WEAKLY-SUPERVISED SEMANTIC LABELING OF MIGRATED SEISMIC DATA

A Thesis
Presented to
The Academic Faculty

By

Yazeed Alaudah

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Electrical and Computer Engineering

Georgia Institute of Technology

August 2019

Copyright © Yazeed Alaudah 2019

WEAKLY-SUPERVISED SEMANTIC LABELING OF MIGRATED SEISMIC DATA

Approved by:

Professor Ghassan AlRegib, Advisor
School of Electrical and Computer Engineering
Georgia Institute of Technology

Professor James H. McClellan
School of Electrical and Computer Engineering
Georgia Institute of Technology

Professor Mark Davenport
School of Electrical and Computer Engineering
Georgia Institute of Technology

Professor Zhigang Peng
School of Earth and Atmospheric Sciences
Georgia Institute of Technology

Professor Ying Zhang
School of Electrical and Computer Engineering
Georgia Institute of Technology

Date Approved: May 2, 2019

I dedicate this dissertation to my family. To my incredible father Khalid, my loving mother Aljoharah, my wonderful wife Shahd, and my little princess Yara.

Thank you for everything ♥.

ACKNOWLEDGEMENTS

I am incredibly fortunate and lucky for all the circumstances, opportunities, and people who have led me to pursue my Ph.D. degree, and have helped me along the way. I'm very grateful for being able to work under the supervision of Prof. Ghassan AlRegib who was more than just a Ph.D. advisor, he was a mentor and a dear friend. He greatly shaped the way I think about research and life, and I have learned a great deal from him along the way. It was a great honor for me to have him as an advisor.

I am also grateful to my thesis committee: Prof. James McClellan, Prof. Mark Davenport, Prof. Ying Zhang, and Prof. Zhigang Peng for their insightful comments and discussion. My Ph.D. would have been lacking without my internships at Panasonic Automotive Innovation Center in 2016, Mitsubishi Electric Research Labs (MERL) in 2017, and Airbus Aerial in 2018. These internships gave me the opportunity to work on a diverse set of problems and helped me further develop the skills that I relied on in my research. I'm grateful for working with Dr. Jin Woo Jung at Panasonic, Dr. Tim Marks at MERL, and Madhav Desetty at Airbus Aerial for their valuable guidance, feedback, and insights.

I would also like to thank all former and current members of the OLIVES lab and the Center for Energy and Geo Processing (CeGP) for their friendship. I will always cherish the wonderful and supportive environment that they helped create. I am especially thankful to my dear friends Motaz Alfarraj, Charlie Lehman, Yuting Hu, Chih-Yao Ma, Min-Hung (Steve) Chen, Mohit Prabhushankar, Jinsol Lee, Gukyeong Kwon, and former members: Amir Shafiq, Mohammad Aabed, Tariq Alshawi, and Zhen Wang. I'm also grateful for the postdocs, Dr. Haibin Di, Dr. Ashraf Alattar, and Dr. Zhiling Long. A special thanks to my very dear friend Dr. Can Temel for always being there for me when life got tough. I am incredibly grateful for his friendship.

I would like to extend my appreciation and thanks to everyone in the Saudi Students Association at Georgia Tech. I will always hold dear all the wonderful times we spent together. I would especially like to thank Abdullah Al-Shehri, Saeed Al Abri, Thamer Al-Guthami, Ahmad Baubaid, Motaz Alfarraj, and Mohammad Al-Hassoun. I also would like to thank my Alma Mater, King Fahd University of Petroleum and Minerals (KFUPM) for the generous scholarship opportunity to obtain my Ph.D.

I am incredibly grateful for all the support I have received from my family. No words can do them justice. To my father Khalid, my mother Aljoharah, my brother Yasir and all my siblings, thank you from the bottom of my heart. Finally, I cannot thank enough my dear wife Shahd, for her love, patience, and support during this journey while being half the world away from her family.

TABLE OF CONTENTS

Acknowledgments iv

List of Tables xi

List of Figures xiii

List of Symbols or Abbreviations xvii

Summary xxi

1 Introduction 1

1.1 Motivation 1

1.2 Overall Approach 11

1.3 Outline 14

2 Similarity-Based Image Retrieval 16

2.1 Overview 16

2.2 Background 17

2.3 The Curvelet Transform 21

2.4 Method 1: Histogram of Curvelet Coefficients 23

2.5 Method 2: Truncated Curvelet Singular Values 24

2.6 Results 27

2.6.1 Retrieval experiment 28

2.6.2 Clustering experiment 30

2.7 Seismic Image Retrieval 33

2.8 Summary 34

3 Structural Interpretation with Weak Image-Level Labels 38

3.1 Overview 38

3.2 Labeling with Image-Level Labels 39

3.2.1 Training Stage 40

3.2.2 Prediction Stage 41

3.3 Comparison with Various Texture and Multiresolution Features . . 44

3.4 Results 52

3.5 Summary 55

4 Weakly-Supervised Label Mapping 60

4.1 Overview 60

4.2 Background 61

4.2.1 Convolutional Neural Networks 62

4.2.2 Matrix Completion and Factorization 67

4.2.3 Summary 69

4.3 Non-Negative Matrix Factorization 74

4.4 Sparsity and Orthogonality Constraints 75

4.5 Multiplicative Update Rules 76

4.6 Extracting the Labels 78

4.7 Results 80

4.8 Summary 85

5 Structural Interpretation with Weak Pixel-Level Labels 89

5.1 Overview 89

5.2 Background 90

5.3 Proposed Method 91

5.3.1 Network architecture 92

5.3.2 Adapting the loss function for weak labels 93

5.4 Results 97

5.5 Summary 102

6 Stratigraphic Interpretation with Weak Pixel-Level Labels . . 105

6.1 Overview 105

6.2 Background 108

6.3 A 3D Geological Model of the Netherlands F3 Block 111

6.3.1 The geology of the F3 block 112

6.3.2 The modeling process 115

6.3.3 The 3D geological model 117

6.4 Deconvolution Network Baseline 120

6.5 Experimental Setup 123

6.5.1 The geological model 123

6.5.2 The train/test split 125

6.5.3 Obtaining weak labels 126

6.5.4 Training the models 129

6.6 Results 131

6.6.1 Fully-Supervised Results 131

6.6.2 Weak vs. Strong Supervision 136

6.7 Summary 141

7 Conclusion 144

7.1 Contributions 146

7.2 Future Research Suggestions 148

8 Thesis Products 150

8.1 Invention Disclosures 150

8.2 Magazine Articles 150

8.3 Datasets 150

8.4 Journal Articles 151

8.5 Conference Papers 151

Appendix A Evaluation Metrics 155

A.1 Retrieval 155

A.2 Clustering 156

A.3 Semantic Segmentation 157

Appendix B Derivation of Multiplicative Update Rules 159

B.1 Multiplicative Update Rule for $\mathbf{W}$ 160

B.2 Multiplicative Update Rule for $\mathbf{H}$ 161

B.3 Constrained Optimization 162

References 162

Vita 180

LIST OF TABLES

2.1 The performance of the different similarity measures in the retrieval experiment. 30

2.2 The performance of the different similarity measures in the clustering experiment 32

3.1 Evaluation of the labeling performance for various texture and multiresolution features. 54

3.2 The percentage of pixels from each class in the manually labeled inline # 380. 55

4.1 A summary of the main CNN-based techniques, and the methods they use to overcome weak-supervision. 72

4.2 A summary of related matrix completion or factorization based techniques proposed in the literature. 73

5.1 A comparison of the labeling results for the method presented in this chapter versus the method presented in Chapter 3 that only uses image-level labels. 102

6.1 The percentage of pixels from different classes in the training set. . . 125

6.2 The accuracy of the weak labels used in this chapter compared to the ground truth labels. The accuracy is computed using different metrics for different values of the normalized confidence threshold $\tilde{\tau}$. 130

6.3 The size and amount of training data for various models. 130

6.4 Results of various strongly-supervised models when tested on both test splits of our dataset. 132

6.5 A comparison of various weakly- and strongly-supervised models when tested on both test splits of our dataset. 140

LIST OF FIGURES

1.1 A three-dimensional view of the Netherlands F3 block [3] showing the inline, crossline, and time section seismic data in addition to data from three well logs. 3

1.2 An example of the different nature of object boundaries in natural and seismic images. 5

1.3 Color information makes it easier to parse complex scenes. 6

1.4 An example of the different nature of annotations in natural images and seismic data. 7

1.5 The exponential growth of the number of images in publically available annotated datasets in the natural image domain. 8

1.6 A comparison of how different machine learning paradigms use annotated training data. 9

1.7 An illustration of the difference between full supervision and weak supervision. 10

1.8 The main elements of our weakly-supervised semantic labeling framework for *structural interpretation*. 13

2.1 A comparison of different image similarity measures for the "Einstein" image altered with different types of distortions. (a) Reference image. (b) Contrast stretch. (c) Luminance shift and (d) Gaussian noise. The measures used are MSE, SSIM [15], and CW-SSIM [16]. Figure adapted from [14] with permission. ©(2009) IEEE. 18

2.2 (a) Steerable pyramid filter bank and (b) Steerable pyramid spectral decomposition with 4 orientations and 4 scales. Figure adapted from [17] with permission. ©(1995) IEEE. 19

2.3 Frequency and spatial viewpoints of a curvelet wedge. 22

2.4 Overall block diagram of Method 2 25

2.5 Sample images from the four classes of the LANDMASS-2 dataset. . . 28

2.6 Receiver operating characteristics (ROC) curves for the various measures used in the retrieval experiments. 29

2.7 The precision at M results for the different classes of the LANDMASS-2 dataset retrieved using various similarity measures. 31

2.8 The results of two-dimensional MDS on different similarity measures. The cyan, blue, green, and red colors correspond to the `horizon`, `chaotic`, `fault` and `salt dome` classes. 35

2.9 The exemplar images of each class of subsurface structures that were used to retrieve the images from the seismic volume. 36

2.10 Sample retreived images from each class of subsurface structures. The first column shows one of the exemplar images for the `chaotic`, `other`, `faults`, and `salt dome` classes. These exemplar images are highlighted in blue, cyan, green, and red respectively. 37

3.1 A block diagram illustrating the weakly-supervised subsurface structure labeling framework described in this chapter. 39

3.2 A weakly-supervised labeled seismic section from the Netherlands North Sea F3 block database. 43

3.3 (a) Block diagram of a 2D 1-level DWT decomposition. (b) A 2-level DWT of a seismic image. 47

3.4 (a) 2-level decomposition for 2D SWT. (b) 3-level SWT of a seismic image. 48

3.5 Gabor filters at two different scales and four different orientations. . . 49

3.6 (a) Steerable pyramid filter bank and (b) Steerable pyramid spectral decomposition with 4 orientations and 4 scales. Figure adapted from [17] with permission. ©(1995) IEEE. 50

3.7 The contourlet filter bank (adapted from [75].) 51

3.8 The nonsubsampled countourlet filter bank. (adapter from [95]) . . . 52

3.9 Results of our image-level labeling framework on inline #380 of the Netherlands F3 block using texture features. The colors blue, green, and red correspond to the `chaotic`, `faults` and `salt dome` classes respectively. 58

3.10 Results of our image-level labeling framework on inline #380 of the Netherlands F3 block using multiresolution features. The colors blue, green, and red correspond to the `chaotic`, `faults` and `salt dome` classes respectively. 59

4.1 A comparison of various scene understanding tasks in computer vision, from coarse (image classification) to fine (instance segmentation). Figure adapted from [108] with permission. All rights reserved ©2017 Elsevier. 62

4.2 An illustration of the fully convolutional network (FCN) architecture. Reprinted, with permission from [116]. All rights reserved ©2016 IEEE. 64

4.3 An illustration of the difference between the outputs of a) FCN and b) DeconvNet. The DeconvNet activation maps are more precise and contain far more fine details compared to FCN. Reprinted, with permission from [117]. All rights reserved ©2015 IEEE. 65

4.4 A plot showing the convergence curves for the multiplicative update rules for $\mathbf{W}$ and $\mathbf{H}$ as well as the overall objective function in equation 4.4. 81

4.5 The effect of the orthogonality term in Equation 4.4 on the final cofficient matrix $\mathbf{H}^{\mathsf{final}}$. On the left, $\mathbf{H}^{\mathsf{final}}$ without the orthogonality term, and on the right matrix $\mathbf{H}^{\mathsf{final}}$ with the orthogonality term. 82

4.6 Results of our weakly-supervised label mapping approach for sample images from each class during different iterations. 83

4.7 The robustness of our label mapping algorithm to mislabeled images for various numbers of feature clusters per class, k. 84

