SEGMENTAÇÃO DE IMAGENS SÍSMICAS BASEADA EM APRENDIZADO AUTO-SUPERVISIONADO E POUCAS AMOSTRAS ROTULADAS

BRUNO A. A. MONTEIRO

SEGMENTAÇÃO DE IMAGENS SÍSMICAS BASEADA EM APRENDIZADO AUTO-SUPERVISIONADO E POUCAS AMOSTRAS ROTULADAS

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação do Instituto de Ciências Exatas da Universidade Federal de Minas Gerais como requisito parcial para a obtenção do grau de Mestre em Ciência da Computação.

Orientador: Jefersson Alex Dos Santos
Coorientador: Hugo Neves de Oliveira

Belo Horizonte

Julho de 2023

BRUNO A. A. MONTEIRO

SEGMENTATION OF SEISMIC IMAGES BASED ON SELF-SUPERVISED LEARNING AND FEW-LABELED SAMPLES

Dissertation presented to the Graduate Program in Computer Science of the Federal University of Minas Gerais in partial fulfillment of the requirements for the degree of Master in Computer Science.

Advisor: Jefersson Alex Dos Santos
Co-Advisor: Hugo Neves de Oliveira

Belo Horizonte
July 2023

Monteiro, Bruno A. A.

D1234p Segmentation of Seismic Images Based on Self-Supervised Learning and Few-Labeled Samples / Bruno A. A. Monteiro. — Belo Horizonte, 2023
xxiv, 77 f. : il. ; 29cm

Dissertação (mestrado) — Federal University of Minas Gerais
Orientador: Jefersson Alex Dos Santos

1. Self-supervised learning. 2. Seismic Image. 3. Semantic Segmentation. I. TÃŋtulo.

CDU 519.6*82.10

Dedicated to all who try.

Acknowledgments

Many thanks to CAPES, CNPq (312102/2017-8, 424700/2018-2, and 311395/2018-0), FAPEMIG, FAPESP (grants #2020/06744-5 and #2015/22308-2), and Serrapilheira Institute (grant #R-2011-37776) for their financial support to this research project.

Resumo

As atuais metodologias de aprendizado profundo tradicionais para interpretação de imagens sísmicas dependem fortemente de grandes quantidades de dados rotulados. Embora muitos volumes sísmicos estejam disponíveis para download em bancos de dados públicos, esses dados não possuem uma interpretação associada. Isso coloca desafios significativos em relação à aceleração da interpretação sísmica. No entanto, esse campo também é de interesse crescente e apresenta muitas oportunidades de melhoria. Há também um interesse crescente em abordar problemas de segmentação com dados rotulados limitados, especialmente em cenários de poucas amostras. Essas metodologias oferecem o potencial para uma resolução mais efetiva do problema dos dados rotulados limitados. Assim como em muitos outros contextos, a interpretação sísmica também pode se beneficiar de métodos de aprendizado auto-supervisionado, que se baseiam em treinamento prévio sem rótulos manualmente anotados e posterior ajuste fino com poucos rótulos. Para demonstrar o potencial desse tipo de abordagem, foram conduzidos uma série de experimentos com três simples tarefas preliminares auto-supervisionadas: previsão de rotação, montagem de quebra-cabeça e previsão de ordem de slices. Também, utilizamos de tarefas prévias com múltiplos objetivos para verificar se a combinação de tarefas forneceria melhor ponto de partida para o ajuste posterior. Essas tarefas exigem que o modelo aprenda características semânticas dos dados, que podem ser usadas como ponto de partida para o ajuste fino em uma tarefa de segmentação semântica, com o objetivo de identificar as diferentes facies litoestratigráficas em seções sísmicas. Nossos resultados para 1, 5, 10 e 20 amostras rotuladas mostraram uma melhoria significativa nas medidas de Interseção-sobre-União para essa tarefa alvo na maioria dos cenários, superando o método de referência. Além disso, aplicamos técnicas de *ensemble* para aprimorar ainda mais o desempenho dos modelos ajustados, obtendo resultados ainda melhores para a tarefa de segmentação. Esses experimentos indicam que a aplicação de métodos SSL pode trazer benefícios substanciais para a interpretação sísmica, especialmente em situações com poucos dados rotulados disponíveis.

Abstract

Current traditional deep learning methods for seismic image interpretation heavily rely on large amounts of labeled data. Although many seismic volumes are available for download through public databases, these data do not have an associated interpretation. This poses significant challenges regarding the accelerating seismic interpretation. Nonetheless, this field is also of growing interest and presents many opportunities for improvement. There is also an increasing interest in addressing segmentation problems with limited labeled data, particularly in few-shot scenarios. These methodologies offer the potential to effectively tackle the limited labeled data issue. As in many other contexts, seismic interpretation can also benefit from self-supervised learning (SSL) methods, relying on prior training without manually-annotated labels and subsequent fine-tuning with few labeled samples. To demonstrate the potential of such an approach, we conducted experiments with three simple self-supervised pretext tasks: rotation prediction, jigsaw puzzling, and a frame-order prediction. Also, we used pretasks with multiple objectives to verify if the combination of tasks would provide the best starting point for the post-fit. These tasks require the model to learn meaningful semantic features about the data that can be used as a start-point for fine-tuning in a semantic segmentation task, aiming at identifying the different lithostratigraphic facies in seismic sections. Our results for 1, 5, 10, and 20 labeled samples (shots) showed significant improvement for Intersection-over-Union (IoU) measurements for the target task in most scenarios, outperforming the baseline method. Additionally, we applied ensemble techniques to further enhance the performance of the fine-tuned models, achieving even better results for the segmentation task. These experiments indicate that the employed SSL methods can benefit seismic interpretation, especially in situations with few available labeled data.

List of Figures

List of Tables

Contents

Chapter 1

Introduction

As petroleum exploration continues over the decades, major oil and gas reservoir discoveries in middle or shallow basins are not expected to occur, leading to exploration toward deep basins [Bjorlykke, 2015]. The prediction of the presence of the oil-hosting rocks is based on the identification of different stratigraphic facies, which allows the distinction between distinct geological formations essential to the comprehension of hydrocarbon migration and imprisonment. The tridimensional acquisition of seismic data results in vast data, which makes the manual interpretation of the seismic sections or volumes slow and biased [Waldeland et al., 2018, Liu et al., 2020a]. Using computational tools that support or automatize this interpretation provides better agility and can reduce the interpreter's subjectivity and bias.

Concomitantly to modern data acquisition processes, the actual computational capacity and the latest methods for pattern recognition brought great advances for many computer vision tasks. These advances are highly related to the development of representation learning methods, which consist of techniques that allow an automatic mapping between raw data tasks such as detection and classification [LeCun et al., 2015]. One major recent breakthrough came with the evolution of representation learning techniques that rely on multiple levels of representation, namely deep learning [LeCun et al., 2015]. These procedures offer a major advantage since they can process data in its raw form, which was a limitation of shallow (conventional) machine learning methods. The advance of deep learning methods made possible the realization of many computer vision tasks using deep network architectures [Jing and Tian, 2019], which also have great potential in seismic interpretation.

Semantic segmentation has an important role in image understanding. It has many applications in traditional visual recognition tasks, such as medical image analysis, autonomous driving, and remote sensing [Lateef and Ruichek, 2019]. Beyond

that, it can ease the interpretation of seismic images, for instance, in the discovery and delimitation of hydrocarbon reservoirs and mineral coal deposits, and better and faster interpretation of stratigraphic data. Examples of the use of deep learning in seismic interpretation include using Convolutional Neural Networks (CNNs) [Krizhevsky et al., 2012] for the classification of salt domes [Waldeland et al., 2018], geological faults/horizon delimitation [Guo et al., 2020a], and facies classification/segmentation [Liu et al., 2020a].

The performance of a deep neural network is highly dependent on the amount of available data during training. As manual labeling for segmentation is a slow, expensive, and biased process, self-supervised learning (SSL) methods are proposed as an alternative for representation learning [Pathak et al., 2016, Noroozi and Favaro, 2016, Gidaris et al., 2018, Chen et al., 2020a, Li et al., 2021, Wang et al., 2021]. SSL consists of pre-training networks via related *pretext tasks* with pseudo-labels obtained from the data itself – that is, not manually annotated. Hence, the network tries to identify relevant features within the target data domain instead of reusing features from a model pre-trained on ImageNet [Deng et al., 2009], since using different data distributions can be harmful [Su et al., 2020] and more accurate image classification does not implicate on better performance on dense prediction tasks [Wang et al., 2021]. During the pre-training phase, the network learns weights relevant to solving the pretext task in the same data domain as the final task. Then its parameters can be leveraged by fine-tuning into a downstream – often low-shot – task, as shown in Fig. 1.1. Using these pre-trained models, one can fine-tune a model with fewer labeled data, increase performance and avoid overfitting [Jing and Tian, 2019].

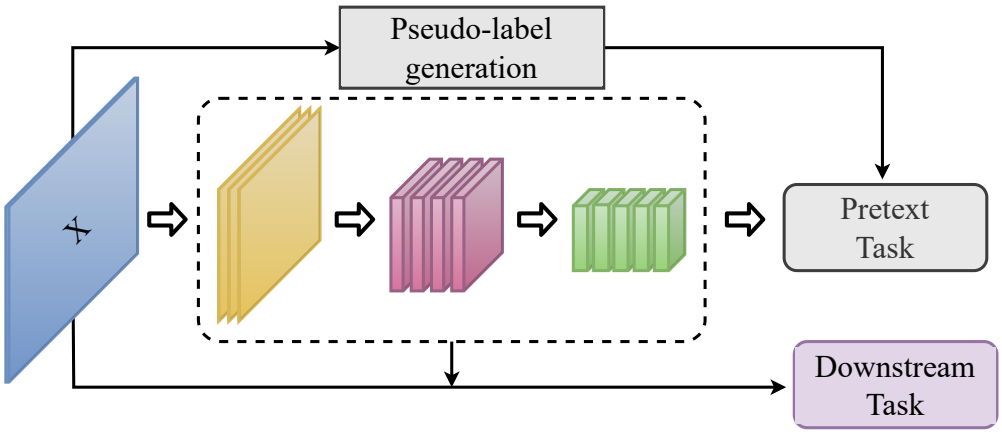


Figure 1.1: SSL pipeline. First, the model is trained in a pretext task within the targeted data domain with pseudo-labels, then the learned weights are repurposed to solve a downstream task. Better viewed in color.

1.1 Motivation

Currently, most mineral deposits and oil- and gas reservoirs that could be easily found have already been discovered [Bjorlykke, 2015]. Therefore, to assist in discovering new natural resources, it is necessary to invest significantly in knowledge and technology that help to understand the origin and location of these commodities. Petroleum geology encompasses a range of disciplines of greater significance in the search and recovery of petroleum and natural gas. Virtually all petroleum deposits occur in sedimentary rocks, so the prediction of the presence of rocks that compose this resource is based on identifying different stratigraphic facies. The facies are a compound of rock characteristics that allow the distinction between other groups. These facies are typically related to sedimentary depositional environments, which determine the reservatory rocks' distribution and composition, whose understanding is essential to comprehending hydrocarbon migration and imprisonment [Bjorlykke, 2015].

Seismic technologies have been employed since the 1900s to measure water depths and detect icebergs. With the advance of this technology, it started being used to find oil fields. Nowadays, seismic surveying composes the most important geophysical method for oil exploration [Mondol, 2010]. By measuring the traveling time of acoustic waves, this method can detect both large and small-scale features of subsurface rocks and estimate their shape and physical properties. There are two main types of seismic techniques, one using refraction and the most common using reflection of acoustic waves. This type of survey acquires data in 2D or 3D, which provides images and volumes of the subsurface after being processed.

These data volumes are called Stack Cubes, which are volumes oriented by perpendicular axes, named Inline and Crossline axes, in the X and Y plane, and Time Samples in the Z-axis (Fig. 1.2). So a seismic section is built using one of the horizontal axes, which shows the lateral distribution of the rock deposit, and the vertical one, which shows the vertical distribution (depth/time). The study of the facies through seismic sections is called seismic stratigraphy (or seismostratigraphy). The analysis of the seismic facies in seismic sections aims to interpret the sedimentary deposition environment by understanding the characteristics shown in the seismic reflection data. The seismostratigraphic facies definition must consider all information constrained within the profiles to identify which parameters characterize it. For that, it is necessary to choose the properties that allow differentiating and interpreting the seismic facies through a multivariate analysis of the profile parameters [Dumay and Fournier, 1988].

Seismic data interpretation composes a critical portion of the geophysical investigation process using seismic reflection. The routine of manually interpreting strati-

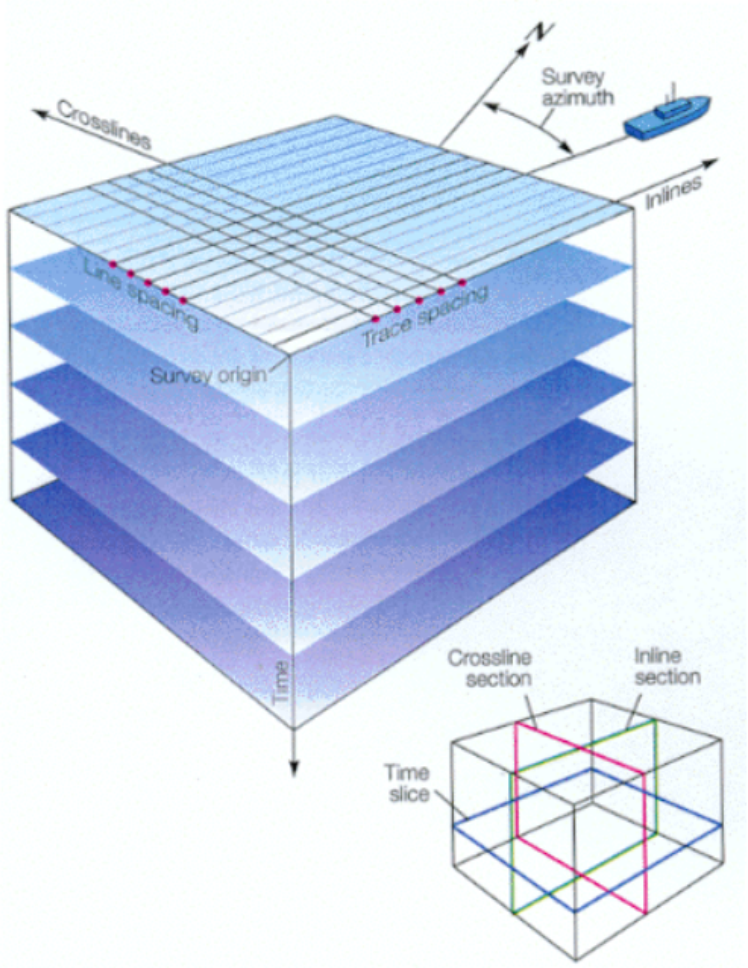


Figure 1.2: Example of the design of tridimensional seismic data acquisition. Image source: [EnergyGlossary, 2023]

graphic facies occurs in collaboration between geologists, geophysicists, and petrophysics. However, due to the large volume of seismic acquisitions, this process demands a highly time-consuming effort. Therefore, there is an actual interest in using machine learning algorithms to help interpret these data.

Typical shallow machine learning methods often suffer from the lack of capability in processing raw data input. To enable a pattern recognition system, a dense knowledge of the application area is often necessary since linking data to the pattern itself requires sensible modeling. Representation learning composes a compound of methods that capacitate the networks to be fed with raw data, in which the data itself will provide the necessary patterns to be automatically discovered for tasks such as detection or classification. Deep learning methods are learning methods that use multiple layers of characteristics abstraction that are extracted by non-linear modules that learn higher semantic features in each layer. Composing enough transformations, a deep network

can learn complex functions that allow the distinction of the essential parameters of the study object [LeCun et al., 2015].

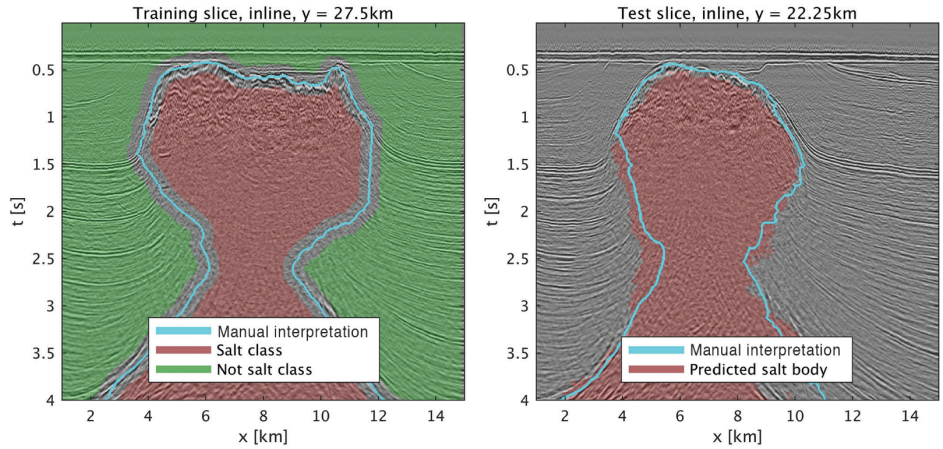


Figure 1.3: Comparison of CNN performance and manual delimited salt dome on the seismic image. Left: Inline section used on the network training phase. Right: Test section with the prediction and its true label. Image source: [Waldeland et al., 2018]

CNNs have played a key role in compelling significant progress within computer vision. They form the foundational framework for numerous contemporary architectures that find utility across diverse applications like image classification and semantic segmentation, as evidenced by Szegedy et al.'s work [Szegedy et al., 2014]. In particular, the significance of semantic segmentation reverberates profoundly in the domain of image interpretation. Its applications span the computer vision field, encompassing essential tasks such as the analysis of medical images, facilitation of autonomous driving, and utilization within remote sensing contexts [Lateef and Ruichek, 2019]. Moreover, the implications of semantic segmentation extend even further, encompassing its utility in interpreting seismic images. As an early work, Waldeland [Waldeland et al., 2018] utilizes CNN for the classification of salt domes in seismic images (Fig. 1.3). Karimpouli [Karimpouli and Tahmasebi, 2019] uses CNN for digital rock image segmentation. Liu [Liu et al., 2020b] conducted a study comparing the classification of seismic facies using both supervised CNN and semi-supervised Generative Adversarial Networks (GANs). Guo [Guo et al., 2020a] applies CNN to detect geological faults and horizon interpretation.

1.2 Objective

Having in mind that learning from few labeled data is a trending topic in deep learning research and that labeling data is a current problem in seismic interpretation, this

study has the following objectives:

- **General objective**:
  - Develop self-supervised learning methods for seismic interpretation
- **Specific objectives**:
  - Test pretext tasks that best generalize visual features from seismic images
  - Verify the performance improvement of models trained from scratch against fine-tuned models using self-supervised tasks
  - Verify the performance advancement brought by ensembling techniques combining models pre-trained in a self-supervised manner
  - Create a routine for semantic segmentation of seismic facies in seismic images using a self-supervised learning approach

1.3 Contributions

The development of this dissertation resulted in a new routine for dealing with the automated segmentation of seismic images, relying on a self-supervised learning approach. Although this method could not achieve state-of-the-art performances, it improved performance compared to a well-defined baseline, utilizing a relatively simple and light method and using less than 2% of the available labels. It also raised many insights for further studies and improvements in self-supervised learning on few-shot scenarios in non-traditional images. This contribution allowed us to have published a summarized version of this work in the IEEE Geosciences and Remote Sensing Letters. The publication is entitled "Self-Supervised Learning for Seismic Image Segmentation From Few-Labeled Samples" [Monteiro et al., 2022][1].

[1] https://ieeexplore.ieee.org/document/9837909

Chapter 2

Literature Review and Related Work

This chapter seeks to provide the necessary support for understanding the proposed approaches. It consists of a literature review regarding deep learning and self-supervised learning methods and also works on applying these techniques in seismic images. Section 2.1 provides a brief explanation of the procedure to obtain the seismic images. Section 2.2 shortly reviews deep learning methods and introduces common SSL approaches. Section 2.3 discusses some applications of such methods to seismic interpretation.

2.1 Brief Seismic Method Background

Seismic surveys use seismic waves to infer the properties of a geological layer. As rocks are a medium with elastic behavior, acoustic waves can propagate through them, and their physical properties can be investigated by observing the changes in the waves' characteristics. In seismic surveys, the waves are usually generated by explosives (on the ground) or compressed air/air guns (on the water). The artificially generated wave propagates through the rocks. When there is a variation between layers, e.g., a change in sediment deposition resulting in distinct geometries, these waves cause many physical phenomena, such as reflection, refraction, diffraction, and scattering. For seismic surveys, seismic reflection plays the most crucial role, as it can be used to register the time required for a wave to travel from a source, hit a target, and return. These recorded signals are the measurements of the two-way travel time, which is when a seismic wave travels from the wave generator, hits the reflection surface, and returns to the receiver. Then, the seismic images are generated by processing the acoustic

response of the reflection of a seismic wave created by a change in the propagation medium, i.e., change of rock properties [Bjorlykke, 2015, Herron, 2011, Nanda, 2021].

Two conditions are required to create a seismic reflection (Fig 2.1). First, there must be an acoustic impedance (AI) contrast between rock types, and second, the interface between these layers must be large enough to be detected. The degree of contrast between the acoustic impedance (Eq. 2.1) of two layers governs the reflection coefficient (Eq. 2.2), producing a stronger or weaker seismic response [Herron, 2011]. The quality of the reflection signal is, among other features, a result of the signal-to-noise ratio and the selection of the wavelet capable of discerning the reflection surfaces separately [Nanda, 2021].

$$AI = V\rho, \tag{2.1}$$

where V and ρ are the compressional-wave velocity and bulk density, respectively.

$$RC = (\frac{AI_2 - AI_1}{AI_2 + AI_1}), \tag{2.2}$$

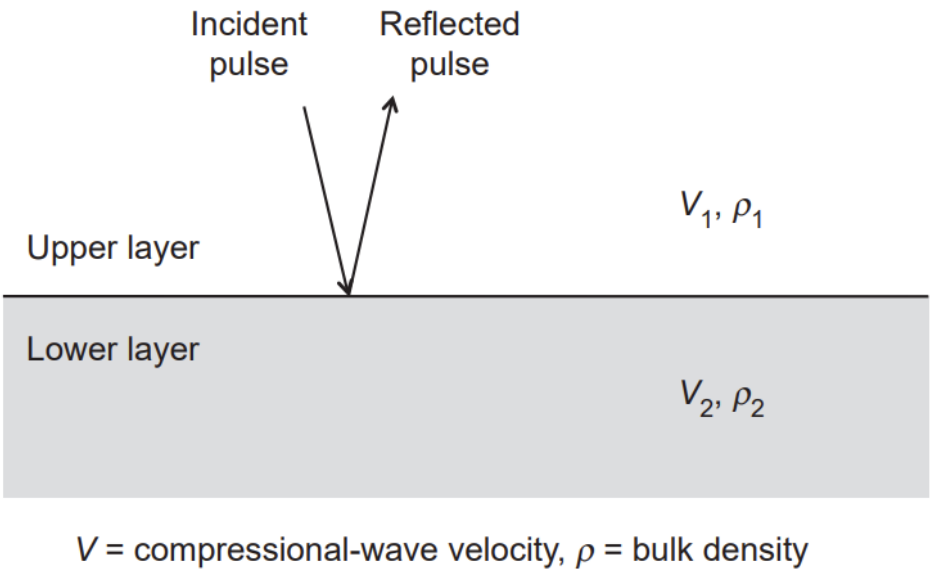


Figure 2.1: Representation of a seismic reflection produced by the incidence of a compressional wave onto the interface of two layers with contrasting acoustic impedance. Image Source: [Herron, 2011]

The acquisition of the seismic signals, their processing, and the interpretation of the seismic data compose an intrinsically related workflow (Fig. 2.2). The seismic response results from the display of the layers and the contrast of acoustic impedance and the reflection coefficient between them. On the other hand, the interpretation tries to accomplish the inverse work (a process called seismic inversion), i.e., by the acquired

seismic response, tries to infer the geological distributions throughout the reflection coefficient and acoustic impedance contrast, among other techniques [Herron, 2011].

