desse contexto, nos últimos anos surgiram diversos trabalhos aplicando redes neurais na tentativa de automatizar essas tarefas, tornando-as mais eficientes e menos subjetivas (WALDELAND et al., 2018; ALAUDAH et al., 2019; SHI; WU; FOMEL, 2019; WU et al., 2019; GAO; WU; LIU, 2021; KAUR et al., 2023). Em particular, o interesse dessa dissertação está em trabalhos que têm aplicado redes neurais convolucionais para a interpretação de fácies sísmicas utilizando segmentação semântica (ALAUDAH et al., 2019; KAUR et al., 2023).

Uma dificuldade para a utilização em larga escala dos dados sísmicos disponíveis é a pouca quantidade de dados interpretados/anotados. No entanto, o recente desenvolvimento de novas técnicas de aprendizado auto-supervisionado tem permitido o treinamento de redes neurais profundas sem a necessidade de anotação. Treinar redes profundas com dados específicos do domínio de interesse é algo que vem sendo aplicado com sucesso na classificação e segmentação de imagens médicas (AZIZI et al., 2022; TANG et al., 2022; HAGHIGHI et al., 2022), e está começando a ser abordado utilizando dados sísmicos (LI et al., 2023).

O presente trabalho consiste na aplicação de uma técnica de aprendizado auto-supervisionado chamada *Masked Autoencoders* (MAE), proposta por He et al. (2022), para pré-treinar Vision Transformer (ViT) (DOSOVITSKIY et al., 2020) com volumes sísmicos. Em seguida, é aplicado o transfer learning para realizar a segmentação semântica de fácies sísmicas. Nessa tarefa, serão empregados modelos que utilizam a ViT como *backbone*. Para avaliar o desempenho dessas redes, foi utilizado o trabalho de referência de Alaudah et al. (2019), que analisou diferentes modelos de redes neurais convolucionais para segmentação de fácies no F3 Block. Na Figura 1.1 são exemplificadas uma imagem sísmica e as fácies anotadas na imagem.




(a) Sísmica (b) Fácies anotadas

Figura 1.1: Exemplo de (a) imagem sísmica e (b) fácies interpretadas. Fonte: Alaudah et al. (2019).

Dada a relevância crescente dos modelos ViTs na área de visão computacional e seu aplicação ainda inicial em problemas envolvendo dados sísmicos, este trabalho visa contribuir para a invetigação da relevância desse tipo de arquitetura nesse domínio de images. Além de avaliar se a ViT é competitiva com as redes neurais convolucionais na segmentação de fácies sísmicas, busca-se avaliar se o treinamento auto-supervisionado pode trazer ganhos importantes para problemas nesse domínio.

Este trabalho está organizado em 6 Capítulos e 2 Apêndices. Neste Capítulo 1 foi feita uma introdução e definidos os objetivos do trabalho. O Capítulo 2 apresenta trabalhos relacionados. O Capítulo 3 apresenta a fundamentação teórica do trabalho: a arquitetura Vision Transformer, a técnica de aprendizado auto-supervisionado Masked Autoencoders e as arquiteturas de segmentação semântica utilizadas. O capítulo 4 detalha a metodologia do trabalho. O Capítulo 5 apresenta os resultados obtidos, comparando-os com outros da literatura. Por fim, no Capítulo 6 serão apresentadas as conclusões e sugestões para trabalhos futuros. O Apêndice A apresenta as configurações utilizadas no treinameto das redes neurais e o Apêndice B uma pequena introdução ao dado sísmico.

# 2
# Trabalhos relacionados

Como não foram encontrados na literatura trabalhos que realizem o treinamento de Vision Transformers (ViTs) com dados sísmicos utilizando aprendizado auto-supervisionado ou que utilizem ViTs em modelos de segmentação de fácies sísmicas, os trabalhos aqui apresentados são: o que introduziu o Masked Autoenconders (MAE), utilizaram redes neurais convolucionais para segmentação de fácies sísmicas no dataset *Facies-Mark*, aplicaram ViTs em imagens sísmicas ou aplicações no contexto de imagens médicas.

## 2.1
## MAE

He et al. (2022) introduziram a técnica de aprendizado auto-supervisionado Masked Autoenconders para realizar o pré-treinamento de ViTs. Os autores criaram um autoencoder assimétrico, utilizando um ViT como enconder e blocos de ViT como decoder, com a tarefa de reconstruir imagens de entrada após a realização de um mascaramento aleatório de alto percentual nessas imagens. A função de custo utilizada foi o erro quadrático médio. Com o ImageNet (DENG et al., 2009) treinaram diferentes ViTs e obtiveram resultados estado da arte para modelos auto-supervisionados em diversas tarefas downstream de visão computacional, como classificação, segmentação e detecção de objetos.

## 2.2
## Sísmica

Alaudah et al. (2019) criaram o dataset *Facies-Mark*, contendo uma parte do dado sísmico F3 Block e um modelo geológico da área, separados em um conjunto de treino e dois de teste. O trabalho criou um benchmark para comparação de modelos de segmentação de fácies sísmicas. Os autores também propuseram uma arquiterura deconvolucional para segmentar o dado sísmico, comparando diferentes estratégias de treinamento: usando patches ou seção sísmica inteira como entrada, com ou sem data augmentation. A arquitetura também foi testada com e sem skip connections. O modelo com melhor performance foi treinado com seções, usando augmentation e skip connections.

Reis (2022) ampliou o trabalho anterior, incluindo atributos sísmicos como entrada de uma arquitetura deconvolucional para segmentar o *Facies-*

*Mark*. Além da amplitude sísmica, foram estudados os atributos Energia, Pseudo Relevo, Fase instantânea e Textura. A abordagem multiatributos apresentou uma performance melhor que a de Alaudah et al. (2019), mostrando a eficácia dessa abordagem.

Tolstaya e Egorov (2022) usaram uma arquitetura UNet, com a EfficientNetB1 como backbone e quatro níveis de encoder e decoder. Utilizaram uma função de custo que combina: Cross-Entropy, Dice e Total Variarion loss. O modelo tem dois canais de entrada: patches das seções sísmicas e um atributo de profundidade, que vai de 0, no topo do volume, a 1, na sua parte mais profunda. Utilizaram aumento de dados, introduzindo uma técnica de distorção de imagens que simula falhas e dobras geológicas. Aplicaram uma estratégia de pseudo-labels, em que um modelo é treinado no conjunto de treino e em seguida metade do conjunto de teste é predito e agregado ao conjunto de treino inicial. O modelo final é treinado no conjunto de dados expandido. Aplicaram essas técnicas aos datasets *Facies-Mark* e *SEAM AI challenge 2020*.

Junior et al. (2023) aplicaram ViTs para o problema de detecção de gás. Para cada um dos dois volumes sísmicos estudados no trabalho, os autores definiram uma região de interesse. Em seguida aplicaram uma janela deslizante para extração de patches, que foram classificados como com ou sem gás, utilizando ViTs. Os hiperparâmetros da ViT e parâmetros da janela deslizante form otimizados com a técnica *Particle Swarm Optimization*.

## 2.3 Imagens Médicas

Xiao et al. (2023) pré-treinaram ViTs com a MAE utilizando um conjunto de 3 datasets de imagens de Raios-X torácicos. Realizaram uma investigação da melhor estratégia de aumento de dados e masking da MAE para o domínio de Raios-X torácicos e também das melhores receitas para fine-tuning dos modelos ViTs. Avaliaram os modelos obtidos investigando a tarefa de classificação multi-label de doenças torácicas em imagens de Raios-X em 3 datasets benchmark, realizando uma extensiva revisão bibliográfica. Para comparação com a MAE, pré-treinaram a DenseNet121, arquiterura dominante na tarefa de classificação multi-label de Raios-X, com outra estratégia de aprendizado autosupervisionado. Os resultados obtidos com ViTs pré-treinadas com a MAE se mostraram competitivos ou melhores que o de CNNs.

Zhou et al. (2022) pré-treinaram ViTs com a MAE para avaliação de 3 problemas distintos: classificação multi-label de doenças torácicas em imagens de Raios-X, segmentação de múltiplos órgãos do abdome por Tomografia Computadorizada e segmentação de tumores cerebrais por Ressonância Magnética.

Eles mostram que o pré-treino agrega valor a todas essas tarefas, obtendo as melhores métricas em diferentes benchmarks.

Matsoukas et al. (2021) compararam a performance de CNNs e ViTs na tarefa de classificação em 3 datasets de imagens médicas de diferentes domínios. Utilizando estratégias variadas para inicialização dos pesos das redes neurais eles observam que as arquiteturas obtêm o mesmo nível de performance. Além disso, observam que as ViTs obtêm um ganho marginal quando utilizam uma técnica de aprendizado auto-supervisionado. Eles concluem que é possível utilizar as ViTs como substitutas às CNNs na tarefa de classificação de imagens médicas seguindo protocolos adequados de treinamento.

# 3
# Fundamentação Teórica

Neste capítulo é apresentada uma visão geral do método de treinamento auto-supervisionado, a arquitetura Vision Transformer, a técnica auto-supervisionada Masked Autoenconders e os Segmentation Transformers. Essas técnicas são a base para o desenvolvimento desse trabalho.

## 3.1
## Aprendizado auto-supervisionado: Visão geral

O recente desenvolvimento de técnicas de aprendizado auto-supervisionado na área de visão computacional tem possibilitado o treinamento de redes neurais profundas sem a necessidade de dados anotados. Essas técnicas têm por objetivo obter representações robustas dos dados, que transferem bem para tarefas *downstream*.

O treinamento de redes neurais é feito através do aprendizado de pesos que minimizam uma função de custo associada a uma tarefa de interesse. No aprendizado supervisionado, os dados são anotados e a função de custo está associada a quão bem a saída da rede casa com as anotações. Na ausência de anotações, a definição da tarefa a ser realizada e a função de custo a ser utilizada é um problema que vem sendo abordado de diferentes formas.

Normalmente, uma tarefa auxiliar (*pretext task*) é definida a partir dos próprios dados. Em um dos trabalhos pioneiros da área (ZHANG; ISOLA; EFROS, 2016), os autores utilizaram a tarefa de colorização: um conjunto de imagens foi transformado para escala de cor cinza e uma rede neural foi treinada para recuperar uma versão colorida das imagens. Como a tarefa requer um entendimento semântico da cena e objetos, ela direciona a criação de boas representações.

Algumas estratégias de sucesso recente buscam treinar redes que produzam representações invariantes a transformações nos dados (*data augmentations*). Essas transformações podem mudar significativamente o aspecto visual do dado, mas devem manter seu significado semântico. A rede neural aprende a gerar uma representação similar para diferentes transformações da mesma imagem, que deve ser diferente da representação de outras imagens. Essa ideia é explorada de diversas maneiras em técnicas de que se encontram na literatura sob o nome de *contrastive representation learning*.

O surgimento da arquitetura Vision Transformer (DOSOVITSKIY et al., 2020) possibilitou novas abordagens de aprendizado auto-supervisionado. Essa

arquitetura é baseada na Transformer (VASWANI et al., 2017) e em seu tremendo sucesso em problemas de processamento de linguagem natural. Grande parte desse sucesso se deve a técnicas de aprendizado auto-supervisionado, que permitiram o treinamento com imensas quantidades de dados textuais. Algumas dessas técnicas, por exemplo, se baseiam na remoção de partes do texto e na predição do mesmo. Com a introdução dos ViTs, essas ideias encontraram ressonância no aprendizado auto-supervisionado em modelos de visão computacional.

O *Masked Autoenconders* (MAE) é uma dessas técnicas. Na MAE, patches das imagens de entrada são aleatoriamente mascarados e através de uma rede encoder-decoder assimétrica, os pixels removidos são reconstruídos. Essa é a técnica utilizada nesse trabalho para o treinamento de ViTs.

## 3.2 Vision Transformers

A arquitetura ViT, Figura 3.1, é muito semelhante ao encoder da rede Transformer original. A seguir, serão detalhados os principais componentes dessa arquitetura, seguindo os trabalhos (ZHANG et al., 2021; DOSOVITSKIY et al., 2020).

A entrada para a ViT ou a Transformer é uma sequência de vetores 1D. Em dados textuais, as palavras de uma sequência de texto são mapeadas em vetores *one-hot*, que tem a dimensionalidade do vocabulário dos dados de treinamento, e em seguida projetadas através de uma camada linear treinável em uma dimensão menor. Esses vetores são chamados de *token embeddings* e são o início do processamento realizado pela Transformer.

A criação da sequência de vetores 1D a partir de imagens $2D$ é mais sutil. Consideremos uma imagem $2D$ com resolução $(H, W)$ e $C$ canais. Ela é dividida em *patches* de resolução $(P, P)$ e $C$ canais, que são redimensionados para 1D. Isso resulta em uma sequência de $N = HW/P^2$ vetores de tamanho $CP^2$, $\boldsymbol{x}_p^1 ... \boldsymbol{x}_p^N$. Desse modo, os *patches* podem ser tratados pela ViT de maneira igual aos *tokens* de uma sequência textual pela Transformer.

A ViT é composta por uma sequência de blocos que operam em vetores de tamanho pré-definido $D$. Desse modo, uma camada linear treinável $\boldsymbol{E}$ projeta os vetores 1D para esse tamanho, e são chamdados de *patch embeddings*. Além disso, um *embedding* especial treinável, $\boldsymbol{x}_{<cls>}$, é incluído na primeira posição da sequência. Como ele interage com todos os outros *embeddings* ao longo do processamento pelos blocos da ViT, sua saída serve como uma representação final da imagem. Essa representação é utilizada, por exemplo, para realizar a classificação da imagem.