4.8 The robustness of our label mapping algorithm to mislabeled images for various feature sparsity levels, ρ_w. 85

4.9 Results of our weakly-supervised label mapping approach for various subsurface structures. 86

4.10 Results of our weakly-supervised label mapping approach for various subsurface structures. 87

4.11 Results of our weakly-supervised label mapping approach for various subsurface structures. 88

5.1 The architecture of the deconvolution network used in this work. White layers are convolution or deconvolution layers. Red layers are max-pooling layers, while green layers are unpooling layers. 92

5.2 An illustration of the difference between cross entropy loss (CE), focal loss (FL), and weak focal loss (WFL) for different values of γ and using $\alpha = 1$. 96

5.3 Fault structures in crossline #1 highlighted using either deconvolution network or FCN-8s, and using either the cross entropy loss (CE) or the weak focal loss (WFL). Green arrows indicate false negatives, while red arrows indicate false positives. 99

5.4 Results using our model to highlight various subsurface structures in inline #350 of the Netherlands F3 block. 100

5.5 A 3D view of the Netherlands F3 block, with our model highlighting the three salt dome structures in the F3 block. 101

5.6 A comparison of the a weakly-supervised labeled seismic section from the Netherlands North Sea F3 block database using models trained with image- or pixel-level labels. 104

6.1 The scale of a typical seismic waveform compared to an outcrop (left), and compared to a wireline log (right). The frequencies used in seismic exploration (10–60 Hz) have long wavelengths, and therefore, the resolution of seismic data is limited to large-scale stratigraphic features. Figure adapted from [167] with permission. ©(2016) Springer. 106

6.2 An example of exposed lithostratigraphic formations in the mountains of northern Ellesmere Island, Canada. Figure adapted from [167] with permission. ©(2016) Springer. Photo credit A. F. Embry. 107

6.3 The publically-available annotated inline of the Netherlands F3 block from Rutherford Ildstad and Bormann [65]. The inline contains partial annotations for nine classes of seismic facies. 109

6.4 The location of the F3 block. Adapted from [180]. 111

6.5 A) A geological cross-section of the North Sea continental shelf along axis A-A'; B) A map of the location of the cross-section. Adapted from [180]. 114

6.6 Locations of the boreholes that were used to create the geological model. 115

6.7 A 3D view of our geological model of the F3 block. 116

6.8 An overhead view of 3D fault planes from three different generations of faults that we have identified in the F3 block. 120

6.9 Two diagonal cross sections of our 3D geological model in Figure 6.7. 121

6.10 A 3D view of the F3 block from above with the Zechstein Group shown in red, while the Chalk Group is shown in a semi-transparent beige color. 124

6.11 The exemplar images of each class of lithostratigraphic units that were used to retrieve the images from the seismic volume. 126

6.12 Sample retrieved images from each class. The first column shows the exemplar image. The remaining columns show the 10th, 50th, 100th, and 200th retrieved images from each class respectively. 127

6.13 Results of the label mapping for each class of lithostratigraphic units. 129

6.14 The results of the different fully-supervised models on inline 200 from test set #1. 133

6.15 Confusion matrices for our two fully-supervised *baseline* models on both test set #1 and #2. 135

6.16 The results of the different weakly- and strongly-supervised models on inline 200 from test set #1. 139

6.17 The results of the different weakly- and strongly-supervised models on inline 400 from test set #2. 141

6.18 Confusion matrices for a fully-supervised and a weakly-supervised model 142

LIST OF ABBREVIATIONS

AP	Average Precision
AUC	Area Under the Curve
CA	Class Accuracy
CAE	Convolutional Autoencoder
CBIR	Content-Based Image Retrieval
CE	Cross Entropy Loss
CLBP	Completed Local Binary Patterns
CLDP	Completed Local Derivative Patterns
CMP	Common Mid-Point
CNN	Convolutionnal Neural Network
CRF	Conditional Random Feild
DWT	Discrete Wavelet Transform
ELBP	Extended Local Binary Patterns
EM	Expectation Maximization
FCN	Fully Convolutional Network
FDCT	Fast Discrete Curvelet Transform
FFT	Fast Fourier Transform

FL	Focal Loss
FPR	False Positive Rate
FWIU	Frequency Weighted Intersection over Union
GLCM	Gray Level Cooccurance Matrix
GPU	Graphical Processing Unit
IU	Intersection over Union
LBP	Local Binary Pattern
LRI	Local Radius Index
MAP	Mean Average Precision
MCG	Multiscale Combinatorial Grouping
MDS	Multidimensional Scaling
MIU	Mean Intersection over Union
MSE	Mean Square Error
MUR	Multiplicative Update Rule
NMF	Non-negative Matrix Factorization
PA	Pixel Accuracy
PSNR	Peak Signal to Noise Ratio
RA	Retrieval Accuracy

ROC	Receiver Operating Characteristics
SCD	Squared-Chord Distance
SGD	Stochastic Gradient Descent
SSIM	Structural Similarity
SVD	Singular Value Decomposition
SVM	Support Vector Machine
SW-WFL	Similarity-Weighted Weak Focal Loss
TPR	True Positive Rate
WFL	Weak Focal Loss

SUMMARY

Deep learning has revolutionized the fields of machine learning and computer vision. However, the availability of annotated data to train state-of-the-art deep networks is one of the main bottlenecks to the successful application of deep learning, especially to applications like seismic interpretation where annotated data is extremely scarce. In this thesis, we develop a weakly-supervised framework for the semantic labeling of large seismic volumes. This framework involves developing a state-of-the-art texture similarity measure and using it for retrieving large numbers of images with high visual similarity to exemplar images for each target class. Images with high visual similarity can be assigned image-level labels matching those of the exemplar images used to retrieve them. A novel weakly-supervised label mapping algorithm, based on orthogonal non-negative matrix factorization, is then used to transform these image-level labels into pixel-level labels that encode the locations of the target classes within each image. Finally, these weak pixel-level labels are used to train deep convolutional networks for the semantic labeling of various seismic structures and lithostratigraphic units within large seismic volumes. A special loss function is introduced to help the networks learn effectively when trained with weak labels. The benefit of this work is that it enables the training and deployment of deep learning models to new application domains—such as seismic interpretation—where sufficient quantities of labeled data are not available, and annotation costs are prohibitively expensive.

CHAPTER 1
INTRODUCTION

1.1 Motivation

In recent years, deep learning has witnessed great successes in wide-ranging applications and has revolutionized the fields of machine learning and computer vision. This success was not only due to the flood of massive quantities of data and the growing use of powerful GPUs, but also the arrival of deep learning models that can achieve state-of-the-art results on a variety of tasks by learning their own hierarchical data representations, and not requiring any hand-engineered features. Despite the overwhelming success of deep learning in various vision tasks; there is, however, a drawback. Deep learning models are often far more complex than traditional machine learning models and can have hundreds of millions of free parameters. This not only means that they need large amounts of computational resources to train these models, but more critically, they require vast amounts of labeled training data. Labeled data can be extremely costly and time-consuming to obtain. In practice, the high cost of obtaining labeled data is a critical bottleneck to the successful application of deep learning to many application domains. This bottleneck is especially true in the field of seismic interpretation, where oil and gas exploration and production (E&P) companies seldom share their data, and where the subjective nature of the interpretation process and the lack of ground truth makes it common for geophysicists to arrive at different interpretations for the same data.

The exploration and production process

Seismic interpretation is only one step in the hydrocarbon exploration and production process. This complicated process can be summarized into four main stages: *aquisition*, *data processing*, *interpretation*, and finally, *feild development and production*. The acquisition step involves performing large-scale seismic surveys inland or offshore by geophysical exploration and production companies to evaluate the prospect of new hydrocarbon reservoirs in various geographical locations. These surveys generate vast amounts of raw data. A single day in a typical seismic survey can generate six terabytes of raw data [1].

In the data processing stage, the raw seismic data obtained from the survey undergoes series of processing steps. These steps include preprocessing, deconvolution, move-out correction, common mid-point (CMP) sorting and stacking, deconvolution and migration [2]. The result of these processing steps is a very large 3D (or sometimes a 4D) seismic volume that reflects the various layers and geological structures in the subsurface of the survey location. Figure 1.1 shows an example of a migrated 3D seismic volume from the well-known Netherlands North Sea F3 block.

Next, the seismic interpretation stage involves the analysis of the migrated seismic volume. The seismic interpretation process can be subdivided into structural, stratigraphic, and lithologic interpretation. Structural interpretation is primarily based on studying subsurface structures, such as faults, fractures, salt domes, and gas chimneys. Stratigraphic interpretation is mainly concerned with the study of depositional environments and sedimentology. Finally, lithologic interpretation involves studying the physical characteristics of rocks —such as porosity, density, and velocity—from seismic well logs.

All these three aspects of seismic interpretation are examined and analyzed by experienced seismic interpreters and geophysicists to understand the geological history of the survey area better and to create a geological model that reflects this

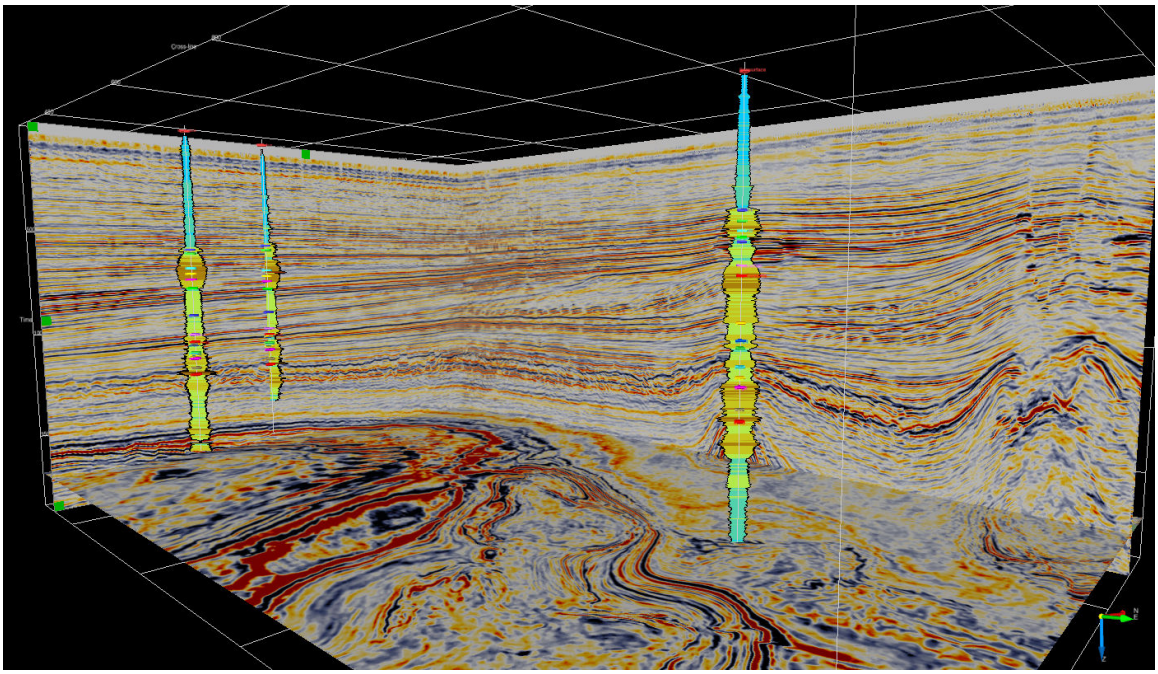


Figure 1.1: A three-dimensional view of the Netherlands F3 block [3] showing the inline, crossline, and time section seismic data in addition to data from three well logs.

history. The combination of the three elements of seismic interpretation allows the interpreters to identify potential locations in the geological model where hydrocarbon reservoirs are likely to be trapped. If the geological model indicates possibilities of hydrocarbon reservoirs that can be economically viable, exploratory wells are drilled, and if the results are positive, work on field development and production starts. In this dissertation, our focus is primarily on structural seismic interpretation. However, in Chapter 6 we extend our work to stratigraphic interpretation as well.

Depending on the size of the seismic survey and the geology of the region, the seismic interpretation process can take from several months to more than a year. Furthermore, with the increasing size of seismic volumes, this process is becoming increasingly more time consuming and costly. Recently, there has been increasing interest in automated or semi-automated interpretation workflows that can help speed up the process of interpreting large seismic volumes. For example, in the case of struc-

tural interpretation, various methods have been recently proposed for detecting and tracking faults, salt bodies, and other subsurface structures within seismic volumes; for a good overview of such methods, see [4]. These techniques can help reduce the time and effort required for interpretation; however, the process of extracting regions within large seismic volumes based on their dominant subsurface structure—so that detection or tracking can be performed on the extracted region—is still done manually. This is one of the leading obstacles to end-to-end automated interpretation workflows. An analogy with natural images would be if a human had to manually extract the location of every traffic sign or pedestrian, before passing them on to algorithms that classify the traffic signs or track the positions of the pedestrians within the frames of a video for example.

