Pontifícia Universidade Católica
do Rio de Janeiro

Maykol Jiampiers Campos Trinidad

Seismic Fault Segmentation using Unsupervised Domain Adaptation

Dissertação de Mestrado

Dissertation presented to the Programa de Pós–graduação em Engenharia Elétrica, do Departamento de Engenharia Elétrica da PUC-Rio in partial fulfillment of the requirements for the degree of Mestre em Engenharia Elétrica.

Advisor : Prof. Marco Aurélio Cavalcanti Pacheco
Co-advisor: Dr. Smith Washington Arauco Canchumuni

Rio de Janeiro
August 2023

Pontifícia Universidade Católica
do Rio de Janeiro

Maykol Jiampiers Campos Trinidad

Seismic Fault Segmentation using Unsupervised Domain Adaptation

Dissertation presented to the Programa de Pós–graduação em Engenharia Elétrica da PUC-Rio in partial fulfillment of the requirements for the degree of Mestre em Engenharia Elétrica. Approved by the Examination Committee:

Prof. Marco Aurélio Cavalcanti Pacheco
Advisor
Departamento de Engenharia Elétrica – PUC-Rio

Dr. Smith Washington Arauco Canchumuni
Co-advisor
Departamento de Engenharia Elétrica – PUC-Rio

Prof. Raul Queiroz Feitosa
Departamento de Engenharia Elétrica – PUC-Rio

Prof. Manoela Rabello Kohler
Departamento de Engenharia Elétrica – PUC-Rio

Dr. Alexandre Cruz Sanchetta
PUC-Rio

Dr. Marcos Sebastião dos Santos
PUC-Rio

Dr. Pedro Juan Soto Vega
IFREMER

Rio de Janeiro, August 10th, 2023

Maykol Jiampiers Campos Trinidad

Graduated in Mechatronics Engineering from the National University of Engineering (UNI, Peru) in 2018. His work is focused on Deep Learning in Computer Vision and Data Science in the industry.

Bibliographic data

Campos Trinidad, Maykol Jiampiers

Seismic Fault Segmentation using Unsupervised Domain Adaptation / Maykol Jiampiers Campos Trinidad; advisor: Marco Aurélio Cavalcanti Pacheco; co-advisor: Smith Washington Arauco Canchumuni. – 2023.

73 f: il. color. ; 30 cm

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

Inclui bibliografia

1. Engenharia Elétrica – Teses. 2. Imagem sísmica. 3. Aprendizado profundo. 4. Segmentação de falhas. 5. Adaptação de domínio não supervisionada. 6. Identificação de reservatórios de petróleo. I. Cavalcanti Pacheco, Marco Aurélio. II. Arauco Canchumuni, Smith Washington. III. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Engenharia Elétrica. IV. Título.

CDD: 621.3

To my parents, for their endless love
and unwavering support.

Acknowledgments

To my advisor, Professor Marco Aurelio, for his trust in me throughout the realization of this work, and I also want to thank my co-advisor, Smith Arauco, for his constant encouragement and collaboration.

To CNPq and PUC-Rio, for the aids granted, without which this work does not could have been accomplished.

I would like to thank at the Applied Computational Intelligence Laboratory (ICA) and Cenpes/Petrobras, partners for 23 years in the research and development of artificial intelligence projects for oil and gas sector.

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001

Abstract

Campos Trinidad, Maykol Jiampiers; Cavalcanti Pacheco, Marco Aurélio (Advisor); Arauco Canchumuni, Smith Washington (Co-Advisor). **Seismic Fault Segmentation using Unsupervised Domain Adaptation**. Rio de Janeiro, 2023. 73p. Dissertação de Mestrado – Departamento de Engenharia Elétrica, Pontifícia Universidade Católica do Rio de Janeiro.

Seismic fault segmentation presents a challenging and time-consuming task in geophysics, particularly in the exploration and extraction of oil and natural gas. Deep Learning (DL) methods have shown significant potential to address these challenges and offer advantages compared to traditional methods. However, DL-based approaches typically require a substantial amount of labeled data, which contradicts the current scenario of limited availability of labeled seismic data. To address this limitation, researchers have explored synthetic data generation as a potential solution for unlabeled real data. This approach involves training a model on labeled synthetic data and subsequently applying it directly to the real dataset. However, synthetic data generation encounters the domain shift problem due to the complexity of real-world geological situations, resulting in differences in distribution between synthetic and real datasets. To mitigate the domain shift issue in seismic fault detection, we propose a novel approach utilizing Unsupervised Domain Adaptation (UDA) techniques. Our proposal involves using a synthetic dataset for model training and adapting it to two publicly available real datasets found in the literature. The chosen UDA techniques include Maximum Mean Discrepancy (MMD), Domain-Adversarial Neural Networks (DANN), and Fourier Domain Adaptation (FDA). MMD and DANN aim to align features in a common n-dimensional feature space by minimizing discrepancy and increasing domain confusion through adversarial training, respectively. On the other hand, FDA transfers the style from real to synthetic samples using Fast Fourier Transform. For the experiments, we utilized a smaller version of UNet and its variant Atrous UNet, which incorporates Dilated Convolutional layers in its bottleneck. Furthermore, DexiNed (Dense Extreme Inception Network), a state-of-the-art model for edge detection, was employed to provide a more comprehensive analysis. Additionally, we studied the application of fine-tuning on labeled datasets to investigate its impact on performance, as many studies have employed it to reduce domain shift.

The final results demonstrated significant improvements in fault detection performance by applying UDA techniques, with up to a 13% increase in evaluation metrics such as Intersection over Union and F1-score on average. Moreover, the proposed approach achieved more consistent detections of seismic faults with fewer false positives, indicating its potential for real-world applications. Conversely, the application of fine-tuning did not show a significant gain in performance but did reduce the training time.

Keywords

Seismic imaging; Deep learning; Fault Segmentation; Unsupervised Domain Adaptation; Oil reservoir identification.

Resumo

Campos Trinidad, Maykol Jiampiers; Cavalcanti Pacheco, Marco Aurélio; Arauco Canchumuni, Smith Washington. **Segmentação de falhas sísmicas usando adaptação de domínio não supervisionada**. Rio de Janeiro, 2023. 73p. Dissertação de Mestrado – Departamento de Engenharia Elétrica, Pontifícia Universidade Católica do Rio de Janeiro.

A segmentação de falhas sísmicas apresenta uma tarefa desafiadora e demorada na geofísica, especialmente na exploração e extração de petróleo e gás natural. Métodos de Aprendizado Profundo (*Deep Learning*) têm mostrado um grande potencial para enfrentar esses desafios e oferecem vantagens em comparação com métodos tradicionais. No entanto, abordagens baseadas em Aprendizado Profundo geralmente requerem uma quantidade substancial de dados rotulados, o que contradiz o cenário atual de disponibilidade limitada de dados sísmicos rotulados. Para lidar com essa limitação, pesquisadores têm explorado a geração de dados sintéticos como uma solução potencial para dados reais não rotulados. Essa abordagem envolve treinar um modelo em dados sintéticos rotulados e, posteriormente, aplicar diretamente ao conjunto de dados real. No entanto, a geração de dados sintéticos encontra o problema de deslocamento de domínio devido à complexidade das situações geológicas do mundo real, resultando em diferenças na distribuição entre conjuntos de dados sintéticos e reais. Para mitigar o problema de deslocamento de domínio na detecção de falhas sísmicas, propomos uma nova abordagem que utiliza técnicas de Adaptação de Domínio Não Supervisionada ou *Unsupervised Domain Adaptation* (UDA). Nossa proposta envolve o uso de um conjunto de dados sintéticos para treinamento do modelo e sua adaptação a dois conjuntos de dados reais disponíveis publicamente na literatura. As técnicas de UDA escolhidas incluem *Maximum Mean Discrepancy* (MMD), *Domain-Adversarial Neural Networks* (DANN) e *Fourier Domain Adaptation* (FDA). MMD e DANN visam alinhar características em um espaço de características comum de n dimensões, minimizando discrepâncias e aumentando a confusão de domínio por meio do treinamento adversarial, respectivamente. Por outro lado, FDA transfere o estilo de amostras reais para sintéticas usando Transformada Rápida de Fourier. Para os experimentos, utilizamos uma versão menor do *UNet* e sua variante *Atrous UNet*, que incorpora camadas convolucionais dilatadas em seu gargalo. Além disso, o DexiNed (*Dense Extreme Inception Network*), um modelo do estado da arte para detecção de bordas, foi empregado para fornecer uma análise mais abrangente. Além disso, estudamos a aplicação de ajuste fino ou *fine-tuning* em conjuntos de dados rotulados para investigar seu impacto no desempenho, pois muitos estudos o têm utilizado para reduzir o deslocamento de domínio.

Os resultados finais demonstraram melhorias significativas no desempenho de detecção de falhas ao aplicar técnicas de UDA, com aumento médio de até 13% em métricas de avaliação como *Intersection over Union* e *F1-score*. Além disso, a abordagem proposta obteve detecções mais consistentes de falhas sísmicas com menos falsos positivos, indicando seu potencial para aplicações no mundo real. Por outro lado, a aplicação de ajuste fino não demonstrou ganhos significativos no desempenho, mas reduziu o tempo de treinamento.

Palavras-chave

Imagem sísmica; Aprendizado profundo; Segmentação de falhas; Adaptação de domínio não supervisionada; Identificação de reservatórios de petróleo.

Table of contents

List of figures

List of tables

List of Abreviations

AP – Average Precision
ASPP – Atrous Spatial Pyramid Pooling
AUPRC – Area Under the Precision-Recall curve
AUC-ROC – Area Under the Curve of the Receiver Operating Characteristic
BCE – Binary Cross-Entropy
CORAL – CORrelation ALignment
CNN – Convolutional Neural Network
CyCADA – Cycle-Consistent Adversarial Domain Adaptation
CycleGAN – Cycle Generative Adversarial Networks
DA – Domain Adaptation
DANN – Domain-Adversarial Neural Networks
DL – Deep Learning
F1 – F1-Score
FCN – Fully Convolutional Network
FFT – Fast Fourier Transform
GAN – Generative Adversarial Network
GLCM – Gray-level co-occurrence matrix
GoT – Gradient of Texture
GRL – Gradient Reversal Layer
GSB – Great South Basin
HED – Holistically-Nested Edge Detection
IOU – Intersection Over Union
ML – Machine Learning
MLP – Multi Layer Perceptron
MK-MMD – Multi Kernel Maximum Mean Discrepancy
MMD – Maximum Mean Discrepancy
ODS – Optimal Dataset Scale
OIS – Optimal Image Scale
RBF – Radial Basis Function
SVM – Support Vector Machine
UDA – Unsupervised Domain Adaptation

1
Introduction

1.1
Motivation

Seismic fault segmentation plays a crucial role in geophysics, particularly in the exploration and extraction of natural resources such as oil and natural gas. The accumulation of these resources in commercially viable quantities relies on various geological factors. These factors encompass the presence of an organic-rich source rock that generates oil or gas, a porous reservoir rock for storage, and the presence of a trap mechanism to prevent hydrocarbon leakage. One of the most common types of traps is fault traps, which result from rock displacement along seismic faults, as depicted in Figure 1.1.

Seismic faults are geological structures that exert a significant influence on subsurface reservoir behavior, controlling fluid flow and impacting hydrocarbon distribution. Therefore, accurate identification and characterization of faults are vital for constructing reliable geological models and facilitating informed decision-making in the Oil & Gas industry.

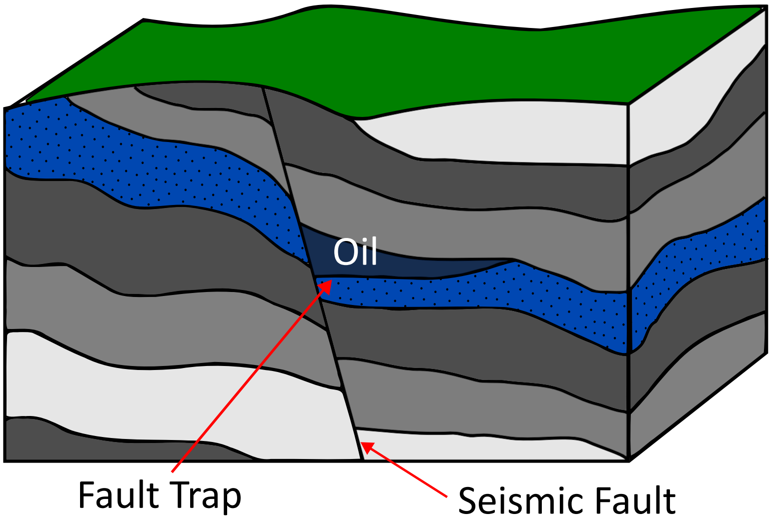


Figure 1.1: Illustration of a fault trap. The blue area depicts water positioned below the oil due to its greater density.

However, the identification of faults from seismic data poses various challenges. Typically, seismic data consists of large volumes of 3D data that represent the subsurface structure of the Earth, as observed in Figure 1.2. A single seismic survey can produce up to one terabyte of data daily, and seismic datasets can quickly exceed many petabytes. Due to the size and complexity of these datasets, advanced computational tools and techniques are required for their processing and analysis. This makes the fault identification process difficult and time-consuming, taking up to months to identify faults throughout the entire volume [1]. Furthermore, the inherent noise and ambiguity in the data further complicate the task. Lastly, manual labeling is highly subjective, and the involvement of multiple expert geophysicists is necessary to minimize errors and biases.

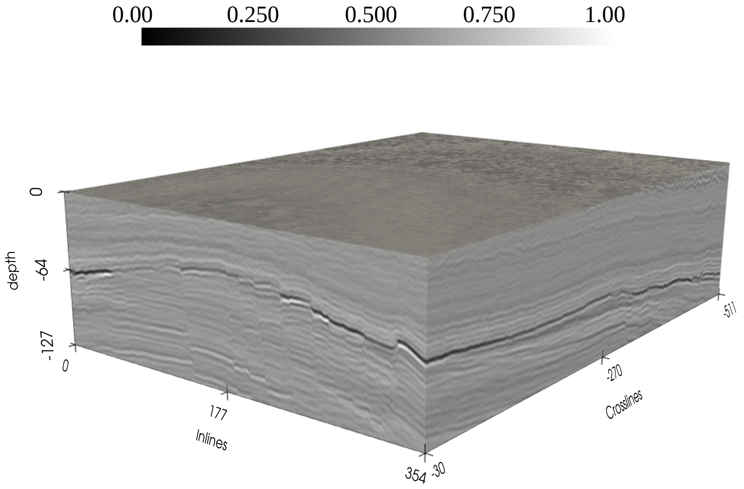


Figure 1.2: 3D real seismic volume employed in [7] for evaluation.

To address the challenges in seismic data analysis, various semi-automated and automated methods have been proposed, based on a wide range of techniques, including attribute extraction [2, 3, 4], Machine Learning (ML) [5, 6], and Deep Learning (DL) [7, 8, 9]. Among these techniques, DL has been found to be particularly effective and has achieved the best results in many applications.

Despite the potential benefits of traditional DL approaches for seismic data analysis, there are still some limitations. As previously mentioned, the primary limitation is the requirement for a substantial amount of labeled data, which can be difficult and time-consuming to obtain [10, 11]. To address this issue, researchers have explored the use of synthetic data generation for seismic faults. The idea is to train robust models on this synthetic data and then apply them to real datasets directly [8, 9, 12]. Nevertheless, due to the complexity of real-world geological situations, differences exist between synthetic and real datasets, such as variations in seismic signal frequency, noise intensity, and fault distribution frequency, making it challenging for models to generalize effectively [13]. This difference in dataset distributions is known as *domain shift* and remains a significant limitation for DL methods.

In order to tackle the domain shift problem, some studies have chosen to employ transfer learning techniques like fine-tuning. Fine-tuning involves using the knowledge gained from a pre-trained DL model on an initial dataset as a starting point for training on another dataset. This process requires manually labeling a significant number of samples from real datasets [11, 14] or using automatic labeling methods [15], which enables the application of fine-tuning with models pre-trained on synthetic data. Although these approaches have demonstrated visual improvements, they still rely on supervised tasks [16] since they heavily depend on generating labeled real samples.

Domain adaptation (DA) is a methodology aimed at improving classifier generalization by minimizing differences between different data distributions. The *source domain* represents the data used for training, while the *target domain* contains data that the model must adapt to. When no labeled samples are available in the target domain, it is called Unsupervised Domain Adaptation (UDA). UDA techniques can generally be categorized into two main groups: feature-level and image-level adaptation [17]. The former aligns features in a common and agnostic n-dimensional feature space, while the latter focuses mainly on style transfer between both domains.

In line with the methodology used in previous studies that applied UDA to the CityScapes dataset [18, 19], synthetic datasets are predominantly utilized as the source domain, whereas real datasets are employed as the target domain. Although some studies have addressed seismic fault segmentation using DA [16, 20, 21], they have applied only one technique per study, and quantitative improvements in real 3D seismic blocks through DA have not been fully explored due to the lack of labels in the real datasets.

In this dissertation, we propose a novel approach for seismic fault segmentation using UDA. Our approach aims to develop end-to-end DL frameworks that incorporates DA techniques to overcome the challenge of domain shift and enhance model robustness and generalization capabilities. We use a synthetic dataset as the source domain and two real seismic datasets as the target domain. We apply three different UDA techniques (feature and image-level adaptation) to allow for better comparison and to determine the most suitable UDA technique for this application. Quantitative improvements are measured using appropriate metrics, taking advantage of the labeled real dataset. Different DL architectures are also explored to assess their impact on the application of UDA techniques. Our approach surpasses the limitations of existing methods and demonstrates superior performance in challenging scenarios, indicating its potential for practical applications. As a complementary part of the study, fine-tuning experiments are conducted using the two labeled datasets to explore the benefits of leveraging multiple domains for enhanced model generalization.