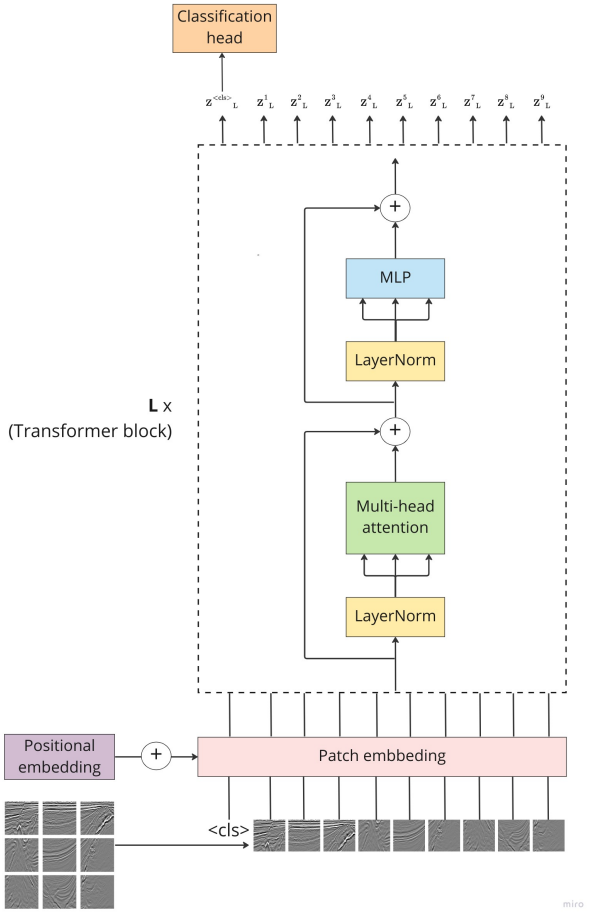



Figura 3.1: Arquitetura da Vision Transformer. A imagem de entrada é dividida em patches de tamanho fixo, que são linearizados e transformados em embeddings. Após a adição de positional embeddings, os vetores são processados por uma sequência de blocos da ViT.

Para preservar a informação do ordenamento dos *patches*, um vetor adicional é somado a cada *patch embedding*. Esses vetores são conhecidos como *positional embeddings*, $\boldsymbol{E}_{pos}$, e podem ser fixos ou aprendidos ao longo do treinamento. Esses *embeddings* iniciais servem como entrada para os blocos da ViT.

A ViT é composto por $L$ blocos, que podem ser divididos em duas subcamadas. A primeira composta por uma operação de *multi-head self-attention* (MSA) e a segunda por um *multilayer percpetron* (MLP), ambas precedidas de uma opereção de *Layer Normalization* (LN). Nas duas subcamadas existem conexões residuais. A MLP possui duas camadas com função de ativação *Gaussian error linear unit*, GELU.

Podemos visualizar um resumo dessa descrição na Figura 3.1. Formal-

mente, seguindo (DOSOVITSKIY et al., 2020) temos que:

$$\boldsymbol{z}_0 = [\boldsymbol{x}_{<cls>}; \boldsymbol{x}_p^1\boldsymbol{E}; ...; \boldsymbol{x}_p^N\boldsymbol{E}] + \boldsymbol{E}_{pos}, \quad \boldsymbol{E} \in \mathbb{R}^{(CP^2)\times D},\ \boldsymbol{E}_{pos} \in \mathbb{R}^{(N+1)\times D} \tag{3-1}$$

$$\boldsymbol{z'}_\ell = \text{MSA}(\text{LN}(\boldsymbol{z}_{\ell-1})) + \boldsymbol{z}_{\ell-1}, \qquad \ell = 1...L \tag{3-2}$$

$$\boldsymbol{z}_\ell = \text{MLP}(\text{LN}(\boldsymbol{z'}_\ell)) + \boldsymbol{z'}_{\ell-1}, \qquad \ell = 1...L \tag{3-3}$$

Após o processamento por todos os blocos da ViT, a representação final $\boldsymbol{z}_L^{<cls>}$ é utilizada, por exemplo, na classificação de imagens. As outras representações, $\boldsymbol{z}_L^1 ... z_L^N$, são utilizadas, por exemplo, na segmentação semântica.

A operação MSA exige uma explicação mais detalhada, devido a sua importância e novidade em relação às CNNs convencionais. Primeiro, será analisada a *self-attention* (SA). A SA realiza uma média ponderada dos *embeddings* de entrada, que tem pesos determinados por projeções lineares dos próprios *embeddings*. Essas projeções são aprendidas ao longo do treinamento da rede.

Seja uma sequência de entrada de vetores 1D, cada um deles de tamanho D, dada por $\boldsymbol{x} = [\boldsymbol{x}_1 ... \boldsymbol{x}_N]$. São definidas 3 projeções:

$$\boldsymbol{q} = \boldsymbol{x}\boldsymbol{U}_q, \boldsymbol{k} = \boldsymbol{x}\boldsymbol{U}_k, \boldsymbol{v} = \boldsymbol{x}\boldsymbol{U}_v, \qquad \boldsymbol{U}_q, \boldsymbol{U}_k, \boldsymbol{U}_v \in \mathbb{R}^{D\times D_h}, \tag{3-4}$$

Onde $D_h$ é a dimensão em que são projetados os vetores 1D. Na prática, esse valor é escolhido de modo que quando multiplicado pelo número de cabeças (*heads*) $h$ da MSA seu resultado seja igual a $D$, ou seja, $h \cdot D_h = D$. Na literatura, essas projeções são chamadas de *query*, *key* e *value*.

Agora, define-se a matrix de pesos, ou matrix de atenção (*attention matix*):

$$A = \text{softmax}\left(\boldsymbol{q}\boldsymbol{k}^\top / \sqrt{D_h}\right) \qquad A \in \mathbb{R}^{N\times N}. \tag{3-5}$$

E finalmente,

$$\boldsymbol{h} = SA(\boldsymbol{x}) = A\boldsymbol{v}. \tag{3-6}$$

Isso define a saída de uma *head*. Como são utilizadas $h$ delas na MSA, é necessário concatená-las. Isso é feito e em seguida aplicada uma nova projeção treinável:

$$MSA(\boldsymbol{x}) = [SA_1(\boldsymbol{x}); SA_2(\boldsymbol{x}); \cdots; SA_h(\boldsymbol{x})]\ \boldsymbol{U}_{msa}, \quad \boldsymbol{U}_{msa} \in \mathbb{R}^{h\cdot D_h \times D}. \tag{3-7}$$

Em cada bloco, a MSA permite uma interação complexa entre todas as partes

da imagem de entrada e não existe a limitação do campo receptivo das CNNs. Outra característica importante da ViT, que a diferencia das CNNs, é que não existe uma diminuição de resolução espacial ao longo dos blocos da rede.

Neste trabalho será utilizada o valor $P = 16$ em todos os modelos ViT. A Tabela 3.1 apresenta informações adicionais sobre as configurações dos modelos utilizados nesse trabalho (STEINER et al., 2021). Essa escolha se deve à predominância dos mesmos na literatura de Vision Transformers.

| Modelo | Camadas | emb_dim (D) | heads | parâmetros |
|---|---|---|---|---|
| ViT-S | 12 | 384 | 6 | 22.2M |
| ViT-B | 12 | 768 | 12 | 86M |
| ViT-L | 24 | 1024 | 16 | 307M |

Tabela 3.1: Configuração de modelos ViT.

O modo usual de treinar ViTs é através do problema de classificação supervisionada do ImageNet (DENG et al., 2009). Para as redes maiores, como a ViT-L, essa é uma tarefa difícil, que envolve uma escolha cuidadosa dos hiperparâmetros.

## 3.3 Masked Autoenconders

A técnica MAE é utilizada para treinar ViTs de maneira autosupervisionada. Seguindo o procedimento da ViT, cada imagem de entrada é divida patches regulares e em seguida um alto percentual deles é mascarado. Na configuração padrão 75%. A MAE tem como *pretext task* a reconstrução da imagem original a partir dos patches não mascarados. Com a reconstrução de uma imagem descaracterizada, que não permite uma operação simples de extrapolação de informação entre patches, ela consegue obter uma representação robusta dos dados de treino.

A arquitetura da autoencoder é assimétrica, tendo como encoder uma ViT padrão, vide Seção 3.2. Ela produz uma representação latente dos dados de entrada. O decoder é composto por 8 blocos de ViT, responsável pela reconstrução do sinal original. Após o treinamento da MAE o decoder é descartado e utiliza-se o encoder para as tarefas de interesse. Na Figura 3.2 apresentamos a arquitetura da MAE.

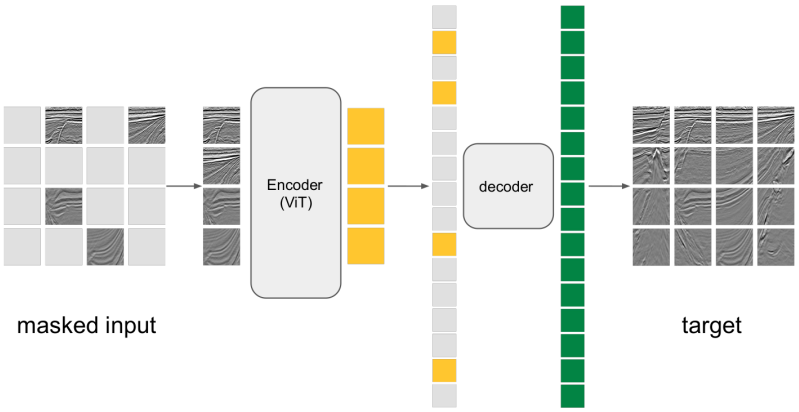



Figura 3.2: Arquitetura da MAE. Adaptado de He et al. (2022).

O encoder processa apenas os patches não mascarados. Após seu processamento, os patches mascarados, *mask tokens*, são re-introduzidos para o processamento pelo decoder. Os *mask tokens* são representados por um vetor treinável e compartilhado, que indica a presença de um patch faltante que precisa ser predito. Os *positional embeddings* são somados novamente nessa etapa, para introduzir informação espacial aos *mask tokens*. Na MAE os *positional embeddings* são vetores fixos e não treináveis.

Após o processamento pelos blocos de ViT, os vetores de saída são projetados para a dimensão do número de pixels em cada patch, $P \times P \times 3$, e em seguida redimensionados para a reconstrução final da imagem.

A função de custo utilizada no treinamento da MAE é o Erro quadrático médio (MSE) entre a imagem original e a imagem reconstruída, no espaço dos pixels, e calculada apenas nos patches originalmente mascarados.

## 3.4 Segmentation Transformers

Os *Segmentation Transformers* (SETR) (ZHENG et al., 2021) são arquiteturas para segmentação semântica que utilizam o ViT como enconder para extração de features, combinado com diferentes decoderes convolucionais. Um aspecto importante da utilização do ViT como enconder é que ele não tem a perda de resolução característica das CNNs e a limitação do campo receptivo, possibilitando uma nova perspectiva para o problema de segmentação.

Esses decoders operam na saída de blocos da ViT, como exemplificado na Figura 3.3. Cada bloco, como visto na seção 3.2, produz como saída uma sequência de vetores de mesma dimensão $D$. Considerando o tamanho do patch $P = 16$ e não levando em conta a representação do embedding $\boldsymbol{x}_{<cls>}$, tem-se um total de $\frac{H \times W}{256}$ vetores de tamanho $D$. A primeira etapa

dos decoders é redimensionar essas representações de saída para um feature map $3D$ convencional, de dimensões $\frac{H}{16} \times \frac{W}{16} \times D$.

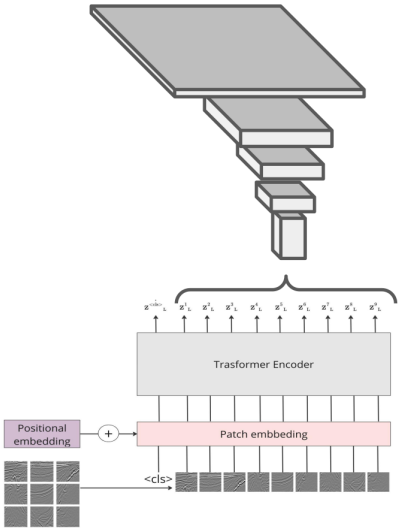



Figura 3.3: Ilustração da SETR

.

A seguir, serão explicados os principais componentes de cada um dos decoders propostos por (ZHENG et al., 2021): *Naive upsampling*, *Progressive UPsampling* (PUP) e *Multi-Level feature Aggregation* (MLA).

Para deixar o texto mais claro, quando for citada a operação de convolução, a não ser a última de cada decoder, ela inclui a seguinte sequência de operações: convolução, *batch normalization* e função de ativação *Rectified Linear Unit* (ReLU).

### 3.4.1 Naive upsampling

Esse decoder opera nos vetores de saída do último bloco do ViT. Ele aplica uma operação de convolução com kernel 1x1 e 256 canais de saída, seguida de uma operação de upsampling que quadruplica a resolução das features. Na sequência, é aplicada uma convolução com kernel 1x1 e número de canais de saída igual ao número de classes do problema ($N_c$), seguida de uma nova operação de upsampling que quadruplica novamente a resolução. Tem-se por fim uma feature map $H \times W \times N_c$.

### 3.4.2 PUP

Esse decoder opera nos vetores de saída do último bloco do ViT. Ele possui 4 camadas. Cada uma delas aplica uma operação de convolução com kernel 3x3 seguida de uma operação de upsampling que duplica a resolução das features. As três primeiras convoluções produzem 256 canais de saída. A última convolução produz um número de canais de saída igual a $N_c$. Esse modelo é chamado de SETR-PUP e está ilustrado na Figura 3.4.

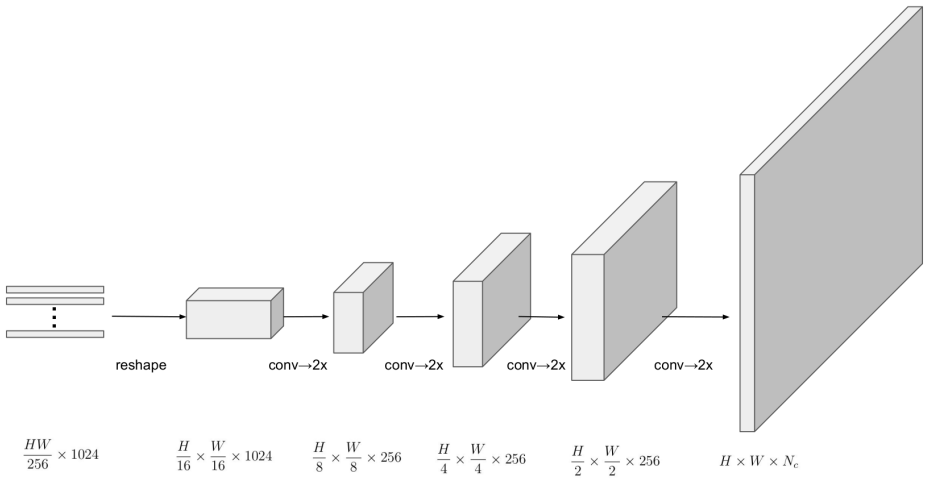



Figura 3.4: Decoder SETR-PUP. Adaptado de Zheng et al. (2021).