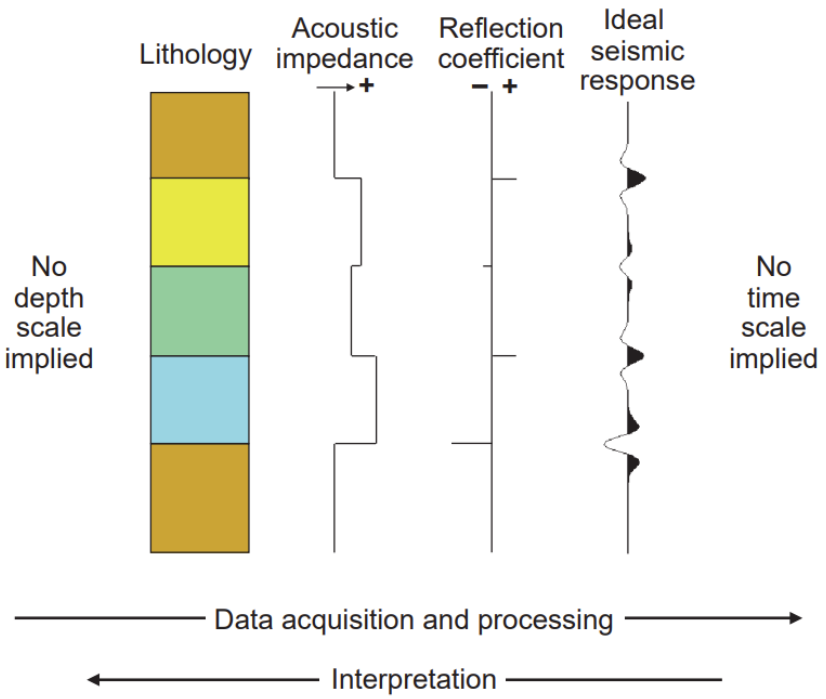


Figure 2.2: Relationship between seismic data acquisition, processing, and interpreting. Image Source: [Herron, 2011]

2.2 Literature Review on SSL for Image Recognition

In this section stands the introduction to the basic concepts required for the development of this research, beginning with some definitions, then explanations about pretext tasks applied in self-supervised learning and their evaluation of target tasks. And to close the chapter, we brought a set of deep learning methods related to interpreting seismic facies.

Supervised Learning In a supervised learning approach, for each $\boldsymbol{x}_i$ in a dataset (X, Y), there is a human-annotated label y_i. One tries to find a function capable of approximating the distribution of Y given the input X: $P(Y|X)$. For that, one searches a function for a set of weights Θ that maps an example x_i into their respective label y_i.

For a set of labeled training data X, the optimal weights Θ_* can be updated through the derivative of a loss function such as L:

$$L(\hat{Y}, Y) = \frac{1}{N}\sum_{i=1}^{N} loss(f(x_i), y_i), \tag{2.3}$$

The loss function $L(\hat{Y}, Y)$ computes the error between the predicted results $\hat{Y} = f(X)$ and the true labels Y.

This approach has obtained satisfactory results in many tasks [Krizhevsky et al., 2012, Simonyan and Zisserman, 2014, He et al., 2016]. However, data collection and annotation can be slow, expensive, and biased to the interpreter. Semi-supervised and self-supervised methods are one way to come around this problem by identifying data attributes and annotating them using only the intrinsic information contained in the data.

Semi-Supervised Learning It groups the learning methods that use a small amount of human-annotated labels and a larger amount of unlabeled data. Consider a labeled dataset X and an unlabelled dataset, for which each x_i sample in training set N has a corresponding human-annotated label, and z_i in training set M has none. For this method, the loss comprises both losses from the labeled and unlabeled datasets, as follows.

$$L(Z, X) = \frac{1}{N}\sum_{i=1}^{N} loss(f(x_i), y_i) + \frac{1}{M}\sum_{i=1}^{M} loss(f(z_i), F(z_i, x_i)), \tag{2.4}$$

$F(z_i, x_i)$ is a task-specific function that connects each unlabelled training data z_i to the labeled dataset X.

One approach to semi-supervised learning consists of the usage of self-supervised training through solving pretext tasks followed by supervised fine-tuning with a smaller labeled dataset, also called "unsupervised pre-train, supervised fine-tune" [Chen et al., 2020b].

Self-Supervised Learning Unsupervised learning refers to learning methods that do not need any human-annotated labels. A compelling case of this approach is called self-supervised learning, which is when pseudo-label are generated for each x_i sample through predefined transformations, and use these pseudo-labels as an artifice to train the network. Like supervised methods, SSL trains with data X and its pseudo-label P and uses loss functions to measure the distance between the predicted label and

the pseudo-label. So, for a training data X, the weights Θ for model $f(X)$ can be optimized with loss functions L such as:

$$L(P, X) = \frac{1}{N} \sum_{i=1}^{N} loss(f(x_i), p_i), \tag{2.5}$$

where P are the pseudo-labels associated with X, and $\hat{Y}$ are the predicted label.

The most widely used SSL approaches relate to discriminative or generative learning [Chen et al., 2020a]. Generative methods learn how to create and model an output given a latent input space, while discriminative methods learn data representation through objective functions similar to supervised learning but without human-annotated labels. Among the discriminative methods, one branch relies on contrastive learning, based on using similarity functions to bring similar samples together and push distinct samples apart. The other branch, which will be the focus of this study, relies on heuristics to design pretext tasks, with strong control of the data transformation and creation of labels.

It is relevant to distinguish that few-shot learning aims to learn representations that generalize to tasks where only a few images are available [Su et al., 2020]. Meanwhile, self-supervised learning benefits from large unlabelled datasets for improvement of generalization capability, relying on a few samples only for fine-tuning when performing semi-supervised learning.

2.2.1 Pretext Tasks

Many usual self-supervised learning methods follow the procedures of trying to solve pretext tasks [Noroozi and Favaro, 2016, Doersch et al., 2016, Gidaris et al., 2018, Su et al., 2020]. They are optimized by minimizing the error between the model's output and the pseudo-label generated automatically without any human-annotated label. It is expected for the network to learn interesting semantic features of the data to solve this task so one can perform transfer learning and evaluate another downstream task or even use this model with no extra training in a zero-shot context, e.g., [Kirillov et al., 2023].

There is still great effort in finding the appropriate pretext task, and its choice is based on the application for which the model is being pre-trained. These SSL pretext tasks aim to build a model capable of generalizing the discriminative information about the data and learning interesting visual features. One must not ignore that pre-training a model for such tasks creates bias and can be beneficial in some cases and harmful in others [Goodfellow et al., 2016, Jaiswal et al., 2021]. To avoid this type of heuristics,

many contrastive learning methods have been proposed as possible solutions that surpass the capability of state-of-the-art supervised methods, e.g., [Chen et al., 2020a]. [Su et al., 2020] conduct a discussion about the role of self-supervised methods and the employed pretext tasks on few-shot meta-learning and present a technique that can select images to be used on these tasks based on the distance between the domains of the pretext task and downstream task.

Based on the data attributes used to design the pretext tasks, it is common to aggregate pretext tasks into four categories: generation-based, context-based (Innate relationship), self-prediction (free semantic label-based), and contrastive (Fig. 2.3).

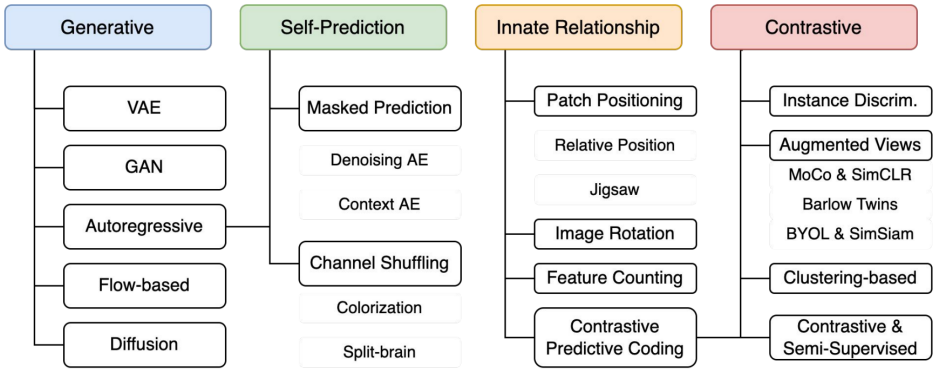


Figure 2.3: Taxonomy of self-supervised learning pretext tasks based on their intrinsic attributes. Image Source: [Weng and Kim, 2021]

2.2.1.1 Generation-Based Methods

The Generation-based pretext tasks aim at learning visual features of the data through the generation of images or videos, e.g., image colorization, image inpainting, image generation with GANs, etc (see Fig.2.4). Usually, in these tasks, instead of building or reconstructing a whole image or input, the model is fed with a partial input and then tries to complete the information.

Figure 2.4: Generation-Based Methods examples. Image Source: [Liu et al., 2020c]

Image Colorization Proposed by [Zhang et al., 2016], provides a gray-scale version of the original image as input to the network and tries to learn to colorize it by using as a pseudo-label the original version. The network receives an input X and learns a mapping $\hat{Y}$=F(X) using the multinomial cross-entropy loss between the predicted and ground truth colors. The main idea is that to map plausible colors to the image, and so the network must learn to extract visual features so it can recognize the objects and group its pixels.

Super Resolution [Ledig et al., 2017] present the SRGAN, a generative adversarial network (GAN) for image super-resolution (SR), which is capable of recovering a high-resolution version for 4x upscaling factors given a blurred input. It works with two networks: one generator that enhances the low-resolution input using an L2 loss plus the feature similarity between prediction and original sample; the other is the discriminator, which is trained to differentiate between the super-resolved images and original photo-realistic images through binary classification loss.

Image Inpainting In the image inpainting task [Pathak et al., 2016], given an incomplete image, randomly cropped, the model tries to fill the void. Concomitantly a discriminator is trained to predict whether the whole image is original or inpainted. The generator network must learn semantic features of the whole image as the structure of the objects and their colors, so that it can produce a plausible hypothesis for the missing part(s) (Fig.2.5).

2.2.1.2 Context-Based Methods - Innate Relationship

As a representation learning problem, with this task, the network tries to build useful intermediate representations that can be used in downstream tasks via transfer learning.

Image Jigsaw Puzzle The task consists of giving an image divided into segments for the network to predict the relative positions of two patches or the permutation performed given all image tiles (Fig. 2.6). [Noroozi and Favaro, 2016], showed by training an adapted ConvNet (CFN – context-free network) to solve jigsaw puzzles that it is possible to learn both a feature mapping of object parts as well as their correct spatial arrangement. The task is solved through the observation of all input tiles at once. It requires that the model learn spatial context information as the relative position (random permutations are provided to avoid absolute positioning shortcuts)

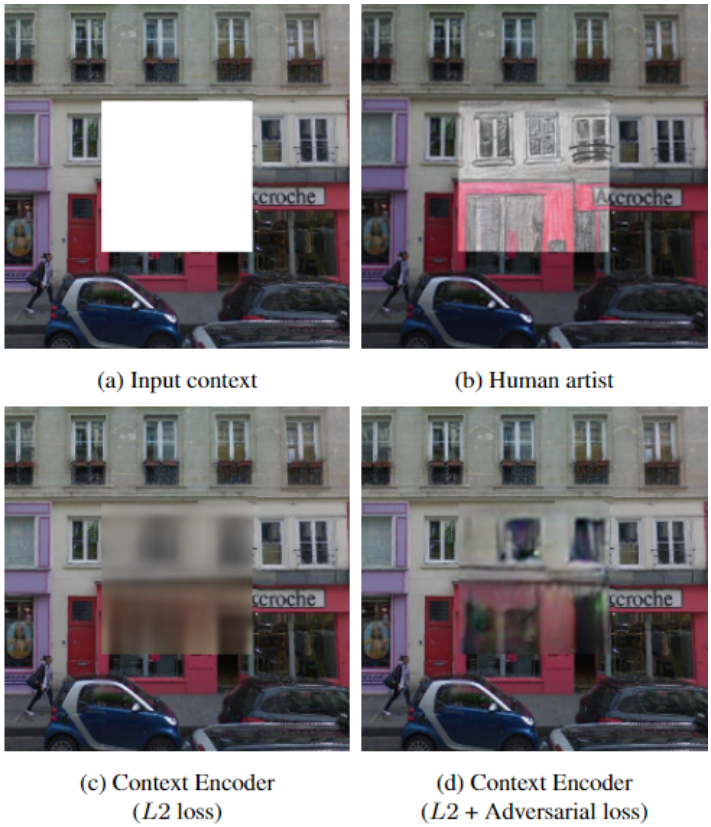


Figure 2.5: Image Inpainting as pretext task. On the top left is the original image with a missing part. On the top right, the label. On the bottom, two results obtained by [Pathak et al., 2016] using context encoder. Image Source: [Pathak et al., 2016]

and contour detection. They perform this task by cutting the input image into nine tiles and randomly permuting them so the network must infer the permutation sequence. Since nine tiles give 9!=362.880 possible permutations, hamming distance is employed to select only a subset of the possibilities, for the model would not be able to predict all the random occurrences. So the training phase is fed only with sufficiently large average Hamming distance configurations.

Geometric Transformation These approaches are proposed to learn image representations by training neural networks to recognize the geometric transformation. At first, a small set of discrete geometric transformations is defined. Then, each geometric transformation is applied to each image on the dataset, and the produced transformed images are fed to the model trained to recognize each image's transformation.

In [Gidaris et al., 2018], the authors propose to train the model on the image classification task of recognizing one of the four image rotations (0°, 90°, 180°, 270°). They argue that for a model to recognize the rotation transformation applied to an

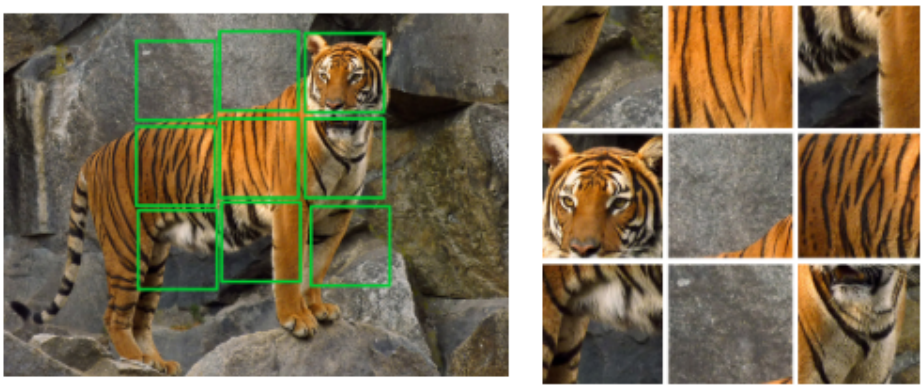


Figure 2.6: Visualization of the Jigsaw Image Puzzle. On the left the original image and the selected patches. On the right is the applied permutation. Image Source: [Noroozi and Favaro, 2016]

image, it will require understanding the concept of the objects depicted in the image, such as their location, type, and pose. So, predicting rotation transformations improves the results for classification image tasks. Similarly, the authors in [Jing et al., 2018] propose to use the same approach for classifying videos. Thus, it performs one of the four image rotations (0°, 90°, 180°, 270°), and finally, trains a model to recognize which rotation was performed in the video.

2.2.2 Temporal Context Structures

Videos encompass various sequential frames and contain valuable spatial and temporal information. The intrinsic temporal information present in videos can serve as a supervision signal for self-supervised feature learning [Jing and Tian, 2019]. The CNN can be trained to identify frame sequence order [Lee et al., 2017] or to check if the input frame sequence is in the correct order [Misra et al., 2016]. [Lee et al., 2017] employ temporal coherence as a supervisory signal by formulating representation learning as a sequence sorting task. They utilize shuffled frames, arranged in non-chronological order, as inputs, then train a convolutional neural network to sort these shuffled sequences (Fig. 2.7). This sorting task requires understanding the statistical temporal structure of images, and training with this proxy task allows for learning rich and generalized visual representations. Their learned representation's effectiveness is validated using the proposed method for pre-training on high-level action recognition problems.

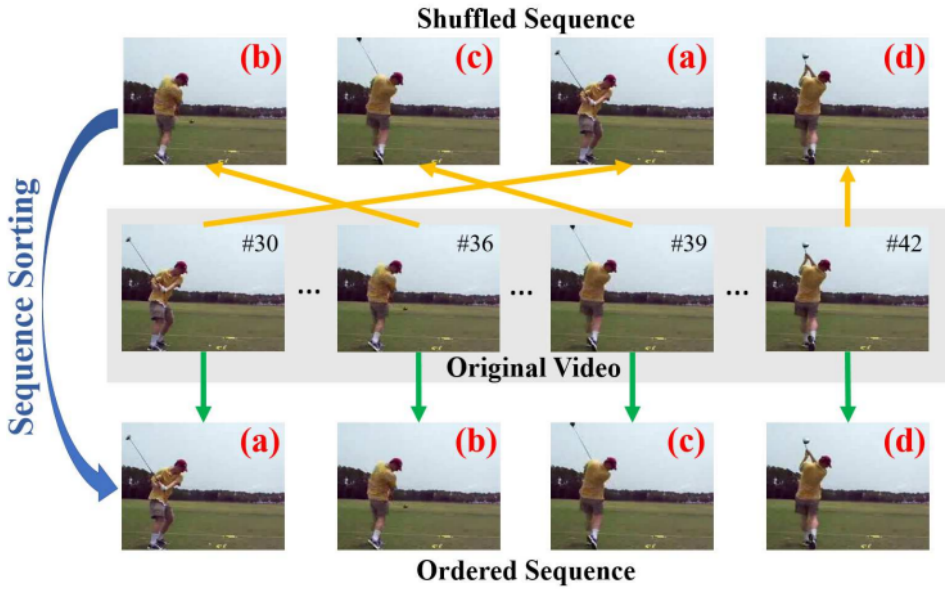


Figure 2.7: Representations learning by sequence sorting. Image Source: [Misra et al., 2016]

2.2.3 Clustering

These approaches try to group similar elements into clusters in a new latent space estimated by the model and group similar elements near the embedding space. Trying to solve the absence of labels, DeepCluster[Caron et al., 2018] proposes to leverage clustering to yield pseudo-labels and asks a discriminator to predict images' labels. This approach uses K-means to cluster encoded representation and produces pseudo labels for each sample. Then, the discriminator predicts whether two samples are from the same cluster and back-propagates to the encoder.

Similarly, Local Aggregation (LA) [Zhuang et al., 2019] uses a pseudo-label approach based on k-means, but unlike DeepCluster, the samples are not assigned to a mutual-exclusive cluster in LA. In addition, LA employs an objective function that directly optimizes a local soft-clustering metric. These modifications primarily boost the performance of LA representation on predictive tasks. Recently, some approaches have used functions that estimate pseudo-soft-label through a new representation of the data based on contrastive learning, as can see in [Tian et al., 2020, Chen et al., 2020a]. These instance discrimination-based methods have gotten rid of the slow clustering stage. It introduced efficient data augmentation (i.e., multi-view) strategies to boost the performance, as shown in Fig. 2.8.

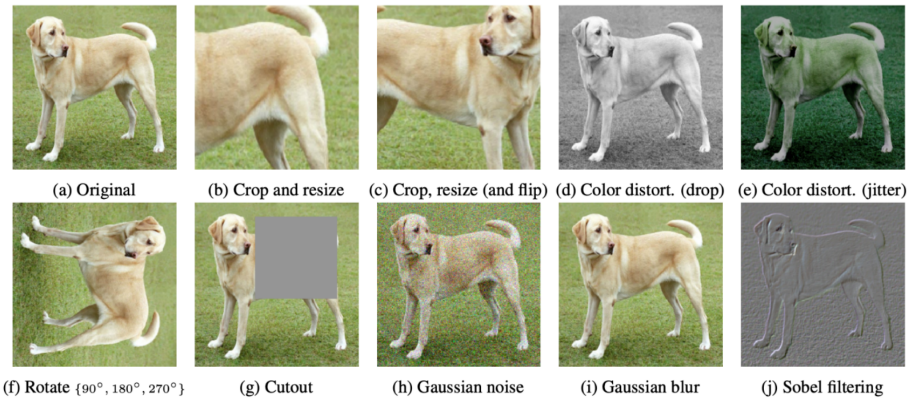


Figure 2.8: Data augmentation strategies applied by [Chen et al., 2020a]. Image Source: [Chen et al., 2020a]

2.2.4 Downstream Tasks

Despite great advances in self-supervised learning, especially in contrastive learning, data labels remain an important factor when training machine learning models for specific targets, especially dense prediction tasks. SSL models are powerful feature extractors, but there is still a gap between the objectives defined for SSL tasks and the aimed final tasks, and therefore semi-supervised learning is often employed to bridge this disparity [Liu et al., 2020c]. Usually, when applying a self-supervised learning pipeline, it is common to split the process into two steps: the pretext tasks and the downstream tasks. These downstream tasks evaluate the learned parameters during the self-supervised stage, using the pre-trained model as a starting point for fine-tuning the desired task, such as image classification, object detection, semantic segmentation, etc [Jing and Tian, 2019].

2.2.4.1 Image Classification

Image classification consists of the prediction of one class given an image input. Many revolutionary architectures were designed for this task, such as AlexNet [Krizhevsky et al., 2012], ResNet [He et al., 2016], and GoogLeNet [Szegedy et al., 2014]. When applying image classification as an evaluation task, the self-supervised pre-trained model works as a feature extractor that has stacked to its output layers a linear classifier that will be responsible for relating the learned representation to the desired classes. The models' comparison can be to distinguish the performance of self-supervised models and tasks or to compare the performance against a fully supervised model. Modern networks such as SimCLR v2 [Chen et al., 2020b] show that with 10% of ImageNet labels, their ResNet-50 can achieve better results than the state-of-the-art supervised

network on the same task.

2.2.4.2 Object Detection

Object detection consists of locating one or more objects, with a bounding box and its class, given an input image (Fig. 2.9). When it is applied as a downstream task of self-supervised methods, usually the pre-trained model is the base for the Fast-RCNN [Girshick, 2015], which will be fine-tuned to the detection task in a semi-supervised way [Jing and Tian, 2019].

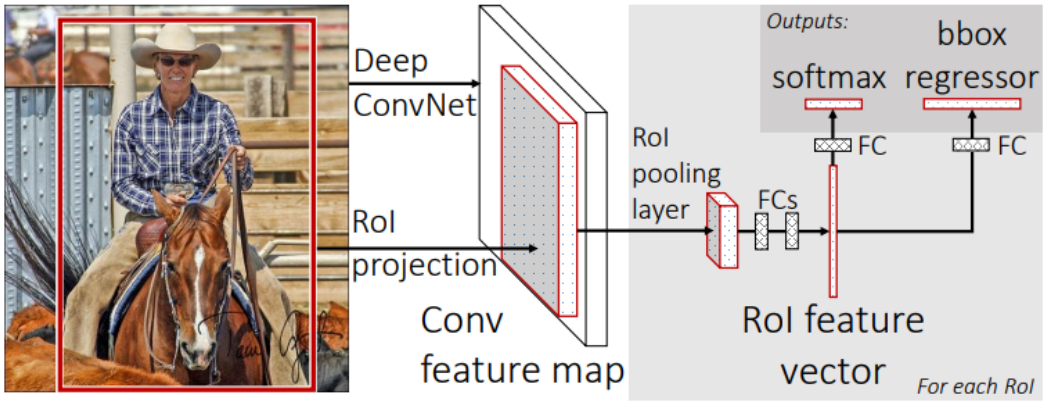


Figure 2.9: Pipeline of the Fast-RCNN for Object Detection proposed by [Girshick, 2015]. Image Source: [Girshick, 2015]

2.2.4.3 Semantic Segmentation

Semantic segmentation, the task of classifying each pixel of an image, plays an important role in image understanding and has several applications. Usually, consolidated architectures such as ResNet and AlexNet are used as base networks to extract the features. Still, instead of a fully-connected layer for image classification, transposed convolutions are applied, allowing a pixel-wise classification as in the FCN [Long et al., 2015a], (Fig. 2.10), SegNet [Badrinarayanan et al., 2017], or U-Net [Ronneberger et al., 2015]. When this task is used for target evaluation, the pre-trained model parameters are used as initialization points for fine-tuning with the label adjustment. Using metrics such as Jaccard's Score (IoU), one can compare the performance of a baseline network, such as training FCN from scratch with fine-tuning a pre-trained model with the self-supervised paradigm.

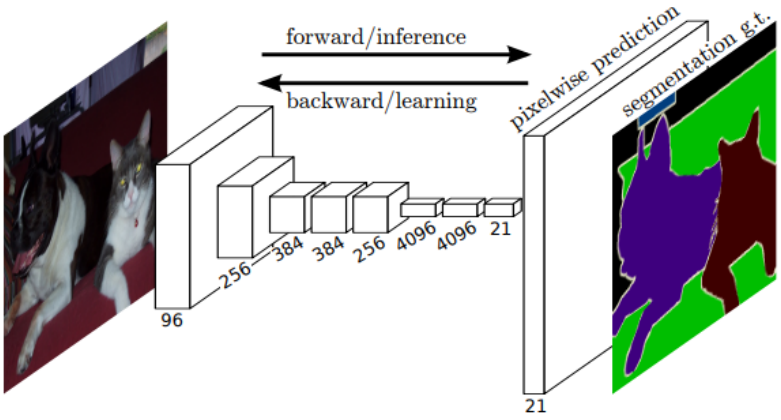
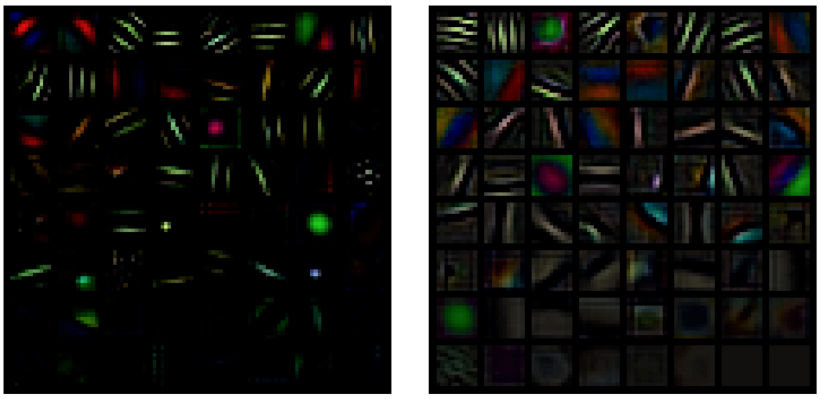


Figure 2.10: FCN architecture applied to dense predictions for pixel-wise classification. Image Source: [Long et al., 2015a]

2.2.4.4 Qualitative Evaluation

Three approaches are often used to visualize the learned parameters: kernel visualization, feature map visualization, and image retrieval visualization [Jing and Tian, 2019].