1.2
Objectives

1.2.1
General Objective

The objective of this study is to assess UDA techniques in semantic segmentation for seismic fault detection, with the aim of minimizing the negative impact of domain shift.

1.2.2
Specific Objectives

- Implement different UDA techniques, such as feature-level and image-level adaptation, for seismic fault segmentation.
- Evaluate the quality of the seismic fault segmentation results of applying UDA techniques using visual inspection.
- Compare the performance of the seismic fault segmentation results of applying UDA techniques with traditional supervised learning approaches using accurate quantitative metrics.
- Assess the generalization capability of supervised learning approaches by fine-tuning on diverse datasets.

1.3
Contributions and Novelties

The main contributions of this work are the following:

- Implementation and evaluation of UDA techniques for the semantic segmentation of seismic faults. This task presents a greater challenge compared to image classification and can serve as a benchmark for future research in the field.

- It was explored the effectiveness of different types of UDA techniques. This allows for a more complete inspection of the UDA-based approach.
- Furthermore, it was investigated the performance of three architectures for seismic fault segmentation: U-Net, Atrous UNet, and DexiNed. This provides insights into the suitability of different DL networks for the proposed task.
- It was utilized a different approach to apply fine-tuning for generalization to new seismic datasets.
- The code[1] for the proposed methods is publicly available, along with detailed descriptions of the preprocessing and training protocols employed. This enables further research to conduct comparisons by applying other UDA techniques for fault segmentation.

Overall, this dissertation makes several contributions to the field of UDA-based seismic fault segmentation, including the use of multiple datasets, the exploration of different techniques, and the investigation of various network architectures.

---

[1] Available code repository: https://github.com/mkl04/UDAforFaultDetection

2
Related Works

As mentioned earlier, the identification of seismic faults has been of significant importance to the Oil & Gas industry for numerous years, as it facilitates the exploration and discovery of new reservoirs. Thus, different techniques have been developed to achieve this identification, which can be classified into two categories: seismic attributes-based methods, and ML-based methods [11].

2.1
Seismic attributes-based methods

Also known as handcrafted features-based methods, these methods were the first to be used for seismic fault identification and were based on the extraction of certain features or characteristics of different types in mainly 2D sections. These attributes can be divided into three groups: edge detection, geometric, and texture [5, 14]. Basic edge detection techniques, such as coherence [2], semblance [22], variance [23], and chaos [24], are still used for relatively simpler seismic interpretation, as they capture the high-frequency variations and gradients that are indicative of fault edges. Geometric attributes aim to capture the geometric properties and patterns associated with faults. These attributes include measurements such as fault length and are obtained by quantifying the geometric variations of seismic reflectors using reflector curvature [3] or flexure [25]. Lastly, textural attributes are more challenging to extract compared to other types of attributes. These attributes rely on statistical analysis, where the spatial distribution of intensity levels surrounding a pixel is estimated. Several texture attributes have been suggested, including directionality, smoothness, and edge content [26], gray-level co-occurrence matrix (GLCM) contrast and homogeneity [4], and gradient of texture (GoT) [27], among others. Although widely used in remotely sensed data, texture-based techniques face the issue of detecting line-like edges while dealing with spatial resolution statistics [11].

In seismic interpretation, seismic attributes-based methods face several limitations when dealing with large-scale seismic data obtained through 3D surveying technologies. These methods are highly time-spatially complex, computationally intensive, and often struggle to achieve high accuracy, making them impractical for industrial deployment [1]. Some seismic attributes such as semblance, coherence and variance are highly sensitive to noise and insufficient for detecting faults [7]. Moreover, preprocessing of seismic datasets is necessary to achieve a certain signal-to-noise level before applying feature extraction methods. However, due to the sensitivity of these methods to noise, it is often necessary to manually adjust parameters by trial and error, taking into account the unique geological structure and data quality of different datasets [9].

2.2
ML-based methods

Initially, many researchers chose to use the attributes or features extracted by applying the techniques described in the previous section and complement them with classic ML techniques, such as Random Forest, Support Vector Machine (SVM) or even Multi Layer Perceptron (MLP). However, in some cases, defining labels indicating the presence of faults was necessary. The initial approach involved a classification procedure, where each section of the 3D seismic volume was divided into small patches, and it was assigned whether it belonged to a fault or not based on a minimum percentage of seismic fault pixels present in the patch.

For example, in [5], the authors extracted 14 attributes from the three mentioned groups (edge detection, geometric, and texture), which were used as input variables to the SVM model to classify between two sets of manually selected samples where one set indicates the presence of seismic faults and the other does not. Similarly, in [6], a multi-attribute SVM was applied, but using another group of attributes called descriptors, which include Local Binary Pattern and its variants [28]. There have also been some attempts to apply unsupervised algorithms, such as in [29], where another list of attributes was extracted, and the different pixels were clustered using the Self-Organizing Maps algorithm. Although combining attributes with ML improved the results, the drawbacks of time-consuming feature extraction and complexity remained.

Following the success of Convolutional Neural Networks (CNN) in image classification tasks across different domains, researchers began extracting features from seismic data at the image level. Specifically, they focused on texture attributes while allowing the model to learn and generate new filters automatically during training. These techniques are commonly referred to as DL-based methods, which use multiple convolutional layers to improve validation scores at the cost of higher computational requirements. For example, in [8], it was trained a basic CNN using manually labeled sections from

Great South Basin (GSB), which is a very used dataset for identification of faults. Their network architecture was based on LeNet-5 [30], as illustrated in Figure 2.1. Similarly, in [31], it was employed three sections along each orthogonal axis to estimate the pixel values at their intersections. To accomplish this, they concatenated the three sections as input to their customized CNN and used an auto-picking method to label its data.

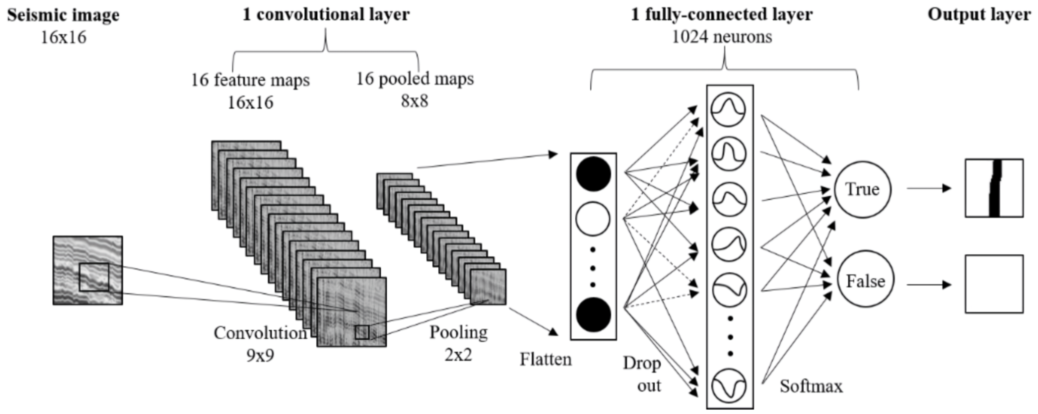


Figure 2.1: Basic CNN classifier used for fault detection (extracted from [8]).

A common challenge observed in these research studies was the lack of large volumes of high-quality labeled data due to the time-consuming process of manual labeling and the unavailability of publicly shared datasets. To overcome this limitation, the use of synthetic data generation techniques based on the workflow of [32] was adopted. This approach involves training models using synthetic seismic data and subsequently evaluates their performance on real datasets. For instance, in [33], synthetic seismic sections were generated along with their corresponding seismic faults for classification, similar to Figure 2.1. In another study [34], the focus was not only on fault classification but also on predicting the inclination angle of the generated seismic faults.

Furthermore, in [7], a more advanced approach was pursued by shifting from image classification to pixel-level classification or *Semantic Segmentation*. In this approach, 3D seismic blocks with multiple faults along their three axes were generated. The trained model, named FaultSeg3D, was a simplified version of the well-known UNet architecture [35], adapted to 3D inputs as illustrated in Figure 2.2. This approach resulted in more continuous fault predictions, which were further enhanced using other architectures such as ResUNet [36] and UNet++ [12].

As previously mentioned, transfer learning was one of the proposed solutions to address the domain shift arising from the distribution differences between synthetic and real data [13], where the synthetic samples are the ones

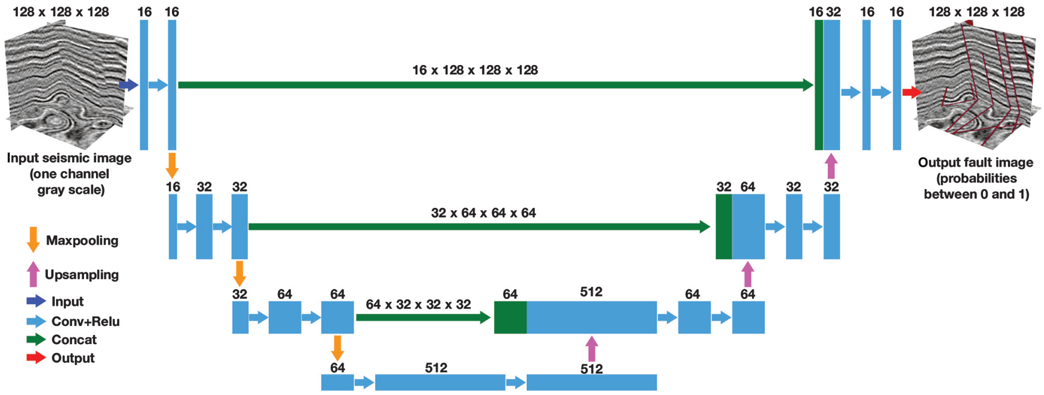


Figure 2.2: FaultSeg3D architecture for 3D fault segmentation (extracted from [7]).

generated in the prior study [7]. In [14], some sections of the real dataset were manually labeled, and two transfer learning techniques were applied. The first approach involved training the entire network with initial weights set to the pre-trained model weights, rather than random ones (full fine-tuning). The second approach froze the weights of the convolutional layers, using them only as a feature extractor. The extracted features were then fed into an SVM or MLP to classify the presence or absence of seismic faults. Although studies have been conducted for automatically generating labels in real datasets to apply fine-tuning, as in [15], the potential improvement depends on the amount of available labeled data and the domain shift between datasets [37].

Before exploring studies that have applied Domain Adaptation (DA) to mitigate the domain shift, it is important to acknowledge the work in [9]. They introduced a new labeled real dataset named Thebe and made specific modifications to address the task of edge detection, resulting in significant performance enhancements on datasets such as GSB and the one utilized in FaultSeg3D. A comprehensive summary of other traditional DL-based studies on seismic faults up until 2020 can be found in [1].

According to literature [38], the first paper that applied UDA for Semantic Segmentation was [19], which utilized domain adversarial training for global domain alignment and class-aware constrained multiple instance loss for transferring spatial layout. This approach has been extended to various applications, including faults detection. For instance, in [20], the authors utilized Domain Adversarial Neural Network (DANN) [39] as a domain adaptation method for a model based on UNet [35]. They used the same synthetic data generated in [7] as the source domain for training and a real dataset as target domain. Since there were no labels, the evaluation was visual, and this feature-level adaptation achieved the detection of much finer seismic faults.

Later, the same author propose a progressive transfer learning to reduce the impact of data distribution differences and update the training dataset with predicted intermediate data and pseudo-labels. The next intermediate data is processed progressively, reducing the distance between the updated training data and the target prediction data. The experimental results show that this method is effective in improving the performance of fault detection in seismic data with limited labeled samples [21].

In another study [16], the problem was addressed using Cycle Generative Adversarial Networks (CycleGAN) [40], which falls under image-level adaptation techniques. CycleGAN involves generating data using the style of the samples in both domains. The results showed great style transfer in real datasets, which could be further explored using the datasets presented in this dissertation to evaluate its performance using metrics.

The latest paper, to the best of our knowledge, that addresses bridging the domain discrepancy between synthetic and real seismic datasets is presented in [41]. They propose an alternative data standardization technique to make both the source domain and target domain distributions similar. To achieve this, they apply a series of transformations such as z-score normalization and equal frequency normalization. Additionally, to leverage the advantages of 3D-based models that capture more spatial context information, they use multiple sections as inputs, treating them as channels, and utilize the labels from the middle section as the target. This approach was called 2.5D inputs and improves visual results without significantly consuming GPU memory and reduces training time.

3
Fundamentals

3.1
Seismic Data Interpretation

Seismic data interpretation is of great importance in the Oil & Gas industry, particularly in the field of seismic exploration. Through the utilization of advanced technology and geophysical techniques, seismic interpretation plays a vital role in gaining a deep understanding of the fundamental principles necessary to comprehend the subsurface of the Earth and identify valuable energy resources.

Seismic exploration begins by utilizing reflective seismic waves to predict subsurface structures. This intricate process involves a series of critical steps, including the acquisition and processing of seismic data. These data processing steps encompass various techniques like denoising to remove unwanted noise and interpolation for filling in missing data gaps.

Additionally, inversion techniques are employed, which include migration and imaging methods. Migration techniques help transform the recorded seismic data into a spatial representation, allowing for a more accurate depiction of subsurface features. Imaging techniques refine these representations further, enhancing the quality and resolution of the subsurface images. Furthermore, seismic exploration also involves interpretation tasks, such as fault detection and facies classification [42]. The procedure of exploration geophysics is summarized in Figure 3.1.

Geo-scientists analyze seismic data to unravel the complexities of subsurface structures and identify potential areas where oil and gas deposits may be trapped. Trap identification is a critical initial step in evaluating prospects and is integral to any exploration or assessment program. A trap can be defined as a geometric arrangement of rock that enables the significant accumulation of hydrocarbons beneath the surface of the Earth [43].

Traps can be broadly categorized into three types: structural traps, stratigraphic traps, and combination traps that exhibit both structural and stratigraphic elements. Structural traps can further be subdivided into fold traps (anticlinal), traps associated with geological faults, traps associated with

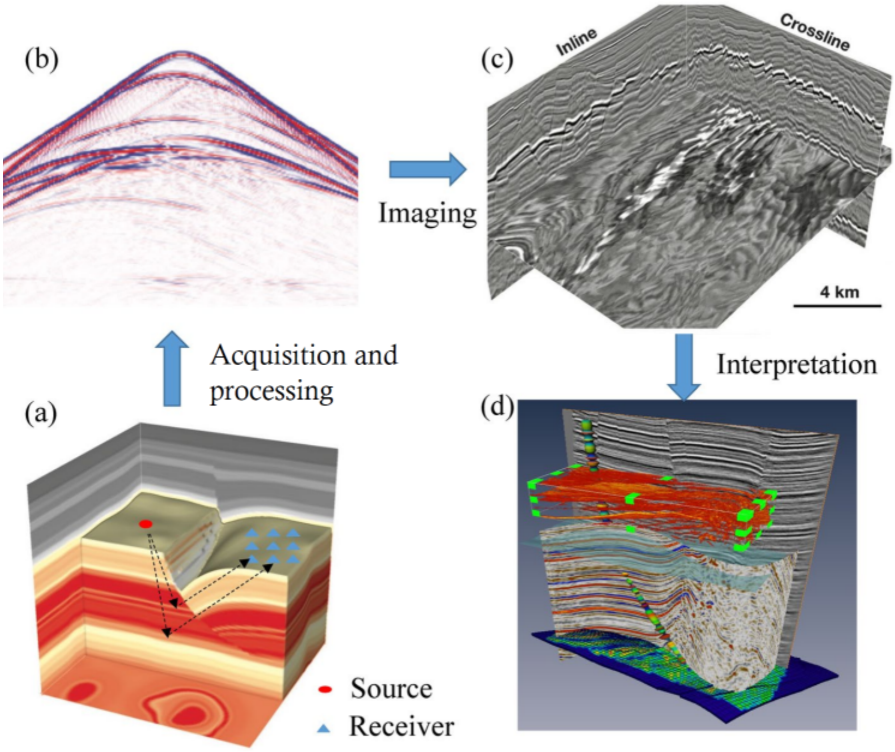


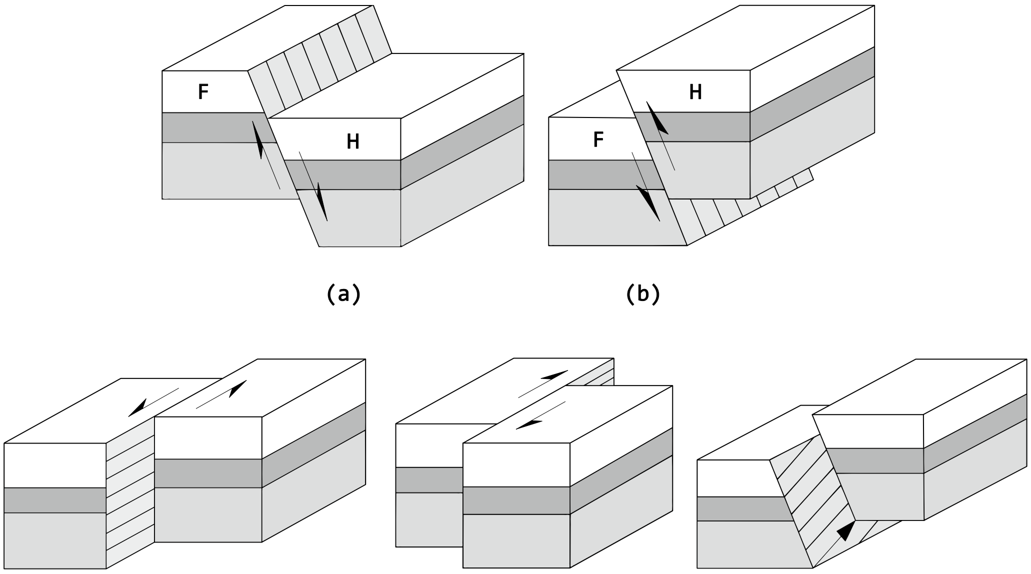
Figure 3.1: The procedure of exploration geophysics. (a) Seismic waves are propagated downward to the reflector and then are recorded by the receivers. (b) Velocity inversion method as preprocessing. (c) The seismic imaging result. (d) Seismic Interpretation. Figure extracted from [42].

piercement features (such as salt bodies), and combination traps that require elements of both faults and folds for their effectiveness. The significance of identifying fault traps and the utilization of ML tools for this purpose is emphasized here due to their substantial advantages over alternative technique [1].