### 3.4.3 MLA

Esse decoder opera nos vetores de saída de 4 blocos distintos da ViT, agregando features de diferentes níveis de representação. Para a ViT-L, os blocos escolhidos são os 6, 12, 18 e 24.

A saída de cada bloco é redimensionada para $\frac{H}{16} \times \frac{W}{16} \times D$. A seguir, cada um deles passa por um operador de convolução com kernel 1x1 e 256 canais de saída. Esses resultados passam por uma operação de agregação de cima para baixo, através de uma soma em cascata, ilustrada na Figura 3.5.

Na sequência, cada feature map passa por duas convoluções de kernel 3x3 e 128 canais de saída e finalmente um operação de upsampling que qradruplica a resolução das features. Os 4 feature maps resultantes são concatenados na dimensão do canal e passam por uma convolução final com kernel 3x3 e número de canais de saída igual a $N_c$. Finalmente é realizado um novo upsampling que qradruplica a resolução das features. Esse modelo é chamado de SETR-MLA e está ilustrado na Figura 3.5.

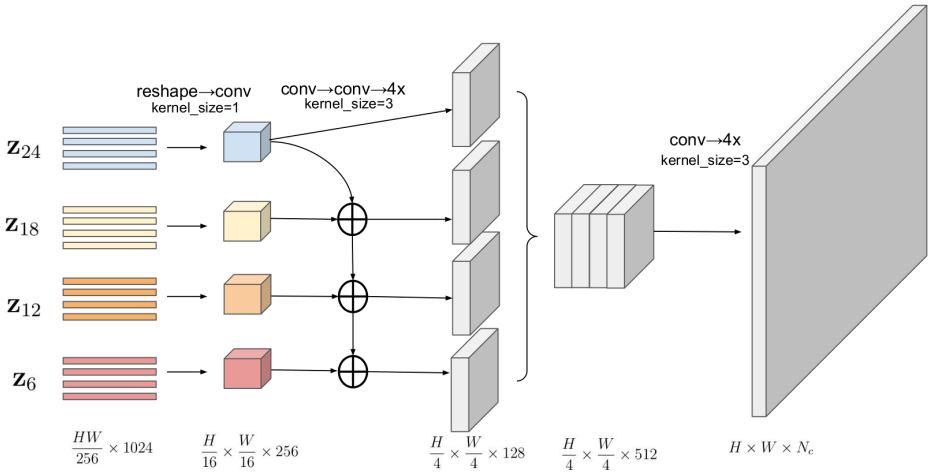



Figura 3.5: Decoder SETR-MLA. Adaptado de Zheng et al. (2021).

### 3.4.4 Função de custo

A função de custo utilizada para o treinamento da SETR é a *Cross-entropy* (CE). Cada pixel de uma imagem pertence a uma de $N_c$ classes. Um pixel da classe $q$ pode ser representada por um vetor de tamanho $N_c$, que tem o elemento $q$ igual a 1 e os outros iguais a zero. Esse vetor será representado por $\mathbf{y}$. A rede neural por sua vez, calcula uma função de probabilidades para a classificação de cada pixel, utilizando a função softmax após a saída da última camada da rede. Essa saída é representada por uma vetor de tamanho $N_c$, $\hat{\mathbf{y}}$, onde $\hat{\mathbf{y}}_q$ é a probabilidade do pixel pertencer à classe $q$. Com essas definições, pode-se escrever a função de custo CE como:

$$\mathcal{L}_{CE} = -\frac{1}{N}\sum_{n=1}^{N}\sum_{k=1}^{N_c}\mathbf{y}_k log(\hat{\mathbf{y}}_k), \tag{3-8}$$

onde a soma em $n$ é sobre os pixels da imagem e $N$ o número total de pixels.

Os autores do trabalho utilizaram cabeças auxiliares para ajudar na convergência do treinamento das diferentes versões da SETR. Todas utilizam a CE como função de custo. Essas cabeças atuam em camadas intermediárias da ViT e tem uma estrutura similar ao decoder PUP. A função de custo total utilizada pode ser escrita, como:

$$\mathcal{L} = \mathcal{L}_{CE}^{head} + \lambda\sum_{i=1}^{n_{aux}}\mathcal{L}_{CE}^{auxiliary}, \tag{3-9}$$

onde $n_{aux}$ é o número de decoders auxiliares.

# 4
# Metodologia

O método utilizado nesse trabalho é dividido em duas etapas: pré-treinamento de ViTs com volumes sísmicos através da MAE e segmentação de fácies sísmicas com Segmentation Transformers. A seguir será apresentada uma visão geral da metodologia. Na sequência, detalham-se as suas etapas.

## 4.1
## Visão geral

A primeira etapa da metodologia utilizada nesse trabalho está representada no fluxograma da Figura 4.1. Nessa etapa, é realizado o pré-treinamento de ViTs com volumes sísmicos através da MAE. Para isso, inicialmente são selecionados alguns volumes sísmicos, que em seguida são pré-processados e de cada um deles extraídos patches de tamanho fixo. Esses patches formam o dataset que será utilizado para o pré-treinamento de ViTs através da MAE. As ViTs passam então por uma etapa de avaliação qualitativa, em que se inspeciona visualmente a reconstrução de imagens sísmicas mascaradas aleatoriamente, seguindo o trabalho de He et al. (2022), e analisa-se a representação de imagens sísmicas em dimensionalidade reduzida.

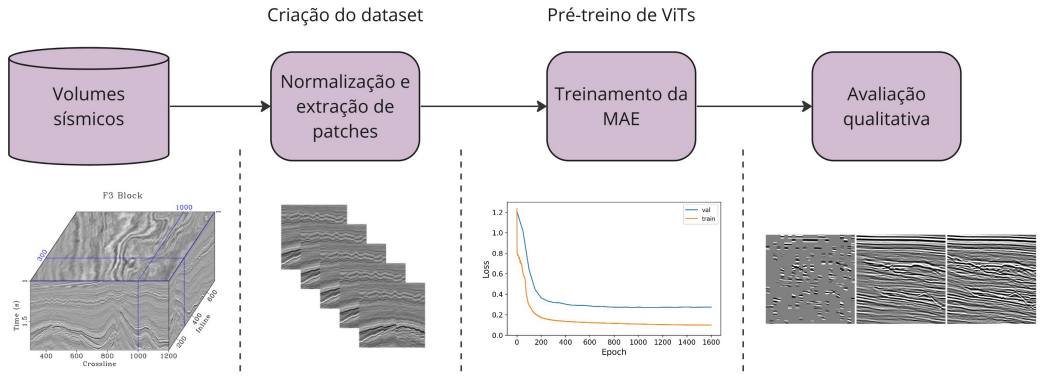



Figura 4.1: Etapa 1: Pré-treinamento de ViTs com dados sísmicos.

Na segunda etapa, representada no fluxograma da Figura 4.2, é realizada uma avaliação quantitativa do pré-treinamento, em que é medido o desempenho da rede pré-treinada na tarefa de segmentação semântica de fácies sísmicas utilizando Segmentation Transformers. Os modelos de segmentação são treinados com dois tipos de inicialização do enconder: ViT resultante do pré-treinamento com a MAE e dados sísmicos (etapa 1) e ViT que vem da tarefa de classificação supervisionada do ImageNet (DENG et al., 2009). Cal-

culadas as métricas, a comparação dessas inicializações serve para avaliarmos se há ganho quantitativo com o treinamento auto-supervisionado.

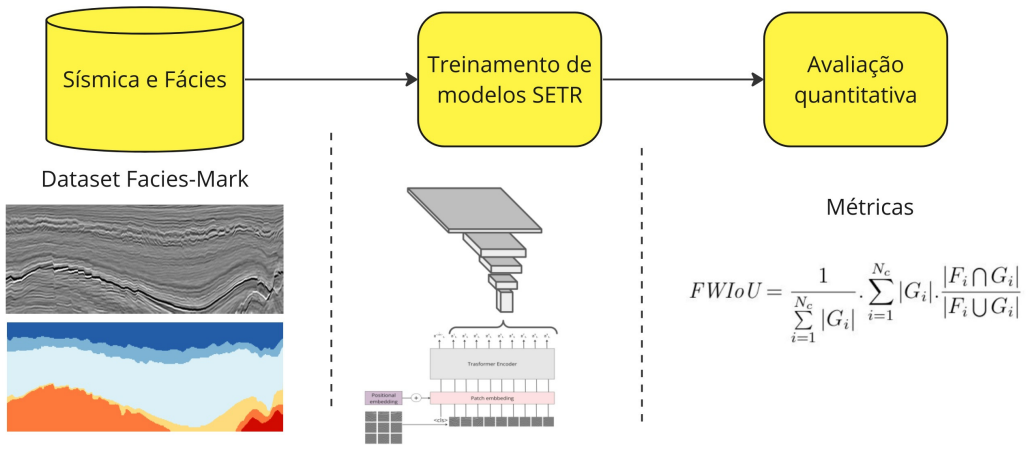



Figura 4.2: Etapa 2: Segmentação de fácies sísmicas com Segmentation Transformers

## 4.2 Etapa 1: MAE

A etapa 1 começa com a seleção de alguns volumes sísmicos. A seguir, cada um dos volumes, $x$, é normalizado por sua média, $\mu$, e desvio padrão, $\sigma$, da seguinte maneira:

$$x_{norm} = \frac{x - \mu}{\sigma} \tag{4-1}$$

Essa operação diminui o efeito da grande variação das amplitudes sísmicas entre os dados.

Na sequência, selecionam-se algumas linhas de cada um dos volumes. Evita-se pegar todas as linhas de cada um dos dados devido à grande redundância e coerência espacial dos mesmos. A partir dessas linhas realiza-se a extração dos patches que irão compor o dataset de treinamento da MAE. Essa estratégia está ilustrada na Figura 4.3.

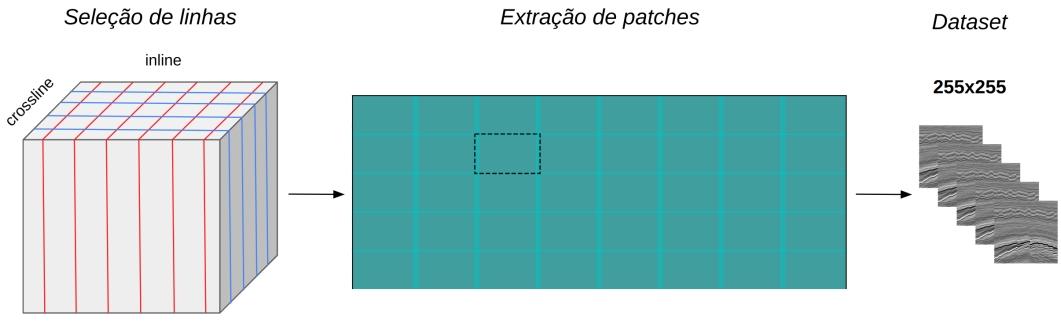



Figura 4.3: Criação do dataset para treinamento da MAE.

### 4.2.1 Extração de patches

A extração de patches de cada linha sísmica selecionada seguiu o método proposto por Claerbout e Fomel (2008). De forma sucinta, seguindo a notação dos autores, dada uma imagem 2D com dimensões nwall = (nwall1, nwall2), quer-se dividi-la em um conjunto de patches de tamanho nwind = (nwind1, nwind2), permitindo overlap. Escolhe-se o número de patches em cada direção, npatch = (npatch1, npatch2). Assim, pode-se determinar todos os cantos superiores esquerdos dos patches, os corners, através dos Algoritmos 1 e 2.

**Algoritmo 1:** LineToCart

**Entrada:** npatch,i
**Saída:** pos

```
pos ← [0, 0];
para axis ← 0 até 1 faça
    pos[axis] ← i mod npatch[axis]
    i ← i ÷ npatch[axis]
```

**Algoritmo 2:** PatchesCorners

**Entrada:** nwall, nwind, npatch
**Saída:** corners

```
para ipatch ← 0 até npatch[0] × npatch[1] − 1 faça
    ii ← LineToCart(npatch, ipatch)
    jj ← [0, 0];
    para axis ← 0 até 1 faça
        se npatch[axis] = 1 então
            jj[axis] ← 0;
        senão se ii[axis] = npatch[axis] − 1 então
            jj[axis] ← nwall[axis] - nwind[axis];
        senão
            jj[axis] ←
              ii[axis] × int((nwall[axis] − nwind[axis])/(npatch[axis] − 1))
    corners ← jj
```

O Algoritmo 1 apenas transforma uma coordenada linear em cartesiana. Essa posição cartesiana é utilizada pelo Algoritmo 2 para determinar os cantos superiores esquerdos de cada um dos patches. Os patches criados a partir dessa informação preenchem todo o dado, incluindo cantos e bordas.

Na Figura 4.4, são apresentados os patches extraídos de uma imagem 2D seguindo o algoritmo descrito. Um patch está ilustrado com linha tracejada de cor preta. As regiões de overlap entre os patches aparecem com cor mais intensa. Neste exemplo, os parâmetros utilizados foram nwall=(1929,1261),

nwind =(255,255) e npatch=(8,5).

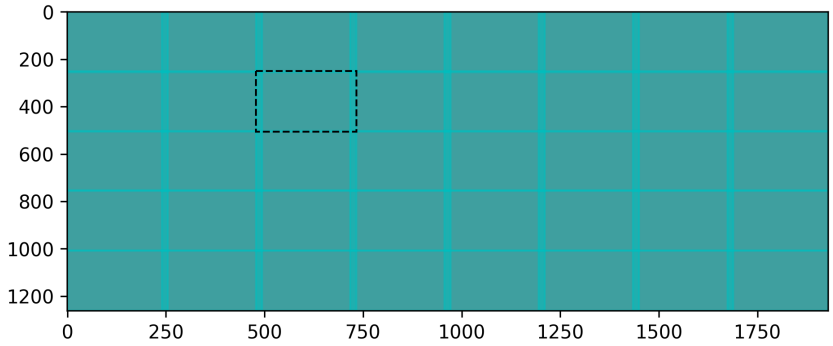



Figura 4.4: Ilustração da extração de patches de uma imagem 2D.