Kernel Visualization allows us to see the learned weights on the kernels of the convolutional layers. The shape of the kernels and the comparison against supervised methods can indicate the capability to learn through self-supervised methods (Fig. 2.11).



(a) **Supervised** (b) **Self-supervised to recognize rotations**

Figure 2.11: Kernel visualization obtained through rotation prediction task by [Gidaris et al., 2018]. a) Kernel visualization obtained using the supervised paradigm; b) Kernel visualization obtained by their SSL model. Image Source: [Gidaris et al., 2018]

Feature Map Visualization shows the activations learn by the network. Higher activations indicate that the layer/network is giving more attention to the specific region given that input (Fig. 2.12).

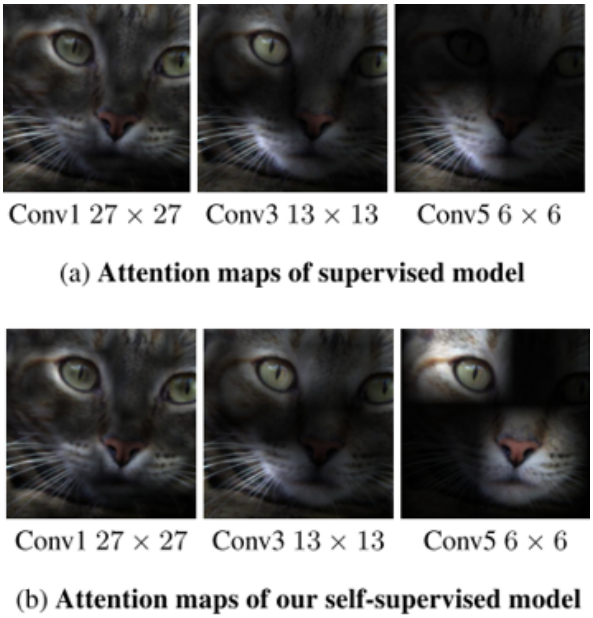


Figure 2.12: Attention Maps obtained through rotation prediction task by [Gidaris et al., 2018]. a) attention maps obtained using the supervised paradigm; b) Attention maps obtained by their SSL model. Image Source: [Gidaris et al., 2018]

Nearest Neighbor Retrieval is most relevant for similar images, which result in similar data representation on the latent space. This technique gives the K-nearest neighbors of the given representation on the latent space.

2.2.5 Architectures

CNN consists of the primary approach for most of the problems involving learning representation from images [LeCun et al., 2015]. Hence, this type of architecture is also strongly present on SSL when working with images. This section briefly describes some predominantly used CNN for image classification and semantic segmentation, for they have added great insights into the computer vision area and are strong candidates for good performance on self-supervised approaches.

AlexNet [Krizhevsky et al., 2012]: It is responsible for a massive breakthrough that has grown the state-of-the-art ceiling in computer vision applications. It contains eight learned layers — five convolutional and three fully connected (Fig. 2.13). A big part of SSL techniques for visual representation learning approaches uses this architecture. Nevertheless, this network employs a considerable scale of parameters and tends to overfit, which makes it necessary to apply techniques such as data augmentation, dropout, and normalization.

GoogLeNet [Szegedy et al., 2014]: Also named InceptionV1, it is a 22-layer

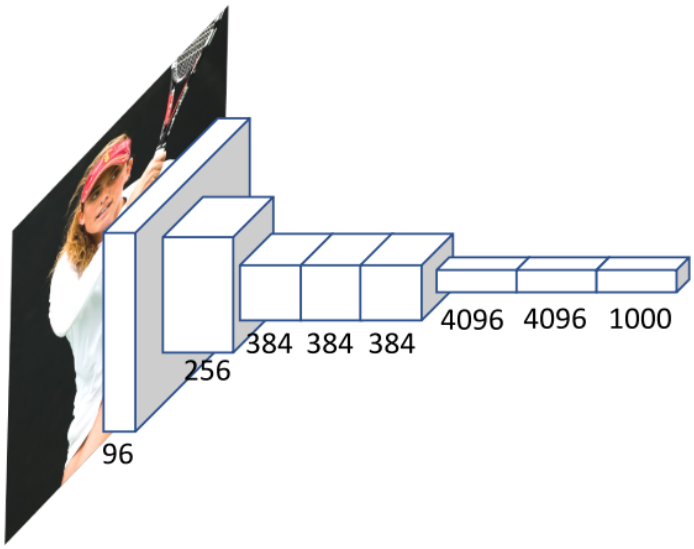


Figure 2.13: AlexNet architecture showing each layer number of channels. Image based on AlexNet architecture by [Jing and Tian, 2019]

convolutional network that tried to achieve higher performances not only by making the architecture deeper but also wider through the usage of the Inception modules. These modules use four parallel convolution layers, which use 1x1 convolutions to guarantee a dimension reduction before higher kernel convolution to increase memory efficiency.

ResNet [He et al., 2016]: After the VGG [Simonyan and Zisserman, 2014] demonstrated the improvement that very deep networks could proportionate, many issues came along, such as exploding gradients and vanishing gradients. To overcome this, the ResNet [He et al., 2016] was proposed as applying the concept of skip connections in residual blocks (Fig. 2.14), which sends the feature map from a previous convolutional layer into the next one, so the activation in the next layer is at least as strong as the previous one. Using this technique allowed for building a network with over one hundred layers. However, despite the increased number of layers, by avoiding fully-connected layers, this network significantly decreases the number of parameters. Combining the reduced parameter size and the superior performance resulted in many variations of ResNet that have been applied to computer vision tasks.

DenseNet [Huang et al., 2018]: All previously mentioned networks employ architectures that receive an image as input in the first layer and pass a feature map to the next ones. It is understood that the bottom layers are responsible for learning low semantic features, such as lines, and the top ones learn higher semantical features, such as complex forms. When using deeper ConvNets, this might result in memorizing low semantic features by the last layer so that it can accomplish the task, but this can

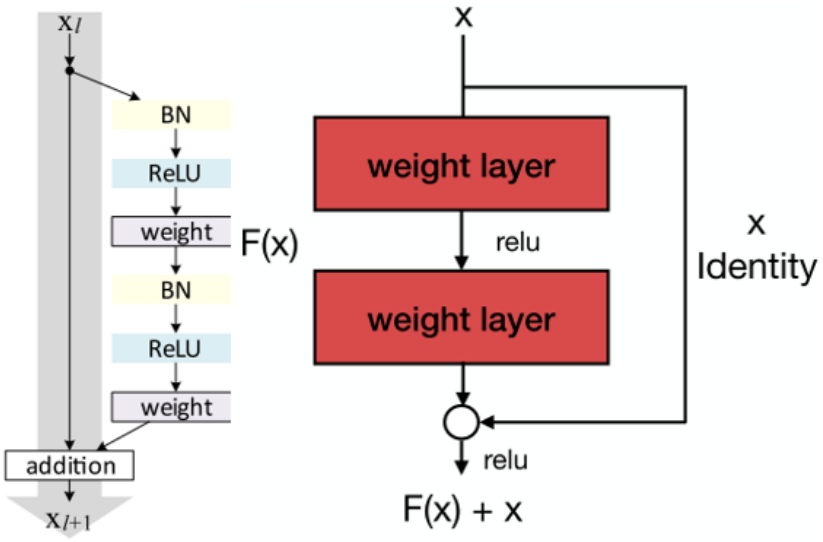


Figure 2.14: Residual Block with skip-connections used in ResNet networks. Left: Proposed Residual Block on [He et al., 2016]. Right: Representation of a common Residual Block. Image Source: [Jing and Tian, 2019]

compromise the performance. To overdue this issue, [Huang et al., 2018] proposed the DenseNet, which instead of only providing feature maps hierarchically, gives as input to each convolutional block the entire set of learned features, so the final layers can inherit the low-level features and focus on learning the more semantic and complex components.

FCN [Long et al., 2015b]: proposed the first fully connected convolutional network, which by the time made a breakthrough on the state-of-the-art semantic segmentation task, surpassing over 20% of the previous network, making this architecture standard for segmentation tasks. They have adapted classification networks (AlexNet, VGG, GoogLeNet) into fully connected, which take an arbitrary size for input and returns an image on the same shape with the classification per pixel. In order to recover the spatial information lost during downsampling in the encoder, the decoder performs a convolutional upsampling using skip-connections that combine the spatial information from the corresponding layer from the encoder with the semantic information from the deeper layers. The architecture is shown in Fig. 2.10.

SegNet [Badrinarayanan et al., 2017]: proposed the SegNet architecture, an encoder-decoder structure with fully connected layers that has the same topology as VGG16, substituting the linear fully connected layers into a decoder followed by a pixel-wise classification layer. The innovation brought up by SegNet is how the decoder upsamples the feature maps inputted to it. Unlike FCN, SegNet transfers the information directly instead of convolving them. The decoder takes the pooling in-

dices obtained in the max-pooling step of the analogous encoder to perform non-linear upsampling, sparing the need for learning to upsample (Fig. 2.15). Thus, the sparse upsampled maps are convolved with learnable filters, which produce dense feature maps. This network was designed to be more efficient than FCN regarding memory and computational time and achieved competing results with lesser parameters with FCN benchmarks on semantic segmentation datasets.

U-Net [Ronneberger et al., 2015]: The U-Net architecture is important because of its efficacy in image segmentation tasks, notably in biomedical image analysis. It has been used successfully to segment numerous structures in medical imaging, such as cells, organs, and tumors. It is known for its U-shaped design, consisting of an encoder and a corresponding decoder path, allowing it to learn end-to-end features and process images of any size. This architecture utilizes contracting and expanding pathways to capture hierarchical features and generate pixel-wise segmentation maps. Skip connections connect feature maps from the encoding path to the decoding path, enabling the network to preserve spatial details and use local and global information.

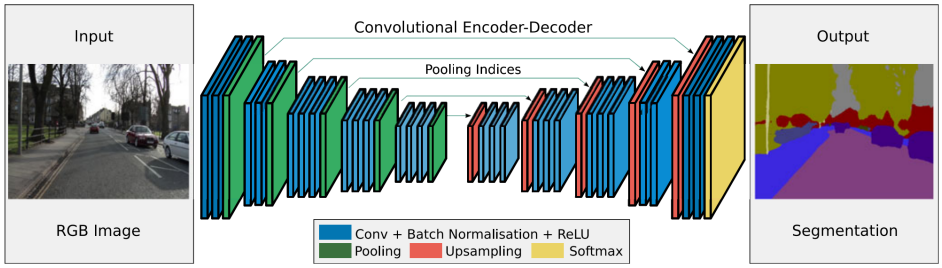


Figure 2.15: Architecture of the SegNet. It consists of a fully convolutional network in which the decoder upsamples its input using the transferred pool indices from its encoder to produce sparse feature maps. Image Source: [Badrinarayanan et al., 2017]

2.3 Related Work

In geosciences, comprehending Earth's inner structure is one of the main goals, either for the pure understanding of the geological systems but also for developing more accurate models used to represent, explain and predict the location and formation of natural resources. To acquire evidence of subsurface rock formations, it makes necessary to rely on indirect approaches such as geophysics methods since direct information sources only rely on investigations of outcrops, samples, and boreholes, which are only sometimes available. Using seismic reflection methods provides acoustic images of the subsurface, which supply essential 3D information on tectonics, stratigraphy facies, and basins structure.

These studies deliver data from large spatial areas, requiring vast processing and time-consuming interpretation. Hence, this field of study could benefit from deep learning models, able to identify faults, salt bodies, and delimit seismic facies. For example, [Wrona et al., 2021] conducted an introduction study aimed at geoscientists with walkthrough concepts and applications about deep learning applied to seismic interpretation. They performed experiments on fault classification and segmentation, salt and horizon segmentation, showing a step-by-step configuration of experiment setup, and with available codes at [1]. They also endorsed that most studies in the area do not make their data, codes, or experimental setup available, turning the experiments unfriendly to replication. To avoid that, we opened our codes[2], as well as detailed the methodology employed, described in section 4.2.

This section brings a set of works on interpreting and detecting geological features in seismic images utilizing neural networks. The subsections are split according to the research goal, starting with a brief introduction to seismic interpretation with deep learning, salt classification, facies segmentation, structure delimitation, and self-supervised applications. Table 2.1 shows a summary of the work related to this research[3], crossing the learning approach versus the target objective.

2.3.1 Salt Classification

Correctly delimiting salt bodies is relevant because they can be reservoir seals and allow an accurate building of velocity models around them [Waldeland et al., 2018]. Usually, these bodies present complex shapes that show a distinct visual aspect compared to surrounding rocks. DL algorithms can be a big time saver, as the delimitation in a 3D view and along hundreds or thousands of seismic sections can be challenging and time-consuming. Therefore, this was the object of study for a few cases as [Waldeland et al., 2018], [Zeng et al., 2019], [Zhou et al., 2020]. In this research, the learning approach is fully supervised, utilizing manually annotated salt bodies as labels.

Aimed at geoscientists with deep learning interest [Waldeland et al., 2018] provided a brief introduction to CNNs with an application example for salt classification on a volume of a seismic survey from the Barents Sea with available codes[4]. They used a CNN to classify the center pixel of a 65x65x65 volume (always centered in one sec-

[1] `github.com/thilowrona/seismic_deep_learning`

[2] `github.com/brunoaugustoam/SSL_Seismic_Images`

[3] list of related studies in continuous development: `github.com/brunoaugustoam/deep_seismic_images_papers`

[4] `https://github.com/waldeland/CNN-for-ASI`

Table 2.1: Related Work comparing available studies for supervised and self-supervised (Context-based pretext tasks and Contrastive Learning) approaches, considering the main task being solved.

Application / Method	**Supervised**	***Geometric Transformation***	***Patch Positioning***	***Clustering***	***Contrastive***
Image Classification / Object Detection	[Krizhevsky et al., 2012], [Simonyan and Zisserman, 2014], [Szegedy et al., 2014], [He et al., 2016]	[Gidaris et al., 2018], [Doersch and Zisserman, 2017], [Su et al., 2020]	[Noroozi and Favaro, 2016], [Doersch et al., 2016], [Su et al., 2020], [Chen et al., 2021]	[Caron et al., 2018] , [Zhuang et al., 2019], [Chen et al., 2021]	[He et al., 2019], [Chen et al., 2020a], [Caron et al., 2021]
Semantic/ Instance Segmentation	[Long et al., 2015a], [Badrinarayanan et al., 2017], [Chen et al., 2017], [Kirillov et al., 2023]	[Gidaris et al., 2018], [Li et al., 2021]	[Kim et al., 2018]		[He et al., 2019], [Li et al., 2021]
Salt Delimitation	[Waldeland et al., 2018], [Zeng et al., 2019], [Di and AlRegib, 2020], [Wrona et al., 2021]				
Facies Segmentation	[Di, 2018], [Chevitarese et al., 2018], [Liu et al., 2020a] , [Alaudah et al., 2019], [Tolstaya and Egorov, 2022]	**Ours**, [Monteiro et al., 2022]	**Ours**, [Monteiro et al., 2022]		[Li et al., 2022], [Su-Mei et al., 2022]
Structure Delimitation	[Wu et al., 2019], [Yang and Sun, 2020], [Guo et al., 2020b], [Wrona et al., 2021]			[Aribido et al., 2021]	

tion) as salt and not salt. Despite visually coherent results and an individual attribute representation, they presented no evaluation metric. Similarly, [Zeng et al., 2019][5] built a model using the U-Net [Ronneberger et al., 2015] structure as the backbone for the task of salt binary segmentation. Three seismic sections were employed, two for training and one for testing. The best-reported validation IoU score was around 75%. Adopting a more antique approach, [Di and AlRegib, 2020] compared the interpretation of a salt body by traditional MLP against CNN, both fully-supervised testing on two expert-labeled sections. They showed the vast superiority provided by CNN for this task.

[Wrona et al., 2021] also provided an introduction to deep learning applied to 3D seismic interpretation. One of their examples was the usage of a 3D U-Net to map salt bodies labeled in 2D from a 3D volume. They used ten sections total to generate 1 million cubes of (16x16x16) and reported precision, recall, and F1 score of 0.91 on the test set (10%) for supervised training. It also provided an image of the predicted 3D volume showing the recognized salt structures, but for that, no evaluation metric was presented.

2.3.2 Facies Segmentation

Interpreting seismic and lithological facies in seismic images is a significant part of understanding depositional environments, basins stratigraphy, etc. Automating this task with deep learning techniques has been pursued by researchers and companies to avoid laborious and time-consuming manual interpretation. Defining classes in a pixel-wise manner in a seismic image is comparable to a classical semantic segmentation task, and a few authors used segmentation networks for segmentation, e.g., [Chevitarese et al., 2018, Alaudah et al., 2019, Souza et al., 2020, Liu et al., 2020a, Su-Mei et al., 2022].

[Di, 2018] used CNNs to segment seismic images using manually interpreted classes of the Netherlands F3 Dataset. They interpreted and segmented twelve seismic patterns/seismofacies using four inline sections. [Zhao, 2019] compared two interpretation methods, a patch-based model for binary classification of each class and an encoder-decoder architecture for semantic segmentation.

[Chevitarese et al., 2018] and [Salles Civitarese et al., 2018] conducted two complementary investigations that focused on the classification and segmentation of lithofacies. Their work proposed a network named Danet, which combines a traditional CNN architecture with elements inspired by the VGG [Simonyan and Zisserman, 2014] and

[5] `https://github.com/mallerao/Seismic_CNN_Saltbody`

ResNet [He et al., 2016] models. They explored some network configurations, creating different versions to evaluate their impact on performance on the F3 Netherlands and Penobscot datasets. Moreover, in their study, [Salles Civitarese et al., 2018] utilized the trained network for supervised classification as a pre-training stage for subsequent segmentation tasks. Specifically, they replaced only the final classification layer with a segmentation layer, adapting the network to perform segmentation tasks.

The labeled dataset introduced by [Alaudah et al., 2019] provides a geological interpretation of the F3 Netherlands Dataset and is the basis for their proposed benchmark. They employed a convolutional neural network (CNN)-based encoder-decoder architecture to build two baseline models. One model was trained using patches of seismic data, while the other utilized sections that preserved contextual information. Interestingly, the model trained on contextual sections performed better than the patch-based model.

Focusing on recoverable hydrocarbon zones (also called leads), [Souza et al., 2020] used a U-Net architecture with binary segmentation mapping and post-processing to enhance detection. They used pre-processing for data cleaning, patch generation (80x80), and data augmentation. Also, they applied image reconstruction, thresholding, and outlier removal as post-processes.

Unlike the previously cited investigations, [Puzyrev and Elders, 2020] attempted to classify the seismic facies in an entirely unsupervised approach, combing a convolutional autoencoder (CAE), PCA, and clustering. The idea is to compress the seismic data into latent space learned by the CAE and then use this data to extract the principal components used for clustering through k-means. Each tile (96x48) was then classified into one of the classes, and by overlapping a weighted sliding window, each pixel was classified. Despite showing a few manually interpreted sections, it presented no evaluation metric.

[Guazzelli et al., 2020] investigated using orthogonal planes as input for convolutional neural networks for facies classifications in 3D seismic cubes. Their study involved subsampling the seismic cube to generate subcubes and focused on classifying the central point corresponding to the intersection of the 2D slice used to construct the subcube. [Liu et al., 2020a] compared the performance of a supervised CNN against a semisupervised GAN on a synthetic dataset and the F3 seismic dataset. The supervised network is trained as a usual supervised CNN for multiclass classification. The main difference in this study is the application of semisupervised GAN for classification. First, they trained a usual generator-discriminator but with eight discriminant classes. Then, for the inference phase, the generator is discarded, and the discriminator is used for classification. Also interesting about this paper is the application of t-SNE

visualization of the extracted features.

Using a reduced number of equally spaced samples from the F3 Dataset, [Wang and Chen, 2021] employed a strategy of padding the images before cropping them and utilizing sliding windows to obtain the segmentation for the entire image. While achieving exceptional results, the authors merged imbalanced classes of the dataset instead of addressing this issue directly, and their method was only tested on a small testing set. [Tolstaya and Egorov, 2022] performed an ablation study using a UNet-based model to examine the effects of incorporating domain-specific attributes and employing a pseudo-labeling approach in the training process. In their research, they utilized a pseudo-labeling method that involved selecting predictions made by the untrained model on unlabeled data, which were then used as labels for training the model, relying on the innate ability of the U-Net architecture to obtain relevant spatial features.

[Abid et al., 2022] conducted a study to explore the potential of utilizing ensembling techniques in deep neural networks. They trained two models based on the DeepLabv3-Plus architecture [Chen et al., 2014] and two models based on the SegNet architecture. Furthermore, they trained these models with and without augmentation, resulting in four models. For the final prediction, they employed an ensemble approach by averaging the probabilities of each class at every pixel across the four models.

[Wang et al., 2023] adopted a two-stage training process, starting with an unsupervised stage where they pre-trained a model to reconstruct the input data. In the supervised step, they froze the encoder weights and trained a new decoder for lithofacies segmentation. To evaluate the effectiveness of their approach, they assessed the performance of models trained in few-shot regimes. They compared it with the version of a supervised approach in the same scenario, achieving competitive results. This methodology to demonstrate efficiency in few-shot scenarios was also employed by [Su-Mei et al., 2022]. They proposed an innovative method that utilized five well-labeled seismic profiles as examples. They assessed the similarity between the data in these ground truth profiles and the 3D volume using cosine similarity at the beginning of the training. They defined subsets of similar sections related to each reference-labeled section using a similarity threshold. the same labels were then assigned To each subgroup, and the entire subset was used for a supervised training stage using the same labels. This procedure showed remarkable results in a big test set of the Parihaka dataset.

[Li et al., 2022] followed a similar approach by employing self-supervised techniques as a pre-training task and fine-tuning their models for the target task. They conducted a study using contrastive learning to combine representations of similar images and push apart different examples. They employed transfer learning techniques to utilize the pre-trained weights from the first stage for the final segmentation task in

scenarios with limited labeled data, achieving promising results.

2.3.3 Structure Delimitation

Structural analysis is of major relevance for understanding basin evolution and Oil and Gas migration and accumulation. Structural information, such as fault, fracture detection, and horizon tracking, are commonly interpreted in seismic surveys. Fault detection, structure-oriented smoothing with edge-preserving, and estimating reflection orientations are correlated interpretation processes. [Wu et al., 2019] developed a multitask simplified U-net for those processes, in which the fault detection output is also used for structure-oriented smoothing, and both are also inputted to normal vector estimation.

Targeting horizon tracking [Yang and Sun, 2020] and also fault tracking [Guo et al., 2020b], applied CNN architectures with softmax classification to identify those geological structures in seismic images. The former used a patched approach on 2D seismic sections to obtain the probability of the presence of a horizon on the central pixel of each tile. The latter used subcubes as input to predict the class (Horizon, Fault, or None) of the central point of the volume. [Wrona et al., 2021] in its introductory study, performed fault classification, segmentation, and horizon classification. They used 2D and 3D U-net architectures for supervised training manually annotated by their team, using class-balanced samples with the interest class centered on each patch or cube.

2.3.4 Self-Supervised Learning for Geological Features Interpretation on Seismic Images

In the classic seismic interpretation workflow, the attributes are extracted by deterministic methods from the data. They are employed in unsupervised algorithms such as K-means and Self-organizing maps that cluster similar features [Aribido et al., 2020]. Despite adopting machine learning methods, these approaches are subject to feature selection and do not compose a deep learning framework. Although deep self-supervised representation learning for natural images is a hot topic and is advancing at high speed, the application of these methods for remote sensing [Li et al., 2021], and seismic image interpretation is not following the same rhythm.

Seismic image processing has also started stepping into self-supervised learning methods, e.g., denoising in [Yu and Ma, 2021] and [Chen et al., 2022]. Despite that, the interpretation of seismic images applying SSL techniques is, until this study, barely

explored, being restricted to the self-supervised annotation of seismic images [Aribido et al., 2020] and self-supervised delineation of geologic structures [Aribido et al., 2021]. First, they used latent space projection with no labels and GAN enhancement to obtain a map of annotations for four typical seismic facies. In the second, they proposed a two-framework application. One performs hierarchical clustering from a seismic volume, and the other uses the obtained clusters as input for an SSL framework to identify horizons, faults, salt domes, and chaotic structures.