3.1.1
Geological faults

Geological or seismic faults are defined as plane surfaces where the relative movement of tectonic units occurs. These faults can either facilitate fluid flow or act as boundaries that create hydrocarbon structural traps or reservoir compartments. Faults are of great importance in hydrocarbon exploration and field development, making it essential to understand the complex connections between fault networks and fractures [1]. They are also a primary consideration in reservoir characterization [5].

The recognition of faults can be made by observing the presence of different rock types across a generally flat surface, which exhibit textures and structures that are typically produced by shearing. Fault planes can either be vertical or inclined at an angle, known as the dip, which is the acute angle



(a) (b)

(c) (d) (e)

Figure 3.2: Types of fault, where arrows indicate direction of movement. (a) normal fault; (b) reverse fault; (c) sinistral fault; (d) dextral fault; (e) oblique-slip (sinistral-reverse) fault. Figure adapted from [44].

between the fault plane and a horizontal surface. Faults are classified based on the direction the crustal blocks have been displaced parallel to the fault plane. Dip-slip displacement occurs when there is vertical movement oriented parallel to the dip of the fault plane, while horizontal offset parallel to the fault plane is referred to as strike-slip displacement. Oblique-slip faults have components of both dip-slip and strike-slip displacement.

Dip-slip faults are categorized as normal or reverse depending on the relative movement of the crustal blocks bounding a fault plane. Normal faults occur when the hanging wall moves down relative to the footwall, while reverse faults have the opposite sense of displacement. On the other hand, as per convention, a strike-slip fault exhibits a left-lateral displacement, commonly known as sinistral displacement, when the crustal block opposite the position of observer moves to the left. In contrast, a right-lateral displacement, known as dextral displacement, takes place when the opposite crustal block is shifted to the right from the perspective of observer [44]. In Figure 3.2, the described types of faults are shown.

3.2
Semantic Segmentation

Prior to delving into the definition of semantic segmentation, it is pertinent to provide a brief overview of the concept of Deep Learning (DL), which is a subfield of ML that involves the use of artificial neural networks with multiple layers, enabling the learning of complex data representations. DL has achieved success in various applications, including image and speech recognition, natural language processing, and robotics [45].

3.2.1
State-of-the-art

The boom of DL in computer vision began with the application of CNN [46] in the 2012 ImageNet competition, which involved classifying 1.3 million high-resolution images into 1000 different classes where CNN surpassed other solutions. However, it should be mentioned that these had already been presented in [30] to perform classification on the well-known MNIST dataset of handwritten digits.

To identify different objects within an image, pixel-level classification or semantic segmentation was necessary, since image classification alone was not sufficient for the task. One of the proposed approaches at the beginning was to generate several small subsets from input image, known as patches, and classify the pixel in the center of each patch. After being trained, the inference was performed throughout the entire image and the output was reconstructed. However, this process was very expensive computationally and did not return good results.

It was not until 2014 that was introduced the first ideas of applying downsampling at the beginning, and then converting all fully connected layers to convolutions to return to the original dimensions of the input image at once, turning the networks into a Fully Convolutional Network (FCN) [47]. This type of CNN architecture is also commonly referred to as an encoder-decoder structure and enables the extraction of essential features or filters without the need for an excessive number of parameters. An example is illustrated in Figure 3.3, which belongs to the architecture SegNet [48].

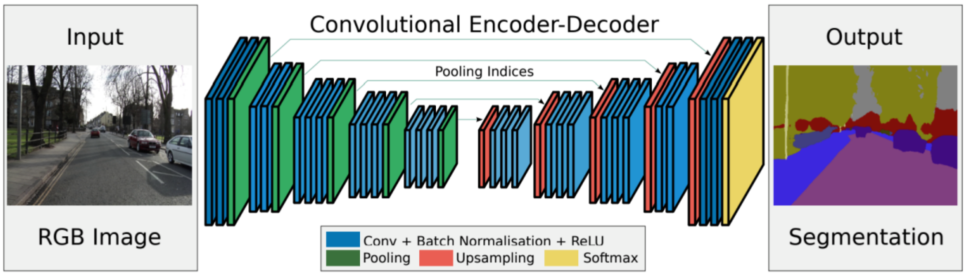


Figure 3.3: Convolutional encoder-decoder architecture for semantic segmentation (extracted from [48]).

A representative diagram illustrating the distinctions between image classification and semantic segmentation is presented in Figure 3.4. As depicted, these tasks address different questions. Image classification focuses on determining the presence or absence of a seismic fault in the input image, along with a certain probability. On the other hand, semantic segmentation provides a probability map of the same size as the input image, allowing the separation of pixels that are detected as seismic faults based on a threshold.

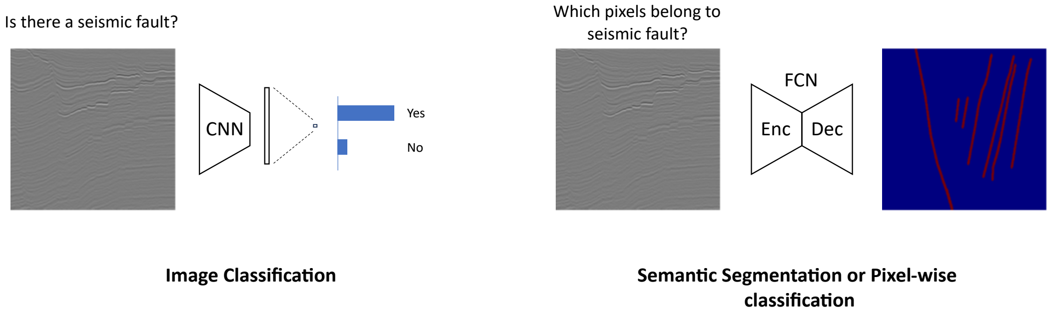


Figure 3.4: The differences between image classification and semantic segmentation.

Next, it will be described the networks used for this study, along with a brief explanation of why they were considered as options.

3.2.1.1
U-Net (UNet)

UNet was chosen as the base model due to its impressive performance and relatively simple implementation. It is a type of CNN commonly used for semantic segmentation tasks that was introduced in 2015 [35]. The architecture of UNet consists of an encoder-decoder structure with skip connections between corresponding layers of the encoder and decoder. The encoder part is composed of multiple convolutional layers with pooling operations to downsample the input image, while the decoder part upsamples the feature maps to produce the segmentation mask. The skip connections help to preserve the spatial information of the image and improve the accuracy of the segmentation.

UNet has been widely used in various applications such as biomedical image segmentation, satellite image analysis and also for fault segmentation [9, 7]. It has shown promising results and outperformed other segmentation methods in many cases.

3.2.1.2
Atrous U-Net (Atrous UNet)

This is a customized network that was originally presented during a Kaggle competition[2] and later utilized in one of our papers, where its superior performance in segmenting seismic images for Facies identification was demonstrated [49]. The network, known as Atrous UNet, was also employed by [50] for satellite image segmentation. This architecture follows an encoder-decoder format with a modification in the bottleneck section, incorporating a series of Atrous or Dilated Convolutions with varying dilation rates (r), similar to the Atrous Spatial Pyramid Pooling (ASPP) [51]. Unlike traditional convolution, which employs a fixed-sized kernel, dilated convolution introduces gaps in the kernel to expand its receptive field without increasing the number of parameters. This enables the network to capture features at various scales, making it particularly advantageous for tasks such as image segmentation. For instance, as depicted in Figure 3.5, the filter on the right-hand side operates much like a 5×5 kernel while utilizing the same number of parameters as a 3×3 kernel.

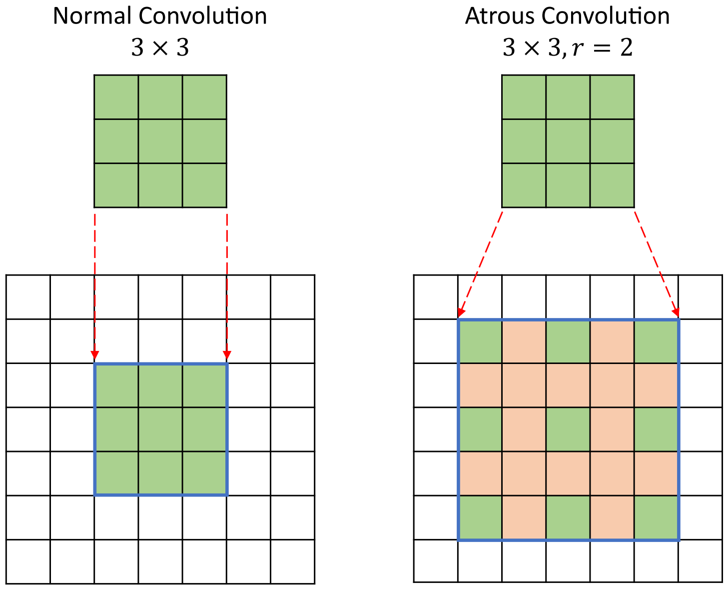


Figure 3.5: Both convolution layers have a kernel size of 3×3 and the same number of parameters. The Atrous or Dilated Convolution has a dilation ratio (r) of 2, which allows it to capture more contextual information without increasing the number of parameters.

---

[2] Code repository: `https://github.com/lyakaap/Kaggle-Carvana-3rd-place-solution/`

3.2.1.3
Dense Extreme Inception Network (DexiNed)

In the paper [9], a shift from semantic segmentation to edge detection was proposed for seismic fault analysis due to the thin nature of seismic faults. To address this, more robust networks such as Holistically-Nested Edge Detection (HED) [52] were utilized. However, further research revealed that DexiNed had already surpassed HED in different datasets such as BSDS500 [53], so it was decided to employ DexiNed for this study. DexiNed introduced the Inception module, which comprises parallel convolutional layers with different kernel sizes. This architecture enables the network to capture information at multiple scales and extract finer details, thereby enhancing its accuracy in detecting edges and boundaries. Similar to the Atrous UNet, DexiNed utilized dilated convolutions.

3.2.2
Evaluation Metrics

Three widely used metrics to assess labeled datasets for semantic segmentation tasks are Intersection Over Union (IOU), F1-score (F1), and Average Precision (AP). These metrics are derived from two fundamental measures: Precision (P) and Recall (R). Precision calculates the fraction of retrieved information that is relevant, whereas Recall estimates the fraction of relevant information that is retrieved by the models:

$$P = \frac{TP}{TP + FP} \tag{3-1}$$

$$R = \frac{TP}{TP + FN} \tag{3-2}$$

where, for this research, True Positive (TP) are pixels that both the machine and the interpreter consider to be fault pixels, False Positive (FP) are pixels that the machine considers to be fault pixels while the interpreter does not, and False Negative (FN) are pixels that the interpreter classifies as fault pixels while the machine does not.

Since some applications require both a good Precision and Recall, F1-score provides a balanced evaluation of the two metrics when both are considered equally important. The F1-score is calculated as the harmonic mean of Precision and Recall:

$$F1 - score = \frac{2 \times P \times R}{P + R} \tag{3-3}$$

The IOU score is another important metric in semantic segmentation. It measures the overlap between the ground truth and the prediction, derived from Precision and Recall as follows:

$$IOU = \frac{intersection}{union} = \frac{TP}{TP + FN + FP} = \frac{1}{\frac{1}{R} + \frac{1}{P} - 1} \tag{3-4}$$

Furthermore, Average Precision (AP) calculates the Area Under the Precision-Recall curve (AUPRC) graph in binary classification tasks, taking into account all possible threshold values. As the model assigns a probability value between 0 and 1 to each pixel, the AP summarizes the overall performance of the classifier. Unlike the widely used AUC-ROC (Area Under the Curve of the Receiver Operating Characteristic), Average Precision (AP) evaluates the effectiveness of a model's positive classification, specifically its capability to accurately classify seismic faults as positive cases. The distinction between these two curves is illustrated Figure 3.6, where the AP score provides a more realistic assessment of the model's performance.

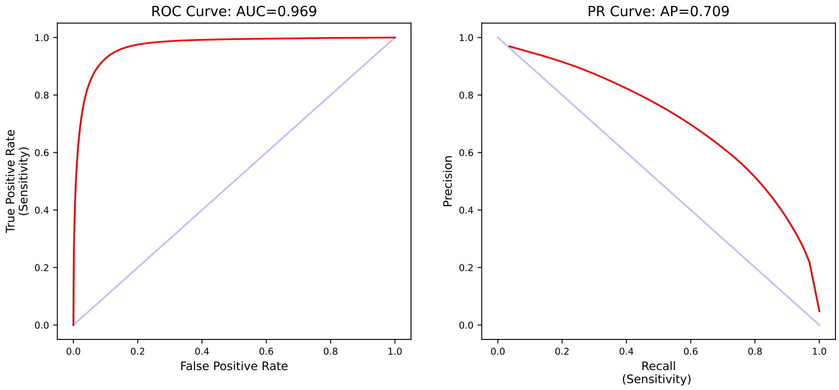


Figure 3.6: The difference between both curves: AUC-ROC (left) and AUPRC (rigth).

While in the described curves, all possible thresholds are considered, in tasks such as edge detection, knowing the optimal threshold has a greater impact both on the metrics and visually. For this reason, Optimal Dataset Scale (ODS) and Optimal Image Scale (OIS) are the most widely used and representative evaluation metrics for assessing edge detection [54]. ODS, also known as global best, calculates the F1-score using the best fixed threshold across the entire dataset, while OIS selects the best threshold for each image and then calculates the F1-score. Based on these definitions, OIS will always be greater than or equal to ODS.

3.3
Domain Adaptation for Seismic Fault Detection

Domain Adaptation (DA) techniques emerged as a consequence of the high cost of acquiring labeled data, which is necessary for training DL models. To have a clear understanding of the concepts, it is necessary to first introduce their notation.

Let $D_S = X_S \times Y_S$ represent a collection of paired samples and annotations ($X_S = \{x_i\}_{i=1}^{M}$ and $Y_S = \{y_i\}_{i=1}^{M}$, respectively). These samples are drawn from a source distribution $P_S(X, Y)$, which is the source domain. In the context of semantic segmentation for seismic faults, x and y correspond to images and their pixel-wise annotations, respectively. Here, $x \in R^{H\times W\times 1}$ and $y \in R^{H\times W\times 1}$, where (H, W) is the image size. Note that x has only one channel as it represents gray-scale input images. Let $D_T = X_T = \{x_i\}_{i=1}^{N}$ be a set of unlabeled samples obtained from a target distribution P_T, representing the target domain, such that $P_S \neq P_T$ due to the domain shift. In the UDA setup, both sets are available during training ($D = D_S \cup D_T$). The ultimate objective is to train a model that performs effectively on samples from the target distribution [38].

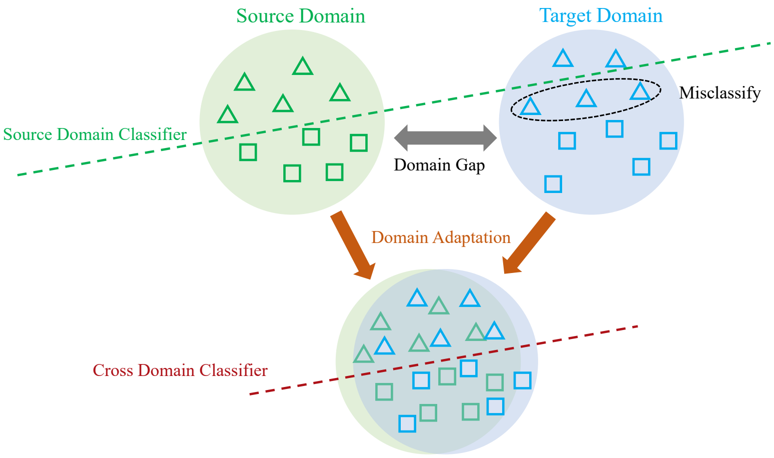


Figure 3.7: Representation of Domain Adaptation to mitigate the domain shift problem between the source and target domains (extracted from [55]).

The process of adapting a model trained in a source domain to a target domain, with the aim of reducing the domain gap and improving performance, is referred to as Domain adaptation and is depicted in Figure 3.7. When this adaptation is achieved without using labeled target samples, it is called Unsupervised Domain Adaptation (UDA) [38]. Given the complexity and nonlinearity of DL models, numerous algorithms for UDA have been developed in recent years, such as metric alignment-based and adversarial learning-based algorithms [55], whose definitions are represented in the techniques that were used.

UDA techniques can be broadly classified into two main types. The first type, known as feature-level adaptation, focuses on reducing the domain gap by aligning features in a common and agnostic n-dimensional feature space. The second type, image-level adaptation, aims to transfer the visual style from the target domain to the source domain. These groups are also referred to as *representation matching* and *image translation*, respectively [37].

As proposed in this dissertation, the goal is to utilize at least one technique from each main type of UDA to enable a direct comparison and determine which approach would be most suitable for seismic fault segmentation.