### 4.2.2 Aumento de dados

Durante o treinamento da MAE existe uma etapa de aumento de dados. Para cada imagem, é realizado o corte de uma janela em posição aleatória, com dimensões com uma escala entre 0.2 e 1 da original, seguido de um reescalonamento para o tamanho de entrada para a rede. Por fim, a imagem é espelhada horizontalmente com probabilidade 0.5.

Na Figura 4.5 ilustramos o aumento de dados para um patch, em nove execuções diferentes. A coluna da esquerda contém um patch fixo. As outras imagens são exemplos do resultado do aumento de dados para diferentes execuções do conjunto de operações.

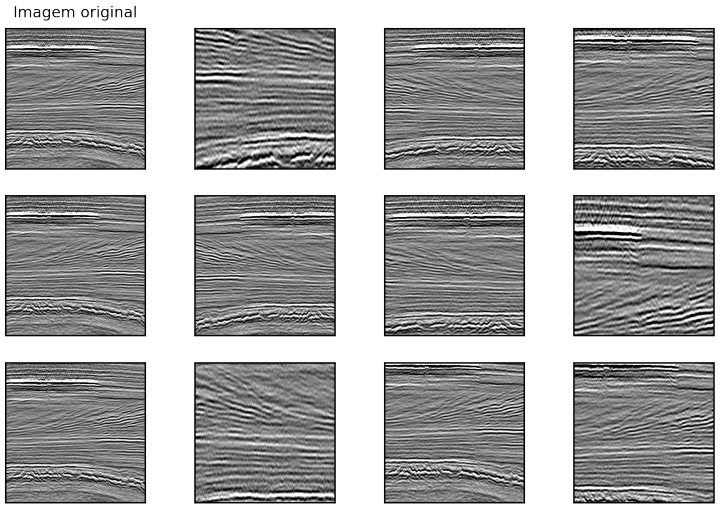



Figura 4.5: Ilustração do processo de aumento de dados utilizado no treinamento da MAE.

## 4.3 Etapa 2: Segmentação semântica

A seguir são descritos o aumento de dados e as métricas utilizadas na etapa 2 da metodologia.

### 4.3.1 Aumento de dados

Para cada iteração de treinamento do modelos de segmentação, a seguinte sequência de aumento de dados é aplicada em cada imagem: redimensionamento do eixo de profundidade para 512 e do eixo horizontal mantendo o *aspect ratio*, mudança da escala da imagem por um fator entre 0.5 e 2.0, corte aleatório de tamanho $512 \times 512$, seguido de espelhamento horizontal com probabilidade 0.5. Esse processo está exemplificado na Figura 4.6.

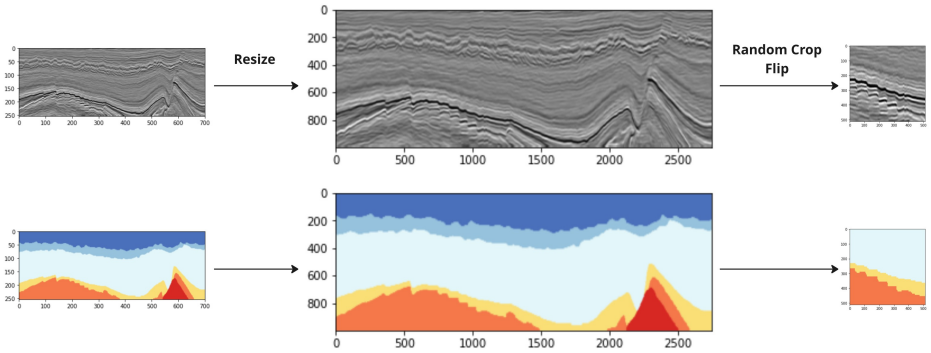



Figura 4.6: Ilustração do processo de aumento de dados utilizado no treinamento da SETR.

Na etapa de teste, as imagens passam pelo mesmo redimensionamento, e em seguida divididas em patches de tamanho $512 \times 512$ com *stride* de $341 \times 341$. Na região de *overlap* entre os patches é tomada a média das saídas.

### 4.3.2 Métricas

Foram utilizadas cinco métricas com objetivo de avaliar a metodologia deste trabalho e realizar comparações com trabalhos da literatura (ALAUDAH et al., 2019; REIS, 2022): *pixel accuracy* (PA), *class accuracy* (CA) para cada uma das classes, *mean class accuracy* (MCA), *mean intersection over union* (mIoU) e *frequency weighted Intersection over Union* (FWIoU).

Sejam $N_c$ o número de classes do problema, $G_i$ o conjunto de pixels que pertencem à classe i e $F_i$ o conjunto de pixels classificados como da classe i. O conjunto de *pixels* classificados corretamente é $F_i \bigcap G_i$.

PA é o percentual de pixels de todas as classes que são corretamente classificados. Utilizando $|.|$ para denotar o número de elementos de um conjunto, podemos escrever:

$$PA = \frac{\sum_{i=1}^{N_c} |F_i \bigcap G_i|}{\sum_{i=1}^{N_c} |G_i|} \tag{4-2}$$

A CA para a classe i ($CA_i$) é a razão entre a quantidade de pixels da classe $i$ que são corretamente classificados e o total de pixel dessa classe:

$$CA_i = \frac{|F_i \bigcap G_i|}{|G_i|} \tag{4-3}$$

A MCA é definida como a média da acurácia por classe sobre todas as classes:

$$MCA = \frac{1}{N_c} \sum_{i=1}^{N_c} \frac{|F_i \bigcap G_i|}{|G_i|} \tag{4-4}$$

A métrica *Intersection over Union* (IoU) da classe $i$ é definida como o número de elementos da interseção de $G_i$ e $F_i$ dividido pelo número de elementos de sua união. Essa métrica mede a sobreposição entre os dois conjuntos. Temos que:

$$IoU_i = \frac{|F_i \bigcap G_i|}{|F_i \bigcup G_i|} \tag{4-5}$$

A média da IoU em todas as classes é a mIoU:

$$mIoU = \frac{1}{N_c} \sum_{i=1}^{N_c} \frac{|F_i \bigcap G_i|}{|F_i \bigcup G_i|} \tag{4-6}$$

Para evitar que essa métrica seja excessivamente sensível às classes pouco frequentes é comum balancear cada classe por seu tamanho. A métrica resultante é a FWIoU:

$$FWIoU = \frac{1}{\sum_{i=1}^{N_c} |G_i|} . \sum_{i=1}^{N_c} |G_i| . \frac{|F_i \bigcap G_i|}{|F_i \bigcup G_i|} \tag{4-7}$$

# 5
# Resultados

Nesse capítulo são apresentados os resultados obtidos com a metodologia proposta no trabalho. Primeiro são apresentados os dados utilizados para o treinamento da MAE e dos modelos de segmentação semântica. Em seguida, são discutidos os experimentos e resultados obtidos na fase de pré-treinamento das ViTs com a MAE. Por fim, analisamos os resultados e métricas obtidas na segmentação de fácies sísmicas, que são comparadas com as métricas de dois trabalhos relacionados da literatura.

## 5.1
## Datasets

Nesta seção apresentamos os dados utilizados nas duas etapas da metodologia. Na primeira etapa foram utilizados 4 volumes sísmicos e na segunda o dataset *Facies-Mark*.

### 5.1.1
### Etapa 1

No pré-treinamento da MAE foram utilizados 4 volumes sísmicos, sendo 1 deles utilizado na etapa de validação. A seguir, as dimensões desses volumes serão representadas como número de amostras em profundidade × crosslines × inlines. São eles:

1. volume privado, pertencente à Petrobras, com dimensões: $1261 \times 1929 \times 1237$;

2. Waka-3D, dado offshore da Nova Zelândia, tornado público pela New Zealand Petroleum and Minerals, com dimensões $499 \times 5342 \times 571$;

3. F3 Block, dado offshore da Holanda, tornado público pela dGB Earth Sciences, na versão de (ALAUDAH et al., 2019), com dimensões $255 \times 701 \times 401$;

4. Parihaka Basin, dado offshore da Nova Zelândia, tornado público pela New Zealand Crown Minerals, com dimensões $1006 \times 590 \times 782$.

O volume Parihaka foi utilizado apenas na validação. Na preparação dos dados, o volume Waka-3D foi recortado para evitar regiões com buracos ou muito ruidosas, com o objetivo de não introduzir características muito diferentes das presentes nos outros 3 volumes. Esse recorte está ilustrado na Figura 5.1. Depois, cada um dos volumes foi normalizado por sua média, $\mu$, e desvio padrão, $\sigma$, da seguinte maneira:

$$x_{norm} = \frac{x - \mu}{\sigma} \tag{5-1}$$

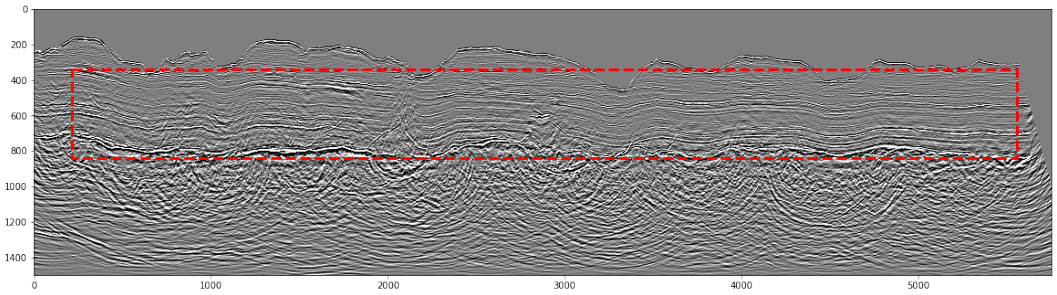
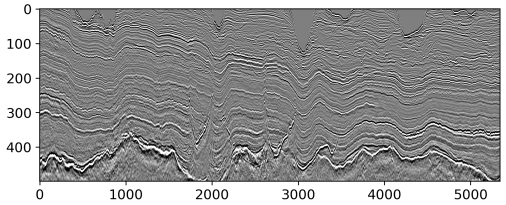
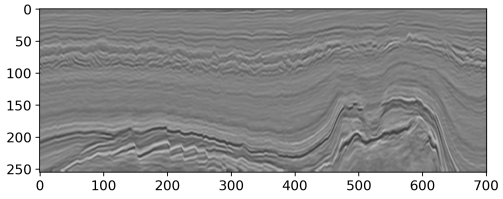



Figura 5.1: Ilustração de recorte realizado no volume Waka-3D, para evitar regiões com buracos ou muito ruidosas. Inline 1000.

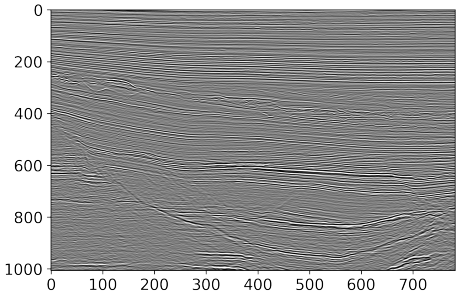
Na Figura 5.2 é apresentada uma seção inline de cada um dos volumes públicos, todas na mesma escala de cor.




(a) Waka-3D




(b) F3 Block




(c) Parihaka

Figura 5.2: Exemplo de inline para cada um dos volumes sísmicos públicos utilizados: (a) Waka 3D, (b) F3 Block e (c) Parihaka.

A seguir, foi realizada a extração dos patches para o treinamento da MAE. Utilizando a nomenclatura do Capítulo 4, a dimensão escolhida para os patches foi de nwind =(255,255). 255 é a menor dimensão presente nos dados e foi escolhida com o intuito de evitar a necessidade de padding. Como a MAE tenta reconstruir o dado original mascarado, isso introduziria uma característica artificial ao dado. Na Tabela 5.1 estão os valores de npatch utilizados em cada um dos dados, nas direções inline e crossline.

| Volume sísmico | npatch crossline | npatch inline |
|---|---|---|
| Privado | (5,5) | (8,5) |
| Waka-3D | (3,2) | (21,2) |
| F3 | (2,1) | (3,1) |
| Parihaka | (3,4) | (4,4) |

Tabela 5.1: Parâmetro npatch utilizado para cada um dos dados.

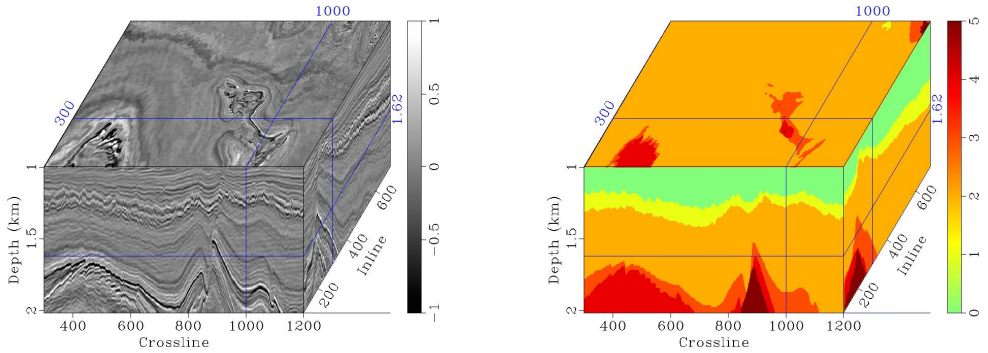
Devido às diferenças nas dimensões dos volumes, para esse trabalho foi preciso equilibrar a composição dos patches. No caso do F3, foram utilizadas todas as inlines e crosslines; para o Waka-3D foram selecionadas as inlines (crosslines) com uma amostragem de uma em cada 4 (5); para o dado privado foi utilizada uma amostragem de uma em cada 5 (5) inlines (crosslines); para o Parihaka uma amostragem de 1 em 4 nas duas direções. Na Tabela 5.2 é apresentado o número de patches extraídos de cada volume.

| Volume sísmico | Número de patches |
|---|---|
| Privado | 19570 |
| Waka-3D | 12420 |
| F3 | 2605 |
| Parihaka | 3772 |
| Treino | 34595 |
| Validação | 3772 |

Tabela 5.2: Número de patches por volume sísmico.