As an attempt to deal with the problems related to the lack of plenty available labeled data, we conducted this study focusing on a self-supervised approach and few-shot fine-tuning. The methods employed are described in section 3, and the first experiments and results were published in IEEE Geoscience and Remote Sensing Letters [Monteiro et al., 2022], available at `https://ieeexplore.ieee.org/document/9837909`.

Chapter 3

Proposed Approaches

To identify stratigraphic facies in seismic sections, one must assume that different layers corresponding to each class are differentiable by the aspects recorded in the images. So one model capable of distinguishing the aspects from the image should be capable of discerning the rock groups from seismic images. As discussed in Section 2.3, this has been done already.

With that in mind, this study aims to tackle another problem regarding seismic data; the lack of labels. Thus it seeks to build a self-supervised framework that can extract the features of the seismic images without using its labels and then evaluate the models' ability to generalize to new seismic images after fine-tuning. The selected datasets described below are adequate for the tasks for they have a good signal/noise ratio, well-defined semantic labels, plenty of volume of data, and some available benchmarks for comparison.

In this chapter, we will introduce the frameworks that have been developed as well as the pipeline for comparison. Our workflow entails pre-training the network in several pretext tasks and fine-tuning it using a few labeled parts. The performance of three traditional SSL frameworks for semantic segmentation of seismic images was then compared against training a network from scratch. Fig. 3.1 summarizes the proposed approach, showing the pipeline with the pretext tasks applied to pre-train the backbone as a feature extractor on the same data domain from the targeted segmentation task. The numbers in Fig. 3.1 show the procedure steps to be followed. We have employed three pretext tasks in the context of seismic images.

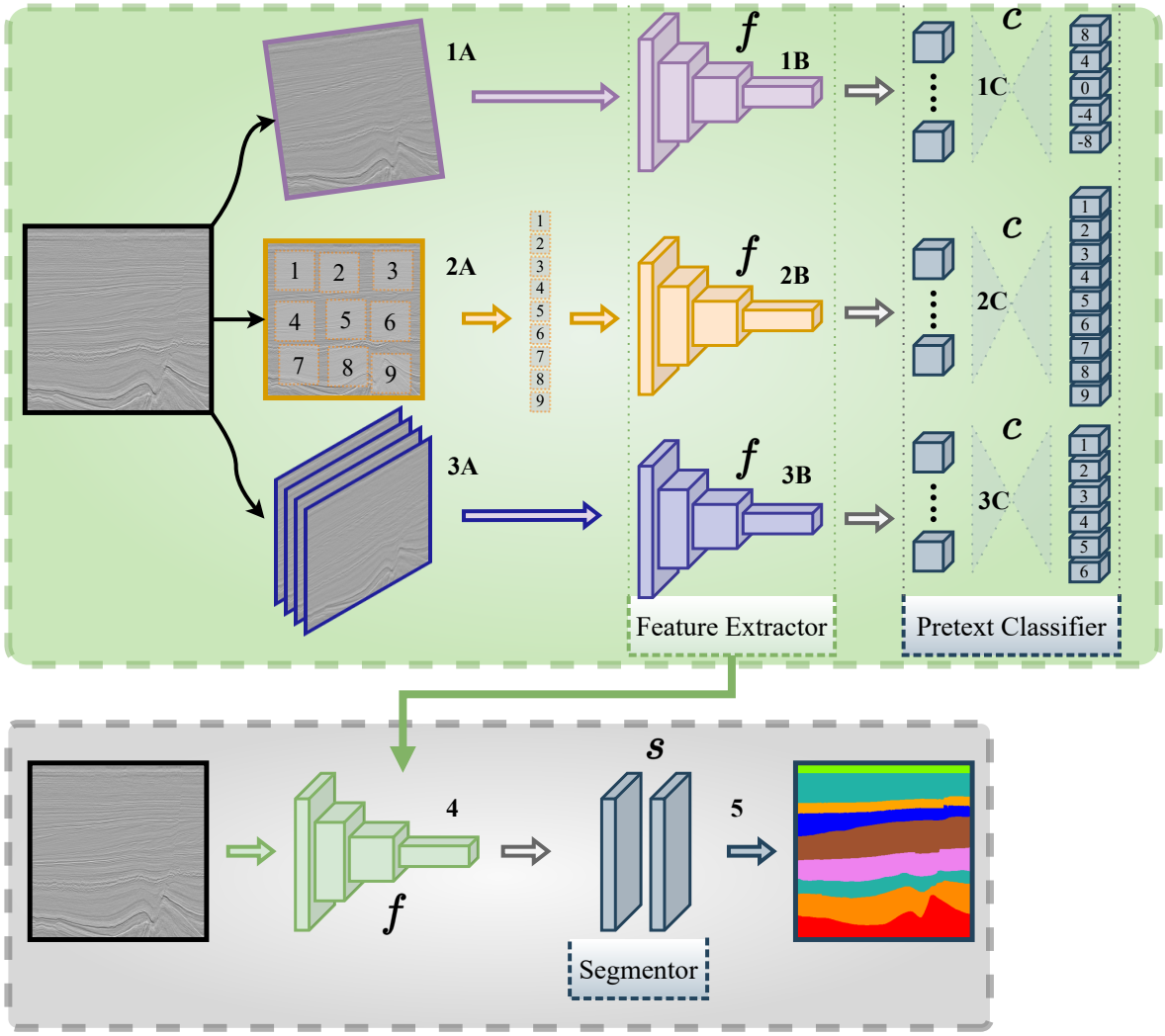


Figure 3.1: Overview of the proposed approach. The upper green portion shows the backbone f and pretext classifier c trained conjointly for each pretext task. The lower gray portion shows the few-shot fine-tuning of f into the segmentation task through a pixel-wise classifier s. Better viewed in color.

3.1 Rotation

The first, based on [Gidaris et al., 2018], consists of training the network to predict the rotation performed in the input image. As discussed in 2.2.1.2, the basic idea behind using image rotations as a set of geometric transformations is that it is virtually impossible for a ConvNet model to effectively perform the rotation recognition task unless it has first learned to recognize object features. As stated in [Gidaris et al., 2018], to recognize the rotation angles, the model has to identify pertinent structures in the image, detect object types and orientations and relate it to the applied geometric transformations. Then during fine-tuning, we leverage these semantic attributes to help

identify the pixel-wise categories.

In the study by [Gidaris et al., 2018], only natural images were used, and 90° rotations were performed to the original image. Unlike the discussed application on natural images, many geological features would lose semantic information if much rotated. Hence, instead of perpendicular operations, we only applied small rotations, preserving the data structure. This attempts to learn smooth distinctions between images while focusing the model's attention on actual geological structures.

In our approach, the original image is rotated randomly into one of the five possible angles (-8,-4, 0, 4, 8 degrees. Fig. 3.2), then the network must identify which of the possible rotations was applied. As shown at the top of the diagram, we rotated the image at a modest angle (1A - Fig. 3.1), then passed it into the convolutional network alongside its rotation label (1B). For this task, a fully-connected classifier head (1C) tries to predict which of the applied rotation in the original image.

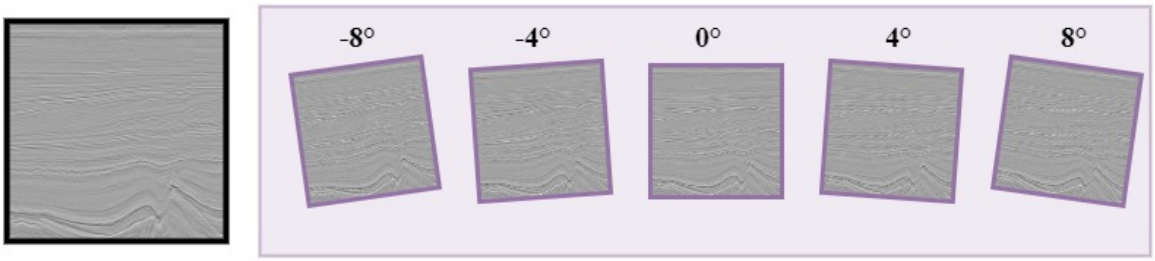


Figure 3.2: Display of the possible rotations applied to each image during the rotation pretext task.

3.2 Jigsaw puzzle

The second, is a jigsaw puzzle, adapted from [Noroozi and Favaro, 2016]. As discussed in 2.2.1.2, this task involves mapping parts of an object into their correct spatial arrangement. For that, the network must learn spatial features that enable the positioning of each task given its data distribution.

As discussed in [Noroozi and Favaro, 2016], it is important to avoid too many possible permutations and trivial permutations. Therefore, we compute five hundred pairs of possible permutations that provide the greatest possible Hamming distance between them. These thousand possible permutations are saved and distributed randomly when training the model.

As demonstrated in Fig. 3.1, we crop the original image into nine tiles, with a small gap between them, and permute them randomly (2A. Fig. 3.3). These crops

are used as input to the convolutional backbone (2B - Fig. 3.1) and then fed to a fully-connected classifier that tries to retrieve the original position of each given tile (2C). Instead of predicting the index related to a single permutation of the nine tiles, as proposed in [Noroozi and Favaro, 2016], we tried to retrieve the individual position of each tile. By doing so, we can observe that the network faces more difficulty differentiating horizontal tiles than vertical ones. It can be explained by the spatial distribution of the geological layers that present horizontal continuity and change their characteristics more often vertically.

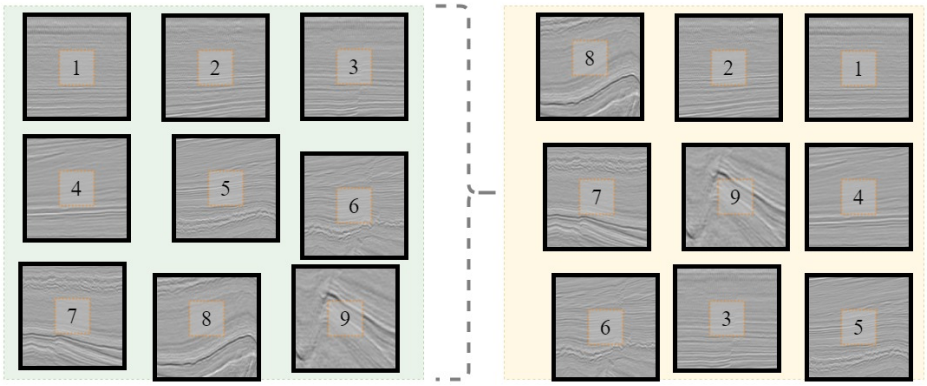


Figure 3.3: Example of the jigsaw puzzling permutation.

3.3 Slice order prediction

Third, drawing inspiration from the application of temporal context structure applied to perform SSL in videos implemented by [Lee et al., 2017], we make use of the sequential enumeration of the sections to develop a "slice order prediction" pretext task. Compared to images, videos offer the benefit of containing an additional time dimension [Lee et al., 2017]. This temporal dimension allows for observing an object's variations over time. By analyzing videos, it is possible to capture the dynamic changes in objects, enabling a more comprehensive understanding of their behavior and characteristics. Thinking about the seismic volume as a continuous mean allows us to think of it as data similar to a video. In this approach, the vertical axis shows the variation of the disposition of geological layers through time, and the lateral changes dispose of the lateral continuity of the layers.

In the seismic volume, each inline and crossline is assigned a numerical identifier relative to its position. As the geological facies have a natural lateral continuity, close sections with close numbers tend to have similar features. This characteristic also

results in significant data redundancy, as the resolution of the seismic survey is designed to capture the continuity of geological layers. By leveraging this inherent structure, we can create a pretext task that involves predicting the position of slices. Hence, the model has to learn the lateral distribution of the layers to know its position within the volume.

First, we provide six key positions equally distributed within the crosslines to be used as pseudo-classes. Then, the model randomly receives a crossline section (3A - Fig. 3.1) and extracts the number of the slice within the volume. Then, passes it through the convolutional backbone (3B) and provides the latent space as input to a classifier (3C) that tries to find the pseudo-class with the smallest distance to the input section within the seismic volume (Fig. 3.4). To predict the spatial position label, the model must learn to recognize the spatial distribution of features within the dataset volume, which can provide a good starting point for fine-tuning.

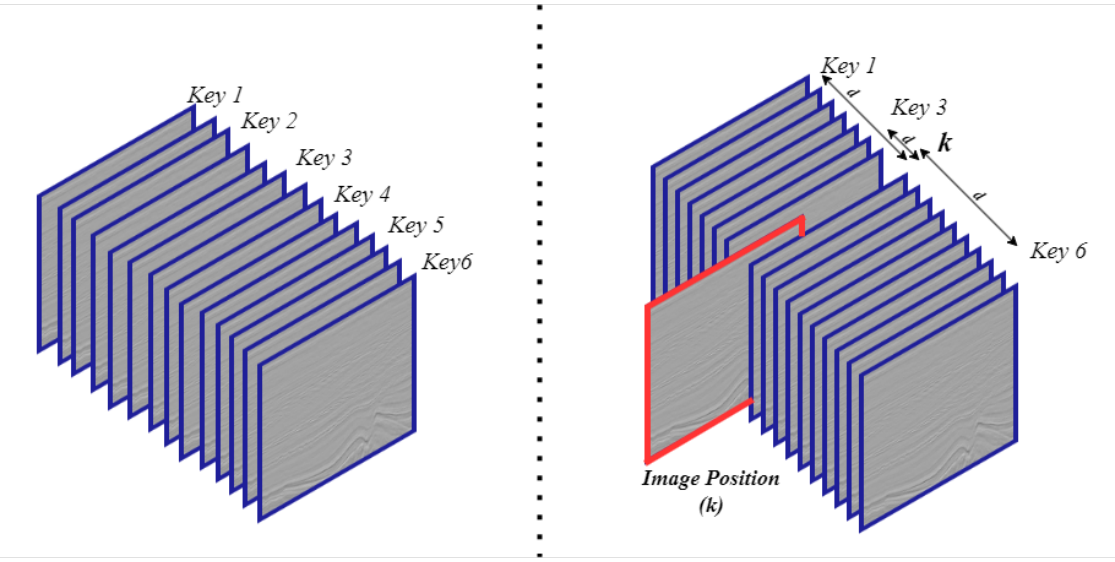


Figure 3.4: Example of the slice order procedure. The key sections are selected equally spaced within the dataset. Each key position is treated as a pseudo-label.

3.4 Multitask

Many studies that use self-supervised learning have adopted multitasking approaches rather than focusing on a single task [Ren and Lee, 2017, Doersch and Zisserman, 2017, Li et al., 2021]. These investigations have found significant improvements by incorporating multitasking, leading to models with performance competitive to the state-of-the-art in their respective domains. The underlying concept behind multitask-

ing is to employ conceptually simple models that encompass a diverse range of tasks. Intuitively, the diversity of tasks helps in learning a diverse set of features [Doersch and Zisserman, 2017].

In this work, we propose a similar approach to Li et al. [Li et al., 2021], where we utilize a single backbone with multiple heads, each specialized in a specific task. In their publication, they employed three highly diverse tasks: inpainting, transform prediction (e.g., geometric transformations like rotation), and contrastive learning. These tasks yielded interesting results. However, due to limitations in time and computational resources, we could not incorporate such tasks into our work. Instead, we focused on combining the tasks described in the previous subsections.

Therefore, we trained two multitasking models, one capable of solving the jigsaw puzzling and the rotation task simultaneously and the other that solves all three pretext tasks together (see Fig. 3.5). In essence, the procedure is the same as the single-task models, but each head of the model solves one task, and the loss is the sum of the losses of each forward pass.

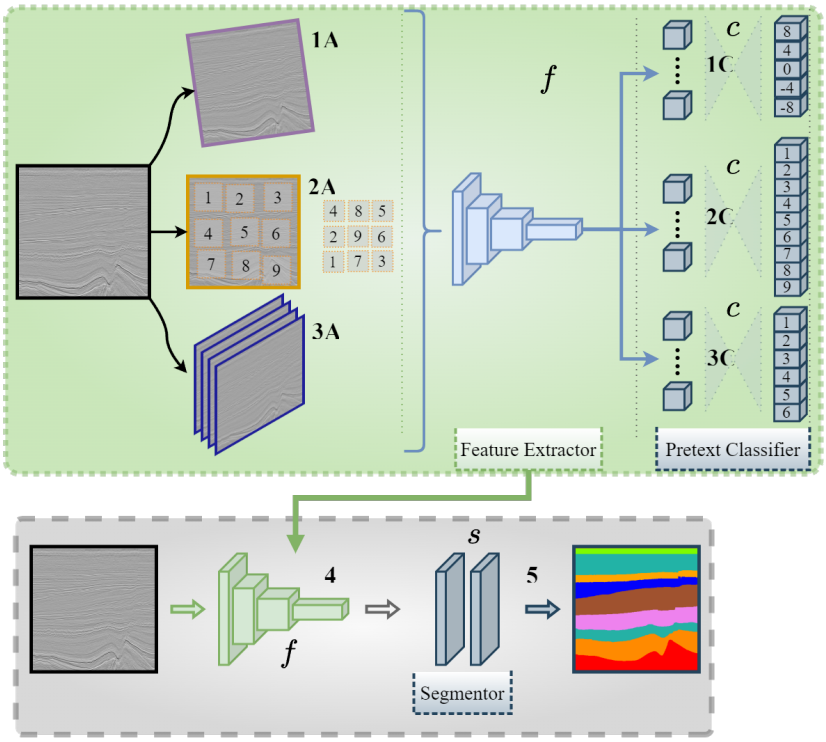


Figure 3.5: Overview of the multitask proposed approach. The upper green portion shows the backbone f, common to all pretext tasks, and pretext classifier c trained conjointly, solving every pretext task at once. The lower gray portion shows the few-shot fine-tuning of f into the segmentation task through a pixel-wise classifier s. Better viewed in color.

3.5 Fine-tuning

Unlike ImageNet [Deng et al., 2009] pre-training, our SSL approach is trained directly with target domain data $\mathbf{X}$ and its automatically computed pseudo-labels $\tilde{\mathbf{Y}}$. SSL can leverage common supervised loss functions such as Cross-Entropy $\mathcal{L}_{CE}$ to minimize the discrepancies between the predicted label $\hat{\mathbf{Y}} = c(f(\mathbf{X}))$ and $\tilde{\mathbf{y}}$. So, for a training batch $\{\mathbf{x}, \mathbf{y}\} \sim \{\mathbf{X}, \mathbf{Y}\}$, the weights $\theta^{(f)}$ for the backbone f and $\theta^{(c)}$ for the pretext classifier c are optimized conjointly using loss $\mathcal{L}_{CE}$:

$$\ell(f, c, \mathbf{x}, \tilde{\mathbf{y}}) = \arg\min_{\theta^{(f,c)}} \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}_{CE}(c(f(x_i)), \tilde{y}_i), \tag{3.1}$$

where $\tilde{\mathbf{y}}$ are the pseudo-labels associated with $\mathbf{x}$.

After training f and c for the pretext tasks, we leverage the extracted features (Fig. 3.1) from f in the fine-tuning stage (4). For that, we replace the pretext classifier c with a randomly initialized segmentor s (5), transforming it into a Fully Convolutional Network able to yield pixel-wise class predictions in the same shape as the input image. The key idea of the SSL pretext task consists of pre-training the backbone f on the same data domain of the final task. This way, f attempts to learn a set of parameters $\theta^{(f)}$ that are domain-dependent and discriminative, used to feed the segmentor s on fine-tuning. To further enhance our procedure, we also resort to an ensemble method, capable of providing better performance by combining predictions of distinct networks [Gawlikowski et al., 2021]. With the trained models, we applied a sum-up, summing the activation outputs a from the fine-tuned models before getting the final prediction (5).

3.6 Ensemble Model

Ensemble model or ensemble learning consists of combining predictions of different trainable models, using rules to obtain a final prediction (Fig. 3.6), and is a method capable of increasing the final performance of many machine learning models [Ganaie et al., 2021, Gawlikowski et al., 2021, Ju et al., 2018]. Also, through evaluating the variety among the single predictions, ensemble models provide a path to representing model uncertainty [Gawlikowski et al., 2021].

Several ensemble strategies have evolved. Two broadly studied and used techniques of ensemble, bagging, and boosting. The former rely on bootstrap aggregation to reduce the variance of a strong learner, and the latter "boost" the capacity of weak learners [Ju et al., 2018].

Regarding deep learning ensemble strategies (deep ensembles), many fusion strategies try to optimize the rules to combine the single outputs and overcome issues that could hurt the performance [Ganaie et al., 2021]. One of the most common strategies consists of unweighted model averaging, where the outcomes of the base learners are averaged to get the final prediction, which makes sense in the context where the performance of each learner is approximate. Another similar approach consists of counting the votes for each class from each base learner and taking the most votes as the final prediction. Comparing these two methods, majority voting is less biased toward a specific single learner, as each vote counts equally. Therefore, it is less sensitive to big activations [Ganaie et al., 2021].

In this study, we have conducted two ensembling strategies. The first one consists of taking the output of each model, extracting the activation before obtaining the prediction, then summing them up and getting the final prediction out of the strongest summed activation. This approach is similar to the unweighted model averaging and can be sensitive if a particular base learner has a stronger activation than the others. The second approach was majority voting, where each final prediction of each model had equal voting weight, and the final prediction for each pixel was defined as the most voted prediction.

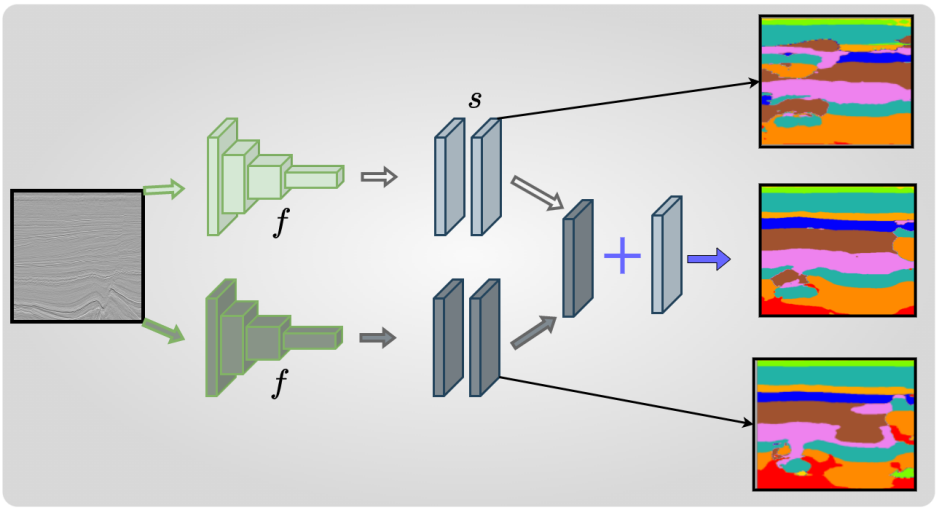


Figure 3.6: Display of an ensemble procedure. Two distinct models produce their output separately, and these outputs are combined to obtain the final prediction.

Chapter 4

Experimental Evaluation

In this chapter, we introduce details on the implementation of our experiments and report the obtained results for both the pretext and downstream tasks, discussing their implications.

4.1 Datasets

The experiments were conducted in two publicly available datasets. One is the Netherlands F3 Interpretation Dataset, interpreted by [Silva et al., 2019][1]. The second is the Parihaka seismic data, released by the "2020 SEG Annual Meeting Machine Learning Interpretation Workshop"[2]. The selected datasets fit the purpose of this study, as they present different geological characteristics for the models to learn, with dense labels available. Both are composed of seismic volumes with more than 1000 slices each, with geological variations within and between the datasets. That way, to perform a full semantic segmentation of the images, the network must learn to distinguish the different textures and features that constitute each class.

4.1.1 Netherlands F3 Interpretation Dataset

The Netherlands F3 dataset is a seismic survey (Fig. 4.1) of approximately 384km2 in the Dutch offshore [Silva et al., 2019]. The F3 block is a famous seismic data volume, for it is open to ready-to-go download at TerraNubis[3] and with labels already separated per inline and crossline section since 2018 by [Baroni et al., 2018][4]. Another aspect

[1] Dataset: `https://zenodo.org/record/1471548#.Yf0Y3-rMKrx`
[2] `https://public.3.basecamp.com/p/JyT276MM7krjYrMoLqLQ6xST`
[3] `https://terranubis.com/datainfo/F3-Demo-2020`
[4] `https://zenodo.org/record/1471548#.Yf0Y3-rMKrx`

that makes this dataset so used is that it is available with the labeled images split into tiles of 64x25 pixels, making it suitable for classification tasks.