3.3.1
Feature-level adaptation

The general UDA methods proposed for image classification aim to align different domains by minimizing some distance metrics, such as Maximum Mean Discrepancy (MMD) [56] or CORrelation ALignment (CORAL) [57], or through adversarial training of a domain discriminator. These methods operate in a latent space to align feature distributions of source and target data. They have been successfully applied in semantic segmentation problems, particularly the adversarial training approach, which has been combined with other techniques to achieve improved results.

3.3.1.1
Maximum Mean Discrepancy (MMD)

The intuition behind this metric is that learn a representation that minimizes the distance between the source and target distributions, then it can be trained a classifier on the source labeled data and directly apply it to the target domain with minimal loss in accuracy [56].

To minimize the distance between the source and target domains, different versions of MMD have been used. The empirical approximation is given by:

$$MMD(X_S, X_T) = \left\| \frac{1}{|X_S|} \sum_{x_s \in X_S} \phi(x_s) - \frac{1}{|X_T|} \sum_{x_t \in X_T} \phi(x_t) \right\|_2 \tag{3-5}$$

where $\phi(.)$ is a representation of the samples in both the source and target domains. This value is minimized and usually accompanies the minimization of the loss of the task to be optimized, such as binary cross-entropy (BCE) for image classification.

Some research has focused on optimizing and enhancing current metrics. For instance, in [58], it was introduced a multi-kernel version (MK-MMD) to reduce the Type II error, while in [59], it was proposed a multi-RBF kernel as the Taylor expansion of the Gaussian function can match all the moments of the two populations. For this research, it was chosen the last one version. Other researchers utilize it to complement their proposed techniques, as evident in [60].

3.3.1.2
Domain-Adversarial Neural Networks (DANN)

The Domain Adversarial Neural Network (DANN) [39] is a topology that utilizes adversarial training to learn latent representations of images that are invariant across domains. Specifically, it minimizes the discrepancy between two latent probability distributions, which are parameterized by a *feature extractor* ($G_f(., \theta_f)$) as shown in Figure 3.8.

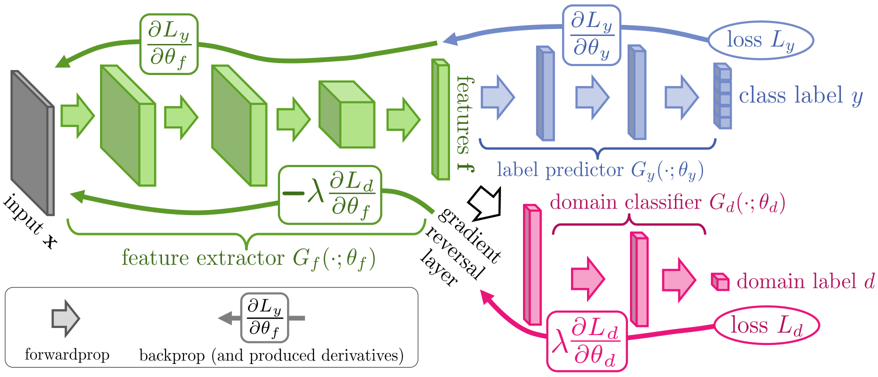


Figure 3.8: DANN architecture (extracted from [39]).

To achieve UDA, a *domain classifier* ($G_d(., \theta_d)$) is added and connected to the feature extractor using a Gradient Reversal Layer (GRL), which has no associated training parameters. The GRL acts as an identity transformation during forward propagation, but during backpropagation, it multiplies the gradient by a negative constant ($-\lambda$). This multiplication ensures that the gradients flowing through the GRL are reversed and, thus, contribute to minimizing the domain discrepancy. The *label predictor* ($G_y(., \theta_y)$) then estimates the labels of the input samples which are used to compute the classifier loss ($\mathcal{L}_y$). The total loss is given by the following equation:

$$E(\theta_f, \theta_y, \theta_d) = \mathcal{L}_y(\theta_f, \theta_y) - \lambda \mathcal{L}_d(\theta_f, \theta_d) \tag{3-6}$$

where, ($\mathcal{L}_d$) is the loss for the domain classifier. The optimal parameters ($\theta_f^*, \theta_y^*, \theta_d^*$) are obtained during training by updating the parameters in two stages to satisfy the following equation:

$$(\theta_f^*, \theta_y^*) = \underset{\theta_f, \theta_c}{\operatorname{argmin}}\, E(\theta_f, \theta_y, \theta_d^*) \tag{3-7}$$

$$\theta_d^* = \underset{\theta_d}{\operatorname{argmax}}\, E(\theta_f^*, \theta_y^*, \theta_d) \tag{3-8}$$

3.3.2
Image-level adaptation

In contrast to feature-level adaptation, some works perform domain adaptation at the image level by translating source data to the style of the target domain, or vice versa. Many strategies in this category are based on the CycleGAN technique [40], which enables image translation between different domains without paired training data. This neural network tackles the Generative Adversarial Network (GAN) method [61], which involves two networks pitted against each other: a generative model, G, that captures data distribution and a discriminative model, D, that distinguishes between G-generated samples and training data images by predicting a binary label. Through a process of cycle consistency, the network can learn to map images from one domain to another, resulting in consistent-looking images in both domains.

While some techniques have focused on modifying input images and have demonstrated better results on some datasets such as CityScapes [62], this study confines itself to conventional fault detection methods. The objective is to begin the investigation with less complicated techniques and gradually advance to more complex network implementations. Fortunately, there are techniques such as Fourier Domain Adaptation (FDA) that outperformed other techniques like Cycle-Consistent Adversarial Domain Adaptation (CyCADA) [63] with minimum effort for the same CityScapes dataset [18].

The FDA technique [18] starts by computing the Fast Fourier Transform (FFT) of an image from both the source domain and the target domain, extracting both amplitude and phase information. In the subsequent step, the low-frequency components of the target image are translated onto the source image to facilitate this stylistic transition between the two domains. Image selection is done randomly, and the magnitude of spectral signal transfer between domains is controlled using the parameter $\beta \in (0,1)$. The recommended value is 0.3, as a higher value of β can introduce artifacts. After this frequency-based substitution, the inverse FFT is applied to the source image, yielding an image that now possesses a style akin to that of the target domain.

The Figure 3.9 illustrates an example of applying spectral transfer from a CityScapes image, representing the target domain, to a labeled synthetic image from the source domain. These newly generated images will have a style similar to the target domain and will be used to train a model using the labels from the source domain, resulting in more accurate inference in the target domain.

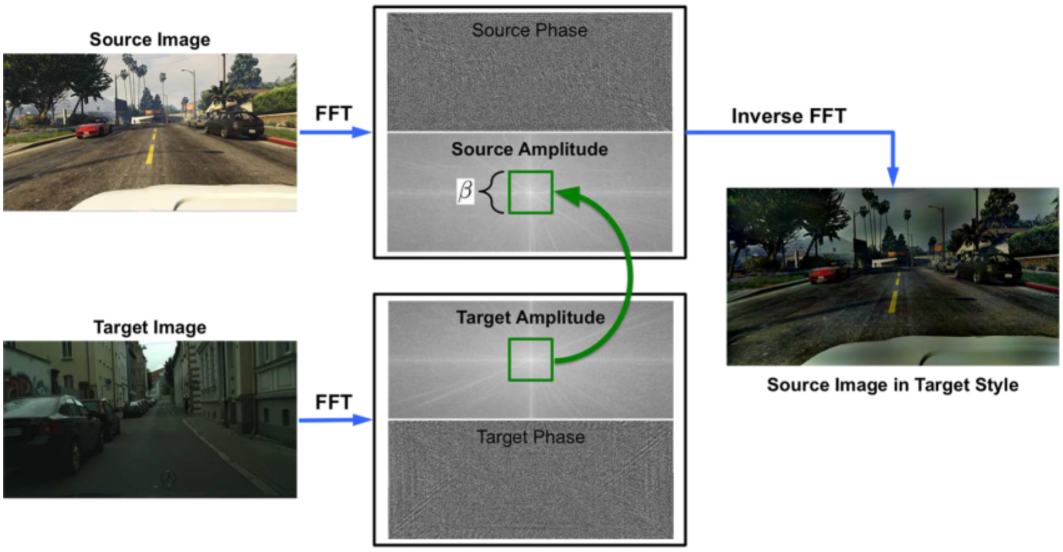


Figure 3.9: Example of spectral transfer for CityScapes dataset (extracted from [18]).

4
Methodology

This chapter provides a comprehensive overview of the methodology proposed for this study. We utilized a total of three datasets, two of which were labeled either manually or through synthetic generation, while the remaining was unlabeled public real dataset [9, 7]. To ensure consistency, a standardized preprocessing technique was applied to each dataset, allowing for the utilization of the same neural network despite their varying dimensions.

Moreover, a detailed explanation of the different training stages is presented. This includes supervised training to establish baselines for each labeled dataset, followed by training using UDA techniques at either the feature-level or image-level. Additionally, as a complement to traditional approaches, fine-tuning is employed to assess its impact on the initial supervised training stage.

Lastly, the inference process is discussed, highlighting the need for image reconstruction techniques, as well as the adoption of various metrics for evaluating the labeled datasets. This section also introduces novel metrics specifically designed for seismic fault detection applications. The overall framework of the proposed methodology is illustrated in Figure 4.1.

4.1
Preprocessing protocol

While many available datasets consist of large volumes, researchers have found in several papers [9, 11, 20, 64] that an effective approach is to divide them into sections for each volume. This methodology provides greater flexibility in defining the input dimensions of our model, as generating voxels can be challenging due to considering size in all three axes. Since the sections generated for each volume still remain large and non-uniform, it was chosen to generate square patches or tiles within each section extracted from the volumes.

Firstly, a *patch size* was defined for all datasets to avoid any complications during model training and inference. For extraction, a sliding window procedure with a 50% overlap in each direction was used, along with reflection padding to ensure inclusion of border pixels. Similar to [9], it was used an additional parameter called *step size*, which involves skipping consecutive

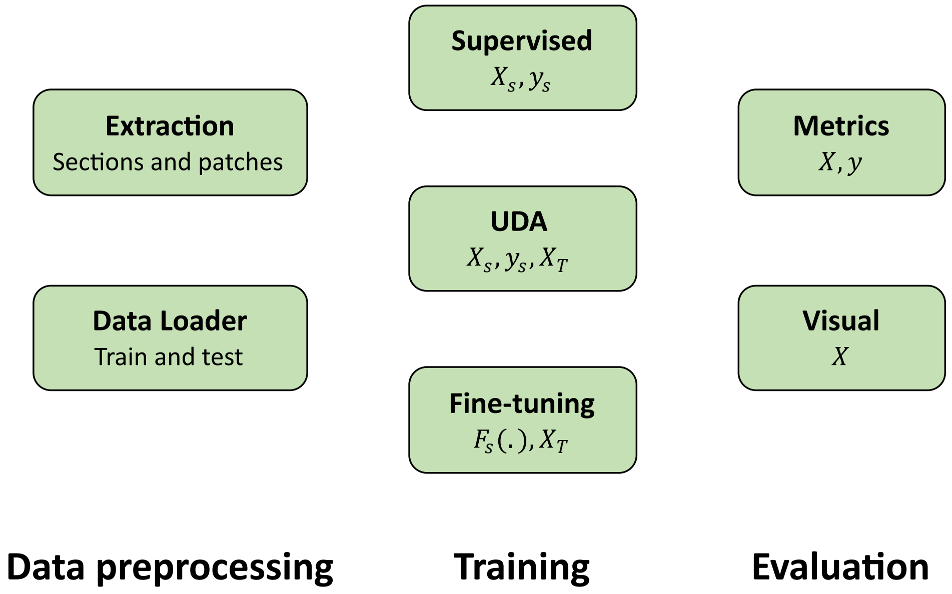


Figure 4.1: The general scheme of the proposed methodology. Where X_S and y_S represents images and labels on source domain, respectively. X_T are dataset on target domain and $F_S(.)$ is the pre-trained model on source domain for fine-tuning on target domain.

sections to reduce the number of samples and thus decrease training time, as neighboring sections share very similar characteristics. To confirm this point, the pixel differences between consecutive sections were computed, revealing an average close to zero.

The authors also suggested utilizing patches in which only 3% of the pixels were classified as faults. However, this contradicts the assumption of not having predefined labels, as mentioned in [65]. After a quick inspection of the patches generated without any filtering, it was observed that some patches had a high percentage of identical values in areas near the ends of the sections, which, in turn, did not contain seismic faults. This could have resulted from complexities in the seismic imaging process due to insufficient seismic signal information. Therefore, only patches where more than 50% of the pixels had distinct values were considered, as this approach helps to avoid selecting patches that were generated inaccurately or contained missing information. The final proposed protocol is illustrated in Figure 4.2, using a step size of 3, which was chosen empirically.

Finally, after extracting and filtering patches from each dataset, data loaders were established for every dataset. These loaders encompassed the train, validation, and test sets, which will be subsequently defined for each respective dataset. This facilitated direct loading and allowed for the modification of certain configurations, such as the number of samples to be used

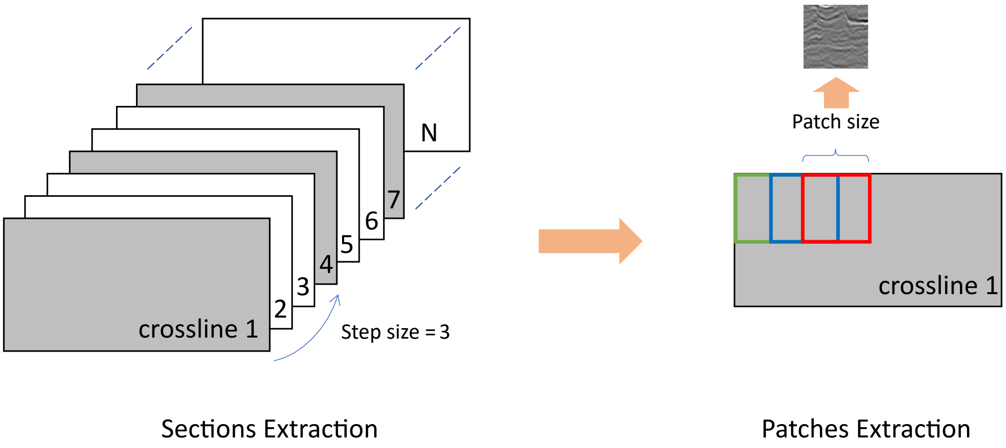


Figure 4.2: Proposed preprocessing protocol.

during training, the standard normalization or z-score, and the application of different combinations of data augmentation.

4.2
Training protocol

For this study, three training stages were conducted to facilitate a comprehensive analysis of the results obtained from traditional and UDA methods. Prior to training, it is important to note that neural networks with an encoder-decoder architecture were selected due to their adaptability to the discussed UDA methods, with the encoder primarily functioning as a *feature extractor* or *backbone*.

The first stage, referred as Supervised Training, enabled the establishment of lower and upper bounds performance for subsequent stages. To achieve this, each labeled dataset, whether synthetic or real, was trained using the selected models (UNet, Atrous-UNet, and DexiNed) on their respective training sets through a supervised learning approach. Depending on the database in which the trained models were assessed, the performance limits were defined. For instance, evaluating a model on the test set of the same dataset on which it was trained yields the maximum attainable value (upper bound). Conversely, assessing the model on a distinct dataset without any Domain Adaptation can be considered as the expected minimum performance without UDA (lower bound). Ideally, the metrics obtained using UDA should be closer to the maximum value than the minimum value.

In the second stage, consistent with the approach used in prior research that applied UDA to the CityScapes dataset [18, 19], synthetic datasets are used as the source domain, while real datasets are used as the target domain in order to apply UDA. The architecture configurations and loss functions were modified based on the chosen UDA methods and adjusted empirically. Among the techniques mentioned in the previous chapter 3, only MMD, DANN, and FDA were used in this study. The configuration of architectures involved determining the placement and necessity of multiple branches for feature-level adaptation techniques. In the case of FDA, no architectural changes were made, as the style transfer is applied prior to the input of patches into the neural network. Evaluation was exclusively carried out on the validation sets of the real datasets.

Lastly, since many studies have applied transfer learning techniques [9, 14, 10], it was also explored the use of one of its most well-known techniques, fine-tuning, to determine its impact on the results. For example, a recent paper [66] conducted a similar analysis using pre-trained models on RGB datasets such as ImageNet [67], COCO [68], and BSDS500 [69, 70], demonstrating that fine-tuning only improves training in specific cases. However, this could also be attributed to the fact that the models were trained on domains unrelated to seismic data and with RGB images. Therefore, it was decided to investigate the effect of fine-tuning by initializing the model weights with pre-trained models on seismic datasets. For all fine-tuning experiments, no layers were frozen to allow them to be optimized as needed (full fine-tuning). It is important to note that this training approach was only applied to labeled datasets because the objective of this application is to determine if this technique would provide any benefits in the presence of labels in our target domain.

During the tuning process, various hyperparameters were considered, including batch size, learning rate (static or dynamic), number of epochs, application of different data augmentation configurations, weighting schemes for loss functions in Domain Adaptation, and certain model modifications such as the inclusion of batch normalization and dropout. All models were implemented using the PyTorch framework (version 1.6) and for training was used a GPU Nvidia Volta V100 32GB.