### 5.1.2 Etapa 2

Para o treinamento e avaliação dos modelos de segmentação de fácies foi utilizado o dataset *Facies-Mark* (ALAUDAH et al., 2019). Os autores do trabalho criaram um modelo geológico 3D de uma porção do F3 Block com base em dados de perfis de 26 poços. Foram definidas 6 fácies: *Upper North Sea*, *Middle North Sea*, *Lower North Sea*, *Chalk/Rijnland*, *Scruff* e *Zechstein*. Na Figura 5.3 é apresentada uma visão do volume sísmico e do modelo geológico.




(a) Sísmica (b) Modelo geológico

Figura 5.3: Visão do dado Facies-Mark: (a) volume sísmico e (b) modelo geológico.

O dado foi então dividido em 3 conjuntos, um de treino e dois de teste, compreendendo as seguintes regiões:

1. Treino: inlines [300,700] e crosslines [300,1000];
2. Teste 1: inlines [100,299] e crosslines [300,1000];
3. Teste 2: inlines [100,700] e crosslines [1001,1200].

Uma visualização dessa divisão é apresentada Figura 5.4.

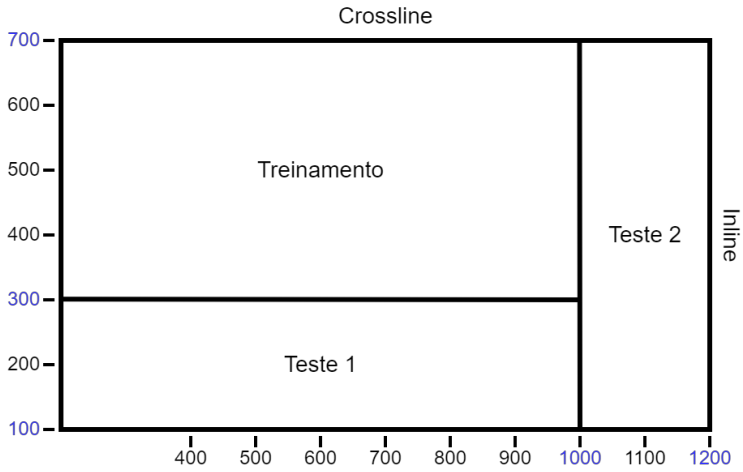


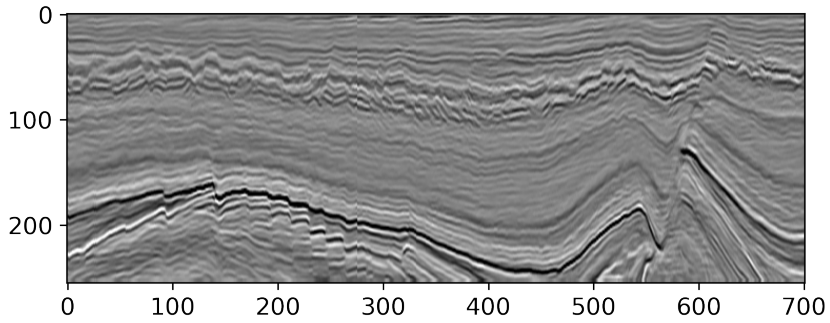

Figura 5.4: Partições do conjunto de dados. Fonte: Nelia Reis

Na Tabela 5.3, é apresentado o percentual das fácies presentes no conjunto de treino. Nota-se que existe um grande desbalanceamento entre elas. As cores associadas a cada fácie na Tabela 5.3 serão as mesmas utilizadas para apresentação dos resultados, seguindo as referências (ALAUDAH et al., 2019; REIS, 2022).

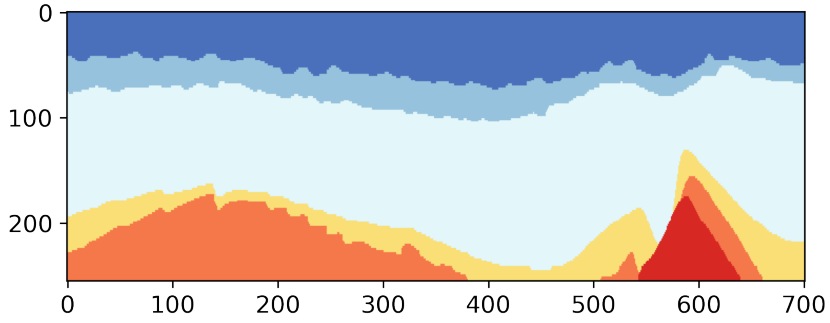
| Zechstein | Scruff | Rijnland/Chalk | Lower | Middle | Upper |
|---|---|---|---|---|---|
| 1.48% | 3.17% | 6.53% | 48.44% | 11.89% | 28.49% |

Tabela 5.3: Percentual de cada classe no conjunto de treino.

Para o treinamento dos modelos de segmentação, foi utilizado o volume de treino, separando 10% para validação. As inlines tem dimensões (profundidade × largura) 255 × 701 e as crosslines 255 × 401. Na Figura 5.5 é exemplificada uma inline do *dataset*, mostrando a sísmica e as fácies correspondentes.



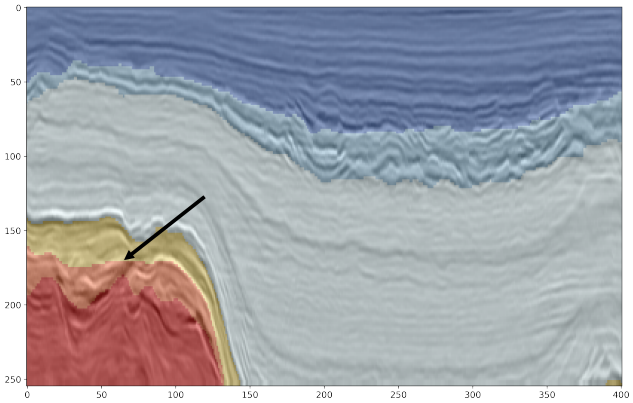

(a) Sísmica



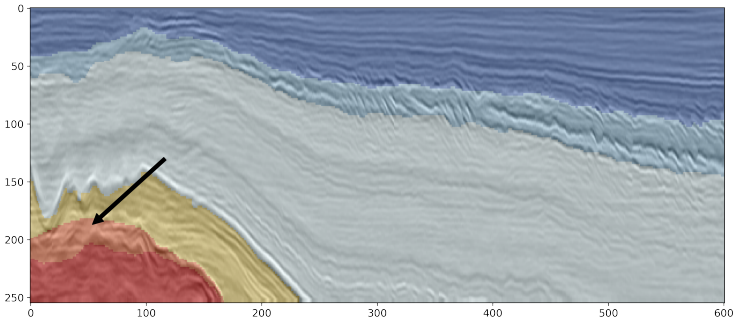

(b) Fácies anotadas

Figura 5.5: Inline 300 do conjunto de treino do *Facies-Mark*.

É importante salientar que, apesar do trabalho cuidadoso realizado por (ALAUDAH et al., 2019), a anotação dos dados de treino e teste é problemática em algumas regiões, principalmente para as fácies menos frequentes. Na Figura 5.6, onde sísmica e fácies estão sobrepostas, é possível ver regiões em que sísmica e anotação não são coerentes. Para servir como padrão para outros trabalhos de interpretação, seria importante uma revisão cuidadosa dessas anotações por especialistas.




(a) Crossline 900 do conjunto de treino




(b) Crossline 1019 do conjunto de test

Figura 5.6: As anotações no conjunto de treino e teste apresentam algumas regiões de qualidade ruim, indicadas nas figuras.

## 5.2 MAE

Nesta seção são apresentados resultados do pré-treinamento com a MAE. Primeiro são analisadas funções de custo de experimentos representativos. Na sequência, é avaliada a qualidade da reconstrução de imagens sísmicas com a MAE, a qualidade das representações de um modelo ViT aplicado no conjunto de teste 2 do *Facies-Mark*, analisadas em dimensionalidade reduzida com o t-SNE. Por fim, são apresentados mapas de atenção de um modelo ViT-S/16.

### 5.2.1 Funções de custo e convergência

Ao longo da realização deste trabalho foram realizados diversos experimentos de pré-treinamento de modelos ViT-S/16, ViT-B/16 e ViT-L/16 com a MAE e volumes sísmicos, usando imagens de entrada com dimensões $224 \times 224$ ou $384 \times 384$ e diferentes estratégias de inicialização dos pesos da rede. O percentual de mascaramento utilizada em todos os experimentos apresentados foi de 75%. A seguir, serão analisadas as funções de custo de alguns experimentos representativos.

Uma primeira observação importante é que os modelos ViT-B/16 e ViT-L/16 não convergiram com inicialização aleatória dos pesos, possivelmente

devido à grande quantidade de parâmetros desses modelos e ao número de imagens utilizadas. Já o modelo ViT-S/16_384 (entrada $384 \times 384$) convergiu com a inicialização aleatória. Quatro desses experimentos estão ilustrados na Figura 5.7, onde apresenta-se a função de custo de treinamento. As curvas preta, vermelha e azul estão associadas aos modelos que não convergiram.

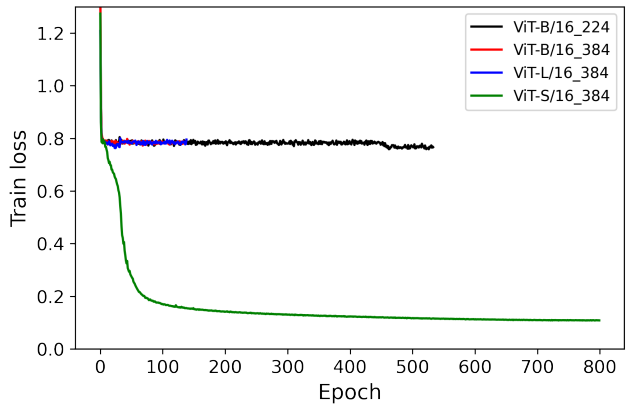
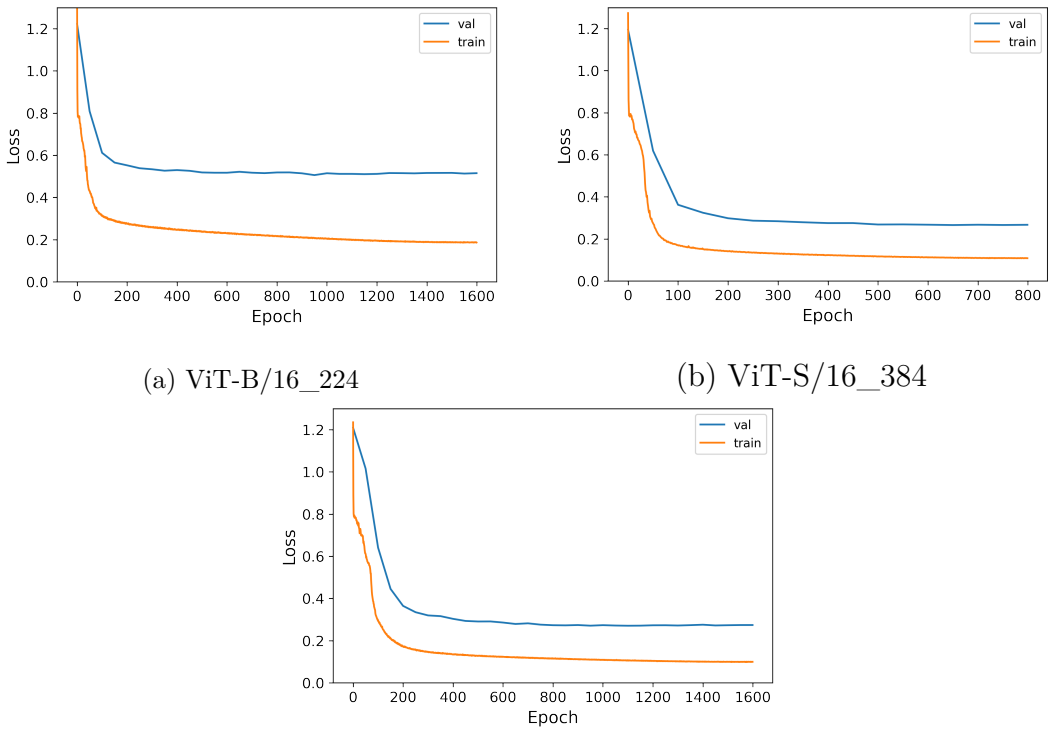



Figura 5.7: Função de custo de experimentos representativos.

Na Figura 5.8, são exemplificadas as funções de custo de três modelos que convergiram. Os modelos ViT-B/16_224 e ViT-L/16_384 foram inicializados com pesos pré-treinados no ImageNet. O modelo ViT-S/16_384 teve inicialização aleatória dos pesos. Pelas curvas, observa-se que os modelos com entrada 384 apresentam melhor convergência. Nas Figuras 5.8(b) e 5.8(c), a função de custo tem um valor final menor e a diferença entre as curvas de treino e validação também é menor.




(a) ViT-B/16_224

(b) ViT-S/16_384

(c) ViT-L/16_384

Figura 5.8: Função de custo do treinamento da MAE para três experimentos.

A seguir, serão explorados resultados qualitativos dos modelos ViT-S/16_384 e ViT-L/16_384. O modelo ViT-L/16_384 foi utilizado como encoder dos modelos de segmentação apresentados na última seção do capítulo.

### 5.2.2 Reconstrução de patches

A análise da reconstrução de imagens é importante para avaliar qualitativamente o resultado do pré-treinamento com a MAE. Na Figura 5.9 são apresentados resultados da reconstrução de imagens pela MAE treinada com o modelo ViT-L/16_384. Para 12 patches do dado de validação escolhidos de modo aleatório, tem-se o tripleto: patch mascarado, reconstrução com o MAE e ground truth. O percentual de patches mascarados é de 75%. Na imagem reconstruída não foram reintroduzidos os patches não-mascarados. Pode-se observar que o resultado da reconstrução tem boa qualidade, captando diferentes características do dado sísmico. Isso indica que esse modelo consegue uma boa representação dos dados sísmicos.