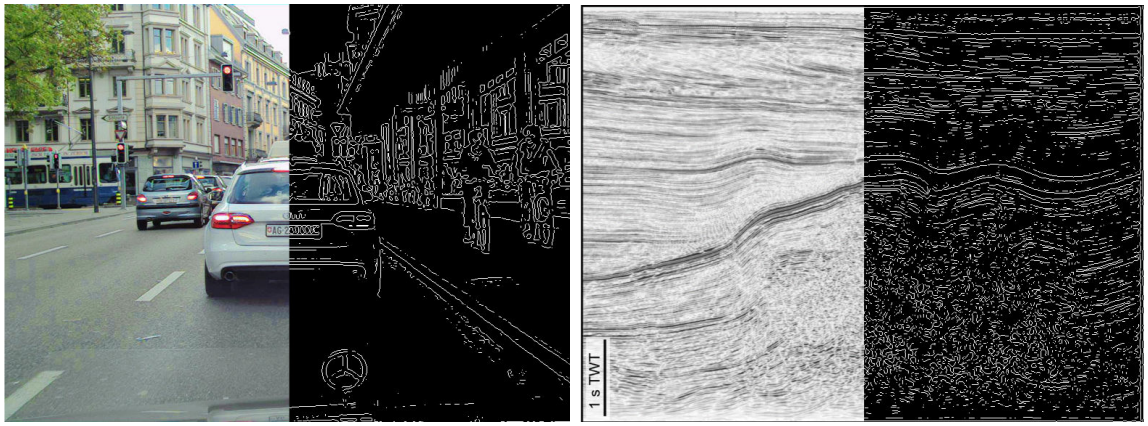
The opportunities and challenges of seismic volume labeling

To automate the process of extracting regions of interest within large seismic volumes, we propose the problem of *seismic volume labeling*. This problem involves the assignment of a class label to every voxel in the 3D seismic volume based on the voxel's subsurface structure (in the case of structural interpretation) or its lithostratigraphic unit (in the case of stratigraphic interpretation). In computer vision, the problem of assigning class labels to each pixel in an image or a video is an established research problem known as semantic segmentation[1]. Semantic segmentation algorithms use either classical techniques based on hand-crafted features or learning-based techniques, commonly based on convolutional neural networks (CNNs), to automatically assign semantic class labels to every pixel in the image. However, these techniques are not directly applicable to seismic data[2]. Seismic data presents challenges that cannot be immediately solved by the existing methods. These challenges include:

[1]The term 'scene labeling' or 'scene parsing' is also used, especially in early papers in the literature.

[2]we review these techniques in Chapter 4.

1. **Poorly-defined boundaries:** Unlike natural images where the boundaries between objects are well-defined, boundaries between subsurface structures in seismic data are either not well-defined or characterized by a change in overall texture rather than a sudden change in amplitude. In addition, the boundaries between different stratigraphic units in seismic volumes are usually defined by seismic horizons, but subject-matter expertise is required to identify these horizons from other horizons within the same stratigraphic units. Figure 1.2 highlights the different nature of object boundaries between natural and seismic images.



(a) (b)

Figure 1.2: An example of the different nature of object boundaries in natural images (a), and seismic data (b). Boundaries between objects in natural images are well defined, and can often be highlighted by simple edge detection techniques. Boundaries between subsurface structures are defined by a change in texture that makes them more difficult to detect.

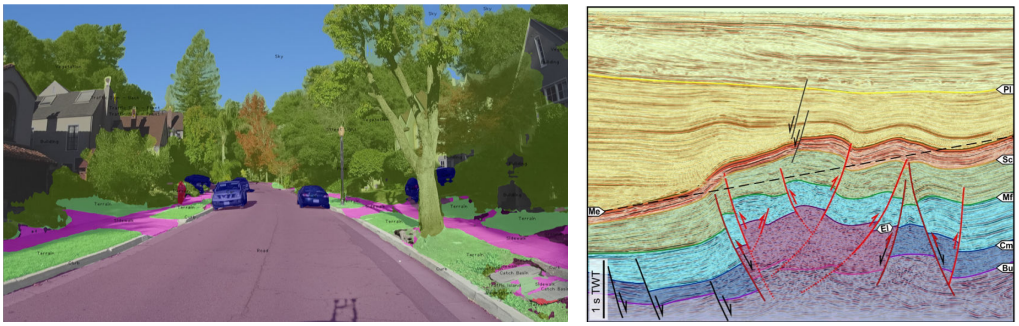
2. **Lack of color information:** As the example in Figure 1.3 shows, natural images have color information that can significantly help in distinguishing different objects and in parsing complex scenes very quickly. Unlike natural images, seismic data is textured in nature and lacks the rich color information that distinguishes various objects in natural images.

3. **Annotation difficulties:** Humans learn to understand their surroundings from

Figure 1.3: Color information makes it easier to parse complex scenes.

a very young age. Therefore, it is very natural for us to distinguish different objects in natural scenes. Online services such as Amazon Mechanical Turk (AMT) help researchers in generating very large scale annotated datasets (such as ImageNet [5]) by outsourcing the annotation process to thousands of annotators worldwide. Even then, obtaining pixel-level annotations is really expensive. For instance, in ImageNet, 14 million images are annotated with image-level labels, a subset of 500,000 images have bounding boxes; but only 4,460 images have pixel-level annotations [6, 7]. For seismic data, the annotation process is a very time consuming and laborious process that requires subject-matter experts, and can not be easily outsourced due to licensing and contractual obligations. Furthermore, the only way to obtain ground truth data in seismic imaging is to drill wells that can cost up to $40 million for land rigs, and at least $200 million for offshore rigs [8]. Therefore, the lack of ground truth data can greatly affect the quality of the annotations, and it is quite normal for different interpreters to

not agree on a single interpretation for the same seismic data. This is especially true in the case of complex subsurface formations that require a high level of experience and understanding of the geology of the survey region. Figure 1.4 illustrates the difference between annotation natural images and seismic data.



(a) (b)

Figure 1.4: An example of the different nature of annotations in natural images (a), and seismic data (b). Unlike natural images, seismic data usually requires subject matter experts to annotate the data, and therefore the annotation process cannot be easily outsourced or expedited. In addition, unlike natural images where it is easy to find the ground truth, the only way to obtain the ground truth in seismic data is to drill wells that can be very expensive.

4. **Lack of large-scale annotated datasets:** There is an abundance of openly available large-scale annotated datasets for a vast range of problems involving natural images and videos (e.g., Pascal VOC [3][9] and CityScapes[4] [10]). Figure 1.5 shows the exponential growth in the size of openly-available large-scale annotated datasets for natural images and videos. For seismic interpretation tasks, there is a severe lack of annotated seismic data for training and well-established benchmarks for testing various learning-based approaches. This annotated-data bottleneck is the leading obstacle to the successful application of deep learning methods, such as CNNs, to the semantic labeling of large seismic volumes. Acquiring large amounts of annotated seismic data is a very challenging task by

[3]http://host.robots.ox.ac.uk/pascal/VOC/voc2012/
[4]https://www.cityscapes-dataset.com/

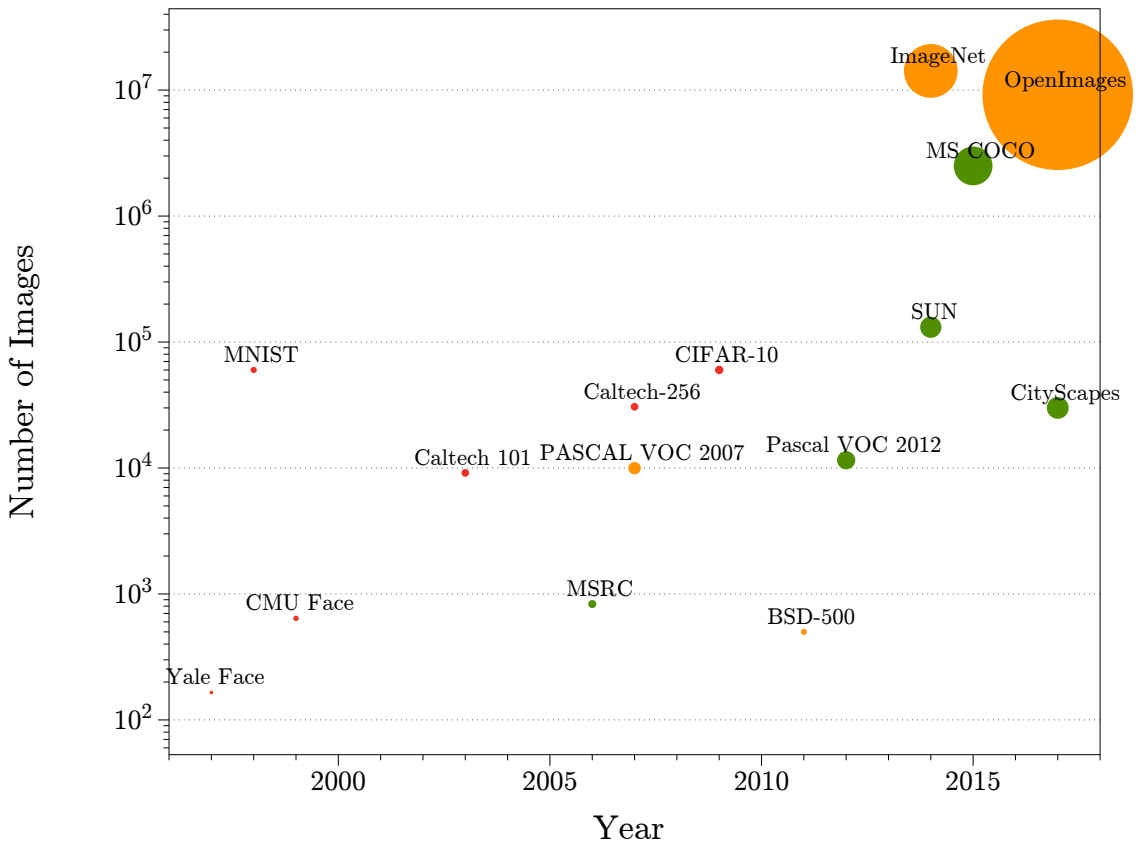


Figure 1.5: The exponential growth of the number of images in publically available annotated datasets in the natural image domain. Red indicates datasets with image-level labels only, orange indicates datasets with an intermediate form of supervision such as bounding boxes, and green indicates datasets with predominantly pixel-level labels. The size of each disk corresponds to the *square root* of the size of each dataset.

itself.

Learning in the absence of sufficient annotated data

In other vision domains, machine learning researchers and practitioners use various methods to overcome the lack of sufficient annotated data. Some use various data augmentation techniques—such as adding random noise, rotations, and cropping random patches from the images—to artificially increase the size of their training data. Others resort to transfer learning by using models pretrained on other datasets and 'fine-tuning' these models to their specific dataset or task. However, all these methods

Supervised	Semi Supervised	Weakly Supervised	Unsupervised
• supervision in the form of strongly-labeled training data	• supervision in the form of strongly-labeled training data for only a subset of the training data • the rest is not labeled	• supervision in the form of weakly-labeled training data • weak labels are noisy and less informative, but are much easier to obtain	• no labeled data required • learns the underlying distribution of the data

Figure 1.6: A comparison of how different machine learning paradigms use annotated training data.

require fully-annotated samples at some point. In many application domains, such as seismic interpretation, obtaining such fully-annotated samples in large-enough quantities is simply unpractical.

Lately, there has been considerable interest in weakly-supervised methods for labeling visual data. Weakly-supervised learning is a machine learning paradigm where the training labels convey less information than the labels desired at the output of the trained model. Another commonly used definition of weakly-supervised learning is that it is a machine learning paradigm where a model is trained using examples that are only partially annotated [11]. Figure 1.6 summarizes the difference between various machine learning paradigms regarding their use of annotated data.