4.3
Inference and Evaluation process

The evaluation process was carried out in two ways depending on whether the dataset was labeled or not. Metrics were used for labeled datasets, while for unlabeled datasets, visual comparisons were made. These evaluations were performed at the section level. To do this, the same technique described by the preprocessing protocol was first used to generate patches, but with a fixed step size of 1 since intermediate sections cannot be ignored during inference. Once the patches were generated, they were inferred using the trained model to obtain their probability map. After this step, the reconstruction of the sections was carried out by averaging the probabilities of overlapping pixels for consecutive patches. Since the quality of the predictions decreased from the center to the edges of the patches, the technique proposed in [9] was considered, where a squared spine window function was applied to merge adjacent patches smoothly. This approach involves generating a 2D matrix, where the intensity of values decreases from the center towards the edges, and then multiplying this matrix with each inferred patch before summing the overlapping regions. The difference between applying and not applying this technique is shown in Figure 4.3 for one of the datasets, confirming that the applied technique delivers a much smoother result.

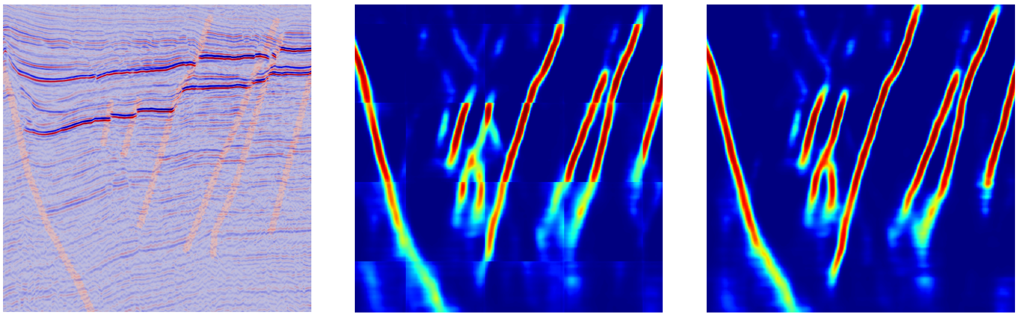

(a) (b) (c)

Figure 4.3: Comparison of Merge Method: (a) Seismic + Fault annotation; (b) Average merge; (c) Smooth merge. Figure adapted from [9].

After the sections have been reconstructed, for the case of labeled datasets, evaluations are conducted using metrics. While there are already common metrics for evaluating models attempting to solve the task of semantic segmentation, such as IOU and F1, these are not the most suitable for our seismic fault detection application. Seismic faults, being thin lines, are frequently not precisely detected by neural networks in the exact pixels where they were labeled. Instead, they are often detected in nearby regions, as illustrated in Figure 4.4. This can result in misleading scores, falsely indicating that the faults were not detected in that particular section. Hence, it is essential to incorporate a tolerance factor to account for such discrepancies. This premise is described in [71], where they proposed the use of more suitable metrics for evaluating seismic fault detections, taking into account a tolerance.

The approach begins by defining a tolerance function $t_f(.)$ that exhibits Gaussian behavior:

$$t_f(d) = e^{-\frac{d^2}{2\sigma^2}} \tag{4-1}$$

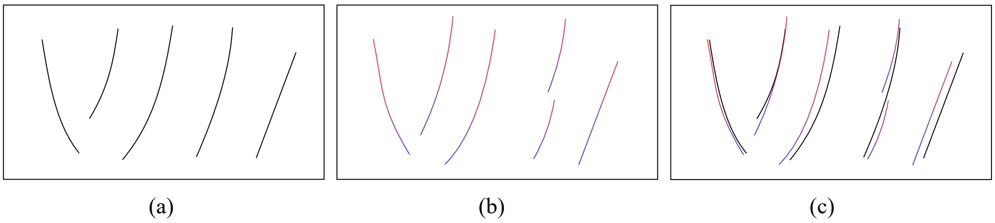


Figure 4.4: Synthetic example of possible difference between ground truth and predictions: (a) ground truth (black curves); (b) prediction (blue-to-red curves); and (c) ground truth/prediction comparison. Figure adapted from [71].

where d is a measure of distance and σ is the parameter that controls tolerance and uncertainty.

To calculate the new Precision and Recall values, taking into consideration this tolerance factor, the process relies on Equations 3-1 and 3-2. It can be observed that the numerator is the same (TP), while the denominators correspond to pixels predicted as positive (N_P) and those belonging to the ground truth (N_{GT}), respectively.

To estimate the numerator TP which represents correct predictions, the distance (D) between predicted positive pixels and the true ones is computed. Subsequently, the function $t_f(d)$ is applied along both axes (m, n). To maintain symmetry between Precision and Recall [71], the calculation of TP varies as follows:

$$P_{new} = \frac{1}{N_P} \sum_{(m,n)\in L_P} t_f(D_{GT}(m,n)) \tag{4-2}$$

$$R_{new} = \frac{1}{N_{GT}} \sum_{(m,n)\in L_{GT}} t_f(D_P(m,n)) \tag{4-3}$$

where:

- L_{GT} : set of pixels assigned to ground truth limits
- $D_{GT}(.)$: distance function from L_{GT}
- L_P : set of pixels assigned to predicted limits
- $D_P(.)$: distance function from L_P

The new F1-Score and IOU are computed using the updated Precision and Recall values in equations 3-3 and 3-4, respectively. These metrics will be used to assess the study. Once the calculations are defined, it is emphasized that a tolerance of $\sigma = 7$ is used, as recommended by the authors on [71], as it has been validated to provide fairer results for seismic fault identification.

From this point on, IOU and F1 will refer to the calculations using this value of tolerance. Although all metrics used in this project could be represented on a scale of $[0-1]$, a scale of 0 to 100% will be employed for better appreciation of their values.

5
Experimental Analysis

This chapter addresses all the details regarding the conducted analyses and the results to be discussed. It begins by describing the involved datasets and additional points to provide a better understanding of the different scenarios in domain adaptation. Next, the different configurations of architectures and hyperparameters used for Supervised and Domain Adaptation training will be specified. The third stage, fine-tuning, applies the same setup as supervised training, so further details are not necessary. Finally, the obtained results for each of the indicated training stages will be discussed.

5.1
Datasets

The proposed method was evaluated on three challenging datasets, both labeled and unlabeled, which are at different scales and distributions based on the analysis of the images. The main characteristics are detailed in Table 5.1 as a summary.

The synthetic dataset presented in [7] has allowed for a sequence of papers and improvements in seismic fault segmentation. For this reason, it was chosen as reference for patch size: 128×128 for all datasets used in this work.

Table 5.1: Main characteristics of used datasets. xl: crossline, il: inline, z: depth.

Dataset	**Data Type**	**Labeling Type**	**Dimensions** ($xl \times il \times z$)
Thebe (TH) [9]	Real	Manual	$1803 \times 3174 \times 1537$
FaultSeg (FS) [7]	Synthetic	Automatic	$128 \times 128 \times 128$
Netherlands F3 (F3) [7]	Real	-	$512 \times 384 \times 128$

5.1.1
Thebe (TH)

This dataset was first used in [9], but its description can be found in [72]. The processed seismic images are derived from a seismic survey conducted in the Thebe Gas Field, located in the Exmouth Plateau of the Carnarvon Basin on the northwest shelf of Australia. The dataset consists of a 3D block with dimensions of $1803[crossline] \times 3174[inline] \times 1537[depth]$, and it was manually labeled (crossline by crossline) by specialists using the *Petrel* software. The authors provided the following split: the first 900 cross-sections for the training set, the next 200 cross-sections for the validation set, and the remaining 703 cross-sections for the test set. After conducting some experiments, it was observed that the manually generated labels using the software were thicker compared to the other datasets. Therefore, patches of size 256×256 were extracted and an additional preprocessing step was performed to reduce them to 128×128 for compatibility with the other datasets during training. Similarly, for inference, a reverse resizing of $128 \rightarrow 256$ was required for the reconstruction of the sections.

5.1.2
FaultSeg (FS)

This dataset was generated synthetically using the method described in [7], aiming to replicate the characteristics of a real seismic dataset. The dataset comprises voxels with dimensions of $128 \times 128 \times 128$, from which 200 blocks were allocated for training and 20 blocks for validation. In this case, the described patch extraction protocol was not required since the dimensions of each section already matched the chosen patch size of 128. Both cross-sections and inline-sections were utilized during training, resulting in a total of $51,200$ samples.

5.1.3
Netherlands F3 (F3)

This dataset is a subset extracted from the widely recognized Netherlands offshore F3 block seismic data, which is made available by the Dutch government through TNO and dGB Earth Sciences. It has dimensions of $512[crossline] \times 384[inline] \times 128[depth]$. The dataset was initially introduced in [7], and subsequently, it has been employed in various studies to assess unsupervised outcomes through visual analysis [20, 41]. For this dataset, a crossline-based split was performed, dividing it into two equal halves to create the training and test sets.

5.2
Supervised Training setup

According to the methodology chapter 4, three architectures were used to perform seismic fault semantic segmentation: UNet, Atrous UNet, and DexiNed, for the three datasets presented.

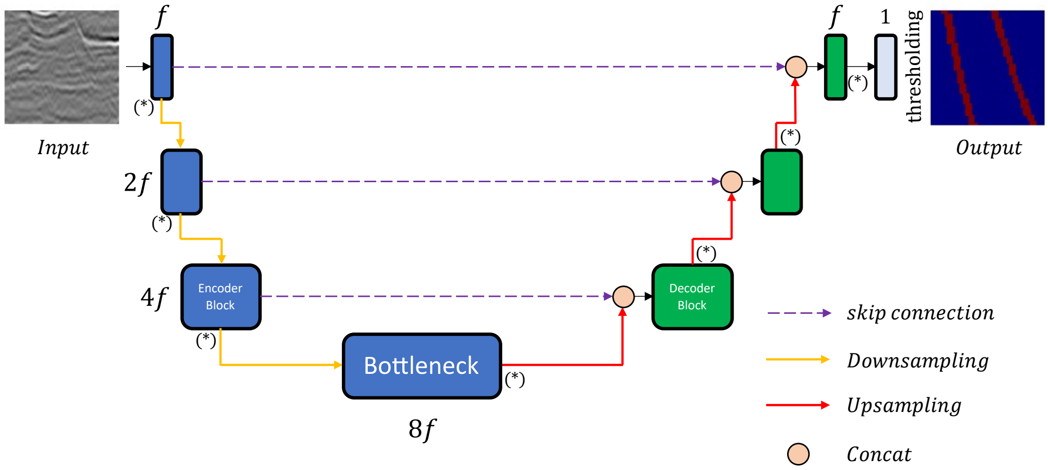


Figure 5.1: UNet-based architectures.

5.2.1
UNet

The first description of the configuration is our custom UNet architecture, since the Atrous UNet is derived from it. It is composed of only three blocks in both the encoder and decoder, as shown in Figure 5.1. These blocks represent two continuous *Conv* layers, where each convolution is accompanied by a *BatchNorm* layer and a *ReLU*(Rectified Linear Unit) activation function (C_n). For more details on the number of filters, stride, and output size at each step, refer to Table 5.2, which shows a smaller version than the one proposed in the original UNet paper [35]. This decision was made because the input image used in this study is 128×128 compared to the 512×512 in the original paper. As part of the experiments, it was also tried using larger filters, but it was not observed significant improvements and instead noticed an increase in the number of parameters and training time per epoch. Therefore, it was remained with this configuration. Additionally, it should be noted that the network was adapted to receive single-channel images since seismic images are gray-scale. Another difference from the original UNet is the use of transposed convolutions *ConvTr* instead of upsampling layers in the decoder.

5.2.2
Atrous UNet

The implementation of this architecture differs from UNet only in the structure of the bottleneck, where a series of consecutive Atrous or Dilated Layers (*AtrConv*) are applied with different dilation ratios (r) that are power-of-2 values. The final output is obtained by summing the outputs from these Atrous Layers. After conducting several experiments, it was determined

Table 5.2: Details of UNet architecture.

	Block level	**Layer**	**Filter**	**Stride**	**Output size**
Input					$128 \times 128 \times 1$
Encoder	Level 1	Conv 1	$3 \times 3/16$	1	$128 \times 128 \times 16$
		Conv 2 (C_1) *	$3 \times 3/16$	1	$128 \times 128 \times 16$
		MaxPooling	2×2	2	$64 \times 64 \times 16$
	Level 2	Conv 3	$3 \times 3/32$	1	$64 \times 64 \times 32$
		Conv 4 (C_2) *	$3 \times 3/32$	1	$64 \times 64 \times 32$
		MaxPooling	2×2	2	$32 \times 32 \times 32$
	Level 3	Conv 5	$3 \times 3/64$	1	$32 \times 32 \times 64$
		Conv 6 (C_3) *	$3 \times 3/64$	1	$32 \times 32 \times 64$
		MaxPooling	2×2	2	$16 \times 16 \times 64$
Bottleneck		Conv 7	$3 \times 3/128$	1	$16 \times 16 \times 128$
		Conv 8 (C_b) *	$3 \times 3/128$	1	$16 \times 16 \times 128$
Decoder	Level 3	ConvTr 3 (CT_3)	$2 \times 2/64$	2	$32 \times 32 \times 64$
		Concat $[C_3 : CT_3]$			$32 \times 32 \times 128$
		Conv 9	$3 \times 3/64$	1	$32 \times 32 \times 64$
		Conv 10 (C_{-3}) *	$3 \times 3/64$	1	$32 \times 32 \times 64$
	Level 2	ConvTr 2 (CT_2)	$2 \times 2/32$	2	$64 \times 64 \times 32$
		Concat $[C_2 : CT_2]$			$64 \times 64 \times 64$
		Conv 11	$3 \times 3/64$	1	$64 \times 64 \times 32$
		Conv 12 (C_{-2}) *	$3 \times 3/64$	1	$64 \times 64 \times 32$
	Level 1	ConvTr 1 (CT_1)	$2 \times 2/32$	2	$128 \times 128 \times 16$
		Concat $[C_1 : CT_1]$			$128 \times 128 \times 32$
		Conv 13	$3 \times 3/64$	1	$128 \times 128 \times 16$
		Conv 14 (C_{-1}) *	$3 \times 3/64$	1	$128 \times 128 \times 16$
Output		Conv 15	$1 \times 1/1$	1	$128 \times 128 \times 1$

(*) branch

that the optimal number of Atrous Layers is 4. To better understand the architecture, it can be used to previous Figure 5.1, where the details of the new bottleneck are provided in Table 5.3.

5.2.3
DexiNed

For the implementation of DexiNed, it was used as it was published in its GitHub repository[3] for the paper [53], with the only modification being the number of input channels changed to 1 since it was originally used for color images. An important aspect to consider about this network is that for the calculation of its loss function, 7 outputs are used, where 6 are from different parts along its architecture, while the seventh is a fusion of the other 6. The architecture is shown in the Figure 5.2, where the outputs of the

---

[3] Code repository: `https://github.com/xavysp/DexiNed`

Upsample Blocks and the Fused edge-map are the mentioned seven outputs used for the loss function. Furthermore, since there is always a significant imbalance between the non-edge and edge ground truth in edge detection tasks, a balanced cross-entropy was also considered, taking into account the frequency between the edge and non-edge ground truth labels.

Table 5.3: Details of bottleneck used on Atrous UNet architecture.

	Layer	**Filter**	**Dilation rate (r)**	**Output size**
Bottleneck	AtrConv 1 (AC_1)	$3 \times 3/128$	1	$16 \times 16 \times 128$
	AtrConv 2 (AC_2)	$3 \times 3/128$	2	$16 \times 16 \times 128$
	AtrConv 3 (AC_3)	$3 \times 3/128$	4	$16 \times 16 \times 128$
	AtrConv 4 (AC_4)	$3 \times 3/128$	8	$16 \times 16 \times 128$
	Sum $[AC_1 : AC_4]$ (C_b)*			$16 \times 16 \times 128$

(*) branch

For training and selection of the best models for each network, different parameter configurations were considered, which are described in Table 5.4. For example, two types of data augmentation were applied: *soft* and *hard*, since this ensured better generalization when inferring on a different dataset than the one used for training. In the case of soft data augmentation, only horizontal and vertical flips were considered, while in the hard data augmentation, more robust versions such as Contrast, Optical Distortion, Sharpen, and Emboss [9] were used. Additionally, different types of loss functions were tested for binary classification. These included the classic Binary Cross-Entropy (BCE) and Weighted BCE, as the problem was highly imbalanced. For DexiNed, its custom loss function was slightly modified to only utilize the last and antepenultimate (second-to-last) outputs out of the original 7 outputs, based on empirical selection. The optimizer used for all experiments was Adam [74], as it is widely used and has shown reliable performance. Finally, different batch sizes were also used, considering the prevention of overfitting from the early stages of training. Therefore, for this supervised training, a maximum of 100 to 150 epochs was set with early stopping based on the loss function and a patience of 10 epochs.

5.3
Domain Adaptation Training setup

In general, for all the UDA techniques used in the study, experiments were conducted prioritizing an equal sample size between the source domain and the target domain. Due to the significant increase in training time and

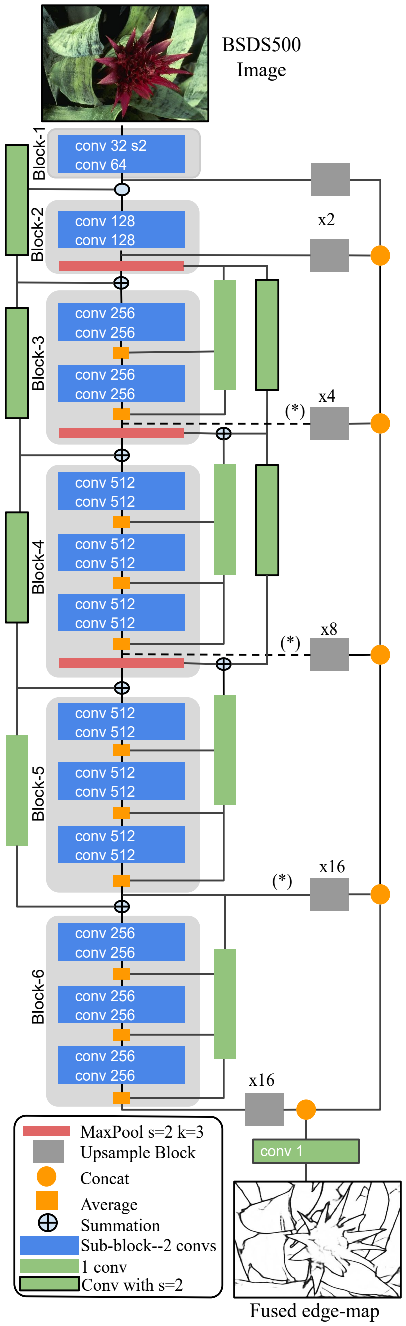


Figure 5.2: DexiNed architecture (adapted from [53]).

memory when loading two datasets simultaneously and passing them through the entire architecture, different sample quantities were tested, such as 48k, 32k, and 16k. It was ensured that reducing the sample size did not have a negative impact on the performance of the models.