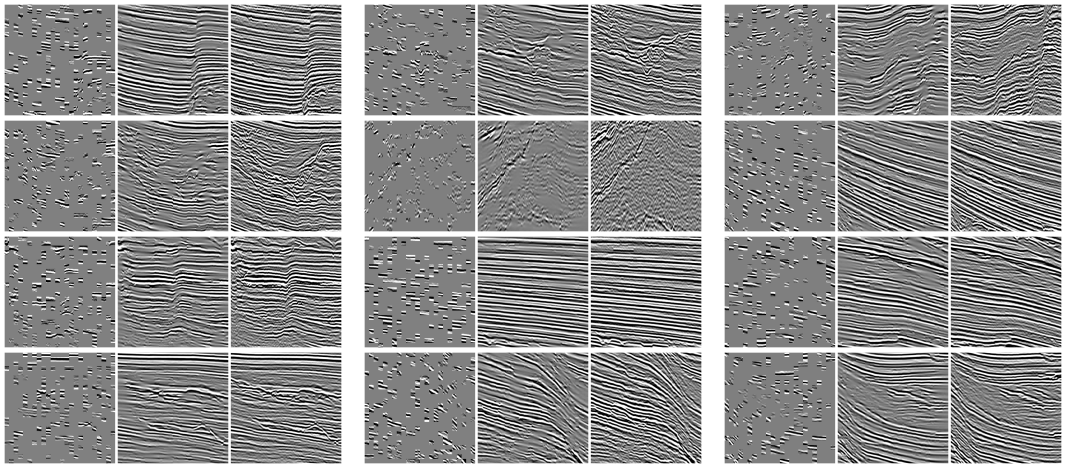


Figura 5.9: Exemplos da reconstrução em imagens do conjunto de validação. Para cada tripleto, temos a imagem mascarada, a reconstrução com o MAE e o ground-truth. O percentual de patches mascarados é de 75%.

Na Figura 5.10 está exemplificada em detalhes a reconstrução de dois patches da Figura 5.9. Pode-se observar que algumas regiões ficam com marcas e outras com uma amplitude pequena.

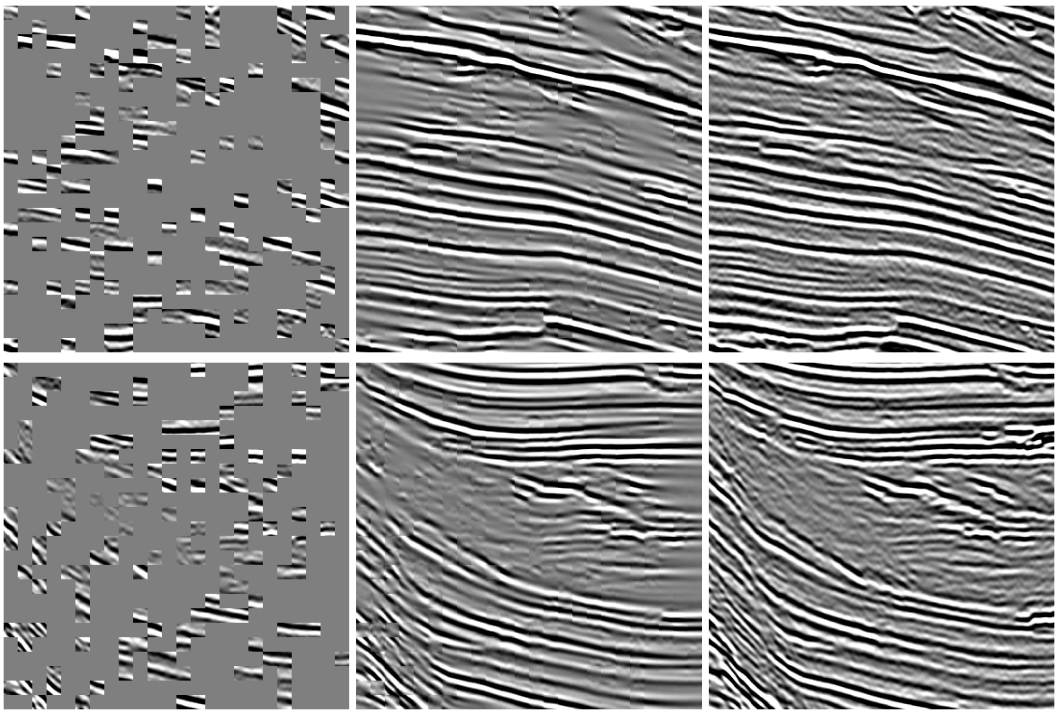


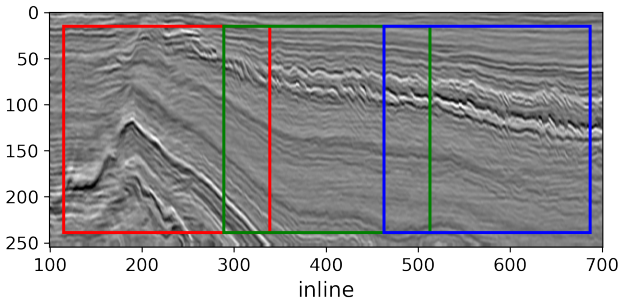
Figura 5.10: Dois exemplos amplificados da reconstrução de imagens do conjunto de validação.

### 5.2.3
### Representações em dimensionalidade reduzida

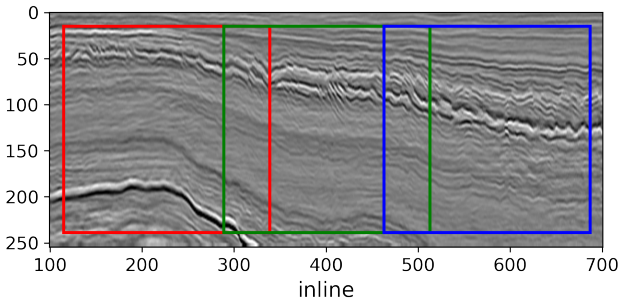
A análise das representações de imagens em dimensionalidade reduzida também é uma técnica utilizada para avaliar a qualidade das representações

obtidas com modelos de redes neurais. Para explorar isso, foi realizado um experimento com o modelo ViT-L/16_384, utilizando o conjunto de teste 2 do Facies-Mark (ver Figura 5.4).

De cada crossline desse conjunto de teste foram extraídos 3 patches adjacentes de tamanho 224x224, como exemplificado na Figura 5.11, onde estão ilustradas as crosslines 1001 e 1101. Os patches na mesma direção crossline estão marcados com a mesma cor.




(a) Crossline 1001

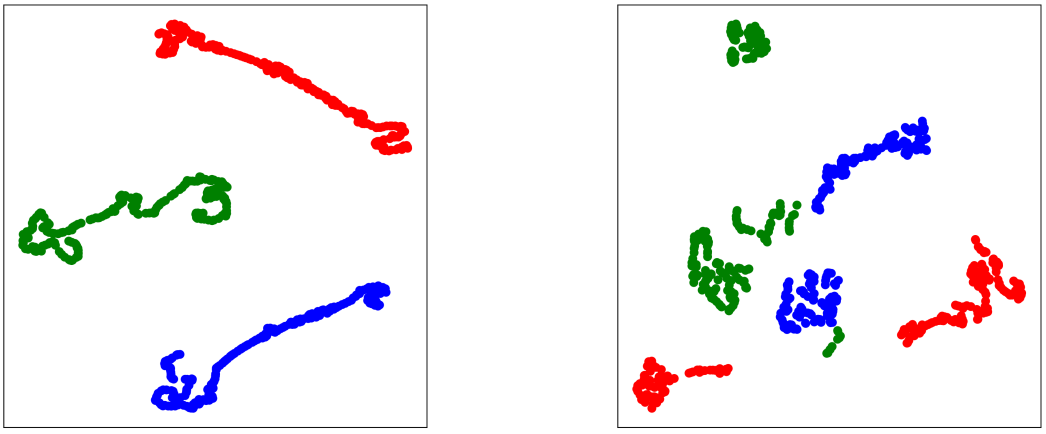



(b) Crossline 1101

Figura 5.11: Crosslines do conjunto de teste 2 do *Facies-Mark*. Separamos 3 patches de cada crossline, exemplificados em vermelho, verde e azul.

Cada um desses patches foi redimensionado para a dimensão $384 \times 384$ e propagado por duas redes ViT-L/16_384: pré-treinada com a MAE e dados sísmicos, ViT Seismic, e pré-treinada na tarefa de classificação do ImageNet, ViT ImageNet. Os patches passaram a ser representados por um vetor 1D de dimensão 1024.

Esses vetores foram em seguida projetados num espaço de duas dimensões utilizando o t-SNE (MAATEN; HINTON, 2008). Pode-se visualizar o resultado na Figura 5.12, onde cada cor está associada a uma região de patches. Como a região de teste é pequena, compreendendo 200 inlines, a variação da imagem sísmica não é grande ao longo dos blocos delimitados por cada cor. Isso pode ser observado na Figura 5.11. O mesmo é esperado das representações dos patches dentro de cada cor. Isso está de acordo com a continuidade da projeção da representação do modelo ViT Seismic para cada uma das regiões observadas na Figura 5.12(a). Para o modelo ViT ImageNet, observa-se uma perda de continuidade, ilustrada na Figura 5.12(a). Esse resultado reforça a qualidade da representação obtida com o pré-treinamento utilizando dados do domínio de interesse.



(a) ViT Seismic (b) ViT ImageNet

Figura 5.12: t-SNE da representação dos patches das crosslines do conjunto de teste 2 do *Facies-Mark* obtidas com os modelos (a) ViT Seismic e (b) ViT ImageNet. Cada cor representa uma região do dado.

### 5.2.4 Mapa de Atenção

A sondagem dos valores da última camada de *self-attention* de modelos ViTs é explorado por (CARON et al., 2021) como forma de avaliar qualitativa e quantitativamente as representações obtidas com treinamento auto-supervisionado. Isso é feito extraindo os valores da matriz de atenção da interação de cada patch da imagem de entrada com o vetor $\boldsymbol{x}_{<cls>}$. Esses valores são redimensionados e passam por um *upsampling* para as dimensões da imagem de entrada, produzindo os mapas de atenção, um para cada cabeça de atenção.

O modelo ViT-S/16_384 possui 6 cabeças, produzindo 6 mapas de atenção. Na Figura 5.13 ilustramos os 6 mapas obtidos com modelo ViT-S/16_384 pré-treinado com a MAE e imagens sísmicas. Utilizamos como entrada um patch do conjunto de teste 1 do *Facies-Mark*. Ele está repetido na primeira coluna das imagens. Nos mapas de atenção é possível ver o destaque de algumas estruturas da imagem sísmica, mas com uma qualidade inferior ao apresentado com imagens naturais por (CARON et al., 2021).

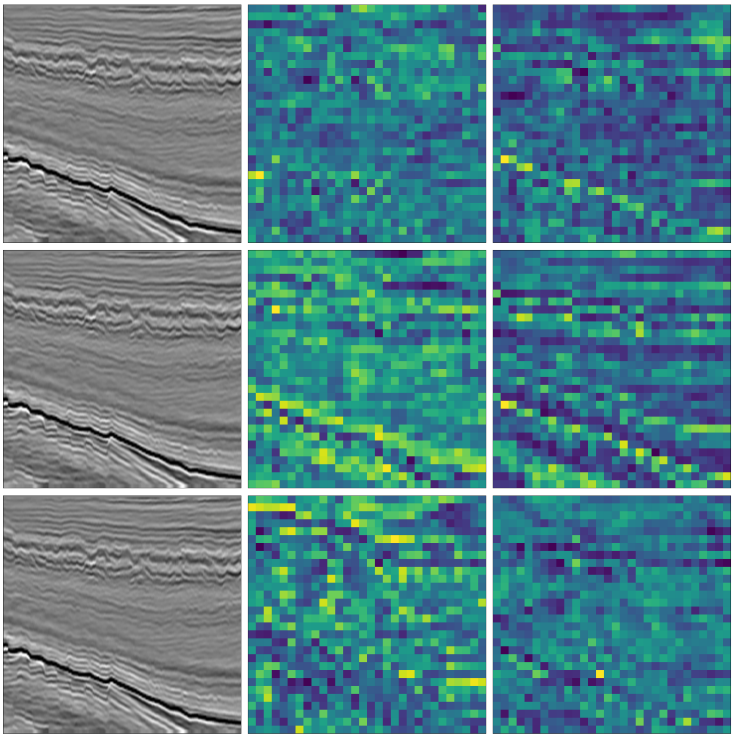


Figura 5.13: Mapas de atenção para o modelo ViT-S/16_384, utilizando como entrada um patch do conjunto de teste 1 do *Facies-Mark*.

Esses resultados encerram a análise qualitativa do pré-treino realizado com a MAE. O conjunto dos resultados apresentados nesta seção nos levam a concluir que foram obtidas boas representações dos dados sísmicos com o MAE. No entanto, a quantidade de dados sísmicos utilizados foi insuficiente para um pré-treino ótimo e indica que um aumento da quantidade e variedade dos mesmos deve ser uma estratégia de sucesso. Também foi observado que a utilização de dados de entrada com dimensões maiores, como 384x384, tende a favorecer a convergência e uma melhor representação dos dados sísmicos.

## 5.3 Segmentação Semântica

Com as ViTs pré-treinadas na etapa 1 da metodologia, segue-se para a etapa de avaliar quantitativamente os modelos através da tarefa downstream de segmentação semântica. Existem duas questões importantes a investigar:

1. O pré-treinamento de ViT com dados sísmicos e MAE traz ganhos quantitativos na tarefa de segmentação semântica?

2. A segmentação semântica utilizando modelos baseados em Vision Transformers é efetiva para dados sísmicos?

A primeira pergunta visa quantificar se o trabalho extra do pré-treinamento retorna ganhos efetivos quando comparado à abordagem padrão, que é partirmos de um *backbone* ViT pré-treinado na tarefa de classificação do ImageNet. Apesar do sucesso dessa última abordagem (ALAUDAH et al.,

2019; KAUR et al., 2023), a utilização de dados específicos do domínio de interesse no pré-treinamento tem apresentado resultados excelentes (AZIZI et al., 2022; TANG et al., 2022; HAGHIGHI et al., 2022). No entanto, sua utilização para o domínio das imagens sísmicas ainda está em estado inicial (LI et al., 2023) e esperamos contribuir para essa discussão.