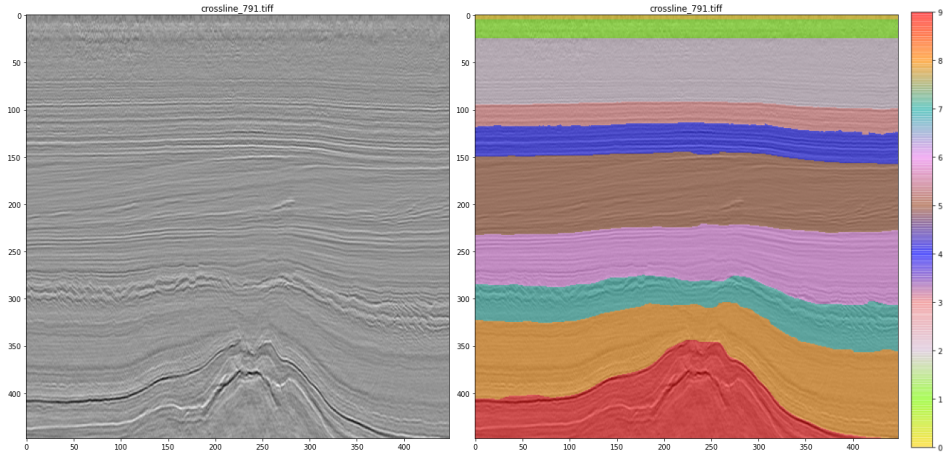


Figure 4.1: Crossline 791 from the F3 Dataset. On the left is the input image, and to the right are the labels over it. Image source: Author composition based on the Dataset released by [Baroni et al., 2018] and interpreted by [Silva et al., 2019]

To make the model evaluation viable, the dataset interpreted by [Silva et al., 2019] was selected, which contains the slices with their respective masks. The pseudo-volume is composed of 651 crossline slices and 951 inline slices, both with 10 classes in total.

According to [Silva et al., 2019], the F3 block is located in the Central Graben Basin, formed by 9 lithostratigraphic groups from the Carboniferous to the Cenozoic, as disposed of in Table 4.1.

4.1.2 Parihaka Seismic Data

This data was collected in 2006 by Pogo Producing Company and provided 2D velocity and stack seismic data from the Parihaka 2D-Opunake tie line, among others. These surveys are offshore Taranaki, North Island, New Zealand. This data was used with the Pogo Producing Company's 3D marine seismic survey in the Taranaki Basin. It consists of non-confidential data released by the New Zealand Petroleum and Minerals (NZPM) [5] and is available at their Online Exploration Database under the survey name "PARIHAKA-TIE-LINES". The interpretation and subsequent labeling of this dataset were released by the "2020 SEG Annual Meeting Machine Learning Interpretation Workshop" [6] and was done by Chevron U.S.A. Inc [Bevc et al., 2020].

[5] `https://www.nzpam.govt.nz/maps-geoscience/exploration-database/`
[6] `https://public.3.basecamp.com/p/JyT276MM7krjYrMoLqLQ6xST`

Class	Group/Period	Description	Thickness(m)
1	North Sea Supergroup / Cenozoic Era	Strong subsidence and halokinesis from the Zechstein Group salt	-
2	Chalk Group / Cretaceous	Chalk and argillites	1800
3	Rijnland Group / Cretaceous	Siliciclastic with erosion and intense halokinesis	1000
4	Schieland, Scruff and Niedersachsen Groups / Jurassic	Lacustrine and clastic with erosion and inversion	1000
5	Altena Group / Jurassic	Marine shales	1600
6	Germanic Trias Group / Triassic	Red shale rocks and siltstone interbedded with sands (lower); anhydrous evaporites (upper)	1800
7	Zechstein Group / Permian	Carbonate and evaporite rocks sometimes with halokinesis	1300
8	Rotliegend Group / Permian	Volcanic rocks and fluvio-lacustrine sediments (lower); fluvial, eolian, and sabkha sediments (upper)	900
9	Carboniferous Group / Carboniferous	Black limestones and clastic rocks	4000

Table 4.1: Summary of the geological facies present on the F3 Dataset.

The 3D seismic data volume and its respective labels (Fig. 4.2) are accessible in the SEG-Y format. These seismic volumes consist of triaxial data, in which each horizontal axis is one geographical direction, and the vertical axis corresponds to the depth/time. The volume can be sliced into 'Inline', 'Crossline', and 'Time stamps', each perpendicular to the other. The training volume and labels consist of 590 inline slices, 782 crossline slices, and 1006 timestamps.

To evaluate the effectiveness of the pretext task on the final target, only the train volume was used since only labels for the training volume were available. It was split into the train, validation, and test sets. The Parihaka dataset contains six lithofacies classes [Bevc et al., 2020]:

1- Basement/Other - Low S/N; Few internal Reflections; May contain volcanics in places.

2- Slope Mudstone A: Slope to Basin Floor Mudstones; High Amplitude Upper and Lower Boundaries; Low Amplitude Continuous/Semi-Continuous Internal Reflectors.

3- Mass Transport Deposit: Mix of Chaotic Facies and Low Amplitude Parallel Reflections.

4- Slope Mudstone B: Slope to Basin Floor Mudstones and Sandstones; High Amplitude Parallel Reflectors; Low Continuity Scour Surfaces.

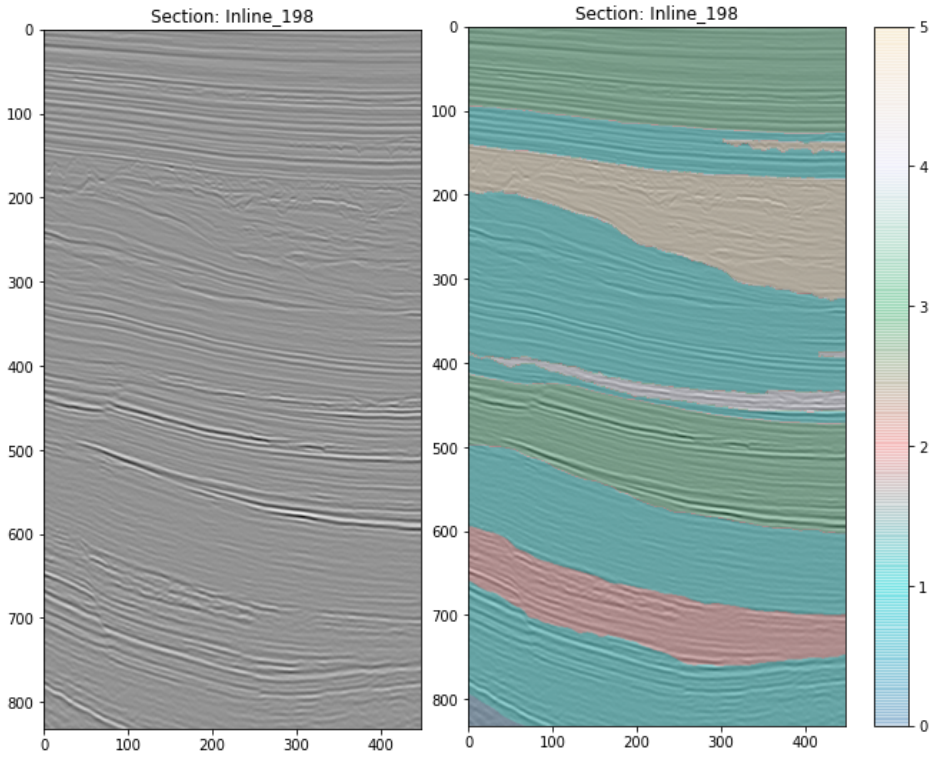


Figure 4.2: Seismic images from the Parihaka Dataset. Both are from the inline 198. To the left is the original image, and to the right are its assigned labels. Image source: Author composition based on the Dataset by [Bevc et al., 2020]

5- Slope Valley: High Amplitude Incised Channels/Valleys; Relatively low relief.

6- Submarine Canyon System: Erosional Base is U-shaped with high local relief. Internal fill is a low amplitude mix of parallel inclined surfaces and chaotic disrupted reflectors. Mostly deformed slope mudstone filled with isolated sinuous sand-filled channels near the basal surface.

4.2 Experimental Setup

Both datasets are composed of crossline and inline slices (seismic sections orthogonal to each other), with ten and six segmentation classes, respectively. After removing sections with problematic data and misclassified labels, we ended up with training/validation/test sections: 916/305/305 in F3 Netherlands and 823/274/274 in Parihaka. The distribution between the three splits was random. To preserve the geological context within the analyzed image, we used the largest possible standard size for each dataset, resulting in 448x448 sized images for the Netherlands F3 dataset and 832x448 for the Parihaka dataset.

As presented in the methods section 3, for the rotation pretext task (R), the original image is randomly rotated into one of the five possible angles (-8,-4, 0, 4, 8 degrees), then the network must identify which of the rotations was applied. To avoid a trivial solution, the rotated image is cropped centrally, so no null border is observable.

For the jigsaw pretext task (J), the original image is cropped into nine regular-sized tiles of size 128x128 in the F3 Netherlands Dataset and 256x128 in the Parihaka Dataset. We leave a small random gap between the tiles to avoid overlap and continuity of the images. The model is requested to determine the original position of each permuted tile.

For the slice order pretext task (F), the model tries to find the class (i.e., the fixed position) with the smallest distance to the input section based on the enumeration of the sections. To do so, we selected six equally distributed positions along the crossline sections and set their index as pseudo-classes. So the forward step consists of giving an input image and extracting its index number (number of the slice within the volume). Then pass the image through the model and try to predict which of the six positions is closest to the one inputted.

All used networks derive from the backbone of the ResNet-50 [He et al., 2016]. For the classification tasks, the output of the backbone is connected to a fully-connected classifier. For the segmentation task, replace the classifier with a segmentation head and utilize bilinear interpolation to retrieve the original input size. Then, two final convolutional layers are used to segment the dataset classes. Dropout was applied after each one of the four-layer sets that compose the 50 layers of the employed backbone network. All training images were augmented using random crop, half-chance horizontal flip, and Gaussian noise addition. Every model was trained utilizing the Cross-Entropy Loss. For fine-tuning, it was modified to consider the class imbalance, i.e., normalized but the pixel count of each class.

We ran a grid search to find the optimal setups for both pretext tasks and fine-tuning. For the pretext tasks, we varied the initial learning rate (0.05, 0.01,0.005,0.001,0.0005), the learning rate scheduler (StepLr and CosineAnnealingLR), and the optimizer (Adam and SGD), adding up to 20 experiments. For fine-tuning, five of the best models of each pretext task were selected. Each was trained, varying the initial learning rate, which was ten times smaller in the pre-trained backbone than in the randomly initialized segmentation layers. For the pre-training, the learning rates dropped to 80% of its value half every five epochs during the 100 epochs of training. In every experiment, the weights were saved if a better accuracy in the validation set was obtained. The final performance is then obtained on the test set.

For fine-tuning, we also conducted a grid search varying the pre-trained models

and learning rates of the backbone and segmentation head. In this case, we separated two sets of experiments. The first was focused on fast convergence, training the models for at most 120 epochs with early stopping. The second was focused on the fine adjustments provided by longer training periods, on which we trained the models for 500 epochs. For both of them, the model weights were saved every time a better mIoU was achieved. As we simulated a few-shot scenario, we only verified the performance on the training set during training. Afterward, the final performance is tested on the unseen test set.

The best setup was then selected for 5-fold cross-validation. For each fold, we randomly selected the same labeled sections for all training algorithms, resulting in paired experiments. For the baseline, we used the same network trained from scratch on the target domain with no pre-training. Codes and demo are available[7].

Following [Sun et al., 2019], we simulated a few-shot scenario considering 1, 5, and 10-shots, plus a 20-shot and 80-shot experiment. This data in the F3 Netherlands Dataset corresponds to approximately 0.1%,0.5%,1.1%, 2.2%, and 8.7% of the training set, respectively. As for the Parihaka Dataset, it corresponds to approximately 0.1%,0.6%,1.2%, 2.4%, and 9.7%, respectively.

For the model ensemble, two main techniques were employed, the sum of activations and the majority voting. The first one can be employed for any combination of two or more models since it consists of summing up the activations of the final layer before obtaining the prediction. The second is better employed if three or more models are available since it consists of obtaining the most voted class for each pixel, and having less than three can result in a draw. In section 4.3, we provide results for each experiment.

The experiments were conducted in the infrastructure of the Pattern Recognition and Earth Observation laboratory inside the Department of Computer Science of the Federal University of Minas Gerais. The models were trained utilizing the Graphics Cards NVIDIA TITAN Xp with 12GB memory. The actual use of memory depends on the input image. The segmentation task for the F3 Netherlands dataset requires ca. 8.5GB of GPU memory. As for the Parihaka dataset, it requires ca. 11.5GB of GPU memory. As the pretext tasks employ sparse prediction, the memory usage is even smaller.

[7] `github.com/brunoaugustoam/SSL_Seismic_Images`.

4.3 Results and Discussion

We have designed and evaluated three pretext tasks based on geometric transformations and contextual information to propel the model to learn the semantic features of seismic images without using manually labeled data. As an attempt to provide a higher variety of information, we also pre-trained the models with multitask attributions combined. We then showed that the designed tasks extracted attributes relevant to a further segmentation task through the comparison of the fine-tuned model against a baseline model trained from scratch in the same conditions of fine-tuning. Our experiments show significant enhancement provided by SSL for semantic segmentation in most few-shot scenarios, with a higher mean Intersection-over-Union (mIoU) and Pixel Accuracy (PA). This improvement is taken even further with the usage of ensembling techniques.

4.3.1 Pretext Tasks

The evaluation of pretext tasks used to pre-train the backbone of our network began by conducting twenty experiments during the grid search phase. Out of these experiments, five were selected to undergo training for semantic segmentation. To determine the best, an investigation was conducted using a 5-shot scenario. Based on the performance observed, the model that exhibited the best results was chosen for further evaluation through a comprehensive five-fold cross-validation process. Table 4.2 exhibits the mean accuracy at each pretext task for the models selected to serve as the backbone for fine-tuning. It can be noticed that for the F3 Dataset, the tasks involving the jigsaw task (i.e., jigsaw, double and multi) presented lower accuracy, demonstrating it to be more challenging to solve. As for the Parihaka, most tasks had a mean accuracy of around 70%, except the jigsaw task itself, which presented lower performance.

In the rotation pretext task, the model with the best performance in the prior stage was also the one with the best performance in the downstream task for both datasets. This suggests that to solve the rotation task adequately, the backbone learned valuable information for the final purpose. Fig. 4.3 shows the confusion matrix results of the selected model for classifying rotation classes in the test set. This image shows that for the F3 Netherlands Dataset, the model could fully predict the rotation applied in the test set with minimum mean error. For the Parihaka Dataset, the errors are concentrated mostly in the adjacent rotation angle, i.e., $\pm 2^{\circ}$.

Fig. 4.4 shows the confusion matrix results on the test set of the jigsaw task for the model selected for fine-tuning. In both datasets, there is a higher error rate between the horizontally neighboring tiles ([1,2,3], [4,5,6], [7,8,9]), which can be explained by

Table 4.2: Summary mean test accuracy for each pretext task considering the models selected as the backbone for the final segmentation task.

Dataset	Task	Mean Accuracy
F3 Netherlands Dataset	Jigsaw	64.37%
	Rotation	99.02%
	Slice	98.40%
	Double (J+R)	72.33%
	Multi (J+R+F)	69.82%
Parihaka Seismic Data	Jigsaw	51.58%
	Rotation	73.82%
	Slice	72.88%
	Double (J+R)	70.36%
	Multi (J+R+F)	67.19%

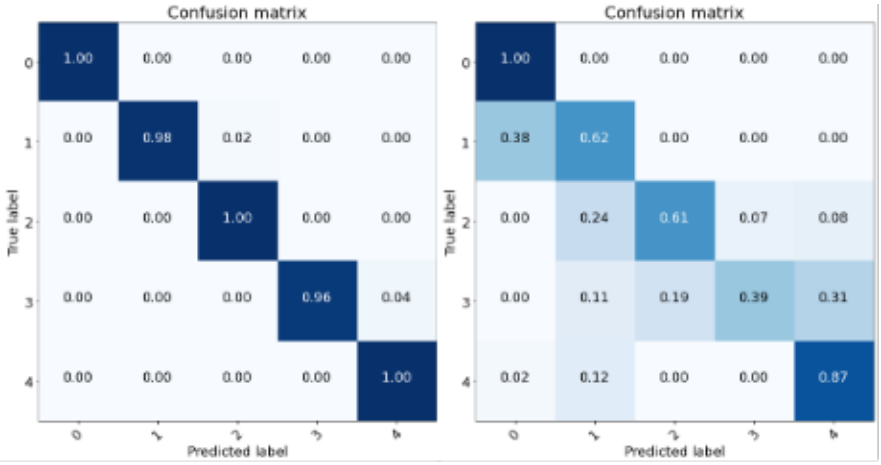


Figure 4.3: Confusion matrix for the rotation pretext task on the test set. To the left, F3 Netherlands Dataset, and to the right, Parihaka Seismic Data. Image source: Author

the nature of the data. These seismic sections show much more geological diversities varying vertically and stronger continuity horizontally, making the task of finding horizontal variations much harder. In the Parihaka Seismic Data, the most suitable model for fine-tuning was also the best for segmentation. In the F3 Netherlands Dataset, the selected model was the third best for solving the jigsaw task but showed better performance during fine-tuning.

Similar to the configuration in the rotation task, the slice order prediction task was solved in F3 Dataset with almost no errors but with more difficulty for the Parihaka Data, as shown in Fig. 4.5. In both datasets, most errors are related to classes adjacent to each other, which in the context of this task, means sections closer to each other within the seismic volume. This makes sense since the data can be very redundant due

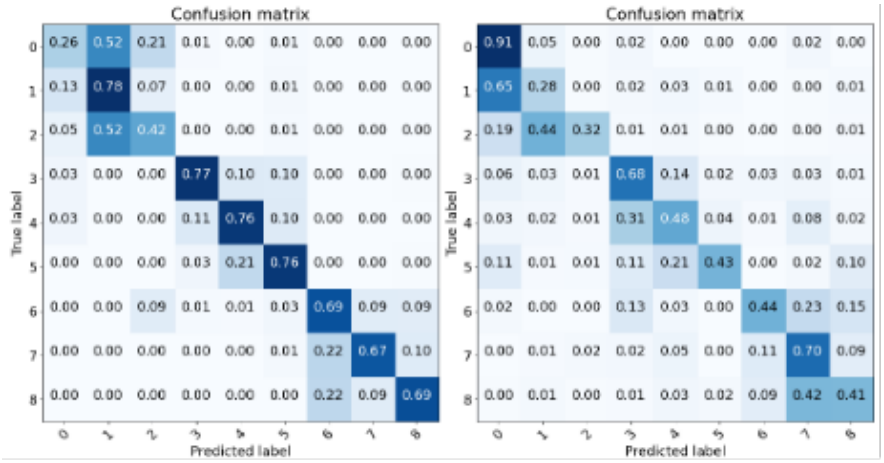


Figure 4.4: Confusion matrix for the test set of the jigsaw puzzling pretext task. To the left, F3 Netherlands Dataset, and to the right, Parihaka Seismic Data.

to the lateral continuity of the geological features. Also, for this task, the model with the best performance in the final task does not match with the best achievement in the pretext task, suggesting that the model that was too good for the previous tasks might be biased and not adapt well to the new purpose.

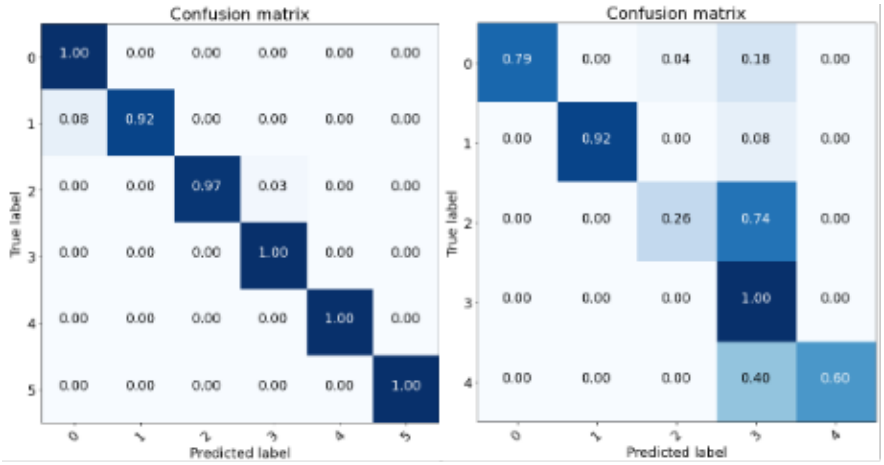


Figure 4.5: Confusion matrix for the test set of the slice order prediction pretext task. To the left, F3 Netherlands Dataset, and to the right, Parihaka Seismic Data.

Looking at the performance for solving the Jigsaw and Rotation at the same time, i.e., Double Task (Table 4.2), we can see that the models are no longer capable of solving both tasks with the same outstanding performance as done individually. Despite that, they solve both tasks together with fair performance. The results for solving the three tasks together, i.e., Multi Task, is similar to solving only two. As shown in Table 4.2, the multitask models can solve all three tasks simultaneously, but

again with a fair performance rather than with the exceptional metrics obtained for each task individually.

4.3.2 Semantic Segmentation

With our experimental protocol, we look forward to demonstrating the improvements brought by SSL methods in a few-shot scenario. With that, we wanted to understand the impact of applying context-based pretext tasks for further seismic image segmentation. As disposed of in section 4.2, we evaluated our method in two datasets considering a common few-shot scenario [Sun et al., 2019] with 1, 5, and 10 shots, plus a 20-shot experiment. In the first round of experiments, the models were trained only for 120 epochs, which was both an attempt to verify the performance achieved with faster training, both also a time limitation due to the deadline to deliver the paper associated with this dissertation. In the second stage, the models were trained longer, for 500 epochs, and with an additional scenario with 80 labeled sections. This simulates a more supervised approach for this task and allows a comparison of performance brought by SSL even in supervised paradigms.

4.3.2.1 First experiments

Table 4.3 shows the mIoU scores and the standard deviation between folds for each experiment in the first round of experiments. Comparing it to the results for the performance in subsection 4.3.1, the best performance in the pretext task does not imply the best performance in the target task. The overall best performance of single models was achieved on pre-trained models relying on the jigsaw pretext task, which had the worst metrics in the first stage of training. This suggests that there must be a balance between feasibility and difficulty in the pretext task. It must be hard enough to avoid trivial solutions and force the model to learn relevant features but not hard to the point that the model cannot solve it, as similarly noticed by [Jing and Tian, 2019, Su et al., 2020].

The employment of ensemble also provided interesting insights, achieving significantly better mIoU and more consistent results, as this technique can improve not only scores but uncertainty and handling out-of-distribution samples [Gawlikowski et al., 2021]. Here, we report results for both sums of activations from the fine-tuned models related to the pre-trained tasks and the majority voting of the combined predictions.

Figures 4.6 and 4.7 show the behavior of mIoU scores varying the few-shot scenario and the paired confidence interval for the F3 Netherlands Dataset and Parihaka Seismic Data, respectively. One should notice that in all but one case, the training

Table 4.3: Summary mIoU results for the five-fold cross-validation and standard deviation between folds for few-shot scenarios trained for 120 epochs. Values in **bold** indicate the best result by dataset. Values in *italic* indicate results significantly better than the baseline.

F3 Netherlands	***1-shot***	***5-shot***	***10-shot***	***20-shot***
Jigsaw (J)	***0.547±0.047***	*0.709±0.021*	*0.762±0.021*	*0.792±0.021*
Rotation (R)	*0.358±0.03*	*0.594±0.012*	*0.696±0.054*	*0.772±0.01*
Slice Order (F)	*0.296±0.016*	*0.497±0.038*	0.562±0.021	0.548±0.046
Ensemble Sum (J + R)	***0.547±0.033***	*0.731±0.013*	***0.777±0.036***	***0.827±0.013***
Ensemble Sum (J + R + F)	*0.516±0.034*	***0.736±0.024***	*0.773±0.025*	*0.803±0.016*
Ensemble Voting (J vs R vs F)	*0.417±0.043*	*0.652±0.013*	*0.732±0.027*	*0.769±0.017*
Baseline	0.167±0.008	0.462±0.029	0.594±0.06	0.643±0.078
Parihaka	***1-shot***	***5-shot***	***10-shot***	***20-shot***
Jigsaw (J)	*0.233±0.013*	*0.403±0.018*	*0.489±0.027*	0.483±0.044
Rotation (R)	*0.182±0.009*	0.342±0.013	*0.446±0.012*	0.516±0.006
Slice Order (F)	*0.225±0.005*	*0.384±0.021*	0.425±0.022	0.482±0.029
Ensemble Sum (J + R)	*0.231±0.014*	*0.452±0.016*	*0.544±0.020*	*0.555±0.025*
Ensemble Sum (J + F)	*0.256±0.007*	*0.459±0.022*	*0.511±0.026*	*0.547±0.021*
Ensemble Sum (J + R + F)	***0.257±0.01***	***0.493±0.019***	***0.547±0.019***	***0.583±0.012***
Ensemble Voting (J vs R vs F)	*0.250±0.008*	*0.451±0.017*	*0.518±0.019*	*0.559±0.013*
Baseline	0.069±0.021	0.324±0.043	0.422±0.02	0.511±0.033

scale positively correlated with the segmentation performance. That is, the mIoU scores improve with increasing the training data, as also observed by [Li et al., 2021]. For most experiments with very limited data, the models trained with SSL methods show a big gap compared to the baseline. This gap is even larger for the jigsaw task, suggesting that it forced the models to learn more relevant features than the rotation or the slice order task, being more adequate for the seismic interpretation context. The mIoU difference diminishes as the available data increases but still holds significant improvement in the F3 Netherlands dataset until the 20-shots experiment. In the Parihaka dataset, it holds up to 10 shots, but the results are not significantly different. As for the predictions obtained through ensembling, the predictions achieved significantly better mIoU in all tested scenarios, demonstrating that fusing deep models can provide better predictions.