For the feature-level UDA techniques, a selection process was first conducted to determine where in the original architecture the branch or multiple branches could be placed for the adaptation process. Tables 5.2 and 5.3 indicate with asterisks the different layers that were considered as

Table 5.4: Hyperparemeter tuning.

Hyperparameter	**Values**
Normalization	z-score (mean=0, std=1)
Data Augmentation	Vertical & Horizontal Flip ShiftScaleRotate, Grid & Optical Distortion, Sharpen, Emboss, Brightness, Contrast and Elastic Transform
Optimizer	Adam
Learning rate	0.0001 or adjustable
Loss function	BCE Weighted BCE
Batch size	8, 16, 32, 64
Epochs (max.)	100 or 150
Early stopping	patience=10

options for extracting features through branches, where the best results were observed in branches close to the bottleneck (C_3, C_b, C_{-3}) in UNet-based architectures. For DexiNed, the branches marked with asterisks in Figure 5.2 were also taken into account, with the center branch (×8 Upsample Block) performing the best.

Different types of architectures were also considered for the branches, starting with a general *Flatten* layer, with or without one or two additional *Dense* layers. Since the size of the feature map was relatively large in many parts, adding a convolutional layer to reduce the number of filters accompanied by max pooling to reduce dimensions was also considered. The overall schemes used for training the feature-level UDA techniques are summarized in Figure 5.3.

Next, the training setups for each of the UDA techniques used will be explained in more detail.

5.3.1
MMD

During the various experiments, different variants of Maximum Mean Discrepancy (MMD) calculations were taken into account. These included the basic version described in [56], the multiple-kernel version known as MK-MMD described in [58], and the version proposed for the Domain Separation Networks (DSN) architecture in [59].

One observation from the various papers applied to semantic segmentation was the possibility of using multiple branches to reduce the domain gap, since minimization of discrepancy would be applied at different levels. There-

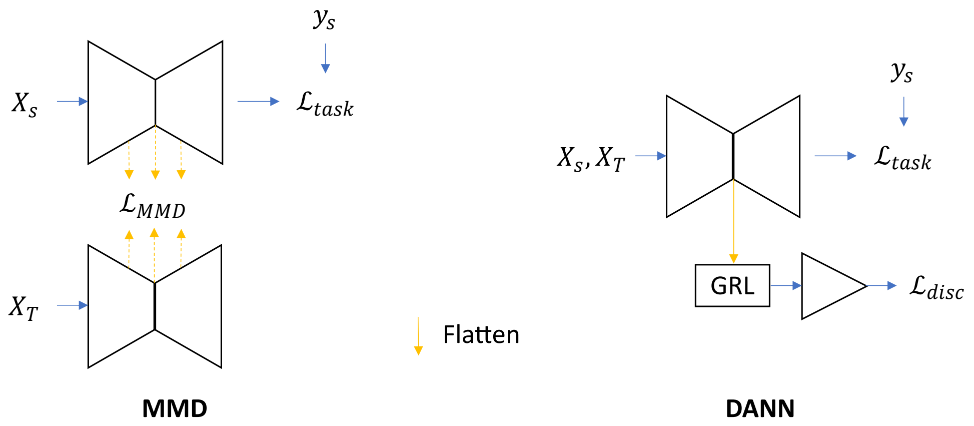


Figure 5.3: Overall schemes used for training the feature-level UDA techniques: MMD and DANN.

fore, our loss function to be optimized or minimized was defined as follows:

$$\mathcal{L}_{MMD}(X_S, X_T) = \sum_{i=1}^{N_{branch}} w_i * MMD_{branch_i}(X_S, X_T) \tag{5-1}$$

$$\mathcal{L}_{total} = \mathcal{L}_{task}(X_S, y_S) + \mathcal{L}_{MMD}(X_S, X_T) \tag{5-2}$$

Here, $\mathcal{L}_{task}$ represents the loss for the source domain or supervised task in semantic segmentation, and w_i denotes the different weights assigned to the MMD loss on branch. It is worth noting that adaptable weights were considered for single-branch applications.

The number of epochs was increased to values between 100 and 150, as some experiments required more time to reach their historical minimum in total loss ($\mathcal{L}_{total}$) for early stopping. Different implementations of learning rates, such as fixed and adjustable, were also considered to prevent overly slow training.

5.3.2
DANN

Unlike MMD, to apply this technique, it was necessary for the branch, or discriminator in this case, to end with a binary classifier that distinguishes between source and target domains. Table 5.5 illustrates an example of how this discriminator needed to be completed in order to generate this new loss and help reduce the domain gap through the adversarial method. It was also necessary to experiment with different values of λ since, according to

Equation 3-6, it serves as a weight for the loss of the discriminator.

As a validation criterion to monitor, the accuracy of the discriminator was also added, where values close to 0.5 indicate that the discriminator can no longer distinguish between samples from the source or target domain, which is the objective. The value of 0.5 is justified by using the same batch size for both domains in each training iteration.

Table 5.5: Implementation details of an example discriminator architecture, in which the output size of the branch is notably large. To address this, convolutional and max-pooling layers were utilized to reduce the output size.

Layer	Filter	Stride	Output size
C_{-3} (*)			$32 \times 32 \times 64$
Conv	$3 \times 3/32$	1	$32 \times 32 \times 32$
MaxPooling	2×2	2	$16 \times 16 \times 32$
Dense + ReLU	100		100
Dense + Sigmoid	1		1

(*) branch

5.3.3
FDA

As mentioned earlier, the application of this technique does not necessitate any changes to the architecture, as it serves as a preliminary step prior to training the neural networks. Specifically, style transfer is applied from the target domain to the source domain, followed by training a model using these new samples along with the source domain labels. After an extensive literature search, two approaches for applying this technique were identified.

The first approach involves defining a fixed set of images from the target domain, from which a random one is chosen to transfer its style for each sample within the source domain batches during training. This was achieved using the Python library Albumentations[4], which already includes an implemented FDA class where the parameter β can also be adjusted. In simple terms, it is applied as a data augmentation technique consistently. This fixed set consisted of 1024 random samples to prevent any bias. No additional samples were used, as it had an impact on the duration of each epoch.

For the second approach, the instructions provided in the original paper [18] were followed. In contrast to the previous approach, here, it is not a fixed set but rather a different batch from the target domain is selected in each training iteration, with the batch size matching that of the source domain. The

[4] Albumentations Python Library: https://albumentations.ai/

use of various batch sizes was also explored to assess their influence on transfer quality, as well as reducing the number of data augmentation techniques. This reduction aimed to avoid confusing the network when understanding the original distribution of the target domain.

In both approaches, for each experiment, a different value of the parameter β was used, which as mentioned above helps us control the magnitude of the spectral signal transfer between domains. Similar to the supervised training stage, it was expected that the loss for the source domain in semantic segmentation would reach its minimum value with early stopping. After conducting multiple experiments, the first approach using β values around 0.1 yielded the best results.

5.4
Results and Discussion

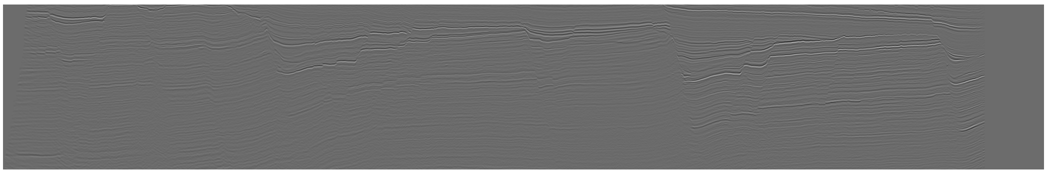
5.4.1
Supervised Training

After training each of the labeled datasets on each network, the results are summarized in Table 5.6. The rows represent the datasets on which the models were trained or the source domain, while the columns represent the datasets on which they were evaluated or the target domain. It is important to note that the values in **bold** indicate the upper bounds, as they were trained and evaluated on the same dataset, representing the ideal scenario where labeled data is available.

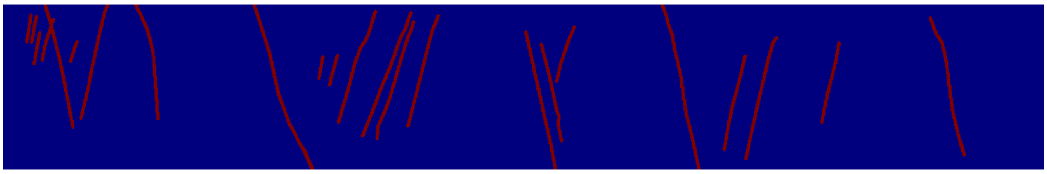
Table 5.6: Results of IOU & F1-score for Supervised stage.

Model	**Target** **Source**	**TH**		**FS**	
		IOU	**F1**	**IOU**	**F1**
UNet	TH	**75.3**	**85.8**	44.8	58.8
	FS	45.7	62.7	**92.5**	**95.4**
DexiNed	TH	**74.7**	**85.5**	66.0	75.4
	FS	37.1	54.0	**93.5**	**96.0**
Atrous UNet	TH	**75.1**	**85.6**	55.5	67.5
	FS	54.1	70.2	**92.8**	**95.6**

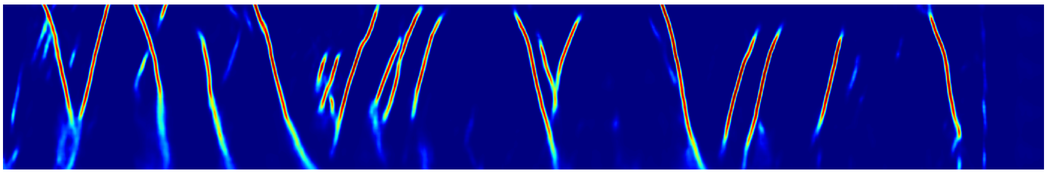
On the other hand, the non-bold values indicate the lower bounds for each pairwise combination. For instance, when inferring on Thebe dataset (TH) using the custom-trained UNet model with FaultSeg dataset (FS), we obtained an $IOU = 45.7\%$ and $F1 = 62.7\%$, which we aim to surpass UDA methods. The analysis of the table also reveals a considerable difference between the distributions of TH and FS, indicating domain shift, as there is a



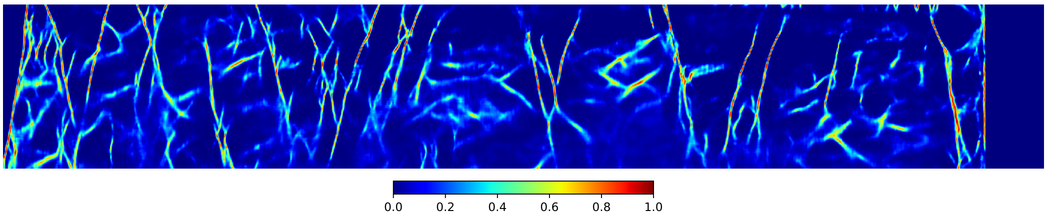
5.4(a): Real seismic section input.



5.4(b): Seismic faults manually identified by experts.



5.4(c): Our best predictions using Atrous UNet trained on TH.



5.4(d): Our best predictions using Atrous UNet trained on FS.

Figure 5.4: The results on the 100th crossline of the test set of TH for the supervised training stage are represented as probabilities ranging from 0 to 1.

significant margin when comparing with their upper bounds. Additionally, it is noteworthy that utilizing the FS dataset contributes to improved inference results across the labeled datasets when using UNet-based architectures.

As a complementary part of this analysis, the visual outcomes for the real datasets are also provided. In the case of the Thebe dataset, the test process focuses on the $100th$ crossline of the test set, as indicated in [9]. Figure 5.4 showcases both the input section and the manually labeled seismic faults. Furthermore, it presents the predictions generated by our best model trained on the same TH dataset (see Figure 5.4(c)), and by using the FS dataset (see Figure 5.4(d)), as it demonstrated the best generalization according to the metrics in Table 5.6. From this last sub-figure, it can be observed that the model successfully recognizes several seismic faults, but it suffers from the common issue of generating numerous artifacts.

Given that we have observed significantly finer seismic faults in comparison to those identified in the benchmark study [9], which primarily employs edge detection-based architectures such as HED, we have conducted an analysis using the same set of edge detection metrics. Table 5.7 showcases a comparison of AP, ODS, and OIS metrics, highlighting the exceptional and comparable performance achieved through the utilization of UNet-based architectures in contrast to the top-performing models from the benchmark study. Moreover, our most proficient model surpassed these benchmark models in all metrics, indicating the establishment of a new State-of-the-Art on the TH dataset. These findings emphasize the significance of not only considering the choice of the model type but also carefully reviewing the data preprocessing and training protocols.

5.4.2
Domain Adaptation Training

5.4.2.1
Source: FS | Target: TH

Within our first DA analysis, the FS and TH datasets were chosen as the source and target domains, respectively, as they fulfilled the synthetic-real condition. The detailed results of the best metrics among all the conducted experiments are shown in Table 5.8, where different behaviors can be observed regarding the utilized UDA techniques and architectures. It is worth noting that the values in the table represent a single execution per configuration due to the extensive time required for training and tuning, making it impractical to conduct multiple executions.

Table 5.7: Results using edge detection metrics for Thebe dataset.

Model	Augmentation	Step size	AP	ODS	OIS
HED [9]	-	1	82.3	80.6	81.1
	Elastic Transform	1	80.1	84.2	84.7
	Hard	10	81.3	83.6	84.1
UNet	Hard	3	89.3	83.2	83.8
DexiNed			90.7	85.0	85.6
Atrous UNet			**91.3**	**86.1**	**86.5**

Starting with the baseline, which represents the case where no adaptation technique is applied, it is evident that the Atrous UNet achieved the best generalization, while DexiNed performed the worst. This discrepancy may be attributed to the deeper architecture of DexiNed, which could have led to overfitting on the training and validation set of the source domain (FS).

Moving on, the application of MMD reveals that it delivered the best result when used with the UNet network, but it did not yield similar improvements when applied to the other two networks. In the case of DexiNed, this could be due to its multiple outputs for calculating the loss function, making it less affected by one or more branches responsible for MMD reduction. As for Atrous UNet, its already strong baseline performance and the relative simplicity of the technique might hinder its further improvement.

A similar scenario is observed in the application of DANN, where improvements ranging from 3% to 4% on average are seen for the UNet and DexiNed networks, but they still struggle to surpass the Atrous UNet baseline. These comparable outcomes to MMD might be attributed to the fact that both methods belong to the feature-level adaptation type of UDA, which has shown challenges when applied to imbalanced data [37].

Lastly, the results obtained by applying FDA show improvements for each network, with DexiNed experiencing an enhancement of up to 13% on average, while UNet and Atrous UNet achieve improvements of 6% and 3%, respectively. This increment can be justified by the fact that FDA focuses more on the aspect of input images (image-level adaptation), which is more relevant in the context of semantic segmentation.

Overall, it could be concluded that the best result obtained by applying any UDA technique (Atrous UNet + FDA) achieves an increase of nearly 20% compared to the baseline with the lowest scores (DexiNed).

Similarly to the previous stage, a visual analysis is conducted using the predicted faults by the architectures that yielded the best scores for each UDA technique, according to Table 5.8. This analysis is in comparison to the results obtained without employing any adaptation technique, utilizing the Atrous UNet architecture, which also proved to be the most effective in this particular scenario. To better distinguish the introduced improvements, a zoom-in is conducted on the 100th cross-section of the test set. This examination displays both the probability map and the identified faults with probabilities exceeding 0.5, as depicted in Figure 5.5. Initially, it can be observed that the application of DA techniques reduces the occurrence of artifacts that might result in false positives. Additionally, employing DA leads to the identification of more continuous faults, thereby instilling greater confidence in the identification and quantification of seismic faults throughout the sections.

Regarding these observed improvements, it can be confirmed that the technique with the highest score, Atrous UNet + FDA, is also accompanied by a reduced amount of noise or artifacts and more continuous faults. Nevertheless, it is worth noting that the visual results obtained with Atrous UNet

Table 5.8: Results using FaultSeg (FS) as source domain and Thebe (TH) as target domain. The values in parentheses indicate a comparison of the application of each DA technique with respect to the scenario without DA (No-adaptation).