A segunda questão tem a ver com o modelo escolhido neste trabalho para segmentar as imagens sísmicas. As redes convolucionais puras apresentaram ótimos resultados nos últimos anos (WALDELAND et al., 2018; ALAUDAH et al., 2019; SHI; WU; FOMEL, 2019; GAO; WU; LIU, 2021; KAUR et al., 2023), mas ainda não existe na literatura uma aplicação de redes com *backbone* do tipo ViT para essa tarefa. Saber se irão funcionar e ser competitivas com as redes convolucionais é outra contribuição da nossa investigação, de grande importância, dada a proeminência que essas redes tem obtido no último ano em todas as tarefas de Visão Computacional.

Na análise a seguir serão utilizados os modelos SETR-PUP e SETR-MLA (ZHENG et al., 2021), descritos no Capítulo 3. Serão apresentados resultados em que a rede ViT-L/16_384 foi utilizada como encoder. Esse encoder foi treinado com os três volumes sísmicos através da MAE ou na tarefa de classificação do ImageNet. Os pesos do modelo treinado no ImageNet foram disponibilizados pelo Google e podem ser obtidos através da biblioteca *PyTorch Image Models* (WIGHTMAN, 2019).

O dataset *Facies-Mark* foi utilizado no treinamento dos modelos de segmentação. Na Tabela 5.4 são apresentadas as métricas calculadas no conjunto de teste do dataset *Facies-Mark* para os modelos treinados neste trabalho. Os modelos serão chamados de SETR MLA Seismic e SETR PUP Seismic quando o encoder tiver sido treinado com a MAE e sísmica, e de SETR MLA IN21K e SETR PUP IN21K quando treinado na tarefa de classsificação do ImageNet. Também apresentamos as métricas de dois trabalhos da literatura que utilizaram o mesmo conjunto de dados, porém com modelos convolucionais (ALAUDAH et al., 2019; REIS, 2022).

A primeira observação é que os modelos SETR apresentam as melhores métricas. Isso mostra que os modelos que utilizam a ViT para segmentação semântica de imagens sísmicas são competitivos com os modelos convolucionais, respondendo a segunda pergunta feita no início da seção.

| Modelo | PA | MCA | FWIoU | mIoU |
|---|---|---|---|---|
| SETR MLA Seismic | 0.938 | **0.884** | 0.893 | 0.7766 |
| SETR MLA IN21K | 0.87 | 0.691 | 0.791 | 0.5914 |
| SETR PUP Seismic | 0.938 | 0.877 | 0.892 | 0.7753 |
| SETR PUP IN21K | **0.94** | 0.879 | **0.894** | **0.7836** |
| Alaudah et al. (2019)[1] | 0.879 | 0.716 | 0.789 | – |
| Alaudah et al. (2019)[2] | 0.905 | 0.817 | 0.832 | – |
| Reis (2022) | 0.912 | 0.801 | 0.857 | – |

[1] Modelo sem aumento de dados
[2] Modelo com aumento de dados

Tabela 5.4: Comparação com resultados de outros trabalhos.

Outra observação importante é que o modelo com melhores métricas foi o SETR PUP IN21K, que não utilizou o pré-treino com imagens sísmicas,

porém com métricas muito similares a SETR PUP Seismic e SETR MLA Seismic. Testes estatísticos adicionais são necessários para afirmar com certeza qual dos modelos tem a melhor performance.

No entanto, quando são comparados apenas os modelos MLA, SETR MLA Seismic e SETR MLA IN21K, vemos que o SETR MLA Seismic apresenta resultados muito melhores, indicando que o pré-treino foi de grande valor para esse caso. Então, respondendo à primeira pergunta feita no início da seção, podemos afirmar que os resultados indicam que o pré-treinamento traz ganhos, porém um estudo com maior quantidade de dados é necessário para uma conclusão. O pré-treino da MAE com dados sísmicos utilizou apenas 34595 imagens, uma quantidade pequena quando comparada às 1,281,167 de imagens do ImageNet, usualmente utilizado em estudos de modelos auto-supervisionados.

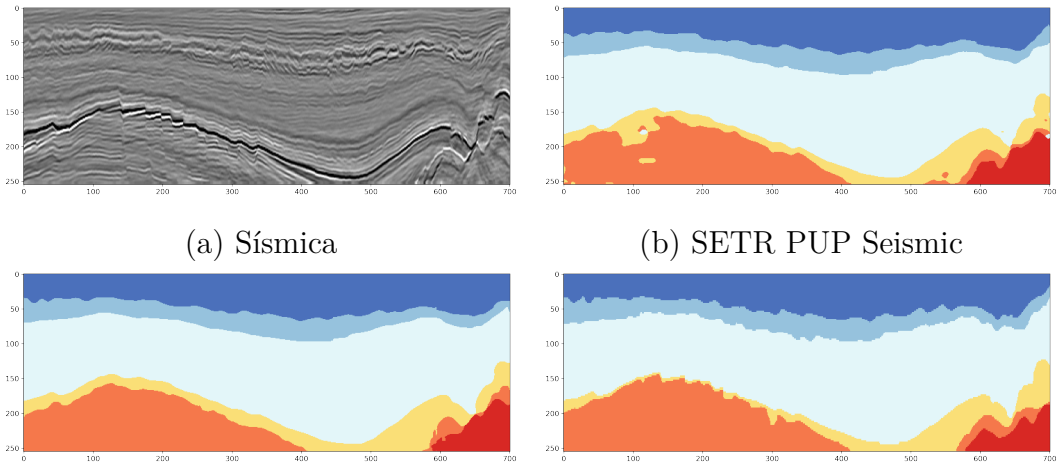
Na Tabela 5.5 são comparadas as acurácias por classe (CA) dos quatro modelos treinados e dos reportados na literatura. Para as classes menos frequentes no *Facies-Mark*, fica claro que a SETR é bem superior em acurácia, mostrando o potencial dessas redes para identificação de padrões sutis nos dados.

| | Acurácia por classe (CA) | | | | | |
|---|---|---|---|---|---|---|
| Modelo | Zechstein | Scruff | Rijnland/Chalk | Lower | Middle | Upper |
| SETR MLA Seismic | **0.874** | **0.782** | 0.764 | 0.977 | 0.93 | 0.977 |
| SETR MLA IN21K | 0.318 | 0.336 | 0.669 | 0.973 | 0.873 | 0.979 |
| SETR PUP Seismic | 0.836 | 0.778 | 0.773 | 0.979 | 0.921 | 0.976 |
| SETR PUP IN21K | 0.844 | 0.715 | **0.826** | **0.987** | 0.92 | **0.98** |
| Alaudah et al. (2019)[1] | 0.219 | 0.539 | 0.744 | 0.95 | 0.872 | 0.973 |
| Alaudah et al. (2019)[2] | 0.602 | 0.674 | 0.772 | 0.941 | **0.938** | 0.974 |
| Reis (2022) | 0.618 | 0.595 | 0.737 | 0.981 | 0.898 | 0.976 |

[1] Modelo sem aumento de dados
[2] Modelo com aumento de dados

Tabela 5.5: Comparação da acurácia por classe.

Na Figura 5.14 é apresentada uma comparação do resultado da segmentação da Inline 200 do conjunto de teste, mostrando a imagem sísmica, as fáceis preditas utilizando o modelo SETR PUP Seismic, o modelo SETR PUP IN21k e o *ground truth*, apresentados nas Figuras 5.14a–5.14d, respectivamente. As fácies preditas pelos dois modelos apresentam boa concordância com o *ground truth*, com menor concordância das fácies menos frequentes no dataset de treino, e melhor resultado para o modelo SETR PUP IN21K. No entanto, os modelos não conseguiram capturar a alta frequência presente nas interfaces das fácies de cor azul. Essa alta frequência está presente em boa parte dessas interfaces ao longo do dataset.

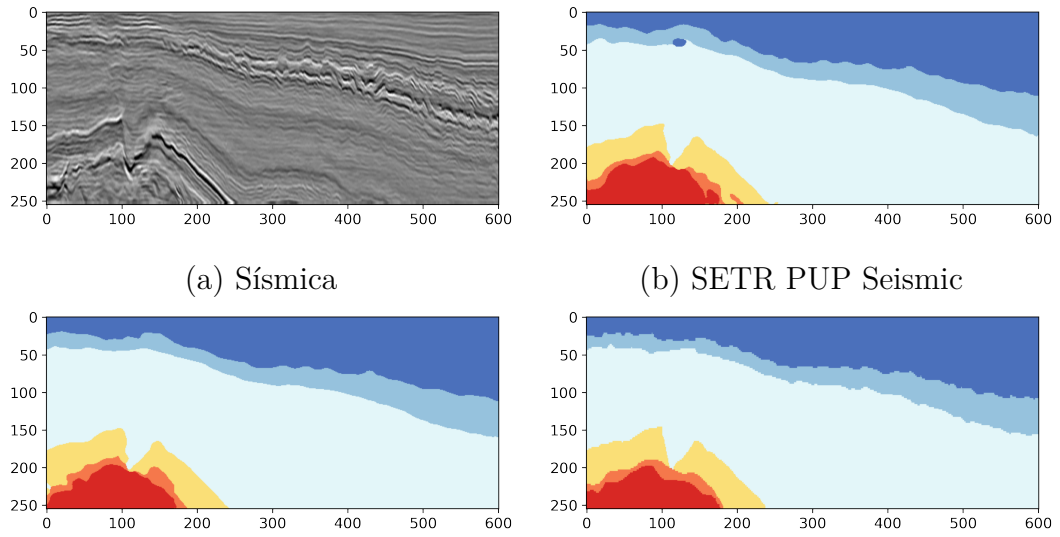


(a) Sísmica (b) SETR PUP Seismic

(c) SETR PUP IN21K (d) Ground Truth

Figura 5.14: Comparação dos resultados para a Inline 200, utilizada nos trabalhos de referência.

Resultados similares podem ser observados na Figura 5.15, para a Crossline 1061. Vale ressaltar novamente a concordância entre as fácies menos frequentes, e que problemas na anotação dos dados podem ter contribuído para uma resolução pior na predição dos modelos em geral.



(a) Sísmica (b) SETR PUP Seismic

(c) SETR PUP IN21K (d) Ground Truth

Figura 5.15: Comparação dos resultados para a Crossline 1061.

Os resultados obtidos são visualmente superiores aos apresentados nos trabalhos de referência para a Inline 200, como está ilustrado na Figura 5.16.

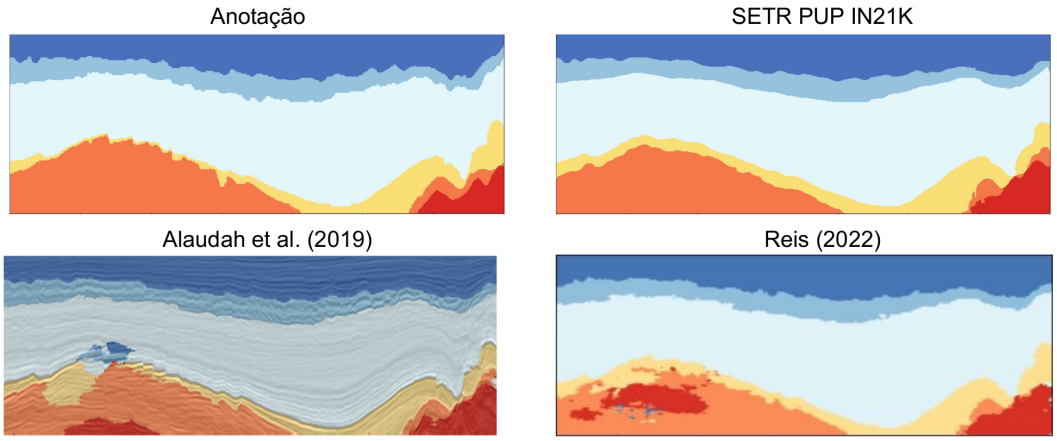
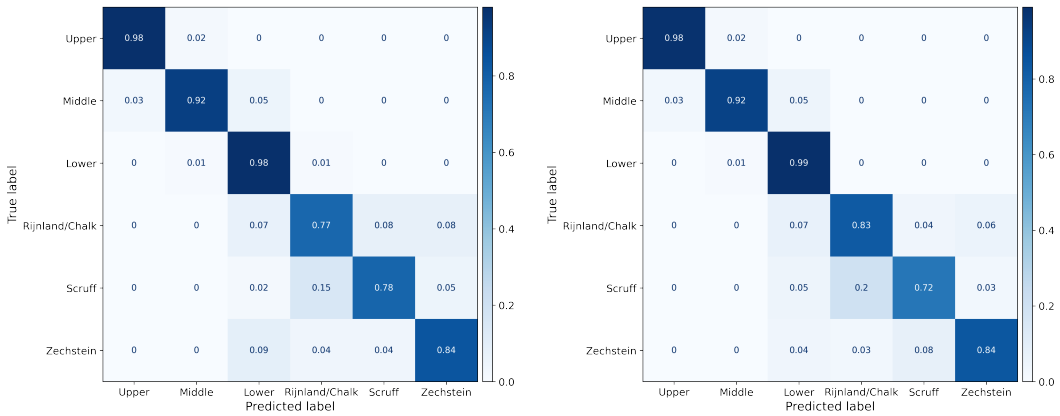



Figura 5.16: Comparação da segmentação da Inline 200 com resultados dos trabalhos de referência.

Por fim, foi calculada a matriz de confusão normalizada para os modelos SETR PUP IN21K e SETR PUP Seismic, apresentadas na Figura 5.17. Observa-se uma boa concordância entre a predição e o *ground truth*. As classes menos frequentes, Zechstein, Scruff e Rijnland/Chalk, se misturam um pouco. Essa classes apresentam uma menor variabilidade sísmica entre si quando comparadas às classes maiores, como podemos ver na Figura 5.6.




(a) SETR PUP Seismic (b)SETR PUP IN21K

Figura 5.17: Matriz de confusão.

Concluindo, podemos afirmar que comparando apenas as métricas apresentadas pelos modelos SETR MLA, o pré-treino com a MAE agregou valor à segmentação. Além disso, os resultados de segmentação de fácies utilizando o modelo SETR mostram que modelos de que utilizam ViT são competitivos com os modelos puramente convolucionais.