Figure 1.7 helps illustrate weakly-supervised learning in the context of a seismic interpretation task. Assume we would like to train a simple binary classifier to classify whether pixels in an image belong to a salt body or not. The model takes an input image similar to the one in the figure and produces the desired output shown, where red denotes salt body, and cyan denotes everything else. To train a *fully*-supervised model, we would need training labels that convey the same information as the desired output, namely, pixel-level labels for all pixels in the training images. If a geophysicist only partially labeled the training images, provided a bounding box, or worst of all,

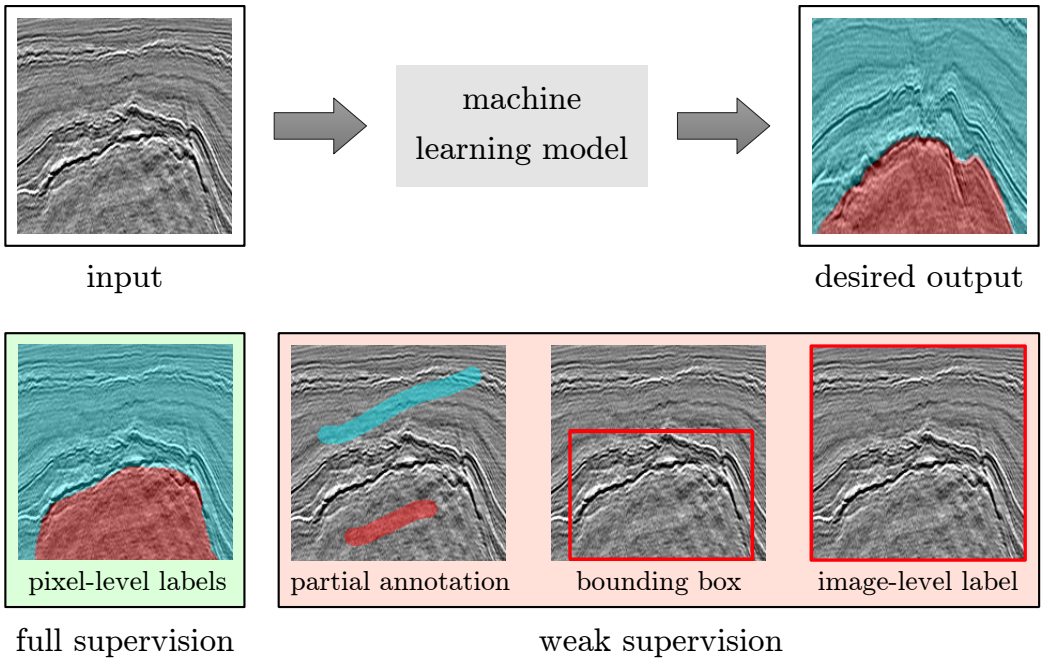


Figure 1.7: An illustration of the difference between full supervision and weak supervision. The second row indicates the form of labels used to train the machine learning model. Red denotes salt body whereas cyan denotes everything else.

just provided an image-level label indicating whether the training image contains a salt body, then our trained machine learning model would be a *weakly*-supervised one. Naturally, weak labels, such as the ones in Figure 1.7 are far easier and less costly to obtain than strong ones. On the other hand, however, they are far less informative and usually lead to poor results compared to their fully-labeled counterparts.

Seismic interpretation is an excellent application domain where weakly-supervised learning can play a significant role in enabling the use of state-of-the-art deep learning models to automate the most time-consuming and laborious interpretation tasks. Therefore, **the objective of this dissertation** is to develop a weakly-supervised framework for the semantic labeling of large visual volumes using state-of-the-art deep learning models and to apply this framework to application domains where large amounts of annotated data are not available. In this dissertation, we use seismic interpretation as our application domain, and we focus specifically on problems related to structural and stratigraphic interpretation.

1.2 Overall Approach

In order to train state-of-the-art deep learning models for the segmentation and labeling of large seismic volumes, vast amounts of training data are required. However, labeled seismic data is extremely limited, and therefore this can be an exceptionally difficult task. To overcome this challenge, we develop a weakly-supervised framework for obtaining large amounts of pixel-level labeled training data, using very minimal input from a seismic interpreter. This framework involves developing an accurate seismic image similarity measure for computing the pairwise similarity of a very large number of seismic images. This similarity measure is then used for retrieving large amounts of images with a high visual similarity to exemplar images selected by an interpreter. These images contain various structures of interest and are retrieved from within large unlabeled seismic volumes. This similarity-based retrieval process provides us with a large number of images with image-level labels that indicate the main subsurface structure within them. A weakly-supervised label mapping technique is then used to transform these image-level labels into pixel-level labels that encode the locations of the target classes within the different images. Finally, these weakly-mapped labels are used to train deep convolutional networks for labeling subsurface structures within large seismic volumes.

While this framework was developed for structural seismic interpretation, we apply the same framework to stratigraphic interpretation where we have created, with the help of an experienced geologist, the largest *annotated* dataset in seismic interpretation and made it publically available.[5] This dataset allows us to compare the performance of our weakly-supervised models with the same models trained on fully-annotated data.

The benefit of our weakly-supervised framework is that it enables the training and deployment of deep learning models to new application domains, such as seismic

[5] `https://github.com/olivesgatech/facies_classification_benchmark`

interpretation, where sufficient quantities of labeled data are not available. While this framework was developed with seismic interpretation in mind, it is not difficult to extend the results of this work to other application domains that suffer similar problems with lack of sufficient annotated data. Towards this end, and to aid fellow researchers, the codes and datasets used throughout this dissertation are publically available[6] online.

[6] `https://ghassanalregib.com/publications/`

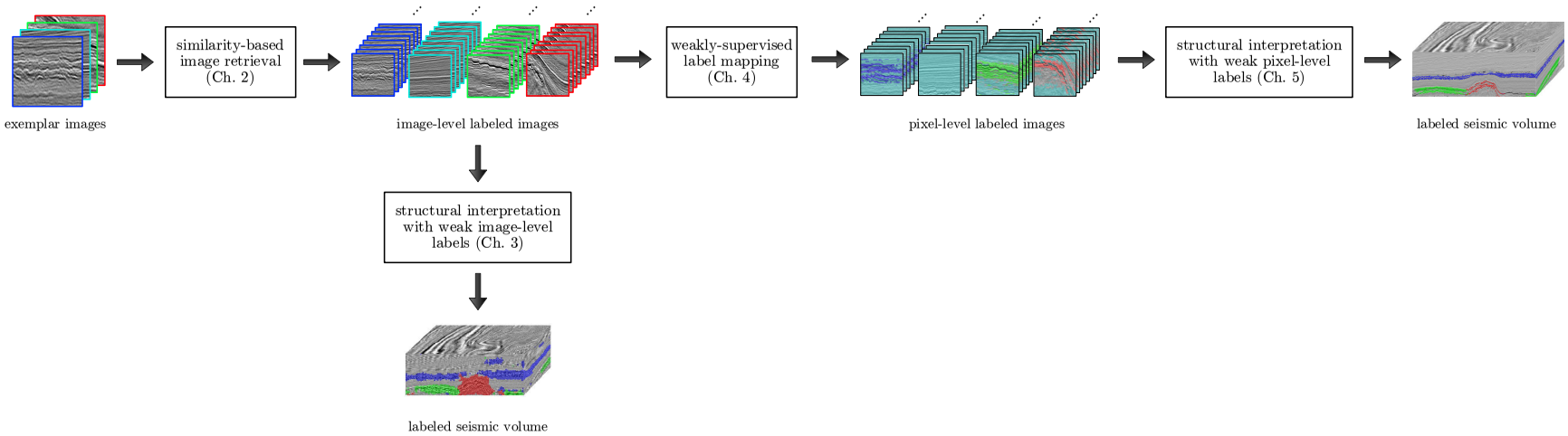


Figure 1.8: The figure shows the main elements of our weakly-supervised semantic labeling framework for *structural interpretation*. Each block represents a chapter in this dissertation. Chapter 3 shown here investigates the semantic labeling of seismic volumes using only weak image-level labels, but is not a central element of our workflow. Also, Chapter 6, not shown in this figure, applies this entire framework to the stratigraphic interpretation problem. Colored image boundaries indicate the image-level labels of different classes of subsurface structures, while colored-pixels indicate pixel-level labels.

1.3 Outline

A visual outline of our proposed framework in shown in Figure 1.8. The rest of this dissertation is organized as follows.

In Chapter 2, we explain the concept of similarity-based image retrieval and review the relevant literature. We then introduce two novel similarity measures, and we conduct detailed experiments that show the superiority of our two measures to others in the literature. We then show how these similarity measures can be used to retrieve a large number of seismic images that contain similar subsurface structures to exemplar images of each target class. Furthermore, we show how these images can be assigned image-level labels based on the experiments we have conducted.

In Chapter 3, we study how effective these image-level labels are in training a weakly-supervised machine learning model for classifying various seismic structures. We introduce a framework for weakly-supervised labeling of seismic sections using only image-level labels. We study the performance of various texture and multiresolution features extracted from these images and compare their performance in labeling structures in the Netherlands North Sea dataset.

In Chapter 4, we introduce a weakly-supervised label mapping algorithm, based on non-negative matrix factorization (NMF), that maps the image-level labels obtained previously to pixel-level labels that encode the locations of the target classes. We show how our proposed algorithm returns confidence values in each predicted pixel-level label. These pixel-level labels can make it easier for machine learning models to classify various subsurface structures since the models do not need to infer the pixel locations of various classes based on the image-level labels of every example. We show how our proposed method is robust to misretreived images, and how it compares to different baseline methods.

Chapter 5 shows how the generated weak pixel-level labels and their associated

confidence values can be used to train deep convolutional neural networks (CNNs) to classify three main subsurface structures in the Netherlands F3 block. We introduce a new network loss function that accounts for the label confidence values, and we show how this new loss function helps reduce false positive classifications. Finally, we compare the results of this approach to the results in Chapter 3 that only uses image-level labels.

In Chapter 6 we extend the proposed framework to seismic stratigraphic interpretation. We introduce a state-of-the-art fully-annotated dataset[7] that we have created with the help of a geologist, and made publically available. This dataset maps the various stratigraphic units in the Netherlands North Sea F3 block. We introduce these stratigraphic units and introduce two baseline deep learning models based on a deconvolution network architecture. We then compare the results of these baselines, when trained on strong versus weak labels, and we show that our framework can be successfully extended to other applications such as seismic stratigraphic interpretation.

Finally in Chapter, 7 we conclude this dissertation with an overall summary of our research, our main contributions, and future research directions.

[7] `https://github.com/olivesgatech/facies_classification_benchmark`

CHAPTER 2
SIMILARITY-BASED IMAGE RETRIEVAL

2.1 Overview

With the tremendous growth of visual content on the web and elsewhere, image retrieval is commonly used for searching and retrieving images from large image databases [12]. Traditional image retrieval systems would often rely on various metadata, such as captions or tags, to retrieve relevant images efficiently. However, since many images may not have such metadata or their metadata might not be sufficient to describe the images accurately, content-based image retrieval (CBIR) is often used to retrieve images based on their visual content, rather than their metadata. Similarity-based image retrieval is a subset of CBIR that aims to retrieve images based on their overall visual similarity to a query image. This "visual similarity" is often computed using a *similarity measure* that quantifies the similarity between the two images.

In seismic interpretation applications, the amount of data is enormous, and its manual annotation is very time consuming and labor intensive. Similarity-based image retrieval can play a significant role in helping to streamline the interpretation process and make it more efficient and less time-consuming. By retrieving images that contain similar structures, seismic images that contain similar subsurface structures can be clustered together. Furthermore, similarity-based image retrieval can be utilized to create large datasets for the classification of seismic images, without requiring any human input other than the selection of the query or 'exemplar' images for each class.

In the following section, we review the relevant literature. Then in sections 2.4 and 2.5, we introduce two seismic image similarity measures that are based on the

curvelet transform [13]. The curvelet transform is described in section 2.3. In section 2.6, we share the results of several experiments and compare our proposed similarity measures to others in the literature. Finally, in section 2.7, we show how these similarity measures are used to retrieve a large number of seismic images that contain similar subsurface structures from within large unlabeled seismic volumes, and show that above some similarity threshold, the retrieved images can be assumed to belong to the same class as the query image[1]. This allows us to assign image-level labels to these retrieved images.

2.2 Background

From image denoising to image quality assessment and super-resolution, many image processing applications use objective measures to quantify the similarity (or dissimilarity) between two images. These *image similarity measures* quantify the similarity between two images, typically as a number between 0 and 1, where a similarity value of 1 means the images are identical. Sometimes, distance measures—such as the Euclidean distance—are used in this context, with a distance of 0 indicating identical images. Throughout this work, distance values are transformed to be in the range $[0, 1]$ to match the similarity values[2].

Traditionally, metrics such as the mean square error (MSE) or the peak signal-to-noise ratio (PSNR) have been used to measure the difference between images, but they have been widely criticized in the image processing community for their poor performance as measures of perceptual dissimilarity [14]. Since metrics such as MSE assume a pixel-to-pixel correspondence between images, they ignore any spatial relationship between the pixels in the images; therefore they perform poorly as measures of perceptual dissimilarity.

Structural similarity (SSIM) is a widely-used similarity measure that improves

[1] With a degree of confidence in this assignment based on the chosen similarity threshold.