	Model	**IOU**	**F1**
Supervised	UNet	75.3	85.8
	DexiNed	74.7	85.5
	Atrous UNet	75.1	85.6
No-adaptation	UNet	45.7	62.7
	DexiNed	37.1	54.0
	Atrous UNet	54.1	70.2
MMD	UNet	54.3 (+8.6)	70.3 (+7.6)
	DexiNed	32.9 (-4.2)	49.5 (-4.5)
	Atrous UNet	53.7 (-0.4)	69.8 (-0.4)
DANN	UNet	49.6 (+3.9)	66.2 (+3.5)
	DexiNed	41.0 (+3.9)	58.1 (+4.1)
	Atrous UNet	53.6 (-0.5)	69.7 (-0.5)
FDA	UNet	51.9 (+6.2)	68.3 (+5.6)
	DexiNed	50.5 (+13.4)	67.0 (+13.0)
	Atrous UNet	57.0 (+2.9)	72.6 (+2.4)

+ DANN identify faults of comparable quality to those mentioned previously.

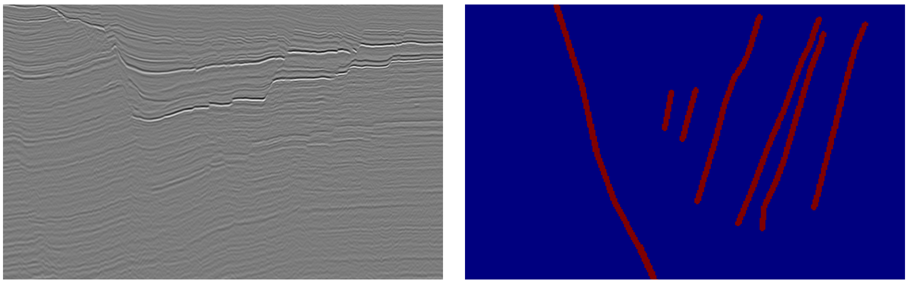
5.4.2.2
Source: FS | Target: F3

Just like in Thebe dataset, the initial step involved direct inference using pre-trained models on FS to establish a baseline. Then, training for each of the three architectures was conducted using the Fourier Domain Adaptation (FDA) technique, which yielded the best results in the previous analysis.

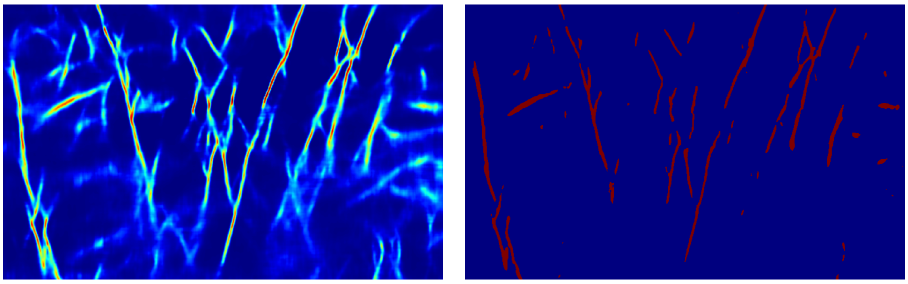
Since this dataset lacks labels, the previously discussed metrics could not be calculated, and evaluations were carried out visually. First, inferences were made on the 30th crossline and 30th inline sections, which are presumed to be two of the three sections visualized in the results of [7] for F3. The UNet network yielded the best predictions without applying domain adaptation, while surprisingly, for the FDA application, the DexiNed architecture exhibited more continuous seismic faults. These predictions are illustrated in Figure 5.6.

The primary observation is that fault identification was easier in the crossline, both with and without domain adaptation, while there was more difficulty in the inline when domain adaptation was not applied. Although more faults were identified in the crossline, it also resulted in false positives.

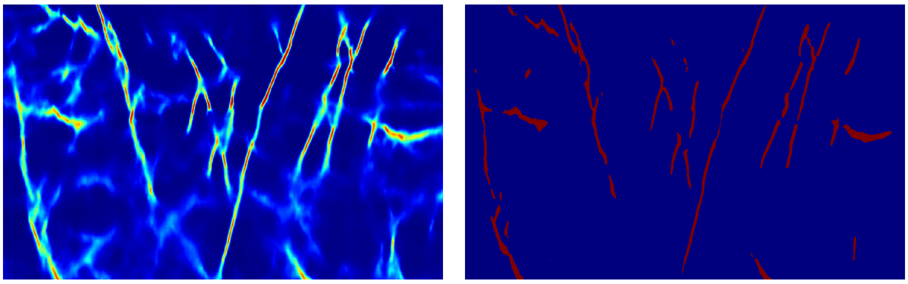
For a more direct comparison with [7], predictions were also made



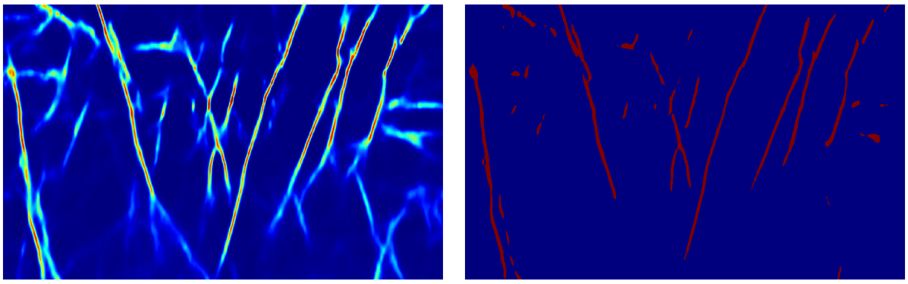
5.5(a): Real seismic section input and identified faults.



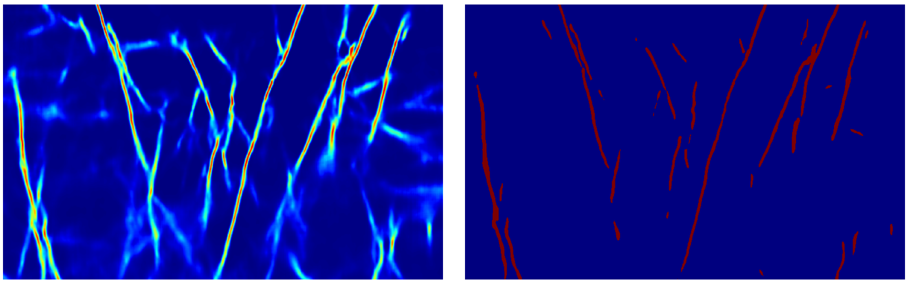
5.5(b): No-adaptation (Atrous UNet).



5.5(c): MMD (UNet).

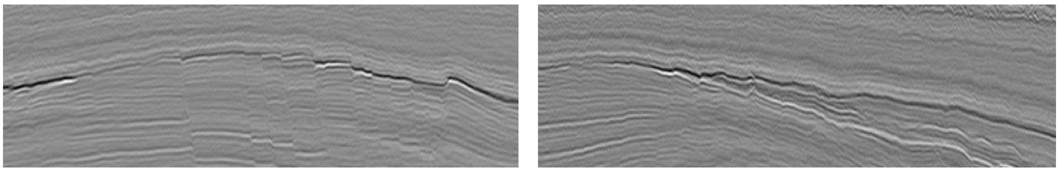


5.5(d): DANN (Atrous UNet).

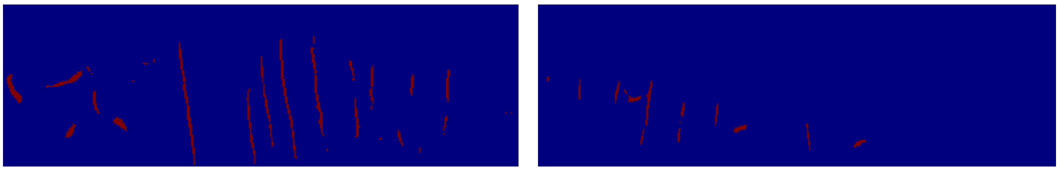


5.5(e): FDA (Atrous UNet).

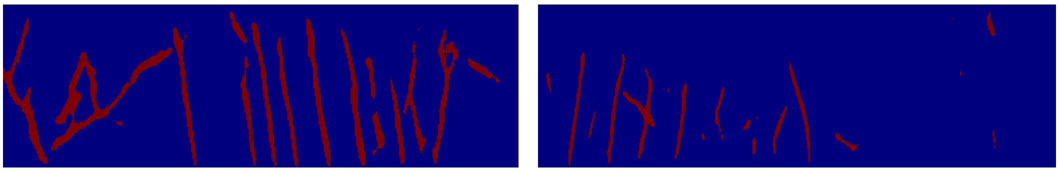
Figure 5.5: The results on part of the 100th crossline of the test set of TH for the DA training stage are represented as probabilities ranging from 0 to 1, and predicted faults (values greater than 0.5).



5.6(a): Real seismic cross-section and inline-section input.



5.6(b): No-adaptation (UNet).



5.6(c): FDA (DexiNed).

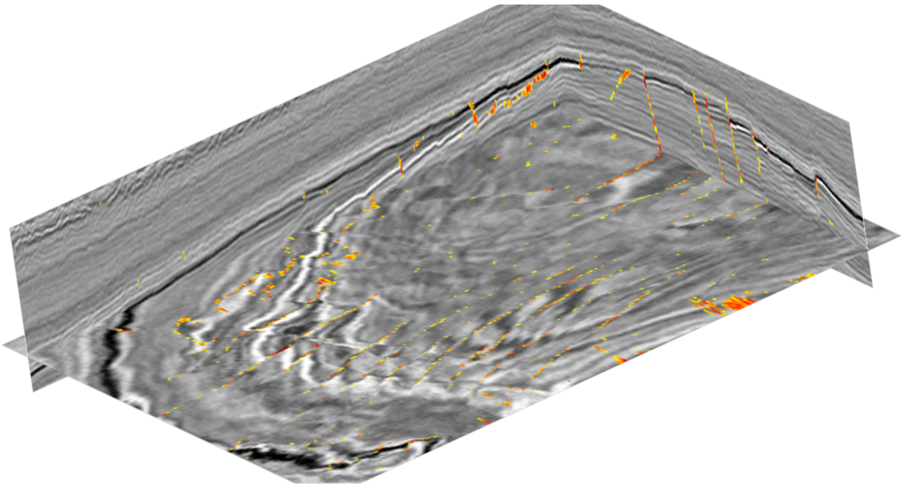
Figure 5.6: Results on the 30th crossline (left) and 30th inline (right) of F3 dataset for the DA training stage are represented as predicted faults.

throughout the entire block along both crossline and inline axes, and the average of these predictions was taken as the final estimation. The results are represented by three sections of the general block, which were plotted in 3D, as shown in Figure 5.7. From this image, it is evident that applying domain adaptation makes the existing faults in the 3D block much more visible.

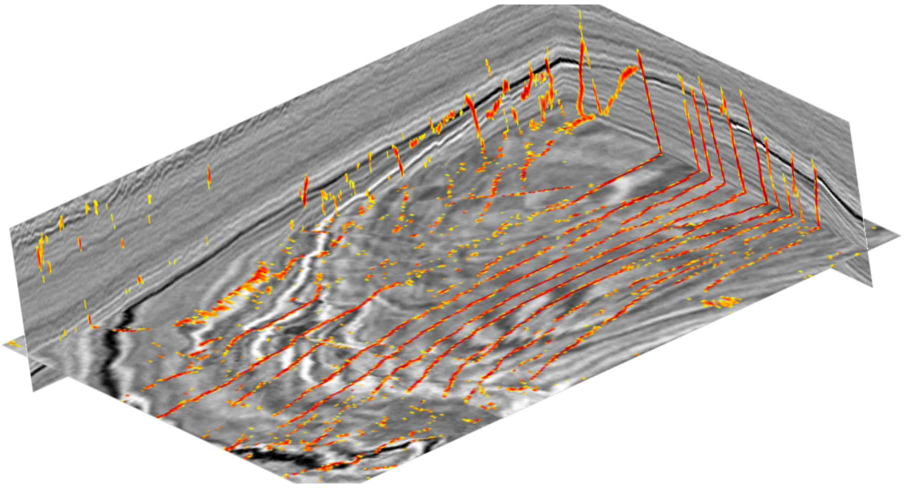
5.4.3
Fine-tuning Training

Finally, as part of the comparative analysis, the use of pre-trained models on another seismic dataset and its potential for significant improvement will be examined. This study serves also as a complementary component to [66], which presents similar tests but utilizing pre-trained models from well-known datasets like ImageNet, BSDS500, and COCO. In this case, the experiments were constrained to the utilization of solely the labeled TH and FS datasets with the three previously mentioned architectures to explore whether pre-trained models with seismic data offer a more effective initialization compared to the previously described datasets.

From Table 5.9 and Table 5.10, similar conclusions can be drawn as stated in [66], where some cases showed improvement, while in others, performance was even reduced. For instance, when using the Atrous UNet architecture, there is a nearly insignificant gain from fine-tuning, while no consistent improvement



5.7(a): No-adaptation (UNet)



5.7(b): FDA (DexiNed)

Figure 5.7: Results on the 30th crossline, 30th inline and 100th depth of F3 dataset for the DA training stage are represented as probabilities ranging from 0.5 to 1.

or deterioration is observed in the other two networks. However, it is worth noting that the number of epochs to reach the local minimum was significantly reduced when fine-tuning was applied, resulting in a shorter training time.

As part of further analysis, it would be beneficial to test with different data proportions to better observe the advantageous qualities of applying fine-tuning.

Table 5.9: Results using fine-tuning with pre-trained model on FS to start the weights of the training for TH.

	$FS \rightarrow TH$			
	Model	**IOU**	**F1**	**# epochs**
Supervised	UNet	75.3	85.8	39
	DexiNed	74.7	85.5	24
	Atrous UNet	75.1	85.6	33
Fine-tuning	UNet	76.0 (+0.7)	86.3 (+0.5)	14
	DexiNed	74.3 (-0.4)	85.2 (3)	13
	Atrous UNet	76.2 (+1.1)	86.4 (+0.8)	19

Table 5.10: Results using fine-tuning with pre-trained model on TH to start the weights of the training for FS.

	$TH \rightarrow FS$			
	Model	**IOU**	**F1**	**# epochs**
Supervised	UNet	92.5	95.4	66
	DexiNed	93.5	96.0	26
	Atrous UNet	92.8	95.6	42
Fine-tuning	UNet	92.1 (-0.4)	95.2 (-0.2)	39
	DexiNed	93.6 (+0.1)	96.1 (+0.1)	31
	Atrous UNet	93.1 (+0.3)	95.7 (+0.1)	33

6
Conclusions and Future Works

In general, the identification of seismic faults aids in fluid flow and serves as boundaries for hydrocarbon reservoir management, significantly influencing the Oil & Gas industry. This process is known to be time-consuming and manual. Among the current methods that enable more accurate automation of fault identification, DL-based approaches have shown promise. However, these methods also face challenges such as the need for a substantial amount of labeled samples to train the model in identifying seismic faults.

In this dissertation, it was proposed the use of techniques of Unsupervised Domain Adaptation (UDA). UDA offers the advantage of transferring knowledge acquired from a source domain to a target domain without relying on labeled samples in the target domain. The three techniques examined were MMD, DANN and FDA, all of which involve two types of domain adaptation: feature-level and image-level. This allowed for a more comprehensive analysis through comparison between the techniques.

6.1
Conclusions

As a first conclusion, it can be stated that the difference between the datasets becomes evident when direct inference is applied without any domain adaptation technique. However, this difference is not pronounced as values of up to 54.16% in IOU and 70.19% in F1 are achieved, as shown in Table 5.8.

Furthermore, it was demonstrated that regardless of whether the network is dedicated to edge detection or not, good metrics can be achieved by simply altering the preprocessing and training approach, as detailed in Table 5.7, where the best results of [9] are surpassed.

Regarding the use of UDA techniques, an improvement was achieved both in terms of metrics (up to 13% on average) and visual quality for the identification of seismic faults with less noise and more continuity in Thebe dataset. It is also observed that the application of FDA technique, being an image-level adaptation approach, easily surpasses feature-level adaptation techniques like MMD and DANN, although the use of a suitable architecture like Atrous UNet also has an impact.

A similar situation was observed in the F3 dataset, where seismic faults were predicted with higher confidence when using FDA, along with the DexiNed architecture.

Finally, it can be observed that fine-tuning does not yield substantial advantages in this particular application, except for the reduction in training time when utilizing all the samples from labeled datasets.

6.2
Future Works

Given that there is evidence indicating that the implementation of 3D input networks has led to improved result consistency across all three axes [7], it should be considered to replicate the study using networks that accept these inputs, such as the classic UNet3D or its improved versions like ResUNet [36] and UNet++ [12].

It should also be considered to attempt using 2.5D inputs as defined in [41], which proposes the use of multiple consecutive sections to estimate the central section to avoid computational cost and maintain smoother inferences.

As observed, image-level-based UDA techniques have shown better results for this application, so the use of methods within this category, such as CyCADA [63], which combines both approaches, could potentially improve the outcomes.

Further analysis could also be conducted when working with different seismic frequencies, as well as optimizing the step size.

For a more focused evaluation on fine-tuning, training with different sample quantities and analyzing the dependency of this technique on data availability could provide valuable insights.

Lastly, exploring the combination of multiple UDA techniques during training to assess if certain methods complement or hinder each other in this specific application could also be a beneficial direction for future investigation.

Bibliography

[1] HAMZEH, A.. **Machine Learning Methods in Fault Detection**. PhD thesis, Politecnico di Torino, 2021.

[2] BAHORICH, M.; FARMER, S.. **3-d seismic discontinuity for faults and stratigraphic features: The coherence cube**. The leading edge, 14(10):1053–1058, 1995.

[3] ROBERTS, A.. **Curvature attributes and their application to 3 d interpreted horizons**. First break, 19(2):85–100, 2001.

[4] EICHKITZ, C. G.; AMTMANN, J. ; SCHREILECHNER, M. G.. **Calculation of grey level co-occurrence matrix-based seismic attributes in three dimensions**. Computers & Geosciences, 60:176–183, 2013.