# 6
# Conclusão e trabalhos futuros

Este trabalho utilizou o método de aprendizado auto-supervisionado Masked Autoencoders para treinar Vision Transformers com dados sísmicos e buscou verificar se esse aprendizado trouxe ganho quantitativo à tarefa de segmentação semântica de fácies sísmicas. Foi realizado um estudo com o dataset *Facies-Mark*, que permitiu a comparação das métricas de segmentação com os trabalhos de Alaudah et al. (2019) e Reis (2022).

Foram utilizados 4 volumes sísmicos para treinamento da MAE, sendo um deles apenas para validação. Esses volumes foram processados e a partir deles gerados um conjunto de 34595 patches para treinamento e 3772 para validação. A avaliação qualitativa do treinamento com a MAE, através da análise da função de custo e reconstrução de patches mascarados mostrou que foram obtidas representações de boa qualidade para o dado sísmico. Isso também foi confirmado por uma análise dessas representações em dimensionalidade reduzida.

Em seguida, foi realizada a segmentação do dataset *Facies-Mark* com os modelos SETR, compostos por um enconder ViT e um decoder convolucional, algo novo no domínio das imagens sísmicas. A segmentação foi realizada utilizando dois modelos diferentes como encoder: ViT pré-treinado com a MAE e dados sísmicos ou ViT pré-treinado na tarefa de classificação do ImageNet.

Os resultados do trabalho mostram que o pré-treino agregou valor à segmentação com o modelo SETR MLA, indicando que essa abordagem é promissora para melhorias de problemas de visão computacional em dados sísmicos. No entanto, a utilização de um conjunto de dados maior e mais diverso é importante para fundamentar melhor a discussão.

As métricas e predições dos modelos SETR PUP e MLA confirmaram a qualidade dos resultados da segmentação com esse novo tipo de arquitetura. A métrica FWIoU apresentou um ganho de 7.45% em relação ao trabalho de Alaudah et al. (2019) e 4.32% em relação ao trabalho de Reis (2022).

A metodologia proposta para pré-treino de ViTs através de aprendizado auto-supervisionado com a MAE utilizando dados sísmicos e as avaliações qualitativa e quantitativa desses modelos é uma contribuição importante desse trabalho. Outra contribuição importante foi a demonstração da eficácia da arquitetura SETR na tarefa de segmentação de dados sísmicos e sua competitividade com as redes convolucionais.

Como direção para trabalhos futuros, seria importante investigar o efeito do treinamento da MAE com uma maior quantidade de dados sísmicos e avaliar a inclusão de aspectos próprios do domínio de interesse na função objetivo. Para os modelos de segmentação, realizar uma otimização dos parâmetros de treinamento e testar novas funções de custo seria uma frente importante de investigação.

# 7
# Referências bibliográficas


ALAUDAH, Y. et al. A machine-learning benchmark for facies classification. **Interpretation**, Society of Exploration Geophysicists and American Association of Petroleum, v. 7, n. 3, p. SE175–SE187, 2019.

AZIZI, S. et al. Robust and efficient medical imaging with self-supervision. **arXiv preprint arXiv:2205.09723**, 2022.

BIONDI, B. **3D Seismic Imaging**. [S.l.]: SEG Books, 2006.

CARON, M. et al. Emerging properties in self-supervised vision transformers. In: **Proceedings of the IEEE/CVF international conference on computer vision**. [S.l.: s.n.], 2021. p. 9650–9660.

CLAERBOUT, J. F.; FOMEL, S. **Image estimation by example: geophysical soundings image construction: multidimensional autoregression**. [S.l.]: Citeseer, 2008.

DELLINGER, J. et al. The garden banks model experience. **The Leading Edge**, Society of Exploration Geophysicists, v. 36, n. 2, p. 151–158, 2017.

DENG, J. et al. Imagenet: A large-scale hierarchical image database. In: IEEE. **2009 IEEE conference on computer vision and pattern recognition**. [S.l.], 2009. p. 248–255.

DOSOVITSKIY, A. et al. An image is worth 16x16 words: Transformers for image recognition at scale. **arXiv preprint arXiv:2010.11929**, 2020.

GAO, H.; WU, X.; LIU, G. Channelseg3d: Channel simulation and deep learning for channel interpretation in 3d seismic images. **Geophysics**, GeoScienceWorld, v. 86, n. 4, p. IM73–IM83, 2021.

HAGHIGHI, F. et al. Dira: discriminative, restorative, and adversarial learning for self-supervised medical image analysis. In: **Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition**. [S.l.: s.n.], 2022. p. 20824–20834.

HE, K. et al. Masked autoencoders are scalable vision learners. In: **Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition**. [S.l.: s.n.], 2022. p. 16000–16009.

JUNIOR, D. A. D. et al. Automatic method based on pso-optimized vision-transformer for gas detection in 2d seismic images. **Revista de Sistemas e Computação-RSC**, v. 12, n. 3, 2023.

KAUR, H. et al. A deep learning framework for seismic facies classification. **Interpretation**, Society of Exploration Geophysicists and American Association of Petroleum, v. 11, n. 1, p. T107–T116, 2023.

KUKREJA, N. et al. Rapid development of seismic imaging applications using symbolic math. In: EAGE PUBLICATIONS BV. **Third EAGE Workshop on High Performance Computing for Upstream**. [S.l.], 2017. v. 2017, n. 1, p. 1–4.

LI, J. et al. Unsupervised contrastive learning for seismic facies characterization. **Geophysics**, Society of Exploration Geophysicists, v. 88, n. 1, p. WA81–WA89, 2023.

MAATEN, L. Van der; HINTON, G. Visualizing data using t-sne. **Journal of machine learning research**, v. 9, n. 11, 2008.

MATSOUKAS, C. et al. Is it time to replace cnns with transformers for medical images? **arXiv preprint arXiv:2108.09038**, 2021.

REIS, N. C. **Classificação de fácies sísmicas utilizando multiatributos sísmicos**. Dissertação (Mestrado em Informática) — Pontifícia Universidade Católica do Rio de Janeiro, 2022.

SHI, Y.; WU, X.; FOMEL, S. Saltseg: Automatic 3d salt segmentation using a deep convolutional neural network. **Interpretation**, Society of Exploration Geophysicists and American Association of Petroleum, v. 7, n. 3, p. SE113–SE122, 2019.

STEINER, A. et al. How to train your vit? data, augmentation, and regularization in vision transformers. **arXiv preprint arXiv:2106.10270**, 2021.

TANG, Y. et al. Self-supervised pre-training of swin transformers for 3d medical image analysis. In: **Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition**. [S.l.: s.n.], 2022. p. 20730–20740.

TOLSTAYA, E.; EGOROV, A. Deep learning for automated seismic facies classification. **Interpretation**, Society of Exploration Geophysicists and American Association of Petroleum ..., v. 10, n. 2, p. SC31–SC40, 2022.

VASWANI, A. et al. Attention is all you need. **Advances in neural information processing systems**, v. 30, 2017.

WALDELAND, A. U. et al. Convolutional neural networks for automated seismic interpretation. **The Leading Edge**, Society of Exploration Geophysicists, v. 37, n. 7, p. 529–537, 2018.

WIGHTMAN, R. **PyTorch Image Models**. [S.l.]: GitHub, 2019. <https://github.com/rwightman/pytorch-image-models>.

WU, X. et al. Faultseg3d: Using synthetic data sets to train an end-to-end convolutional neural network for 3d seismic fault segmentation. **Geophysics**, Society of Exploration Geophysicists, v. 84, n. 3, p. IM35–IM45, 2019.

XIAO, J. et al. Delving into masked autoencoders for multi-label thorax disease classification. In: **Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision**. [S.l.: s.n.], 2023. p. 3588–3600.

ZHANG, A. et al. Dive into deep learning. **arXiv preprint arXiv:2106.11342**, 2021.

ZHANG, R.; ISOLA, P.; EFROS, A. A. Colorful image colorization. In: SPRINGER. **Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part III 14**. [S.l.], 2016. p. 649–666.

ZHENG, S. et al. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In: **Proceedings of the IEEE/CVF conference on computer vision and pattern recognition**. [S.l.: s.n.], 2021. p. 6881–6890.

ZHOU, L. et al. Self pre-training with masked autoencoders for medical image analysis. **arXiv preprint arXiv:2203.05573**, 2022.

# A
# Configurações de treinamento

Neste Apêndice apresentamos as configurações do treinamento da MAE e da SETR.

## A.1
## MAE

A configuração de parâmetros de treinamento seguiu a original do trabalho de He et al. (2022), com exceção do número de épocas e batch size. Os parâmetros utilizados estão na Tabela A.1

| parâmetro | valor |
|---|---|
| otimizador | AdamW |
| base learning rate (blr) | 1.5e-4 |
| weight decay | 0.05 |
| momentum | $\beta_1, \beta_2 = 0.9, 0.95$ |
| learning rate schedule | cosine decay |
| warmup epochs | 40 |
| épocas | 1600 |

Tabela A.1: Configurações de treinamento da MAE.

O batch size efetivo utilizado no treinamento de cada modelo variou devido à memória requerida por cada um deles. O batch size efetivo é igual a (batch size por gpu)×(número de nós)×(gpus por nó). Os experimentos foram executados em 8 nós, cada um deles com 4 GPUs NVIDIA V100. Na Tabela A.2 são apresentados os valores de batch size utilizados.

| modelo | batch size por gpu | batch size efetivo |
|---|---|---|
| ViT-S | 256 | 8192 |
| ViT-B | 128 | 4096 |
| ViT-L | 128 | 4096 |
| ViT-L_384 | 32 | 1028 |

Tabela A.2: Batch size utilizado no treinamento de diferentes modelos.

## A.2
## SETR

A configuração de parâmetros empregada foi a mesma de Zheng et al. (2021), com exceção do número de épocas. O treinamento dos modelos foi realizado em 1 nó com 4 GPUs NVIDIA V100. Os parâmetros utilizados no treinamento das redes estão na Tabela A.3.

| parâmetro | valor |
|---|---|
| otimizador | SGD |
| taxa de aprendizado | 0.001 |
| weight decay | 0 |
| momentum | 0.9 |
| batch size | 8 |
| learning rate schedule | polinomyal decay |
| épocas | 100 |

Tabela A.3: Configurações de treinamento das redes de segmentação.

# B
# Dado Sísmico

Aqui será feito um breve resumo sobre o dado sísmico. Para detalhes, recomendamos Biondi (2006).

O dado sísmico utilizado na interpretação sísmica é o resultado de um processo complexo, caro e demorado. A primeira etapa desse processo é o planejamento e execução da aquisição dos dados de campo. Esses dados podem ser adquiridos em terra ou mar.

Em mar, o tipo de aquisição mais simples envolve um navio sísmico, que carrega uma fonte de sinal, um canhão de ar comprimido, e um conjunto de receptores. Esses receptores são espalhados em diversos cabos com vários quilômetros de extensão. Cada receptor é um sensor de variação de pressão, chamado de hidrofone.

As ondas acústicas enviadas pela fonte sísmica penetram na subsuperfície e refletem nas regiões de grande contraste de propriedades físicas das rochas. Esse sinal refletido é registrado pelos hidrofones e compõe o dado sísmico de campo. Para cada tiro, alguns segundos de sinal são registrados. Esse processo está ilustrado na Figura B.1.

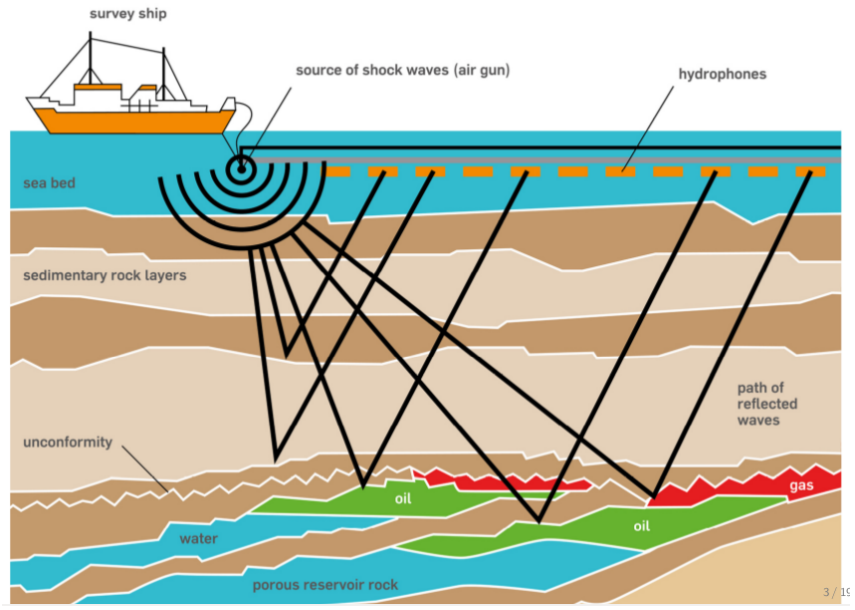



Figura B.1: Ilustração de uma aquisição sísmica marítima. Fonte Kukreja et al. (2017)

Os dados adquiridos passam então por uma etapa de processamento, que pode ser dividido em duas etapas: processamento em tempo e imageamento. De maneira bastante sucinta e simplificada, o processamento em tempo realiza as seguintes operações:

1. coloca o do dado de campo em um grid regular;
2. retira a assinatura da fonte, tornando-a parecida com um sinal do tipo spike;

3. atenua ruídos que degradam a qualidade do sinal sísmico refletido;

4. atenua sinais refletidos que prejudicam a etapa de imageamento.

Ruídos são causados, por exemplo, por ondas de superfície, navios, interferência de outras fontes sísmicas, entre vários outros. Um exemplo de sinal que prejudica o imageamento são as múltiplas, ondas acústicas que reverberam entre a superfície e fundo do mar antes de serem registradas pelos receptores.

A etapa de imageamento envolve a construção de um modelo de velocidades da subsuperfície e a etapa conhecida como migração, que gera a imagem sísmica 3D final. A construção do modelo de velocidades pode ser resumida como um problema inverso, fundamentado na equação da onda, em que se extrai o modelo dos próprios dados sísmicos processados em tempo. A migração, também um problema inverso fundamentado na equação da onda, posiciona os sinais registrados na posição em que foram originalmente gerados, criando um volume 3D com amplitudes associadas ao contraste de propriedades das rochas. Esse é o volume utilizado na interpretação sísmica.