Fig. 4.8 shows the general qualitative evaluation, providing a paired comparison of the segmentation for 5- and 10-shot for the baseline method and the ones pre-trained with self-supervision for both datasets. For the F3 Dataset, the results can already capture the major trend of the geological layers with five labeled sections during fine-tuning, especially considering the jigsaw task and the ensemble model. As for the Parihaka data, likely due to more complex and discontinuous structures, the performance with only 5 labeled sections could not capture important data features. This was only possible using at least 10 labeled samples for training. As we can observe,

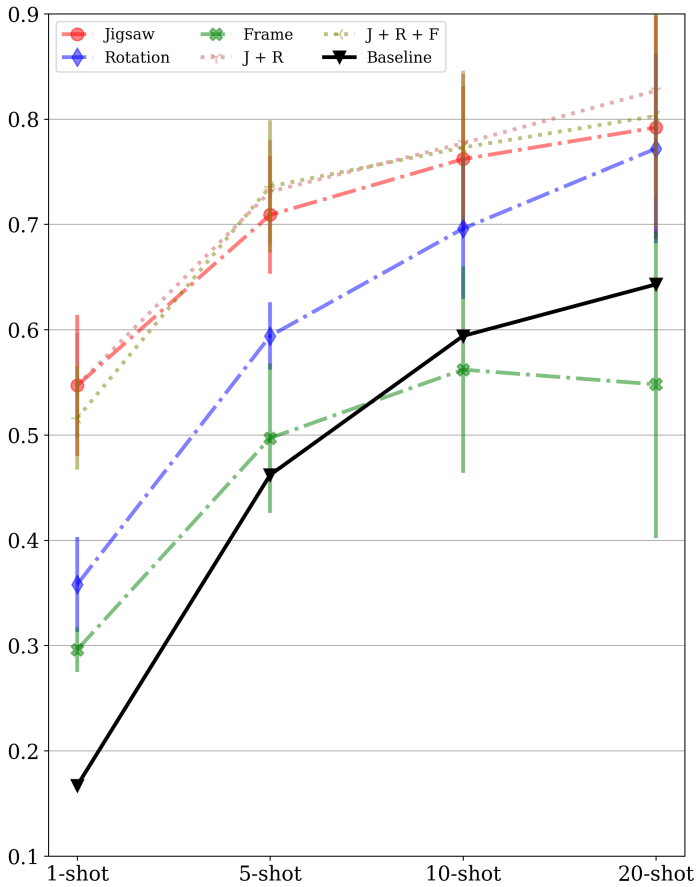


Figure 4.6: F3 Netherlands results - mIoU across the 5 folds for the selected few-shot scenarios with confidence intervals assuming a two-tailed paired t-Student distribution with $p \leq 0.05$. Continuous lines indicate the single pretext task experiments, and dashed lines represent the ensemble models. Experiments were conducted for 120 epochs. Better viewed in color.

there was great improvement brought by pre-training, providing final predictions that can capture data discontinuity and curvature, with even better predictions obtained by ensembling.

4.3.2.2 Second experiments

This round of experiments was similar to the first one. Still, as there was a bigger time availability, the models were trained for 500 epochs to observe if a longer training would provide finer adjustments and better performance. Also, for this stage, we could fine-tune the two models pre-trained in a multitasking manner.

F3 Netherlands Dataset Table 4.4 shows the mIoU metric obtained in the F3 Netherlands dataset, and table 4.5 disposes of the additional pixel accuracy results.

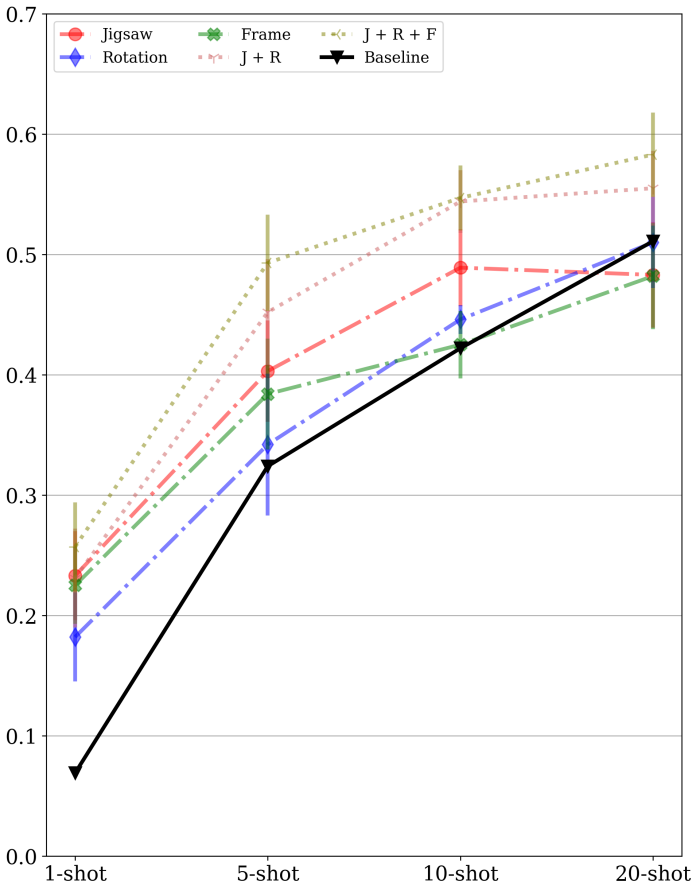


Figure 4.7: Parihaka results - mIoU across the 5 folds for the selected few-shot scenarios with confidence intervals assuming a two-tailed paired t-Student distribution with $p \leq 0.05$. Continuous lines indicate the single pretext task experiments, and dashed lines represent the ensemble models. Experiments were conducted for 120 epochs. Better viewed in color.

As there is a strong imbalance of classes, especially in the Parihaka dataset, the pixel accuracy metric achieves better results in all scenarios. Despite that, it does not reflect an adequate metric for this problem and is provided solely as an additional metric for further comparison against other studies.

As observed in the previous experiments, the best results in single models are associated with the jigsaw puzzling task or the double task, which combines jigsaw and rotation tasks. That reinforces that the jigsaw task was an efficient pre-training technique for segmentation in seismic images, surpassing the baseline in all scenarios and achieving better results than the other pretext tasks. This also reinforces the remarks previously stated, that only achieving better performance on pretext tasks does not reflect better metrics for the target task, as can be seen in Fig. 4.9. This figure compares the performance in the final segmentation task against the previous

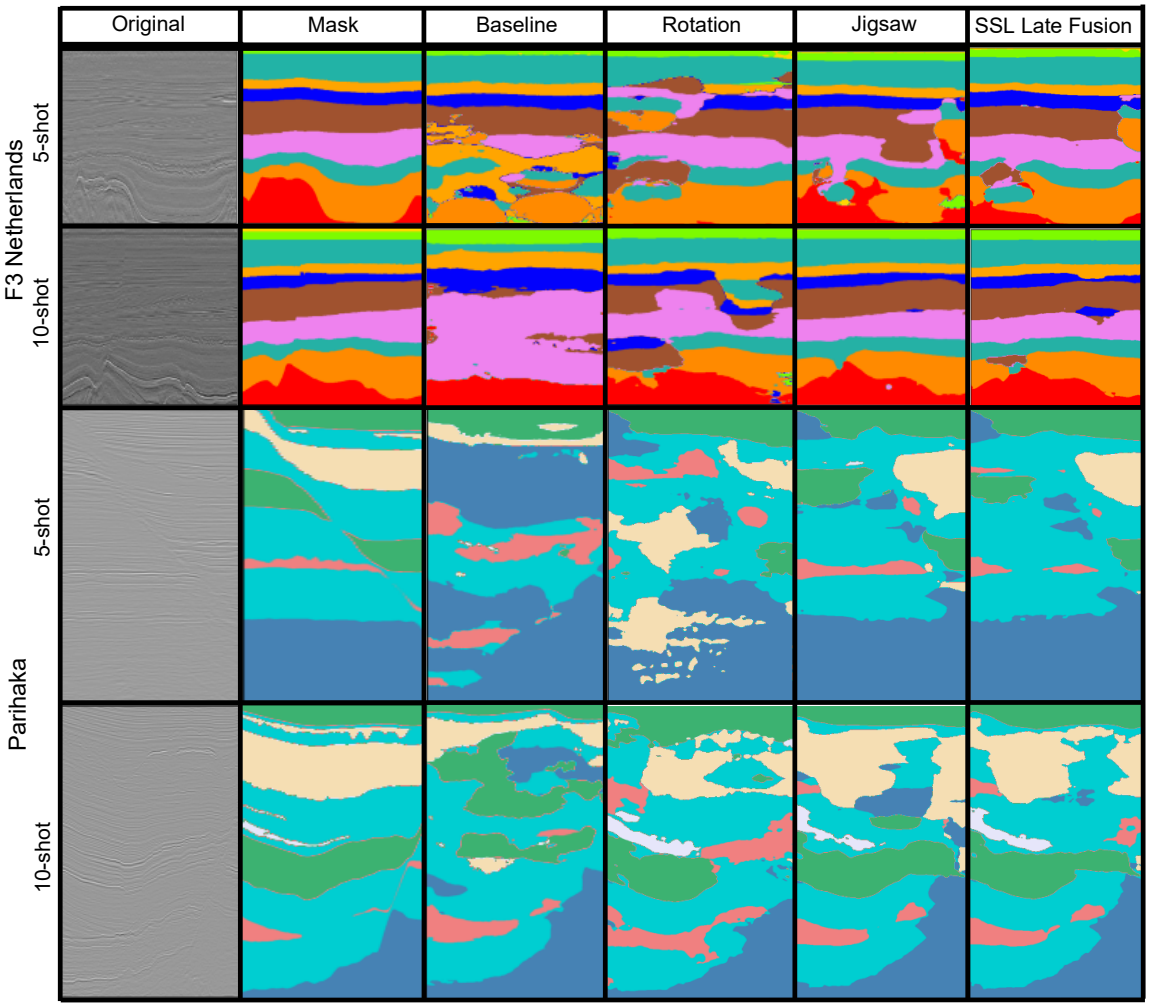


Figure 4.8: Qualitative comparison of segmentation for 5- and 10-shots for the SSL strategies and baselines on F3 and Parihaka trained for 120 epochs. Left to the right, show the original images, labels, results for Baseline and SSL pretexts, and the ensemble (activation sum) on F3 and Parihaka datasets. Better viewed in color.

mean accuracy obtained in the pretext task, considering the distinct number of labeled sections during fine-tuning. As we can see, there is no evidence linking better results in the first to the second stage of training. On the contrary, it suggests that challenging tasks might be more adequate to extract relevant features from the data, as also noticed by [Su et al., 2020].

Looking at table 4.4, we observe an overall improvement when the ensembling techniques are employed. This is especially noticeable when the sum of activations is used instead of the majority voting. Comparing the ensemble results, we see that combining all available models does not imply improvements, and the best results are concentrated on combinations that do not involve using the slice order task, suggesting

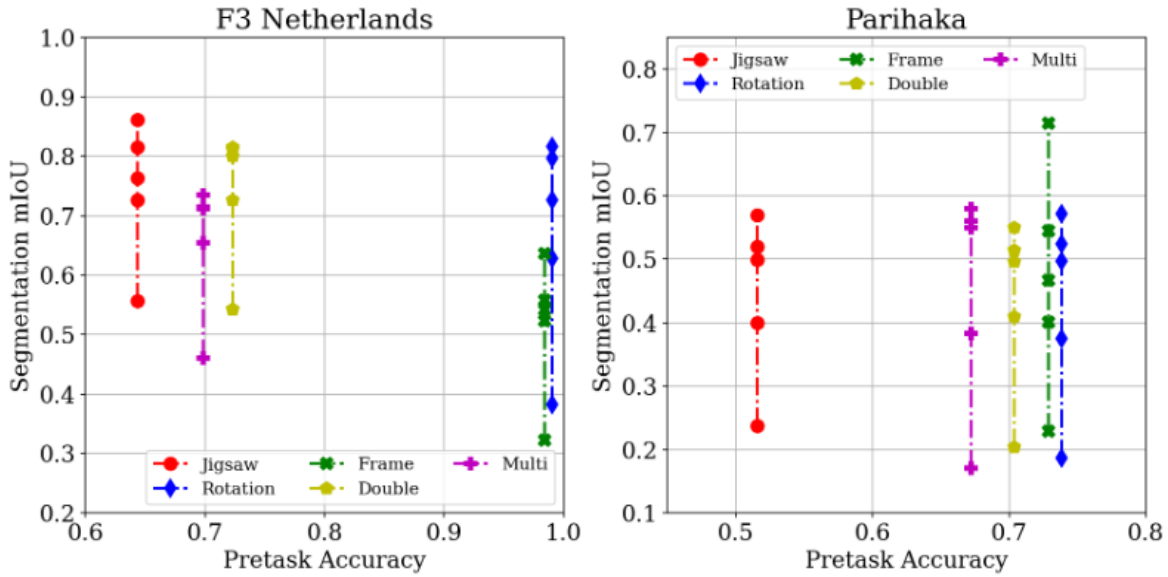


Figure 4.9: Comparison of results obtained in the pretext task against the performance on the final segmentation task for both datasets. The x-axis provides the pretask accuracy, and the y-axis provides the segmentation mIoU. Each marker represents the number of labeled sections used for training during the fine-tuning stage.

that the design of this task was not the most suitable for the target task. Also, one interesting point is that longer training benefits all trained models, including the baseline. The longer training with systematically reducing learning rate allowed the models to refine the weights and achieve better detailing of the predictions.

Table 4.4: Summary mIoU results in the F3 Dataset for the five-fold cross-validation and standard deviation between folds for few-shot scenarios trained for 500 epochs. Values in **bold** indicate the best result by dataset. Values in *italic* indicate results significantly better than the baseline.

F3 - mIoU	***1-shot***	***5-shot***	***10-shot***	***20-shot***	***80-shot***
(J) Jigsaw	*0.557±0.055*	*0.727±0.020*	*0.764±0.040*	*0.815±0.010*	0.861±0.024
(R) Rotation	*0.383±0.020*	*0.628±0.034*	*0.726±0.034*	*0.797±0.013*	0.817±0.007
(F) Slice Order	*0.322±0.019*	*0.523±0.025*	0.558±0.029	0.542±0.046	0.636±0.021
(D) Double [J,R]	*0.543±0.055*	*0.726±0.026*	*0.815±0.055*	*0.812±0.028*	0.800±0.026
(M) Multi [J,R,F]	*0.461±0.027*	*0.655±0.071*	*0.715±0.043*	*0.735±0.044*	0.711±0.016
J + R	*0.565±0.032*	*0.754±0.022*	*0.798±0.034*	*0.851±0.009*	*0.884±0.011*
J + F	*0.513±0.039*	*0.715±0.016*	*0.746±0.041*	*0.772±0.020*	*0.827±0.028*
J + R + F	*0.537±0.038*	*0.752±0.007*	*0.781±0.035*	*0.819±0.01*	*0.858±0.017*
J vs R vs F	*0.441±0.037*	*0.678±0.016*	*0.734±0.04*	*0.786±0.012*	0.827±0.01
J + R + D	*0.640±0.035*	*0.805±0.014*	***0.855±0.027***	***0.892±0.007***	***0.906±0.011***
J vs R vs D	*0.548±0.059*	*0.763±0.017*	*0.820±0.042*	*0.864±0.005*	*0.885±0.012*
J + R + D + M	***0.659±0.037***	***0.816±0.013***	***0.855±0.027***	*0.886±0.008*	***0.892±0.019***
J vs R vs D vs M	*0.593±0.055*	*0.779±0.017*	*0.823±0.039*	*0.865±0.004*	***0.883±0.008***
J + R + F + D + M	*0.653±0.046*	*0.814±0.010*	*0.848±0.029*	*0.877±0.008*	***0.876±0.020***
J vs R vs F vs D vs M	*0.592±0.048*	*0.784±0.017*	*0.827±0.038*	*0.860±0.008*	***0.878±0.012***
Baseline	0.231±0.019	0.521±0.061	0.663±0.067	0.693±0.061	0.797±0.036

Table 4.5: Summary pixel accuracy results in the F3 Dataset for the five-fold cross-validation and standard deviation between folds for few-shot scenarios trained for 500 epochs.

F3 - PA	***1-shot***	***5-shot***	***10-shot***	***20-shot***	***80-shot***
(J) Jigsaw	0.681±0.037	0.833±0.018	0.862±0.028	0.898±0.008	0.925±0.017
(R) Rotation	0.541±0.013	0.755±0.026	0.826±0.023	0.869±0.010	0.878±0.009
(F) Slice Order	0.440±0.023	0.619±0.012	0.668±0.029	0.658±0.046	0.721±0.024
(D) Double [J,R]	0.675±0.053	0.816±0.020	0.876±0.051	0.874±0.022	0.874±0.022
(M) Multi [J,R,F]	0.599±0.020	0.768±0.072	0.814±0.034	0.814±0.043	0.771±0.018
J + R	0.704±0.022	0.855±0.012	0.883±0.020	0.915±0.008	0.935±0.008
J + F	0.640±0.040	0.816±0.018	0.842±0.033	0.864±0.017	0.893±0.026
J + R + F	0.672±0.037	0.85±0.004	0.872±0.023	0.894±0.008	0.914±0.015
J vs R vs F	0.612±0.01	0.799±0.006	0.832±0.007	0.874±0.004	0.892±0.001
J + R + D	0.764±0.028	0.889±0.008	0.919±0.016	0.942±0.005	0.951±0.008
J vs R vs D	0.681±0.006	0.865±0.003	0.900±0.008	0.926±0.002	0.942±0.003
J + R + D + M	0.780±0.038	0.895±0.013	0.918±0.017	0.939±0.006	0.937±0.018
J vs R vs D vs M	0.742±0.009	0.878±0.003	0.905±0.007	0.930±0.001	0.939±0.001
J + R + F + D + M	0.777±0.045	0.895±0.010	0.914±0.021	0.933±0.006	0.922±0.018
J vs R vs F vs D vs M	0.744±0.010	0.878±0.003	0.903±0.007	0.930±0.001	0.926±0.03
Baseline	0.328±0.046	0.642±0.057	0.764±0.064	0.775±0.059	0.856±0.033

Fig. 4.10 displays the plots comparing the number o labeled sections used for fine-tuning and the mIoU obtained for the F3 Netherlands dataset. We can again see a positive correlation between the availability of labeled data and the final performance, although this improvement is not linear. That shows that it can be challenging to provide finer improvements after a particular ceiling, needing a massively increased number of labeled sections to obtain results close to 90% mIoU, for example. At the same time, the 80% mark was achieved with 5 and 10 sections. Focusing on the regime with very scarce data, the 1, 5, and 10-shot experiments can also see a big gap between the baseline and the proposed approaches, suggesting the method's success in few-shot scenarios. Although this gap does diminish as the number of labeled samples is raised, the SSL methods still provide statistically better results in most designs, especially those not involving the slice order task.

Another relevant evaluation of semantic segmentation is the qualitative inspection provided in Fig. 4.11, where we can see some examples of predictions provided by each model considering the number of labeled samples used while training. We can observe that in this dataset, some predictions (Jigsaw, Double, and Ensemble) obtained with one labeled section can capture some of the trends of the layers, especially in the bottom and top layers, but present bad predictions in the middle. Meanwhile, with five and ten labeled samples, we can see more coherent predictions for most predictions, especially employing ensembling.

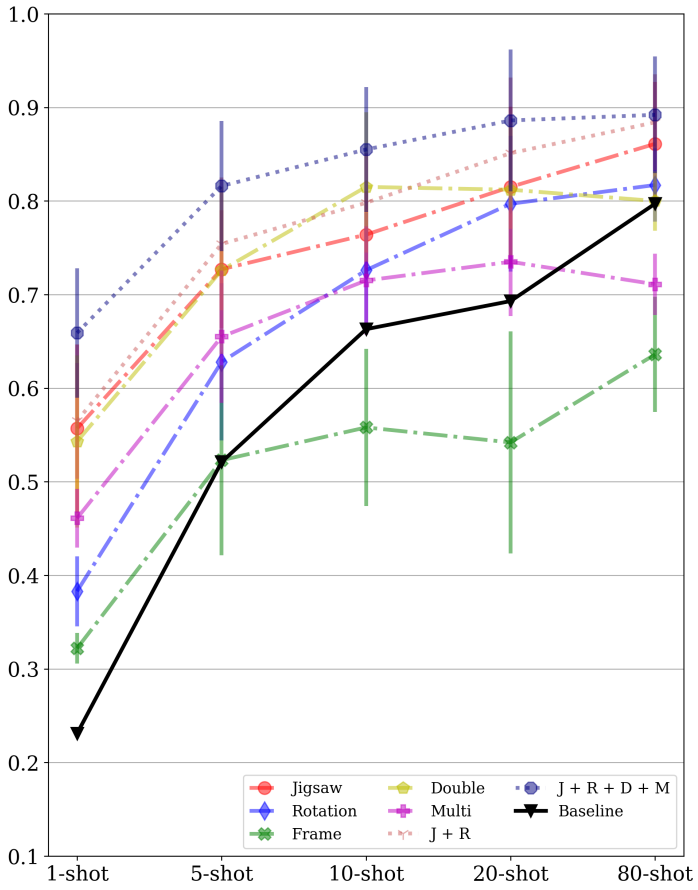


Figure 4.10: F3 Dataset - mIoU across the five folds for the selected few-shot scenarios with confidence intervals assuming a two-tailed paired t-Student distribution with $p \leq 0.05$. Continuous lines represent single-model results, while dashed lines show the ensemble metrics. Better viewed in colour.

Parihaka Dataset Table 4.6 shows the mIoU metric obtained in the Parihaka dataset, and table 4.7 disposes of the additional pixel accuracy results. As observed for the F3 dataset, a strong class imbalance facilitates the accuracy to achieve better results, although it does not reflect an adequate metric for this problem. As we can see in these tables, no consistent model with better performance holds up to all experiments. The jigsaw and slice order tasks performed better with one and five labeled samples, but with ten or twenty sections available, all single models show similar results. As for the slice order task, it achieved the best outcome with eighty labeled samples, surpassing even the ensemble models.

Figure 4.12 presents plots illustrating the relationship between the number of labeled sections used for fine-tuning and the achieved mIoU for the Parihaka dataset. Once again, we observe a positive correlation between the availability of labeled data and the final performance; however, this improvement is not linear. This suggests that

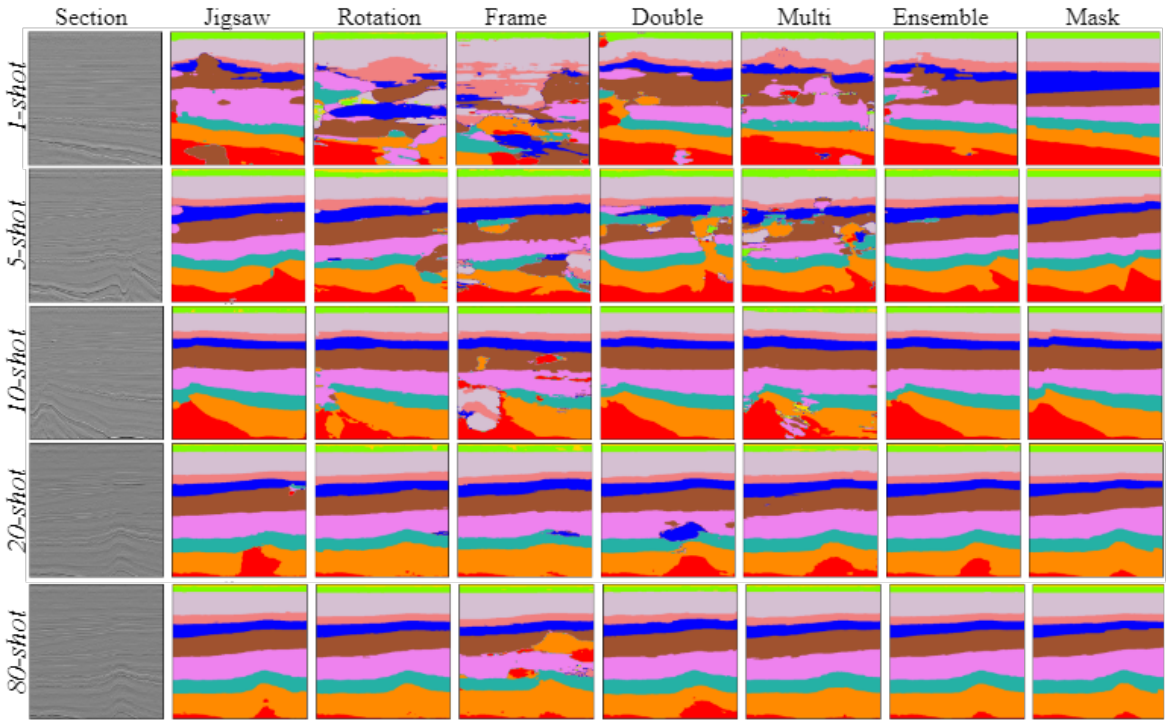


Figure 4.11: Qualitative comparison of segmentation results in the F3 Netherlands dataset trained for 500 epochs. On the y-axis varies the number of sections used for training, and on the x-axis are the models used to obtain the final prediction. The presented ensemble refers to the sum of the activation of all five models. Better viewed in color.

achieving finer improvements becomes increasingly challenging beyond a certain threshold, necessitating a substantial increase in the number of labeled sections to approach, for instance, a 70% mIoU. Unlike observed in the F3 dataset, there is no significant gap between the baseline and the proposed approaches, suggesting the methods did not provide significant advantages. Despite that, as the trained models offer a variety of learned features and can provide different predictions for the same test sections, the combination of the models through the ensemble did bring a significant improvement.