[5] DI, H.; SHAFIQ, M. A. ; ALREGIB, G.. **Seismic-fault detection based on multiattribute support vector machine analysis**. In: 2017 SEG INTERNATIONAL EXPOSITION AND ANNUAL MEETING. OnePetro, 2017.

[6] LONG, Z.; ALAUDAH, Y.; QURESHI, M. A.; HU, Y.; WANG, Z.; ALFARRAJ, M.; ALREGIB, G.; AMIN, A.; DERICHE, M.; AL-DHARRAB, S. ; OTHERS. **A comparative study of texture attributes for characterizing subsurface structures in seismic volumes**. Interpretation, 6(4):T1055–T1066, 2018.

[7] WU, X.; LIANG, L.; SHI, Y. ; FOMEL, S.. **Faultseg3d: Using synthetic data sets to train an end-to-end convolutional neural network for 3d seismic fault segmentation**. Geophysics, 84(3):IM35–IM45, 2019.

[8] DI, H.; WANG, Z. ; ALREGIB, G.. **Seismic fault detection from post-stack amplitude by convolutional neural networks**. In: 80TH EAGE CONFERENCE AND EXHIBITION 2018, p. 1–5, 2018.

[9] AN, Y.; GUO, J.; YE, Q.; CHILDS, C.; WALSH, J. ; DONG, R.. **Deep convolutional neural network for automatic fault recognition from 3d seismic datasets**. Computers & Geosciences, 153:104776, 2021.

[10] ALFARHAN, M.; DERICHE, M.; MAALEJ, A.; ALREGIB, G. ; ALMARZOUQI, H.. **Multiple events detection in seismic structures using a novel u-net variant.** In: 2020 IEEE INTERNATIONAL CONFERENCE ON IMAGE PROCESSING (ICIP), p. 2900–2904. IEEE, 2020.

[11] ALFARHAN, M.; DERICHE, M. ; MAALEJ, A.. **Robust concurrent detection of salt domes and faults in seismic surveys using an improved unet architecture.** IEEE Access, 10:39424–39435, 2020.

[12] YANG, D.; CAI, Y.; HU, G.; YAO, X. ; ZOU, W.. **Seismic fault detection based on 3d unet++ model.** In: SEG INTERNATIONAL EXPOSITION AND ANNUAL MEETING. OnePetro, 2020.

[13] NASIM, M. Q.; MAITI, T.; SRIVASTAVA, A.; SINGH, T. ; MEI, J.. **Seismic facies analysis: a deep domain adaptation approach.** IEEE Transactions on Geoscience and Remote Sensing, 60:1–16, 2022.

[14] CUNHA, A.; POCHET, A.; LOPES, H. ; GATTASS, M.. **Seismic fault detection in real data using transfer learning from a convolutional neural network pre-trained with synthetic seismic data.** Computers & Geosciences, 135:104344, 2020.

[15] YAN, Z.; ZHANG, Z. ; LIU, S.. **Improving performance of seismic fault detection by fine-tuning the convolutional neural network pre-trained with synthetic samples.** Energies, 14(12), 2021.

[16] ZHANG, Z.; YAN, Z.; JING, J.; GU, H. ; LI, H.. **Generating paired seismic training data with cycle-consistent adversarial networks.** Remote Sensing, 15(1), 2023.

[17] HONG, J.; YU, S. C.-H. ; CHEN, W.. **Unsupervised domain adaptation for cross-modality liver segmentation via joint adversarial learning and self-learning.** Applied Soft Computing, 121:108729, 2022.

[18] YANG, Y.; SOATTO, S.. **Fda: Fourier domain adaptation for semantic segmentation.** In: PROCEEDINGS OF THE IEEE/CVF CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION (CVPR), June 2020.

[19] HOFFMAN, J.; WANG, D.; YU, F. ; DARRELL, T.. **Fcns in the wild: Pixel-level adversarial and constraint-based adaptation.** arXiv preprint arXiv:1612.02649, 2016.

[20] ZHOU, R.; YAO, X.; HU, G. ; YU, F.. **Learning from unlabelled real seismic data: Fault detection based on transfer learning**. Geophysical Prospecting, 69(6):1218–1234, 2021.

[21] ZHOU, R.; YAO, X.; WANG, Y.; HU, G. ; YU, F.. **Seismic fault detection with progressive transfer learning**. Acta Geophysica, 69:2187–2203, 2021.

[22] MARFURT, K. J.; KIRLIN, R. L.; FARMER, S. L. ; BAHORICH, M. S.. **3-d seismic attributes using a semblance-based coherency algorithm**. Geophysics, 63(4):1150–1165, 1998.

[23] VAN BEMMEL, P. P.; PEPPER, R. E.. **Seismic signal processing method and apparatus for generating a cube of variance values**, Nov. 21 2000. US Patent 6,151,555.

[24] RANDEN, T.; PEDERSEN, S. I. ; SØNNELAND, L.. **Automatic extraction of fault surfaces from three-dimensional seismic data**. In: 2001 SEG ANNUAL MEETING. OnePetro, 2001.

[25] GAO, D.. **Integrating 3d seismic curvature and curvature gradient attributes for fracture characterization: Methodologies and interpretational implications**. Geophysics, 78(2):O21–O31, 2013.

[26] HEGAZY, T.; ALREGIB, G.. **Texture attributes for detecting salt bodies in seismic data**. In: 2014 SEG ANNUAL MEETING. OnePetro, 2014.

[27] WANG, Z.; HEGAZY, T.; LONG, Z. ; ALREGIB, G.. **Noise-robust detection and tracking of salt domes in postmigrated volumes using texture, tensors, and subspace learning**. Geophysics, 80(6):WD101–WD116, 2015.

[28] OJALA, T.; PIETIKAINEN, M. ; MAENPAA, T.. **Multiresolution gray-scale and rotation invariant texture classification with local binary patterns**. IEEE Transactions on pattern analysis and machine intelligence, 24(7):971–987, 2002.

[29] HUSSEIN, M.; STEWART, R. ; WU, J.. **Unsupervised machine learning techniques for subtle fault detection**. In: 82ND EAGE ANNUAL CONFERENCE & EXHIBITION. EUROPEAN ASSOCIATION OF GEOSCIENTISTS & ENGINEERS, p. 1–5, 2020.

[30] LECUN, Y.; BOTTOU, L.; BENGIO, Y. ; HAFFNER, P.. **Gradient-based learning applied to document recognition**. Proceedings of the IEEE, 86(11):2278–2324, 1998.

[31] XIONG, W.; JI, X.; MA, Y.; WANG, Y.; ALBINHASSAN, N. M.; ALI, M. N. ; LUO, Y.. **Seismic fault detection with convolutional neural network**. Geophysics, 83(5):O97–O103, 2018.

[32] HALE, D.. **Methods to compute fault images, extract fault surfaces, and estimate fault throws from 3d seismic images**. Geophysics, 78(2):O33–O43, 2013.

[33] POCHET, A.; DINIZ, P. H. B.; LOPES, H. ; GATTASS, M.. **Seismic fault detection using convolutional neural networks trained on synthetic poststacked amplitude maps**. IEEE Geoscience and Remote Sensing Letters, 16(3):352–356, 2019.

[34] **Convolutional neural networks for fault interpretation in seismic images**, volumen All Days de **SEG International Exposition and Annual Meeting**, 10 2018.

[35] RONNEBERGER, O.; FISCHER, P. ; BROX, T.. **U-net: Convolutional networks for biomedical image segmentation**. In: MEDICAL IMAGE COMPUTING AND COMPUTER-ASSISTED INTERVENTION–MICCAI 2015: 18TH INTERNATIONAL CONFERENCE, MUNICH, GERMANY, OCTOBER 5-9, 2015, PROCEEDINGS, PART III 18, p. 234–241. Springer, 2015.

[36] LIU, N.; HE, T.; TIAN, Y.; WU, B.; GAO, J. ; XU, Z.. **Common-azimuth seismic data fault analysis using residual unet**. Interpretation, 8(3):SM25–SM37, 2020.

[37] VEGA, P. J. S.. **Deep Learning-Based Domain Adaptation for Change Detection in Tropical Forests**. PhD thesis, PUC-Rio Rio de Janeiro, Brazil, 2021.

[38] CSURKA, G.; VOLPI, R. ; CHIDLOVSKII, B.. **Unsupervised domain adaptation for semantic image segmentation: a comprehensive survey**. arXiv preprint arXiv:2112.03241, 2021.

[39] GANIN, Y.; USTINOVA, E.; AJAKAN, H.; GERMAIN, P.; LAROCHELLE, H.; LAVIOLETTE, F.; MARCHAND, M. ; LEMPITSKY, V.. **Domain-adversarial training of neural networks**. The journal of machine learning research, 17(1):2096–2030, 2016.

[40] ZHU, J.-Y.; PARK, T.; ISOLA, P. ; EFROS, A. A.. **Unpaired image-to-image translation using cycle-consistent adversarial networks.** In: PROCEEDINGS OF THE IEEE INTERNATIONAL CONFERENCE ON COMPUTER VISION (ICCV), Oct 2017.

[41] TANG, Z.; WU, B.; WU, W. ; MA, D.. **Fault detection via 2.5d transformer u-net with seismic data pre-processing.** Remote Sensing, 15(4), 2023.

[42] YU, S.; MA, J.. **Deep learning for geophysics: Current and future trends.** Reviews of Geophysics, 59(3):e2021RG000742, 2021. e2021RG000742 2021RG000742.

[43] BIDDLE, KEVIN T. AND WIELCHOWSKY, CHARLES C.. **Hydrocarbon traps.** In: THE PETROLEUM SYSTEM—FROM SOURCE TO TRAP, chapter 13. American Association of Petroleum Geologists, 1994.

[44] BRYANT, WILLIAM A.. **"Encyclopedia of Natural Hazards**, chapter Fault, p. 317–321. Springer Netherlands, 2013.

[45] GOODFELLOW, I.; BENGIO, Y. ; COURVILLE, A.. **Deep learning.** MIT press, 2016.

[46] KRIZHEVSKY, A.; SUTSKEVER, I. ; HINTON, G. E.. **Imagenet classification with deep convolutional neural networks.** Commun. ACM, 60(6):84–90, May 2017.

[47] LONG, J.; SHELHAMER, E. ; DARRELL, T.. **Fully convolutional networks for semantic segmentation.** In: PROCEEDINGS OF THE IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION, p. 3431–3440, 2015.

[48] BADRINARAYANAN, V.; KENDALL, A. ; CIPOLLA, R.. **Segnet: A deep convolutional encoder-decoder architecture for image segmentation.** IEEE Transactions on Pattern Analysis and Machine Intelligence, 39(12):2481–2495, 2017.

[49] CAMPOS TRINIDAD, M. J.; ARAUCO CANCHUMUNI, S. W. ; CAVALCANTI PACHECO, M. A.. **Towards a benchmark for sedimentary facies classification: Applied to the netherlands f3 block.** In: INFORMATION MANAGEMENT AND BIG DATA: 7TH ANNUAL INTERNATIONAL CONFERENCE, SIMBIG 2020, LIMA, PERU, OCTOBER 1–3, 2020, PROCEEDINGS, p. 211–222. Springer, 2021.

[50] PIAO, S.; LIU, J.. **Accuracy improvement of UNet based on dilated convolution**. Journal of Physics: Conference Series, 1345:052066, nov 2019.

[51] CHEN, L.-C.; PAPANDREOU, G.; KOKKINOS, I.; MURPHY, K. ; YUILLE, A. L.. **Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs**. IEEE transactions on pattern analysis and machine intelligence, 40(4):834–848, 2017.

[52] XIE, S.; TU, Z.. **Holistically-nested edge detection**. In: 2015 IEEE INTERNATIONAL CONFERENCE ON COMPUTER VISION (ICCV), p. 1395–1403, 2015.

[53] SORIA, X.; SAPPA, A.; HUMANANTE, P. ; AKBARINIA, A.. **Dense extreme inception network for edge detection**. Pattern Recognition, p. 109461, 2023.

[54] SUN, R.; LEI, T.; CHEN, Q.; WANG, Z.; DU, X.; ZHAO, W. ; NANDI, A. K.. **Survey of image edge detection**. Frontiers in Signal Processing, 2, 2022.

[55] SHI, Y.; YING, X. ; YANG, J.. **Deep unsupervised domain adaptation with time series sensor data: A survey**. Sensors, 22(15), 2022.

[56] TZENG, E.; HOFFMAN, J.; ZHANG, N.; SAENKO, K. ; DARRELL, T.. **Deep domain confusion: Maximizing for domain invariance**. arXiv preprint arXiv:1412.3474, 2014.

[57] SUN, B.; FENG, J. ; SAENKO, K.. **Correlation alignment for unsupervised domain adaptation**. Domain adaptation in computer vision applications, p. 153–171, 2017.

[58] LONG, M.; CAO, Y.; WANG, J. ; JORDAN, M.. **Learning transferable features with deep adaptation networks**. In: INTERNATIONAL CONFERENCE ON MACHINE LEARNING, p. 97–105. PMLR, 2015.

[59] BOUSMALIS, K.; TRIGEORGIS, G.; SILBERMAN, N.; KRISHNAN, D. ; ERHAN, D.. **Domain separation networks**. Advances in neural information processing systems, 29, 2016.

[60] ZHANG, Y.; ZHANG, Y.; WEI, Y.; BAI, K.; SONG, Y. ; YANG, Q.. **Fisher Deep Domain Adaptation**, p. 469–477. Society for Industrial and Applied Mathematics, 2020.

[61] GOODFELLOW, I.; POUGET-ABADIE, J.; MIRZA, M.; XU, B.; WARDE-FARLEY, D.; OZAIR, S.; COURVILLE, A. ; BENGIO, Y.. **Generative adversarial nets**. In: Ghahramani, Z.; Welling, M.; Cortes, C.; Lawrence, N. ; Weinberger, K., editors, ADVANCES IN NEURAL INFORMATION PROCESSING SYSTEMS, volumen 27. Curran Associates, Inc., 2014.

[62] CORDTS, M.; OMRAN, M.; RAMOS, S.; REHFELD, T.; ENZWEILER, M.; BENENSON, R.; FRANKE, U.; ROTH, S. ; SCHIELE, B.. **The cityscapes dataset for semantic urban scene understanding**. In: PROCEEDINGS OF THE IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION (CVPR), June 2016.

[63] HOFFMAN, J.; TZENG, E.; PARK, T.; ZHU, J.-Y.; ISOLA, P.; SAENKO, K.; EFROS, A. ; DARRELL, T.. **CyCADA: Cycle-consistent adversarial domain adaptation**. In: Dy, J.; Krause, A., editors, PROCEEDINGS OF THE 35TH INTERNATIONAL CONFERENCE ON MACHINE LEARNING, volumen 80 de **Proceedings of Machine Learning Research**, p. 1989–1998. PMLR, 10–15 Jul 2018.

[64] DOU, Y.; LI, K.; ZHU, J.; LI, X. ; XI, Y.. **Attention-based 3-d seismic fault segmentation training by a few 2-d slice labels**. IEEE Transactions on Geoscience and Remote Sensing, 60:1–15, 2021.

[65] NOA, J.; SOTO, P. J.; G, A. O. P. C.; WITTICH, D.; FEITOSA, R. Q. ; ROTTENSTEINER, F.. **Adversarial discriminative domain adaptation for deforestation detection**. ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, -3-2021:151–158, 2021.

[66] AN, Y.; DONG, R.. **Understanding the effect of different prior knowledge on cnn fault interpreter**. IEEE Access, 11:15058–15068, 2023.

[67] DENG, J.; DONG, W.; SOCHER, R.; LI, L.-J.; LI, K. ; FEI-FEI, L.. **Imagenet: A large-scale hierarchical image database**. In: 2009 IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION, p. 248–255, 2009.

[68] LIN, T.-Y.; MAIRE, M.; BELONGIE, S.; HAYS, J.; PERONA, P.; RAMANAN, D.; DOLLÁR, P. ; ZITNICK, C. L.. **Microsoft coco: Common objects in context**. In: Fleet, D.; Pajdla, T.; Schiele, B. ; Tuytelaars, T., editors, COMPUTER VISION – ECCV 2014, p. 740–755, Cham, 2014. Springer International Publishing.

[69] MARTIN, D.; FOWLKES, C.; TAL, D. ; MALIK, J.. **A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics**. In: PROCEEDINGS EIGHTH IEEE INTERNATIONAL CONFERENCE ON COMPUTER VISION. ICCV 2001, volumen 2, p. 416–423 vol.2, 2001.

[70] ARBELÁEZ, P.; MAIRE, M.; FOWLKES, C. ; MALIK, J.. **Contour detection and hierarchical image segmentation**. IEEE Transactions on Pattern Analysis and Machine Intelligence, 33(5):898–916, 2011.

[71] GUILLON, S.; JONCOUR, F.; BARRALLON, P.-E. ; CASTANIÉ, L.. **Ground-truth uncertainty-aware metrics for machine learning applications on seismic image interpretation: Application to faults and horizon extraction**. The Leading Edge, 39(10):734–741, 2020.

[72] AN, Y.; GUO, J.; YE, Q.; CHILDS, C.; WALSH, J. ; DONG, R.. **Deep convolutional neural network for automatic fault recognition from 3d seismic datasets**. Computers & Geosciences, 153:104776, 2021.

[74] KINGMA, D. P.; BA, J.. **Adam: A method for stochastic optimization**. In: Bengio, Y.; LeCun, Y., editors, 3RD INTERNATIONAL CONFERENCE ON LEARNING REPRESENTATIONS, ICLR 2015, SAN DIEGO, CA, USA, MAY 7-9, 2015, CONFERENCE TRACK PROCEEDINGS, 2015.