[2] Specifically, we use $\mathsf{similarity} = \frac{1}{\alpha \times \mathsf{distance}+1}$, where α is a positive constant and $\mathsf{distance} \in [0, \infty)$.

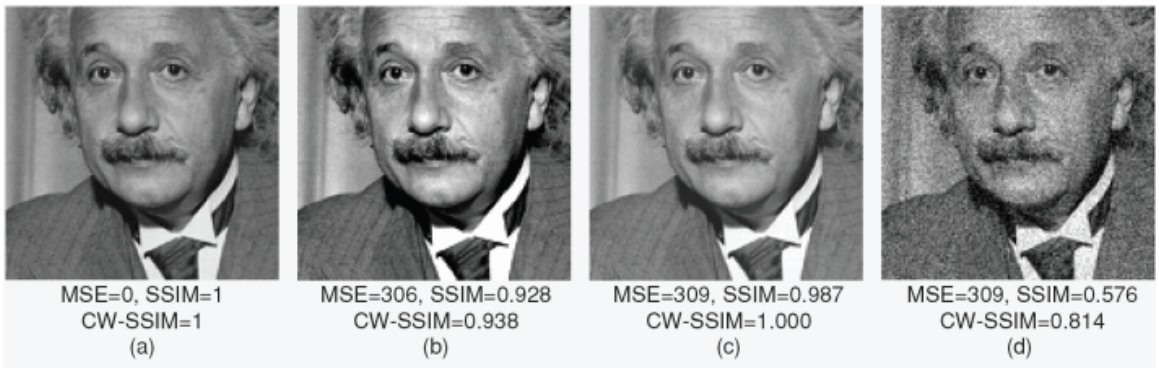


Figure 2.1: A comparison of different image similarity measures for the "Einstein" image altered with different types of distortions. (a) Reference image. (b) Contrast stretch. (c) Luminance shift and (d) Gaussian noise. The measures used are MSE, SSIM [15], and CW-SSIM [16]. Figure adapted from [14] with permission. ©(2009) IEEE.

upon MSE and PSNR by capturing local image structure using low-level local statistics in the spatial domain [15] or the complex wavelet domain (CW-SSIM)[16]. These measures and similar ones proposed in the literature are often used for applications such as image denoising or image quality assessment where the pixel-to-pixel correspondence assumption can be justified. Figure 2.1 shows an image with three different distortions that perceptually degrade the image to various degrees. However, the MSE score for the distorted images is almost identical. The *perceptual* degradation in the images is captured better by the SSIM and CW-SSIM scores.

A different class of similarity measures is *content-based* similarity measures. These measures quantify the similarity between the *contents* of the images without making any assumptions about the location of the content within the image. Such measures usually improve on metrics such as SSIM or CW-SSIM by being translation- or rotation-invariant to some degree. These content-based measures are often used for applications like content-based image retrieval (CBIR) in which the goal is to find images that contain similar visual content to a query image. A particular class of content-based similarity measures is texture-based similarity measures. These measures compare the texture content of images and compute their similarity. This is

of particular interest since our work mainly focuses on seismic images, which are textured in nature.

A well-known example of such measures is a family of structural texture similarity measures (STSIM) which uses subband statistics and correlations in a multiscale frequency decomposition called the steerable pyramid [17]. The steerable pyramid is a multiscale image decomposition developed by Simoncelli *et al.* [17]. As shown in Figure 3.6, the image is first decomposed into highpass and lowpass subbands and then the lowpass band is further decomposed into bandpass subbands of different orientations and a lowpass subband. The lowpass subband is then subsampled and passed as an input to a similar decomposition to obtain details at other scales. The bandpass filters capture details at different orientations and the subsampling allows it to capture details of different scales.

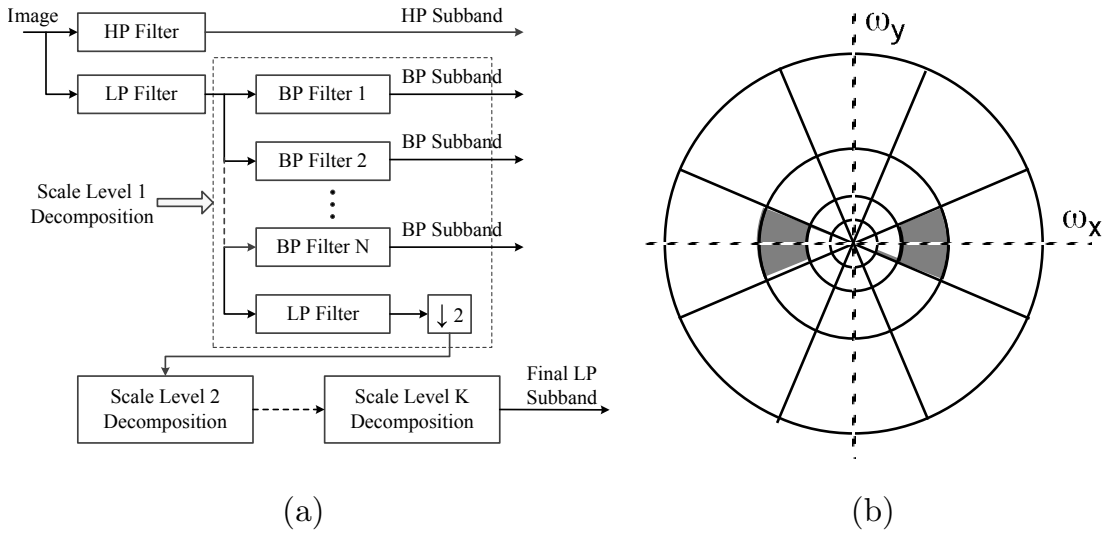


Figure 2.2: (a) Steerable pyramid filter bank and (b) Steerable pyramid spectral decomposition with 4 orientations and 4 scales. Figure adapted from [17] with permission. ©(1995) IEEE.

STSIM-1 was introduced by Zhao *et al.* [18]. It replaces the structure term in the CW-SSIM with terms that compare first-order autocorrelations of neighboring subband coefficients in the steerable pyramid decomposition. Several years later, Zujovic

et al. introduced STSIM-2 [19]. STSIM-2 builds on the success of its predecessor by using a broader set of subband statistics in the steerable pyramid decomposition, such as the cross-correlation of subband statistics at adjacent scales and orientations. Zujovic *et al.* showed that STSIM-2 achieves state-of-the-art results on texture retrieval on the CURET and CORBIS texture datasets [19].

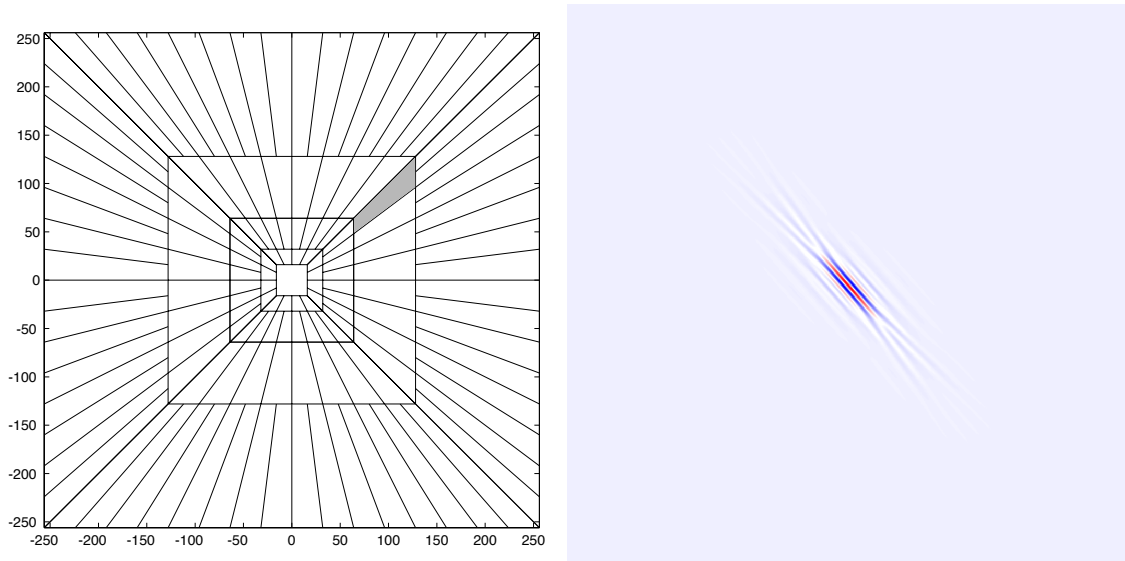
Another example of such multiscale decomposition is the curvelet transform [20], which provides an efficient representation of images with high directional content. Candés and Donoho [20] have shown that images that contain geometrically regular edges are more compactly represented by a curvelet rather than a wavelet decomposition. Recently, some texture-based similarity measures based on the curvelet transform have been proposed in the literature. Zhang *et al.* [21] proposed a rotationally invariant texture similarity measure using first order statistics of curvelet features. A very similar approach was proposed by Arivazhagan *et al.* [22] that used curvelet co-occurrence features in addition to first order subband statistics. Other researchers proposed a rotationally-invariant texture distance measure by fitting generalized Gaussian distributions to the curvelet [23], or wavelet [24], coefficients of the two images; then the Kullback-Leibler divergence is used to compute the difference between the different subband densities. However, such rotational invariance is not suitable for seismic images where the orientation of various subsurface structures is an important feature. Alternatively, Selvan and Ramakrishnan [25] suggested modeling the distribution of the singular values of the wavelet coefficients of the two images as an exponential distribution; then, the Kullback-Leibler divergence between the parameters of the two exponential distributions is used as a similarity measure between two texture images.

With the increasing interest in computational seismic interpretation, similarity measures designed explicitly for seismic images have been proposed. One of the earliest works in this area was the work proposed by Al-Marzouqi and AlRegib [26]

who proposed a seismic image similarity measure based on adaptive curvelets. The scale and angle parameters of the adaptive curvelet transform [27] are found by an optimization algorithm that maximizes the coefficient of variation of the curvelet coefficients[28]. A disadvantage of this approach, however, is that this method requires the curvelets to be adapted to the two images before computing their similarity. Alternatively, the curvelet adaptation algorithm can be run on the entire dataset beforehand. However, this process is computationally expensive, and therefore this method does not suit our application well. Later, driven by the fact that seismic images are textured in nature, Long *et al.* [29] proposed combining the STSIM-1 texture similarity measure [18] with seismic discontinuity maps [30] to obtain a seismic image similarity measure that exploits the texture content of seismic images. While the use of the discontinuity map attribute in SeiSIM allows the STSIM-1 to be more sensitive to seismic data, the discontinuity map is computed in the spatial domain and is very computationally demanding. In sections 2.4 and 2.5 we present two similarity measures, based on the curvelet transform, that achieve state-of-the-art results in terms of retrieval accuracy, while not requiring any parameter selection or expensive spatial domain computations. In the following section, we review the curvelet transform in more detail.

2.3 The Curvelet Transform

The curvelet transform is a directional multiscale decomposition. Candés and Donoho introduced the first generation of the curvelet transform in 2000 [31] then refined it a few years later [32]. Despite their popularity, wavelets fail to compactly represent images with highly directional elements such as curves and edges [33]. To the contrary, curvelet frames have been shown to represent images with geometrically regular edges (such as seismic images) more compactly than other traditional multiscale representations [32]. This is especially true for seismic images where the seismic wavefronts lie



(a) Frequency viewpoint

(b) Spatial viewpoint of the real part of the wedge.

Figure 2.3: Frequency and spatial viewpoints of a curvelet wedge. Adapted from [13]. Copyright ©2006 Society for Industrial and Applied Mathematics. Reprinted with permission. All rights reserved.

mainly along smooth curves. Candés and Donoho [20] have shown that the curvelet transform provides an optimally sparse representation for curve-like structures, such as seismic reflectors when compared to wavelets. The curvelet transform has been successfully used in wide-ranging applications within seismic signal processing from seismic denoising [34] to primary multiple separation [35] and seismic migration [36]. For an image with N number of pixels, the fast discrete curvelet transform (FDCT) allows the computation of curvelet coefficients in $\mathcal{O}(N \log N)$ operations making the curvelet transform not only fast to compute, but also scalable to very large image datasets [37]. The reasons outlined in this paragraph lead us to believe that the curvelet transform would be very suitable for calculating the pairwise similarity of seismic images, and would be efficient enough to allow these similarity computations to be done on a large scale.