Upon examining Table 4.6 and Fig. 4.12, it becomes apparent that using ensembling techniques leads to an overall improvement. Unlike the results in the F3 dataset, combining predictions through majority voting in the Parihaka dataset did bring benefits as good as the sum of activations methods. Moreover, in this dataset, increasing the number of models involved in the ensemble had a positive impact on the final performance, with the ensemble combining all models yielding the best results in most experiments. This impact also reflects the good results of the pre-training with the slice order task, capable of providing better results on its own and, consequently, better ensembles. Another distinct point is that the more extended training did not

imply better outcomes for this dataset, as the results obtained are not significantly different from the first round of experiments. The most significant gains observed in the second round of experiments in the Parihaka dataset are associated with multiple ensembling models instead of combining only the three original pretext tasks.

Table 4.6: Summary mIoU results in the Parihaka Dataset for the five-fold cross-validation and standard deviation between folds for few-shot scenarios trained for 500 epochs. Values in **bold** indicate the best result by dataset. Values in *italic* indicate results significantly better than the baseline.

Parihaka - mIoU	***1-shot***	***5-shot***	***10-shot***	***20-shot***	***80-shot***
(J) Jigsaw	*0.238±0.010*	0.400±0.043	*0.499±0.019*	0.520±0.035	*0.570±0.045*
(R) Rotation	0.187±0.006	0.375±0.013	*0.497±0.012*	0.524±0.006	*0.572±0.031*
(F) Slice Order	*0.229±0.011*	0.401±0.020	0.467±0.021	0.545±0.031	***0.715±0.015***
(D) Double [J,R]	*0.204±0.014*	*0.410±0.016*	*0.496±0.034*	0.515±0.025	0.550±0.024
(M) Multi [J,R,F]	0.170±0.011	0.384±0.025	*0.560±0.039*	0.550±0.032	*0.579±0.013*
J + R	*0.230±0.010*	*0.452±0.033*	*0.562±0.020*	0.577±0.025	*0.619±0.033*
J + F	***0.259±0.009***	*0.456±0.038*	*0.532±0.018*	0.581±0.027	*0.683±0.018*
J + R + F	*0.253±0.009*	*0.452±0.033*	*0.562±0.020*	0.577±0.025	*0.619±0.033*
J vs R vs F	*0.248±0.012*	*0.465±0.026*	*0.554±0.019*	0.587±0.022	*0.665±0.018*
J + R + D	*0.239±0.005*	*0.487±0.019*	*0.577±0.023*	0.588±0.024	*0.636±0.024*
J vs R vs D	*0.248±0.004*	*0.469±0.016*	*0.565±0.024*	0.581±0.020	*0.619±0.023*
J + R + D + M	*0.225±0.009*	*0.479±0.012*	*0.603±0.022*	*0.608±0.023*	*0.661±0.021*
J vs R vs D vs M	*0.224±0.006*	*0.455±0.012*	*0.581±0.017*	*0.595±0.017*	*0.648±0.018*
J + R + F + D + M	*0.240±0.011*	***0.502±0.012***	***0.607±0.015***	***0.621±0.022***	*0.697±0.020*
J vs R vs F vs D vs M	*0.251±0.007*	***0.502±0.018***	*0.603±0.018*	*0.612±0.020*	*0.690±0.016*
Baseline	0.189±0.034	0.402±0.016	0.481±0.026	0.595±0.021	0.564±0.052

Table 4.7: Summary pixel accuracy results in the Parihaka Dataset for the five-fold cross-validation and standard deviation between folds for few-shot scenarios trained for 500 epochs.

Parihaka - PA	***1-shot***	***5-shot***	***10-shot***	***20-shot***	***80-shot***
(J) Jigsaw	0.552±0.014	0.652±0.042	0.747±0.014	0.757±0.027	0.783±0.030
(R) Rotation	0.437±0.012	0.653±0.010	0.753±0.009	0.763±0.007	0.783±0.028
(F) Slice Order	0.496±0.006	0.661±0.020	0.722±0.022	0.782±0.023	0.891±0.010
(D) Double [J,R]	0.479±0.010	0.656±0.016	0.746±0.030	0.757±0.021	0.788±0.011
(M) Multi [J,R,F]	0.476±0.034	0.666±0.024	0.796±0.027	0.777±0.022	0.773±0.015
J + R	0.569±0.014	0.738±0.021	0.806±0.013	0.811±0.016	0.783±0.030
J + F	0.587±0.009	0.721±0.030	0.783±0.013	0.817±0.016	0.871±0.008
J + R + F	0.599±0.011	0.738±0.021	0.806±0.013	0.811±0.016	0.818±0.020
J vs R vs F	0.585±0.004	0.735±0.002	0.799±0.002	0.815±0.004	0.841±0.009
J + R + D	0.588±0.010	0.769±0.013	0.822±0.012	0.821±0.014	0.840±0.012
J vs R vs D	0.582±0.003	0.752±0.003	0.813±0.002	0.811±0.003	0.826±0.003
J + R + D + M	0.586±0.011	0.767±0.008	0.838±0.008	0.832±0.013	0.855±0.011
J vs R vs D vs M	0.570±0.006	0.749±0.004	0.827±0.003	0.824±0.001	0.847±0.002
Baseline	0.466±0.064	0.657±0.020	0.718±0.024	0.811±0.020	0.756±0.047

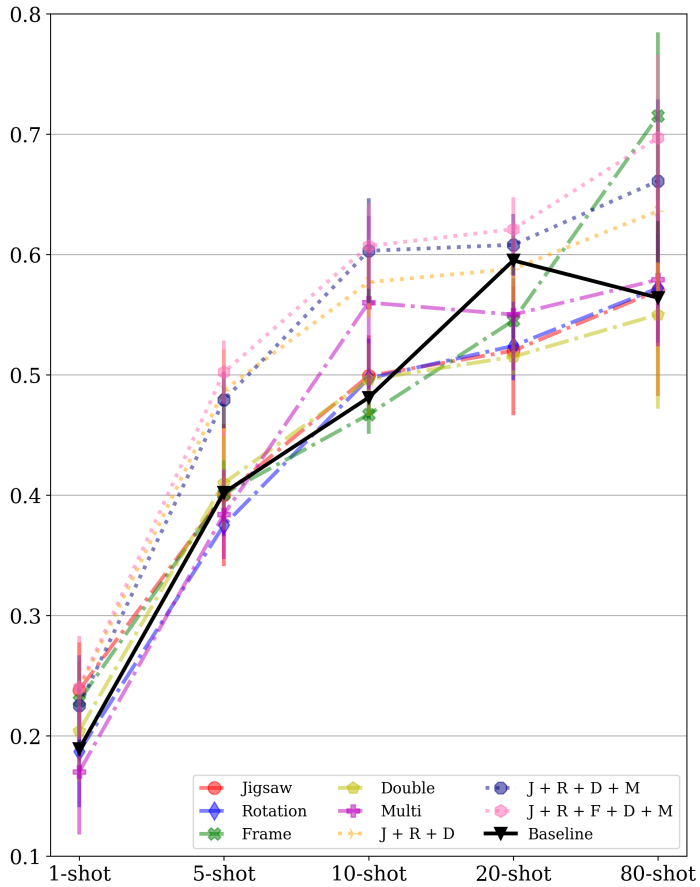


Figure 4.12: mIoU across the 5 folds for the selected few-shot scenarios with confidence intervals assuming a two-tailed paired t-Student distribution with $p \leq 0.05$. Continuous lines represent single-model results, while dashed lines show the ensemble metrics. Better viewed in color.

Fig. 4.13 provides the qualitative inspection of semantic segmentation evaluation in the Parihaka dataset, where we can see examples of predictions each model provides considering the number of labeled samples used while training. We can observe that in this dataset, only the double and multi, especially the ensemble prediction, could capture some of the main features of the lithofacies disposal with only one labeled section. When using five samples, the jigsaw pretext also provides some coherent predictions. Utilizing ten or more labeled sections, we can see that all models can already capture the main trends of the layers. However, only the multitask and the combination of the models could provide predictions with finer details.

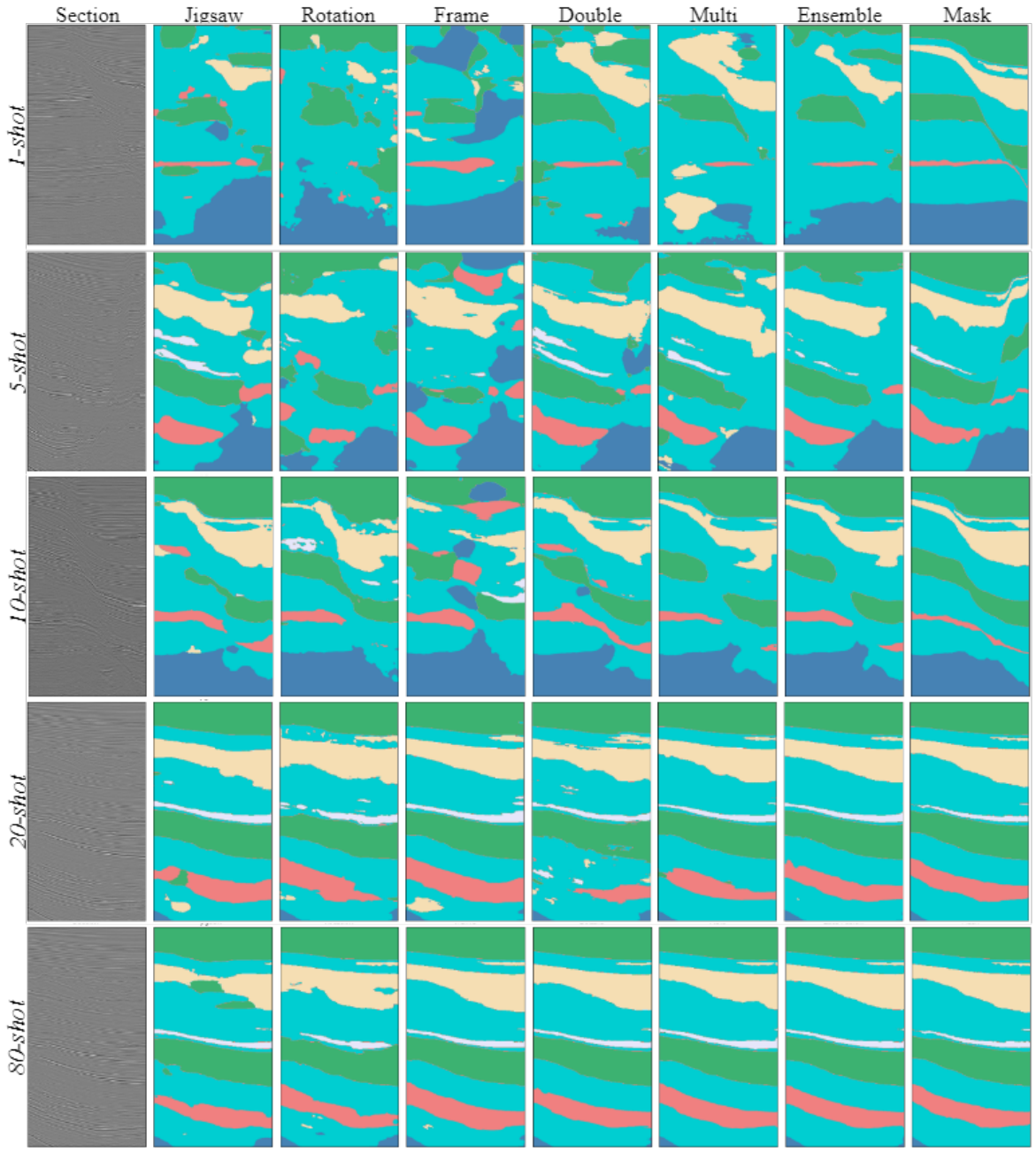


Figure 4.13: Qualitative comparison of segmentation results in the Parihaka dataset trained for 500 epochs. On the y-axis varies the number of sections used for training, and on the x-axis are the models used to obtain the final prediction. The presented ensemble refers to the sum of the activation of all five models. Better viewed in color.

4.3.3 Discussion

The utilization of deep learning algorithms in seismic data analysis has several challenges. One major issue is the complexity of geological features depicted in seismic images. These structures often exhibit multiple possible interpretations, leading to

contrasting interpretations between experts, e.g., interpretations of the F3 dataset by [Silva et al., 2019, Alaudah et al., 2019, MalenovF3,]. Also, unlike other domains of computer vision, there is a scarcity of publicly annotated datasets designed explicitly for seismic interpretation [Alaudah et al., 2019]. Consequently, training and evaluating deep learning models for seismic data become challenging due to the limited availability of labeled data.

At the same time, while some studies may utilize the same dataset and same interpretation, e.g., [Abid et al., 2022, Li et al., 2022, Tolstaya and Egorov, 2022] use the F3 interpretation by [Alaudah et al., 2019], the authors often combine classes from different available datasets, resulting in discrepancies in the evaluation methodology. This lack of consistency makes it difficult to establish a baseline for comparison and interferes with the repeatability of experiments. Moreover, the absence of fixed train/test splits in the public datasets, as in Parihaka Dataset by [Bevc et al., 2020], leads to a wide range of testing sets with varying sizes and assessment techniques. Additionally, different learning paradigms are employed (fully supervised, semi-supervised, and self-supervised), including approaches that utilize only a limited amount of labeled data for training, further complicating fair comparisons of results.

Therefore, instead of only comparing our results to other studies, we adopted a supervised approach baseline as a benchmark. With that, we can observe that our method shows promising results, achieving significantly better results than our baseline and providing more coherent predictions for few-shot scenarios that, for all cases, use less than 2.5% of the available training set and less than 1.5% of all data. In the F3 Dataset, there was an improvement brought by most SSL experiments, both dealing with single models and ensemble predictions. As discussed in subsection 4.3.2, the most successful pretext task in this dataset was the jigsaw puzzling task, most likely as it was the hardest one among the selected, which increases the benefits of self-supervised learning, especially in a small database or grayscale images [Su et al., 2020]. As for the Parihaka dataset, the gains were observed mainly by combining the models via ensemble techniques, which benefit from varying models to create synergies for a final prediction and marginalize false predictions of single models [Gawlikowski et al., 2021]. In this section, we further depict our obtained results and compare them to other studies.

As also noticed by [Su-Mei et al., 2022, Wang et al., 2023], when dealing with few-shot scenarios, there is a relevant impact of the distance between training and testing sections within the seismic volume. This impact is due to the natural character of the geological deposition of sediments, which implicates the horizontal continuity of layers, resulting in more redundant information in closer sections. To investigate

this impact in our study, we plot the mIoU throughout the testing crossline and inline sections in Fig. 4.14 and 4.15.

These figures present performance variation in the test set and the position of the training samples, and also, a grid visualization displaying the training sections' disposal within the volume, viewed from above. One can notice a relevant impact on the performance due to the distance to training samples. However, it can not provide a definitive conclusion, only open the discussion. In the F3 dataset (Fig. 4.14), especially in the five-shot, but also in the ten and twenty-shot experiments, we can see that the random sampling resulted in training sections concentrated in the middle of the volume, thus provoking a notable drop in performance in the borders. The data acquisition and processing itself might also influence this drop, but independently, we can see that when training samples are closer to the edges, the performance does improve. As for the Parihaka dataset (Fig. 4.15), there is an overall improvement in performance closer to training sections, also somewhat concentrated in the center of the volume. This dataset also has a more frequent sudden performance drop, which must be further investigated. However, we suppose to be related to the labels' natural character, which presents many small and discontinuous layers, not so well demarcated in the raw data. To exemplify the impact of the distance on the performance, we also show a qualitative comparison of some predictions obtained far away and close to training sections in Fig. 4.16. In this figure, we can naturally see the improvement brought by increasing the number of examples used. Nevertheless, in many of them, the impact of predicting a sample far away from the training samples is even more robust than the number of training sections. That suggests that independent of the number of training examples, the models could learn the local data distribution but could not generalize to the whole volume, a sign of overfitting in the few-shot scenario. Although this discussion is not yet closed, we believe that another sampling method for training samples would be adequate, either utilizing an equidistant sampling, as in [Su-Mei et al., 2022, Li et al., 2022] or through the expert selection of samples that contain the most representative features of the data.

To address the performance of our models in the F3 Netherlands dataset in comparison to other methods, we compare it to the results presented by [Wang and Chen, 2021] and [Wang et al., 2023], which both applied their methods utilizing the interpretation provided by [Silva et al., 2019]. In their study, [Wang and Chen, 2021] utilized a U-Net model [Ronneberger et al., 2015] and employed a strategy of using a reduced number of equally spaced samples throughout the dataset (32 and 63 samples). They implemented a padding and cropping technique and sliding windows to obtain segmentation for the entire image. While achieving outstanding results, it is worth noting that

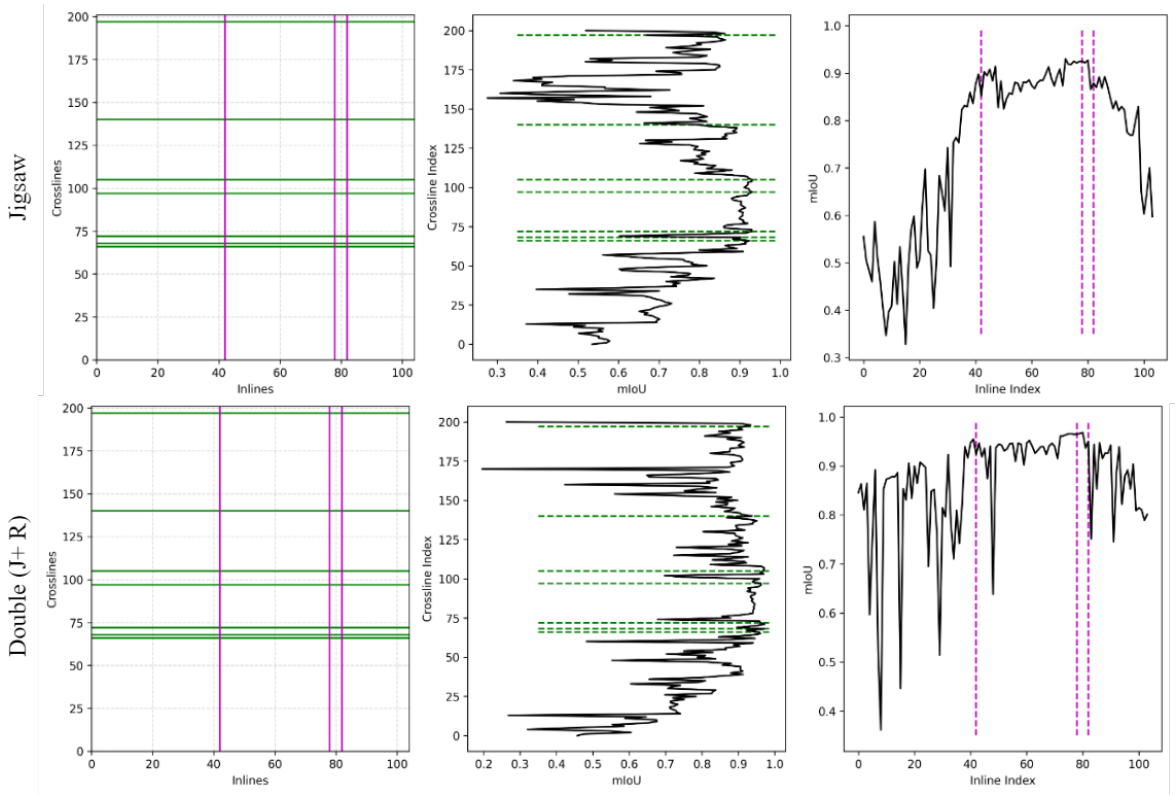


Figure 4.14: F3 Netherlands: Jigsaw and Double (J+R) pre-trained models - Investigation of the impact of the distance between training and test sections on the model's performance. To the left, the top view displays the grid that delineates the seismic volume, and the colored lines show the position of the labeled samples. In the center, the green lines show the position of the labeled crossline sections. To the right, the vertical magenta lines indicate the position of the labeled inline sections.

they chose to merge imbalanced classes of the dataset rather than addressing this issue directly. Furthermore, their method utilized a validation set to select the best models, which would not be possible in a few-shot scenario. Their methods were tested on a relatively small testing set (30 seismic sections), achieving up to 94% mIoU in using either 32 or 63 labeled samples. These results are 4% better than our best results, but our test set had over ten times more testing sections and used no validation set. One point of interest in comparing their study with ours is that they used equally sampled labeled sections throughout the seismic volume instead of using random sampling as we did. As discussed in this section, this sampling provided intriguing investigative insights when addressing Figures 4.14 and 4.15.

In their work, [Wang et al., 2023] implemented a two-stage training process containing an unsupervised stage followed by a supervised stage. During the unsupervised phase, they trained a model to reconstruct the input data. During the supervised stage, they fixed the encoder weights and trained a new decoder designed for lithofacies seg-

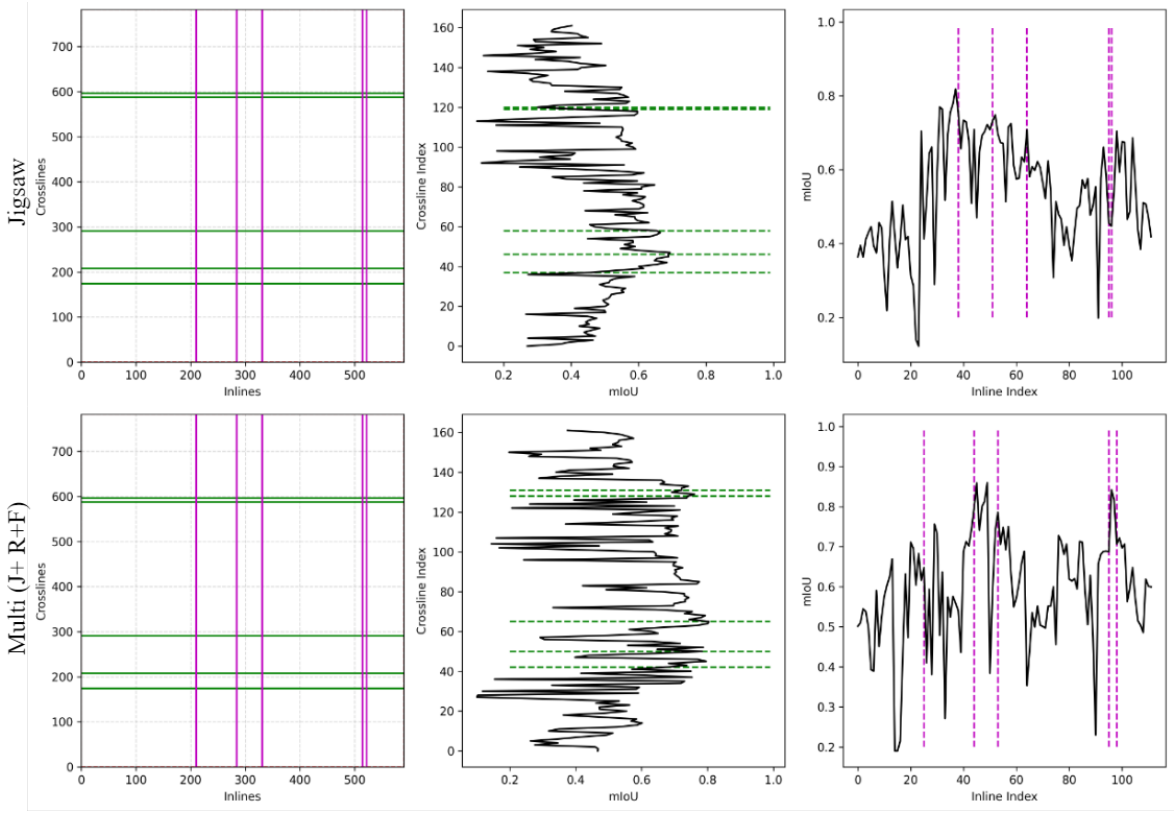


Figure 4.15: Parihaka dataset: Jigsaw and Multi (J+R+F) pre-trained models - Investigation of the impact of the distance between training and test sections on the model's performance. To the left, the top view displays the grid that delineates the seismic volume, and the colored lines show the position of the labeled samples. In the center, the green lines show the position of the labeled crossline sections. To the right, the vertical magenta lines indicate the position of the labeled inline sections.

mentation. To assess the efficacy of their approach, they evaluated the performance of models trained under few-shot regimes, achieving up to 95% mean class accuracy. Although reporting exceptional results, the size and distribution of the test set were not reported, confusing the assessment of the model's capability. As also noticed with our results, their investigation also showed that the benefits of applying semi-supervised methods are more substantial when very few labels, such as two or four labeled sections, are available. Then, these benefits decrease as more labels are at hand, such as eight or sixteen. They have also realized the importance of addressing which sections to label when intending to automate a seismic volume since proximity between sections has an essential correlation with the disposition of the geological layers.