For our purposes here, we present a simplified overview of the FDCT. For a de-

tailed description see [37]. Given an image of size $N_1 \times N_2$, the FDCT divides the Fourier support of the image into J scales and $K(j)$ orientations as is shown in Figure 2.3(a). The total number of scales in the curvelet tiling, J, depends on the size of the image, and is given by

$$J = \lceil \log_2 \min(N_1, N_2) - 3 \rceil, \tag{2.1}$$

where $\lceil \cdot \rceil$ is the ceiling function. The number of orientations at scale $j \geqslant 1$, $K(j)$, is given by:

$$K(j) = 16 \times 2^{\lceil (j-1)/2 \rceil}. \tag{2.2}$$

For scale $j = 0$, there is only one orientation. Curvelet coefficients are then generated by taking the inverse 2D FFT for each wedge (such as the one highlighted in Figure 2.3(b) after multiplying it by a smooth bandpass filter. Since the 2D FFT of real images is symmetric around the origin, only two consecutive quadrants of the Fourier spectrum are necessary for obtaining the curvelet coefficients. Figure 2.3 shows the spatial and frequency representations of a curvelet wedge.

2.4 Method 1: Histogram of Curvelet Coefficients

The first proposed similarity measure [38] is very simple but outperforms the other measures that preceded it in the literature. This method, which we refer to as method 1, is based on computing the sum of the squared chord distance [39] between corresponding histograms of curvelet coefficients, at all scales and orientations. The resulting value is then normalized to generate a similarity value. Given two images, $\mathbf{x}_1$ and $\mathbf{x}_2$, we first normalize the two images by subtracting the mean and dividing by the standard deviation

$$\hat{\mathbf{x}}_i = \frac{\mathbf{x}_i - \mu_{\mathbf{x}_i}}{\sigma_{\mathbf{x}_i}}. \tag{2.3}$$

Then, the squared-chord distance (SCD) [39] is used to calculate the distance between the corresponding histograms of the curvelet coefficients of the two images

for each orientation and scale. We use the SCD here, as opposed to Euclidean or Manhattan distances, because it has been shown repeatedly that SCD outperforms them in various image retrieval tasks [40, 41, 42, 43]. If we let $\mathbf{h}^{(1)}_{j,k}(i)$ be the ith bin in the histogram of the curvelet coefficients of image $\hat{\mathbf{x}}_1$ at scale j and orientation k, and similarly, $\mathbf{h}^{(2)}_{j,k}(i)$ for image $\hat{\mathbf{x}}_2$. Then, we can define the following distance measure:

$$\text{distance}(\hat{\mathbf{x}}_1, \hat{\mathbf{x}}_2) = \sum_{j=0}^{J-1} \sum_{k=1}^{K(j)/2} \sum_{i=1}^{M} \left(\sqrt{\mathbf{h}^{(1)}_{j,k}(i)} - \sqrt{\mathbf{h}^{(2)}_{j,k}(i)} \right)^2, \tag{2.4}$$

where M is the total number of bins in the histogram, J is the number of curvelet scales, and $K(j)$ is the number of orientations at scale j. This distance measure is then converted to a similarity measure by using the following function:

$$\text{Method 1}(\mathbf{x}_1, \mathbf{x}_2) = \frac{1}{\alpha \times \text{distance}(\hat{\mathbf{x}}_1, \hat{\mathbf{x}}_2) + 1}. \tag{2.5}$$

Where α is a constant set to 10. This function is monotonic, so it does not affect the retrieval performance of this measure. Since this method takes the SCD over the *corresponding* histograms in each scale and orientation, it is not scale or rotation invariant. This allows this similarity measure to be sensitive to structures of different orientations and scales, such as small discontinuities or large salt domes. Finally, due to the simplicity of this approach, and its reliance only on the fast discrete curvelet transform, it is significantly faster than other seismic similarity measures we have tested (seven times faster than SeiSIM [29] and almost nine times faster than the adaptive curvelets approach [26]).

2.5 Method 2: Truncated Curvelet Singular Values

The second proposed similarity measure [44] computes the similarity on the singular values of the curvelet coefficients of the two images, after effective-rank truncation.

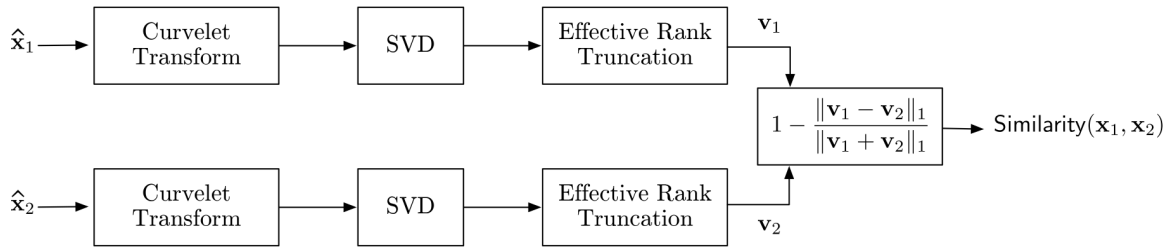


Figure 2.4: Overall block diagram of Method 2

Given the two images we compute feature vectors for the two images separately, and then the similarity of the two images is computed by comparing their corresponding feature vectors. The overall workflow for computing the similarity values using method 2 is shown in Figure 2.4.

To obtain the feature vector for an image, $\mathbf{x}_1$, we first normalize the image by subtracting the mean and dividing by the standard deviation

$$\hat{\mathbf{x}}_i = \frac{\mathbf{x}_i - \mu_{\mathbf{x}_i}}{\sigma_{\mathbf{x}_i}}. \tag{2.6}$$

Then, we apply the fast discrete curvelet transform, and compute its coefficients for all scales, $j = \{0, 2, \ldots, J-1\}$, and orientations, $k = \{1, 2, \ldots, K(j)\}$. The singular values of the curvelet coefficients at every scale j and orientation k are then calculated as $\boldsymbol{\sigma}_{[j,k]} = [\sigma_1, \ldots, \sigma_L]^T$ where $\sigma_1 \geqslant \sigma_2 \geqslant \ldots \geqslant \sigma_L$ and L is the smallest dimension of the coefficients matrix.

Ideally, if the rank of a matrix is r, only the first r singular values are non-zero. However, when we consider the singular value decomposition (SVD) on images that are subject to different types of noise, the number of non-zero singular values is greater than r. In most cases, none of the singular values are exactly zero; even for a rank-deficient matrix. Roy and Vetterli [45] proposed the *effective rank* as a method to estimate the actual rank of a matrix by estimating its effective dimensionality. To

calculate the effective rank, we first compute the normalized singular values as

$$p_i = \frac{\sigma_i}{\|\boldsymbol{\sigma}_{[j,k]}\|_1} \text{ for } i = 1, \ldots, L, \tag{2.7}$$

where $\|\cdot\|_1$ is the ℓ_1 norm. Then, the effective rank is calculated as a function of the entropy of the singular value distribution defined in equation 2.7, that is

$$\mathsf{EffectiveRank} = \exp\left(-\sum_{i=1}^{L} p_i \log p_i\right). \tag{2.8}$$

This results in a real number less than or equal to L with equality if and only if all singular values are equal.

For each set of curvelet coefficients, the $\mathsf{EffectiveRank}$, is calculated as in equation 2.8. A new vector of *effective* singular values is formed by keeping the first $\lfloor \mathsf{EffectiveRank} \rfloor$ singular values, where $\lfloor \cdot \rfloor$ denotes the floor function. The remaining singular values are set to 0. In other words, for scale j and orientation k, we form the vector $\hat{\boldsymbol{\sigma}}_{[j,k]} = [\sigma_1, \ldots, \sigma_{\lfloor \mathsf{EffectiveRank} \rfloor}, 0, \ldots, 0]$. The overall feature vector of image $\hat{\mathbf{x}}_i$ is then obtained by concatenating all $\hat{\boldsymbol{\sigma}}_{[j,k]}$ for all scales and half the number of orientations,

$$\mathbf{v}_i = [\hat{\boldsymbol{\sigma}}_{[1,1]}, \hat{\boldsymbol{\sigma}}_{[2,1]}, \ldots, \hat{\boldsymbol{\sigma}}_{[2,K(2)/2]}, \hat{\boldsymbol{\sigma}}_{[3,1]} \ldots, \hat{\boldsymbol{\sigma}}_{[J,1]}]. \tag{2.9}$$

Finally, the similarity between two images, $\hat{\mathbf{x}}_1$ and $\hat{\mathbf{x}}_2$, is then computed as

$$\text{Method 2}(\hat{\mathbf{x}}_1, \hat{\mathbf{x}}_2) = 1 - \frac{\|\mathbf{v}_1 - \mathbf{v}_2\|_1}{\|\mathbf{v}_1 + \mathbf{v}_2\|_1}. \tag{2.10}$$

Where, $\mathbf{v}_1$ and $\mathbf{v}_2$ are the feature vectors corresponding to $\hat{\mathbf{x}}_1$ and $\hat{\mathbf{x}}_2$. Since the singular values are non-negative by definition, the resulting similarity value is in the range $[0, 1]$ with a value closer to 1 indicating higher similarity.

2.6 Results

To evaluate the performance of our proposed seismic image similarity measures, we devise two experiments. Namely, an experiment on seismic image retrieval and another on seismic image clustering. Both of these experiments are performed directly on the similarity matrices of the various measures. These matrices contain the pairwise similarity values between for all images in the dataset for a specific measure. For example, for a dataset that contains N_s images, the size of the similarity matrix $\mathbf{S}$ is $N_s \times N_s$, where $\mathbf{S}(i, j)$ is the similarity between $\mathbf{x}_i$ and $\mathbf{x}_j$. The ith row of $\mathbf{S}$ represents the similarity values of all images in the dataset compared to $\mathbf{x}_i$.

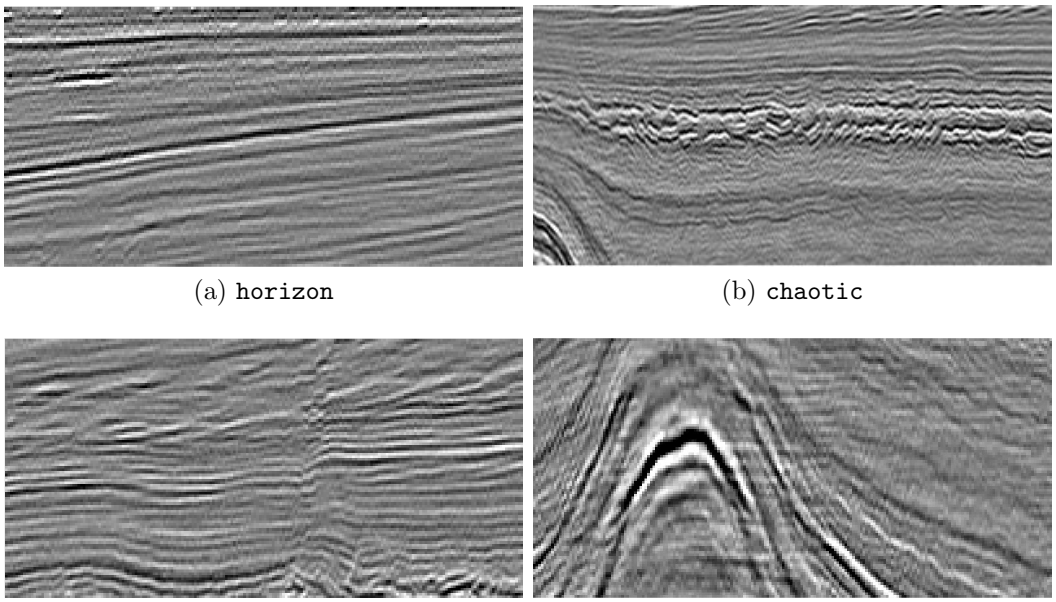
Throughout the similarity-based retrieval experiments, we use the LANDMASS-2 dataset[3] [46] which is comprised of $N_s = 4000$ images of size 99×99 pixels, with their values normalized to be between 0 and 1. These images were extracted from the Netherlands Offshore F3 block and divided equally into four classes according to their dominant subsurface structure. The classes are `horizon`, `chaotic`, `fault` and `salt dome`. Figure 2.5 shows sample images from each class.

In our experiments, we compare the performance of our proposed methods to different similarity and distance measures. The following measures were used in the experiments:

1. Euclidean distance
2. CW-SSIM with default parameters [16]
3. STSIM-1 and STSIM-2 with 4 scales and 8 orientations [19]
4. SeiSIM with 4 scales and 8 orientations [29]
5. Curvelet-based distance measure [38]

The performance of these similarity measures is quantified using commonly used information retrieval and clustering metrics. These metrics and other metrics used throughout this dissertation are detailed in Appendix A. In the following subsections,

[3] https://ghassanalregib.com/landmass/



(a) `horizon` (b) `chaotic`

(c) `fault` (d) `salt dome`

Figure 2.5: Sample images from the four classes of the LANDMASS-2 dataset.

we discuss the results of the retrieval and clustering experiments.

2.6.1 Retrieval experiment

To assess the retrieval performance of our method, we compute the pairwise similarity between all the images in the dataset and use these values to populate the similarity matrix, $\mathbf{S}(i, j)$, for every one of the similarity and distance measures listed above. Then the various retrieval metrics are computed directly on these similarity matrices to test their performance.

Figure 2.6 shows the receiver operating characteristic (ROC) curves for the various measures listed above. ROC curves plot the true positive (correctly retrieved) rate versus the false positive (wrongly retrieved) rate for various threshold values. The closer the curve of a similarity measure to the upper left corner of the plot, the better its retrieval performance. The results in Figure 2.6 show that both our similarity measures significantly outperform the others, with our method 2 slightly outperforming method 1.

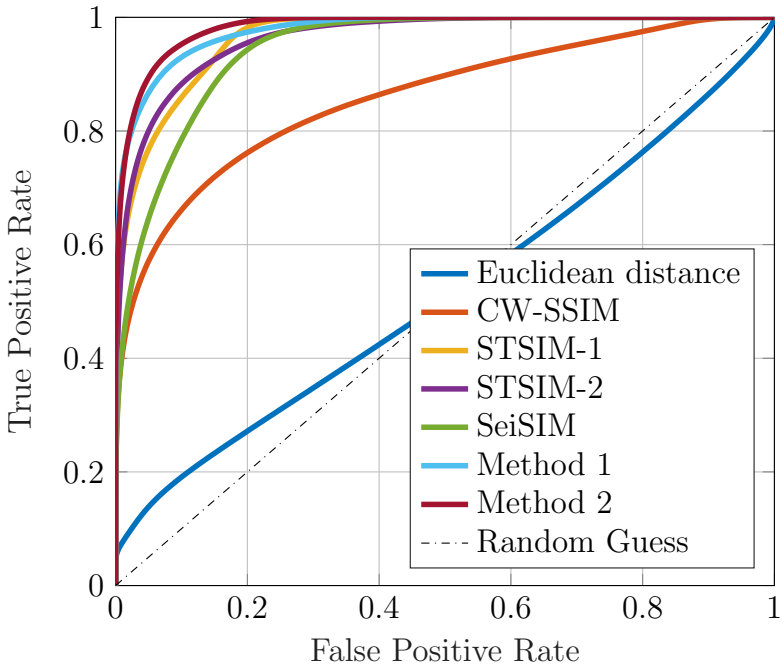


Figure 2.6: Receiver operating characteristics (ROC) curves for the various measures used in the retrieval experiments.

Table 2.1 shows the retrieval performance of the different measures using retrieval accuracy (RA), mean average precision (MAP), and area under the ROC curve (AUC). RA measures the overall percentage of correctly retrieved images. MAP is similar to RA, but considers the rank of the correctly retrieved images. Finally, AUC quantifies the ROC performance of the different measures. In all three performance metrics, our two methods significantly outperform the others. In third and fourth place are STSIM-1 and STSIM-2, the state-of-the-art measures in texture similarity. SeiSIM, a recent similarity measure specifically designed for seismic images, performs surprisingly poorly with only about 82% of the images correctly retrieved.

Figure 2.7 shows the precision at M results for the different classes in the LANDMASS-2 dataset that were retrieved using different similarity measures. The black curves show the similarity at M averaged over the entire dataset, while the colored curves show the results for specific classes of images. The results show that all the similarity measures, except CW-SSIM, correctly retrieve all the `horizon` images in the dataset.

Table 2.1: The performance of the different similarity measures in the retrieval experiment.

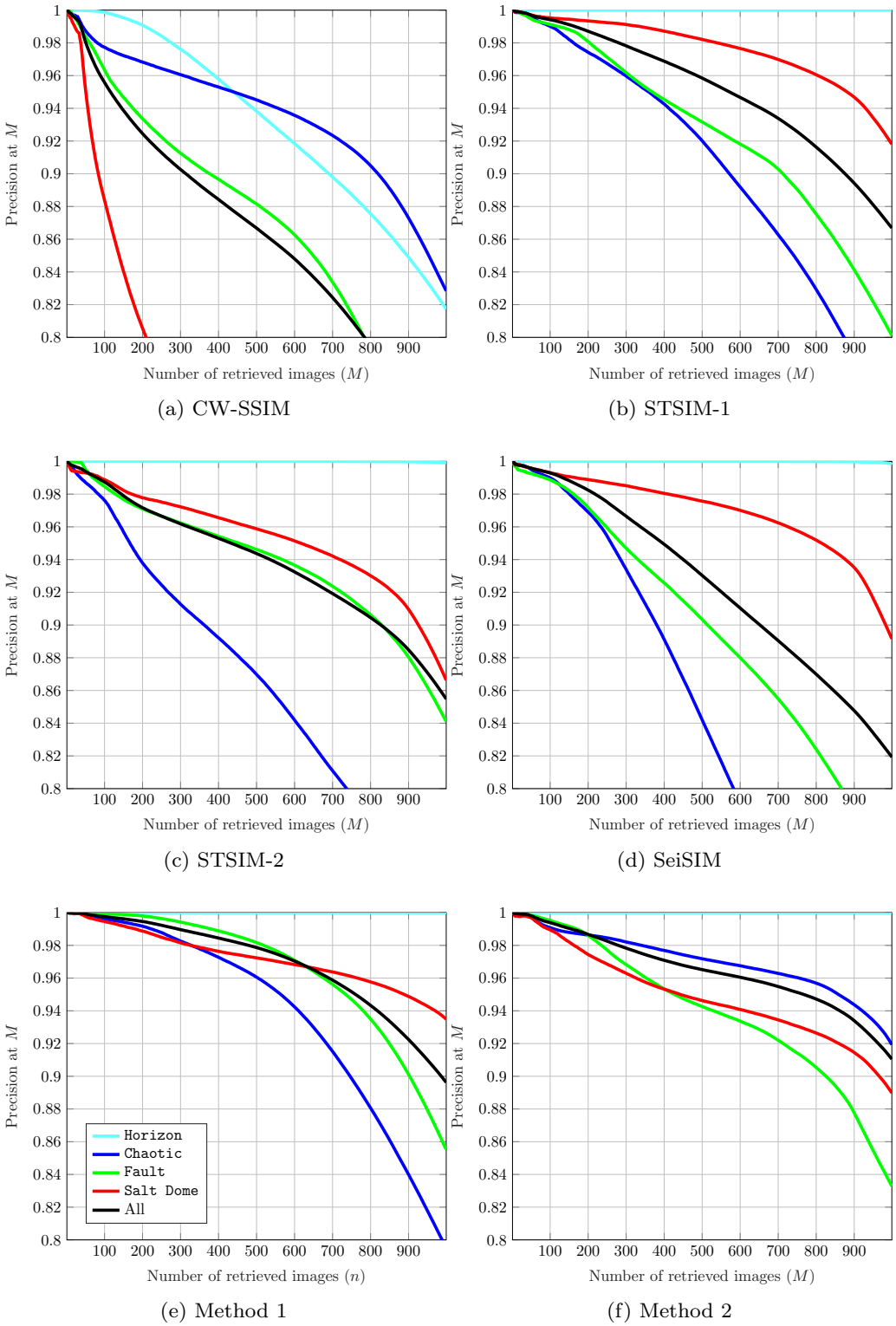
Measure	Features	RA	MAP	AUC
Euclidean distance	N/A	0.345	0.394	0.515
CW-SSIM [15]	Complex Wavelet	0.721	0.806	0.858
STSIM-1 [19]	Steerable Pyramid	0.867	0.926	0.966
STSIM-2 [19]	Steerable Pyramid	0.855	0.910	0.964
SeiSIM [29]	Steerable Pyramid	0.819	0.886	0.945
Method 1 [38]	Curvelet	0.896	0.949	0.978
Method 2 [44]	Curvelet	**0.911**	**0.954**	**0.983**

This is mainly due to the simplicity, and lack of diversity, of the structures in the horizon class. On the other hand, the precision at M curves of the other classes drops at different rates depending on the complexity of their structures and the sensitivity of the similarity measure in capturing these complex structures. The `chaotic` class, in particular, seems to be particularly challenging for many of the similarity measures except our proposed method 2. Also, while our two proposed methods show superior performance to the other similarity measures, method 2, in particular, shows the most consistent performance across all classes, whereas method 1 does not perform well in the `chaotic` class.

2.6.2 Clustering experiment

To further assess the performance of the similarity measures listed above on seismic data, we set up a clustering experiment using the similarity matrix obtained previously. First, the images in the dataset are projected into a two-dimensional Euclidean subspace based on their similarity, such that the distance between images in the projection subspace is inversely proportional to their similarity values. This projection is done using classical multidimensional scaling (MDS) [47]. Then, the projected data points are clustered into four clusters using the k-means algorithm.

The resulting clusters do not necessarily correspond to classes; unless the similarity



(a) CW-SSIM (b) STSIM-1

(c) STSIM-2 (d) SeiSIM

(e) Method 1 (f) Method 2

Figure 2.7: The precision at M results for the different classes of the LANDMASS-2 dataset retrieved using various similarity measures. The cyan, blue, green, and red colors correspond to the `horizon`, `chaotic`, `fault`, and `salt dome` images respectively. The black curve shows the precision at M result averaged over all the classes.

Table 2.2: The performance of the different similarity measures in the clustering experiment

Measure	Features	Rand Index
Euclidean distance	N/A	0.394
CW-SSIM [15]	Complex Wavelet	0.870
STSIM-1 [19]	Steerable Pyramid	0.895
STSIM-2 [19]	Steerable Pyramid	0.877
SeiSIM [29]	Steerable Pyramid	0.888
Method 1 [38]	Curvelet	0.905
Method 2 [44]	Curvelet	**0.970**

measure is very accurate. Therefore, one can use the clustering results to quantify the goodness of the similarity measure. To evaluate the clustering performance, we compute the *rand index* (explained in Appendix A.2) which is a measure of the accuracy of the clustering. We report the rand index results for different similarity measures in Table 2.2. The results of the clustering experiment further validate our conclusion from the retrieval experiment that our similarity measures are superior to other methods in the literature. Our proposed method 2 significantly outperforms other measures in the literature, with a rand index of 0.970 compared to STSIM-1 which achieves a rand index of 0.895.

Also, we show the two-dimensional projection of the data using all the similarity measures in Figure 2.8. The figure shows that using the similarity values to project the dataset into a lower dimensional subspace produces clusters that are almost linearly separable. `Horizon` and `salt dome` classes are separated well from all other classes. However, the `fault` and `chaotic` classes commonly overlap, with our proposed methods showing the least overlap between the two classes. It is important to mention that this is only a two-dimensional projection of the data and that the data is more easily separated in a higher-dimensional space. A two-dimensional projection was chosen partly for the ease of visualization, and to increase the challenge of the classification experiment. These results suggest that our proposed similarity measures

can be used to discriminate the different classes of seismic images with high accuracy.

2.7 Seismic Image Retrieval

The results in section 2.6 show that out of all the similarity measures we tested, our proposed method 2 significantly outperforms the other measures, including the state-of-the-art measure in texture similarity. Therefore, in the remainder of this dissertation, we will always elect to use method 2 for our retrieval purposes. Given a suitable similarity measure, we can retrieve large numbers of images from unlabeled datasets based on their similarity to hand-selected exemplar images that are selected by an experienced interpreter. The process of selecting the exemplars only involves cropping small patches from within an unlabeled seismic volume. These exemplar images are used to represent the interpreter's notion of various subsurface structures. While more exemplar images would undoubtedly lead to better results, we restrict the number of exemplars per class to a maximum of two. This is done to test the effectiveness of our approach, and to keep user input at a minimum.

Our exemplar images represent chaotic layers, faults, and salt domes which are all well-known subsurface structures; we also add an additional class (`other`) to contain examples of subsurface structures that do not belong to the first three (such as horizons, and sigmoidal structures). These classes of subsurface structures were selected because they commonly form traps for hydrocarbons reservoirs and because they are the least controversial to interpret. Other structures such as gas chimneys are more subjective to interpret and require more in-depth knowledge of the geological history of the survey area.