We also compared our results in the Parihaka dataset with other studies [Su-Mei et al., 2022, Li et al., 2022, Tolstaya and Egorov, 2022] that also utilized the interpretation made available by Chevron [Bevc et al., 2020]. With their approach,

[Su-Mei et al., 2022] reached up to 95% pixel accuracy on an extensive test set of the Parihaka dataset, a remarkable result in this dataset. Still, their testing methodology also considered the training and validation sections inside the test set, which obfuscates the actual competence of the model. They have also investigated the impact of the distance between the training and testing sections on the final performance. They developed an approach that considers the continuous lateral variation of geologic strata and utilizes cosine similarity as a metric to quantify the similarity within a seismic data subdomain. They then define one labeled section as the label for all the sections within a similar subset, therefore obtaining more labels without manual annotations.

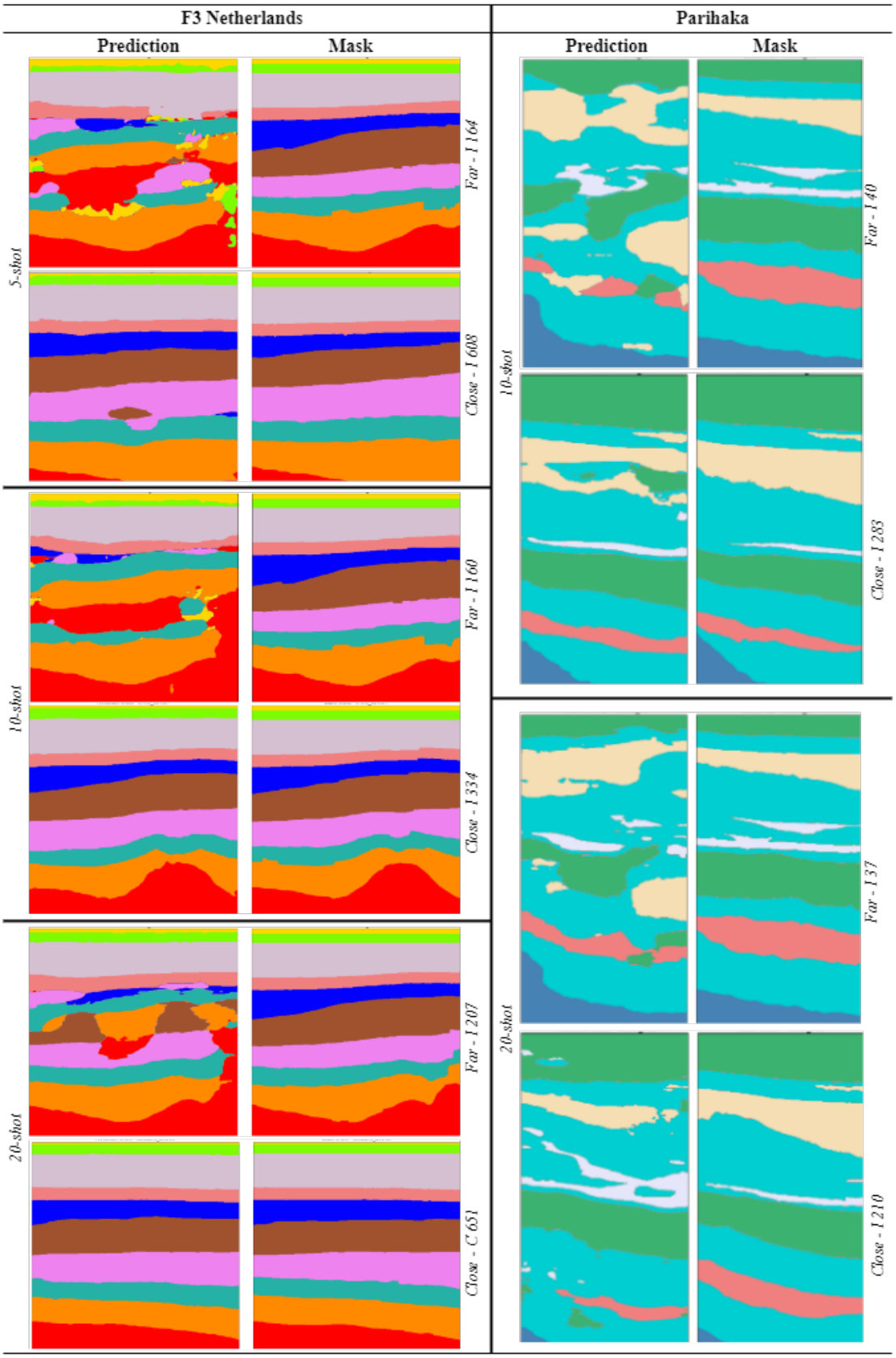


Figure 4.16: Quality inspection of models fine-tuned from the jigsaw pre-training comparing predictions far and close to training samples. For each model trained with five, ten, and twenty labeled sections, we provide an example of prediction far and close to a training sample and the respective mask. To the left is the F3 dataset, and to the right is the Parihaka dataset.

Chapter 5

Conclusion and Future Works

We engage in the currently open deep learning question of working with a few available labels. We applied three classical self-supervised learning (SSL) techniques, namely rotation prediction, jigsaw puzzling, and slice order prediction, to enhance the semantic segmentation of lithostratigraphic facies on two open seismic datasets. The obtained mIoU scores are significantly better than our baseline in many scenarios, reinforcing that pre-trained models can be an alternative when only a few labels are available. This result suggests that choosing appropriate self-supervised pretext tasks can benefit the final model in a few-shot learning setup.

Additionally, we employed deep ensemble methods to create extensive model combinations, resulting in enhanced robustness for seismic segmentation and improved predictions in most experiments. This technique not only allows for the assessment of uncertainty but also boosts overall performance. To the best of our knowledge, this is the first work to fuse SSL methods for enhancing segmentation performance on seismic images.

To further advance the field, it would be recommended to establish a robust benchmarking methodology that can accommodate multiple datasets and their respective interpretations. The benchmark proposed by [Alaudah et al., 2019] stands out as a comprehensive and well-defined framework, despite not being suitable for comparison with our experiment design. Analog approaches should be developed for other available interpreted datasets, such as the Penobscot dataset by [Baroni et al., 2019], Netherlands dataset by [Silva et al., 2019], and Parihaka by Chevron [Bevc et al., 2020].

To summarize, there is substantial room for advancement in deep learning for seismic interpretation. In future research, we would like to include approaches employing contrastive learning as an alternative to manually developed pretext tasks, and pre-training with dozens of unlabeled datasets to obtain important semantic in-

formation without labels. Also, we intend to utilize ways of assessing and minimizing uncertainty. These approaches provide compelling options for addressing the issues of limited labeled data in seismic interpretation.

Bibliography

[Abid et al., 2022] Abid, B., Khan, B. M., and Memon, R. A. (2022). Seismic facies segmentation using ensemble of convolutional neural networks. *Wireless Communications and Mobile Computing*, 2022.

[Alaudah et al., 2019] Alaudah, Y., Michałowicz, P., Alfarraj, M., and AlRegib, G. (2019). A machine-learning benchmark for facies classification. *Interpretation*, 7(3):SE175--SE187.

[Aribido et al., 2021] Aribido, O. J., AlRegib, G., and Alaudah, Y. (2021). Self-supervised delineation of geologic structures using orthogonal latent space projection. *GEOPHYSICS*, 86(6):V497–V508.

[Aribido et al., 2020] Aribido, O. J., AlRegib, G., and Deriche, M. (2020). Self-supervised annotation of seismic images using latent space factorization.

[Badrinarayanan et al., 2017] Badrinarayanan, V., Kendall, A., and Cipolla, R. (2017). Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(12):2481–2495.

[Baroni et al., 2019] Baroni, L., Silva, R. M., Ferreira, R. S., Civitarese, D., Szwarcman, D., and Brazil, E. V. (2019). Penobscot dataset: Fostering machine learning development for seismic interpretation.

[Baroni et al., 2018] [dataset] Baroni, L., Silva, R. M., S. Ferreira, R., Chevitarese, D., Szwarcman, D., and Vital Brazil, E. (2018). Netherlands f3 interpretation dataset.

[Bevc et al., 2020] [dataset] Bevc, D., Halpert, A., Herrmann, F., Power, B., Esmersoy, C., and Fomel, S. (2020). 2020 seg annual meeting machine learning interpretation workshop.

[Bjorlykke, 2015] Bjorlykke, K. (2015). *Introduction to Petroleum Geology*, pages 1--29. Springer Berlin Heidelberg, Berlin, Heidelberg.

[Caron et al., 2018] Caron, M., Bojanowski, P., Joulin, A., and Douze, M. (2018). Deep clustering for unsupervised learning of visual features. In Ferrari, V., Hebert, M., Sminchisescu, C., and Weiss, Y., editors, *ECCV*, pages 139--156, Cham. Springer International Publishing.

[Caron et al., 2021] Caron, M., Misra, I., Mairal, J., Goyal, P., Bojanowski, P., and Joulin, A. (2021). Unsupervised learning of visual features by contrasting cluster assignments.

[Chen et al., 2022] Chen, G., Liu, Y., Zhang, M., and Zhang, H. (2022). Dropout-based robust self-supervised deep learning for seismic data denoising. *IEEE Geoscience and Remote Sensing Letters*, (c):1--1. ISSN 1545-598X.

[Chen et al., 2014] Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., and Yuille, A. (2014). Semantic image segmentation with deep convolutional nets and fully connected crfs. *CoRR. arXiv.*

[Chen et al., 2017] Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., and Yuille, A. L. (2017). Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs.

[Chen et al., 2021] Chen, P., Liu, S., and Jia, J. (2021). Jigsaw clustering for unsupervised visual representation learning.

[Chen et al., 2020a] Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. (2020a). A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597--1607. PMLR.

[Chen et al., 2020b] Chen, T., Kornblith, S., Swersky, K., Norouzi, M., and Hinton, G. (2020b). Big self-supervised models are strong semi-supervised learners.

[Chevitarese et al., 2018] Chevitarese, D., Szwarcman, D., Mozart, R., Silva, D., and Brazil, E. V. (2018). Seismic Facies Segmentation Using Deep Learning*. *AAPG Annual and Exhibition*, 42286(February).

[Deng et al., 2009] Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248--255. Ieee.

[Di, 2018] Di, H. (2018). Developing a seismic pattern interpretation network (SpiNet) for automated seismic interpretation. *arXiv*. ISSN 23318422.

[Di and AlRegib, 2020] Di, H. and AlRegib, G. (2020). A comparison of seismic salt-body interpretation via neural networks at sample and pattern levels. *Geophysical Prospecting*, 68(2):521--535. ISSN 13652478.

[Doersch et al., 2016] Doersch, C., Gupta, A., and Efros, A. A. (2016). Unsupervised visual representation learning by context prediction.

[Doersch and Zisserman, 2017] Doersch, C. and Zisserman, A. (2017). Multi-task self-supervised visual learning.

[Dumay and Fournier, 1988] Dumay, J. and Fournier, F. (1988). Multivariate statistical analyses applied to seismic facies recognition. *Geophysics*, page 1151–1159.

[EnergyGlossary, 2023] EnergyGlossary (2023). Diagram of crosslines, inlines, and a time slice. `https://glossary.slb.com/en/terms/c/crossline`. Accessed: 2023-07-25.

[Ganaie et al., 2021] Ganaie, M. A., Hu, M., et al. (2021). Ensemble deep learning: A review. *arXiv preprint arXiv:2104.02395*.

[Gawlikowski et al., 2021] Gawlikowski, J., Tassi, C. R. N., Ali, M., Lee, J., Humt, M., Feng, J., Kruspe, A., Triebel, R., Jung, P., Roscher, R., et al. (2021). A survey of uncertainty in deep neural networks. *arXiv preprint arXiv:2107.03342*.

[Gidaris et al., 2018] Gidaris, S., Singh, P., and Komodakis, N. (2018). Unsupervised representation learning by predicting image rotations.

[Girshick, 2015] Girshick, R. (2015). Fast r-cnn.

[Goodfellow et al., 2016] Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning*. MIT Press. `http://www.deeplearningbook.org`.

[Guazzelli et al., 2020] Guazzelli, A. B., Roisenberg, M., and Rodrigues, B. B. (2020). Efficient 3d semantic segmentation of seismic images using orthogonal planes 2d convolutional neural networks. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1--8. IEEE.

[Guo et al., 2020a] Guo, Y., Peng, S., Du, W., and Li, D. (2020a). Fault and horizon automatic interpretation by CNN: a case study of coalfield. *Journal of Geophysics and Engineering*, pages 1--10. ISSN 1742-2132.

[Guo et al., 2020b] Guo, Y., Peng, S., Du, W., and Li, D. (2020b). Fault and horizon automatic interpretation by CNN: a case study of coalfield. *Journal of Geophysics and Engineering*, 17(6):1016–1025. ISSN 1742-2132.

[He et al., 2019] He, K., Fan, H., Wu, Y., Xie, S., and Girshick, R. (2019). Momentum Contrast for Unsupervised Visual Representation Learning.

[He et al., 2016] He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep Residual Learning for Image Recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2016-Decem, pages 770--778. IEEE. ISSN 10636919.

[Herron, 2011] Herron, D. A. (2011). *First steps in seismic interpretation*. Society of Exploration Geophysicists.

[Huang et al., 2018] Huang, G., Liu, Z., van der Maaten, L., and Weinberger, K. Q. (2018). Densely connected convolutional networks.

[Jaiswal et al., 2021] Jaiswal, A., Babu, A. R., Zadeh, M. Z., Banerjee, D., and Makedon, F. (2021). A survey on contrastive self-supervised learning.

[Jing and Tian, 2019] Jing, L. and Tian, Y. (2019). Self-supervised visual feature learning with deep neural networks: A survey. *arXiv*, pages 1--24. ISSN 23318422.

[Jing et al., 2018] Jing, L., Yang, X., Liu, J., and Tian, Y. (2018). Self-supervised spatiotemporal feature learning via video rotation prediction. *arXiv preprint arXiv:1811.11387*.

[Ju et al., 2018] Ju, C., Bibaut, A., and van der Laan, M. (2018). The relative performance of ensemble methods with deep convolutional neural networks for image classification. *Journal of Applied Statistics*, 45(15):2800--2818.

[Karimpouli and Tahmasebi, 2019] Karimpouli, S. and Tahmasebi, P. (2019). Computers and Geosciences Segmentation of digital rock images using deep convolutional autoencoder. *Computers and Geosciences*, 126(February):142--150. ISSN 0098-3004.

[Kim et al., 2018] Kim, D., Cho, D., Yoo, D., and Kweon, I. S. (2018). Learning image representations by completing damaged jigsaw puzzles.

[Kirillov et al., 2023] Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A. C., Lo, W.-Y., Dollár, P., and Girshick, R. (2023). Segment anything.

[Krizhevsky et al., 2012] Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In Pereira, F., Burges, C. J. C., Bottou, L., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc.

[Lateef and Ruichek, 2019] Lateef, F. and Ruichek, Y. (2019). Survey on semantic segmentation using deep learning techniques. *Neurocomputing*, 338:321--348. ISSN 18728286.

[LeCun et al., 2015] LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *Nature*, 521(7553):436--444. ISSN 0028-0836.

[Ledig et al., 2017] Ledig, C., Theis, L., Huszar, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A., Tejani, A., Totz, J., Wang, Z., and Shi, W. (2017). Photo-realistic single image super-resolution using a generative adversarial network.

[Lee et al., 2017] Lee, H.-Y., Huang, J.-B., Singh, M., and Yang, M.-H. (2017). Unsupervised representation learning by sorting sequences. In *CVPR*, pages 667--676.

[Li et al., 2022] Li, K., Liu, W., Dou, Y., Xu, Z., Duan, H., and Jing, R. (2022). Contrastive learning approach for semi-supervised seismic facies identification using high-confidence representations. *arXiv preprint arXiv:2210.04776*.

[Li et al., 2021] Li, W., Chen, H., and Shi, Z. (2021). Semantic segmentation of remote sensing images with self-supervised multitask representation learning. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 14:6438–6450.

[Liu et al., 2020a] Liu, M., Jervis, M., Li, W., and Nivlet, P. (2020a). Seismic facies classification using supervised convolutional neural networks and semisupervised generative adversarial networks. *GEOPHYSICS*, 85(4):O47–O58.

[Liu et al., 2020b] Liu, M., Jervis, M., Li, W., and Nivlet, P. (2020b). Seismic facies classification using supervised convolutional neural networks and semisupervised generative adversarial networks. *Geophysics*, 85(4):O47--O58. ISSN 0016-8033.

[Liu et al., 2020c] Liu, X., Zhang, F., Hou, Z., Mian, L., Wang, Z., Zhang, J., and Tang, J. (2020c). Self-supervised Learning: Generative or Contrastive. *arXiv*, pages 1--23. ISSN 23318422.

[Long et al., 2015a] Long, J., Shelhamer, E., and Darrell, T. (2015a). Fully convolutional networks for semantic segmentation.

[Long et al., 2015b] Long, J., Shelhamer, E., and Darrell, T. (2015b). Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[MalenovF3,] MalenovF3. Data from the malenov machine learning seismic interpretation project from conocophillips norge. `https://dataunderground.org/dataset/malenov-f3`. Accessed: 2023-05-19.

[Misra et al., 2016] Misra, I., Zitnick, C. L., and Hebert, M. (2016). Shuffle and learn: unsupervised learning using temporal order verification. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*, pages 527--544. Springer.

[Mondol, 2010] Mondol, N. H. (2010). *Seismic Exploration*, pages 375--402. Springer Berlin Heidelberg, Berlin, Heidelberg.

[Monteiro et al., 2022] Monteiro, B. A. A., Oliveira, H., and Santos, J. A. d. (2022). Self-supervised learning for seismic image segmentation from few-labeled samples. *IEEE Geoscience and Remote Sensing Letters*, 19:1–5.

[Nanda, 2021] Nanda, N. C. (2021). *Seismic data interpretation and evaluation for hydrocarbon exploration and production*. Springer.

[Noroozi and Favaro, 2016] Noroozi, M. and Favaro, P. (2016). Unsupervised learning of visual representations by solving jigsaw puzzles. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 9910 LNCS:69--84. ISSN 16113349.

[Pathak et al., 2016] Pathak, D., Krahenbuhl, P., Donahue, J., Darrell, T., and Efros, A. A. (2016). Context encoders: Feature learning by inpainting.

[Puzyrev and Elders, 2020] Puzyrev, V. and Elders, C. (2020). Unsupervised seismic facies classification using deep convolutional autoencoder.

[Ren and Lee, 2017] Ren, Z. and Lee, Y. J. (2017). Cross-domain self-supervised multi-task feature learning using synthetic imagery.

[Ronneberger et al., 2015] Ronneberger, O., Fischer, P., and Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation.

[Salles Civitarese et al., 2018] Salles Civitarese, D., Szwarcman, D., Silva, R., and Vital Brazil, E. (2018). Seismic facies segmentation using deep learning.

[Silva et al., 2019] Silva, R. M., Baroni, L., Ferreira, R. S., Civitarese, D., Szwarcman, D., and Brazil, E. V. (2019). Netherlands dataset: A new public dataset for machine learning in seismic interpretation.

[Simonyan and Zisserman, 2014] Simonyan, K. and Zisserman, A. (2014). Very Deep Convolutional Networks for Large-Scale Image Recognition.

[Souza et al., 2020] Souza, J. F. L., Santana, G. L., Batista, L. V., Oliveira, G. P., Roemers-Oliveira, E., and Santos, M. D. (2020). Cnn prediction enhancement by post-processing for hydrocarbon detection in seismic images. *IEEE Access*, 8:120447–120455.

[Su et al., 2020] Su, J. C., Maji, S., and Hariharan, B. (2020). When Does Self-supervision Improve Few-Shot Learning? *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 12352 LNCS:645--666. ISSN 16113349.

[Su-Mei et al., 2022] Su-Mei, H., Zhao-Hui, S., Meng-Ke, Z., San-Yi, Y., and Shang-Xu, W. (2022). Incremental semi-supervised learning for intelligent seismic facies identification. *Applied Geophysics*, 19(1):41--52.

[Sun et al., 2019] Sun, Q., Liu, Y., Chua, T.-S., and Schiele, B. (2019). Meta-transfer learning for few-shot learning. In *CVPR*, pages 403--412.

[Szegedy et al., 2014] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. (2014). Going deeper with convolutions.

[Tian et al., 2020] Tian, Y., Krishnan, D., and Isola, P. (2020). Contrastive multiview coding. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XI 16*, pages 776--794. Springer.

[Tolstaya and Egorov, 2022] Tolstaya, E. and Egorov, A. (2022). Deep learning for automated seismic facies classification. *Interpretation*, 10(2):SC31--SC40.

[Waldeland et al., 2018] Waldeland, A. U., Jensen, A. C., Gelius, L.-j., and Solberg, A. H. S. (2018). Convolutional neural networks for automated seismic interpretation. *The Leading Edge*, 37(7):529--537. ISSN 1070-485X.

[Wang and Chen, 2021] Wang, D. and Chen, G. (2021). Seismic stratum segmentation using an encoder–decoder convolutional neural network. *Mathematical Geosciences*, 53(6):1355--1374.

[Wang et al., 2023] Wang, L., Joncour, F., Barrallon, P.-E., Harribey, T., Castanie, L., Yousfi, S., and Guillion, S. (2023). Semi-supervised semantic segmentation for seismic interpretation. *Geophysics*, 88(3):1--57.

[Wang et al., 2021] Wang, X., Zhang, R., Shen, C., Kong, T., and Li, L. (2021). Dense contrastive learning for self-supervised visual pre-training. In *CVPR*, pages 3024--3033.

[Weng and Kim, 2021] Weng, L. and Kim, J. W. (2021). Self-supervised learning self-prediction and contrastive learning. https://neurips.cc/media/neurips-2021/Slides/21895.pdf.

[Wrona et al., 2021] Wrona, T., Pan, I., Bell, R. E., Gawthorpe, R. L., Fossen, H., and Brune, S. (2021). 3D seismic interpretation with deep learning: A brief introduction. *The Leading Edge*, 40(7):524–532. ISSN 1070-485X.

[Wu et al., 2019] Wu, X., Liang, L., Shi, Y., Geng, Z., and Fomel, S. (2019). Multi-task learning for local seismic image processing: fault detection, structure-oriented smoothing with edge-preserving, and seismic normal estimation by using a single convolutional neural network. *Geophysical Journal International*, 219(3):2097–2109. ISSN 0956-540X.

[Yang and Sun, 2020] Yang, L. and Sun, S. Z. (2020). Seismic horizon tracking using a deep convolutional neural network. *Journal of Petroleum Science and Engineering*, 187:106709. ISSN 0920-4105.

[Yu and Ma, 2021] Yu, S. and Ma, J. (2021). Deep Learning for Geophysics: Current and Future Trends. *Reviews of Geophysics*, 59(3):1--36. ISSN 19449208.

[Zeng et al., 2019] Zeng, Y., Jiang, K., and Chen, J. (2019). Automatic seismic salt interpretation with deep convolutional neural networks. In *Proceedings of the 2019 3rd International Conference on Information System and Data Mining*, ICISDM 2019, page 16–20, New York, NY, USA. Association for Computing Machinery.

[Zhang et al., 2016] Zhang, R., Isola, P., and Efros, A. A. (2016). Colorful image colorization. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 9907 LNCS:649--666. ISSN 16113349.

[Zhao, 2019] Zhao, T. (2019). Seismic facies classification using different deep convolutional neural networks. *2018 SEG International Exposition and Annual Meeting, SEG 2018*, pages 2046--2050.

[Zhou et al., 2020] Zhou, H., Xu, S., Ionescu, G., Laomana, M., and Weber, N. (2020). Salt interpretation with u-saltnet. *SEG Technical Program Expanded Abstracts 2020*.

[Zhuang et al., 2019] Zhuang, C., Zhai, A. L., and Yamins, D. (2019). Local aggregation for unsupervised learning of visual embeddings. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6002--6012.