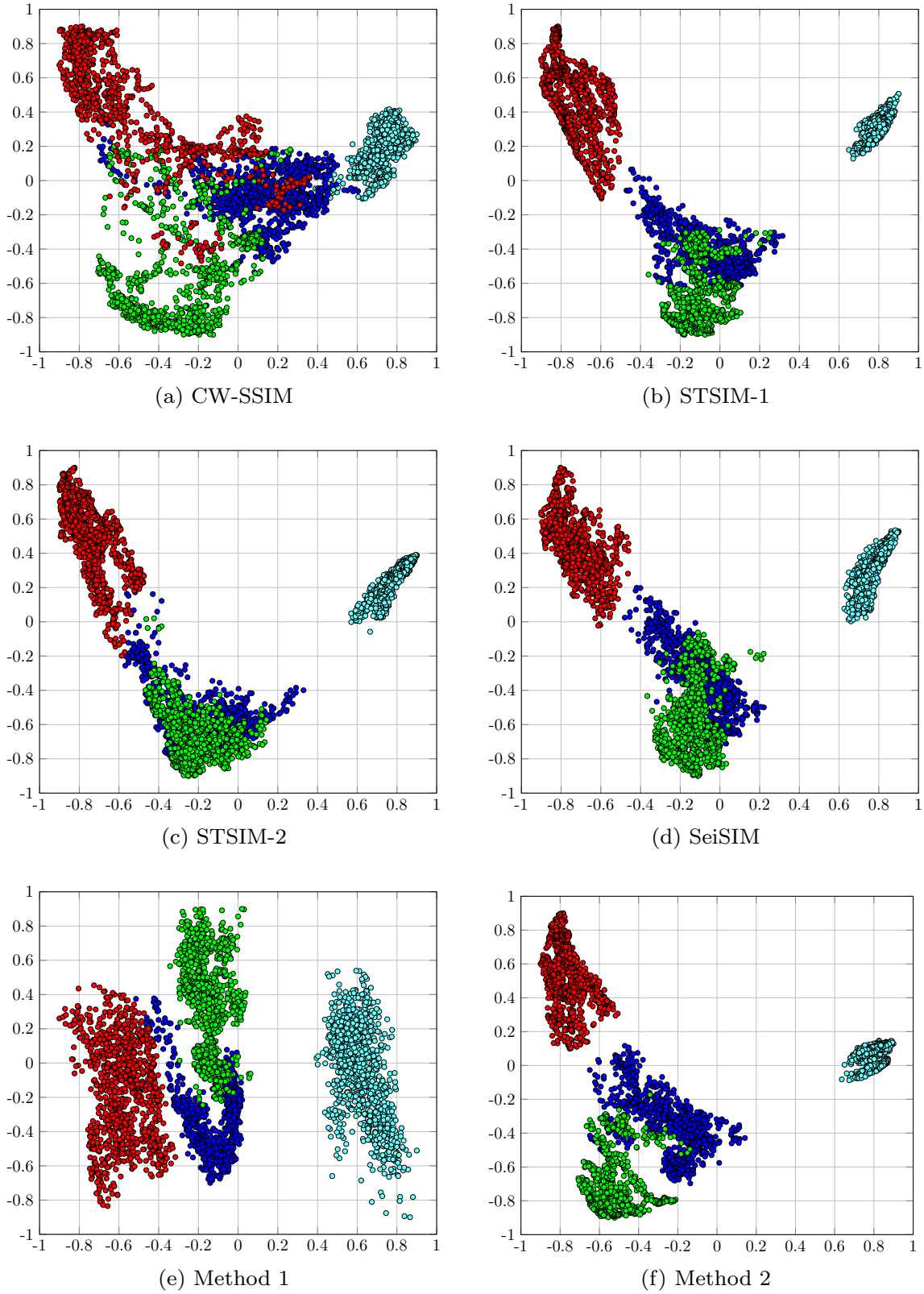
Given these exemplars, shown in Figure 2.9, we search through the entire Netherlands F3 block dataset [3] for images that contain similar subsurface structures. We retrieve $M = 500$ images for each class of subsurface structures. Figure 2.10 shows examples of these retrieved images. Given the results of method 2 in Figure 2.7, we

would expect the precision at $M = 500$ to be at least 95%. This is a valid expectation since the LANDMASS-2 dataset that was used in the experiments section is a tiny subset of the Netherlands F3 block. However, this accuracy level is not necessarily guaranteed since different exemplar images can lead to different results. Given that we expect around 95% of the retrieved images to contain subsurface structures of the same class, we can safely assign these retrieved images 'image-level' class labels that match that of the exemplar image that was used to retrieve them. This is how the similarity-based retrieval process automatically generates image-level labels for a large number of unlabeled images extracted from the seismic volume.

At this stage, we have thousands of seismic images containing different subsurface structures, with image-level labels assigned to them. The next chapter will investigate how these image-level labels alone can be used for the structural interpretation of seismic volumes.

2.8 Summary

In summary, we have proposed two seismic image similarity measures based on the curvelet transform. We have shown that the proposed methods outperform existing methods in the literature in different applications such as seismic image retrieval and clustering. Also, since these methods rely only on the fast discrete curvelet transform, they are computationally efficient and can scale easily to large seismic volumes. Method 2 has shown the best results on all the metrics we have used and has shown the most consistent performance across the different classes of subsurface structures that we have investigated. Therefore, we use this method to retrieve a large number of seismic images from the Netherlands F3 block, given exemplar images that contain seismic structures of interest. These retrieved images can later be used in the automation of structural interpretation by training machine learning models to recognize the common structures in each class of seismic images.



(a) CW-SSIM

(b) STSIM-1

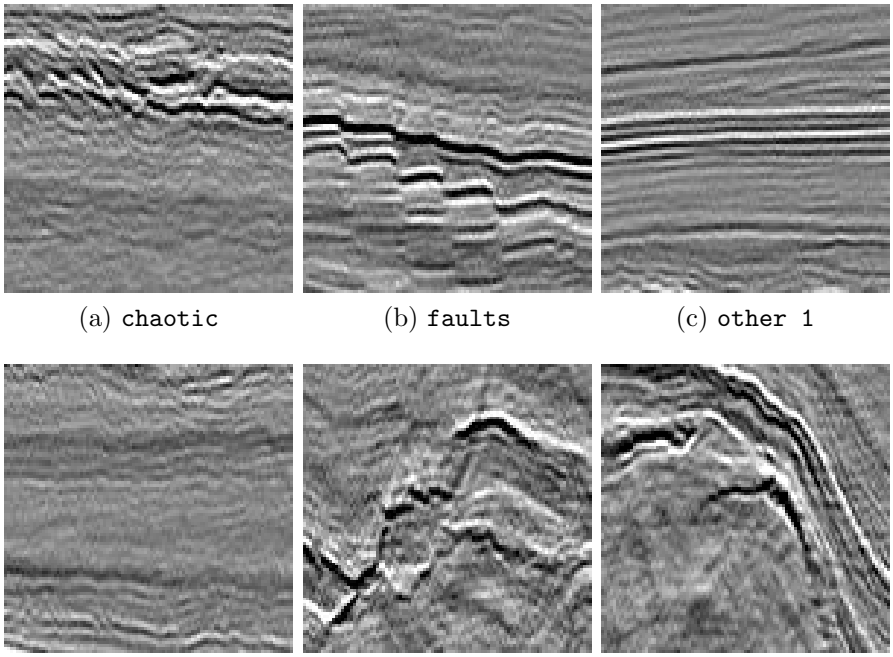
(c) STSIM-2

(d) SeiSIM

(e) Method 1

(f) Method 2

Figure 2.8: The results of two-dimensional MDS on different similarity measures. The cyan, blue, green, and red colors correspond to the `horizon`, `chaotic`, `fault` and `salt dome` classes.



(a) `chaotic` (b) `faults` (c) `other 1`

(d) `other 2` (e) `salt dome 1` (f) `salt dome 2`

Figure 2.9: The exemplar images of each class of subsurface structures that were used to retrieve the images from the seismic volume. One exemplar image was used for `chaotic` and `fault`, and two exemplars were used for `other` and `salt dome`. These images are of size 99×99 pixels and were obtained from the Netherlands Offshore F3 Block [3].

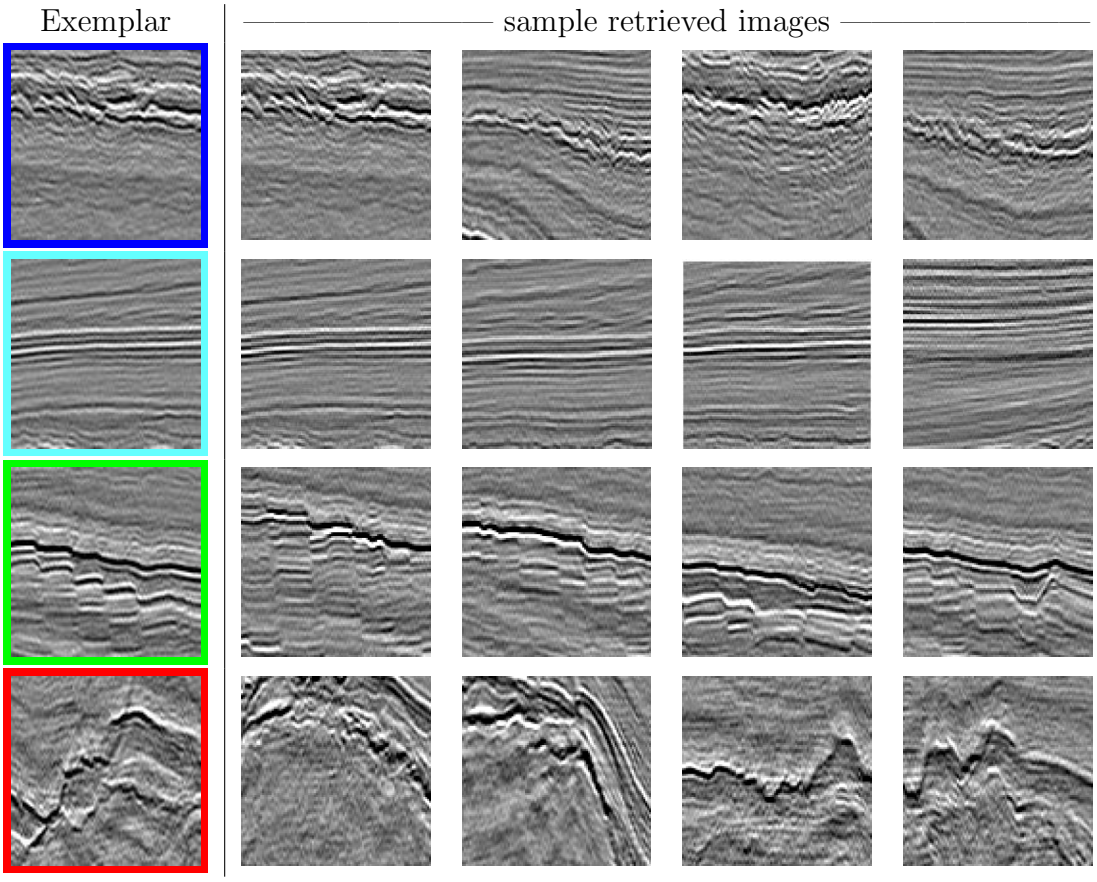


Figure 2.10: Sample retreived images from each class of subsurface structures. The first column shows one of the exemplar images for the `chaotic`, `other`, `faults`, and `salt dome` classes. These exemplar images are highlighted in blue, cyan, green, and red respectively.

CHAPTER 3
STRUCTURAL INTERPRETATION WITH WEAK IMAGE-LEVEL LABELS

3.1 Overview

In the previous chapter, we have introduced a similarity measure that can be used to retrieve large numbers of images similar to an exemplar image from within large unlabeled seismic volumes. We have shown how this similarity-based retrieval process can be used to assign image-level labels to a large number of images. In this chapter, we investigate the use of these image-level labels in the semantic labeling of subsurface structures [48, 49, 50]. Since image-level labels are used to predict subsurface structures on the pixel-level, our trained models are weakly-supervised. We often refer to labels used to train such models as 'weak' labels.

Subsurface structure labeling is the process of classifying the voxels within a seismic volume into one of many predefined structures. While labeling using image-level (rather than pixel-level) labels is not ideal, it serves as a baseline for our later work. In addition, image-level labels are far easier to obtain that pixel-level labels, and therefore can be more useful for applications where obtaining pixel-level labels is too expensive. Finally, some applications (such as object detection) do not require a highly-accurate pixel-level segmentation since more often than not, we are interested in the bounding boxes of individual objects rather than their pixel-level masks. In the case of seismic structural interpretation, such a rugged segmentation can still be useful in extracting seismic subvolumes around various subsurface structures (such as faults or salt domes) and applying advanced computational seismic interpretation techniques on these subvolumes [4].

The remainder of this chapter is organized as follows. Section 3.2 introduces our weakly-supervised framework for subsurface structure labeling using only image-level labels. In Section 3.3, we review several texture and multiresolution attributes that we used in our framework to evaluate the best feature representation that can be used for this application. We present the results in Section 3.4, and we summarize and conclude this chapter in Section 3.5.

3.2 Labeling with Image-Level Labels

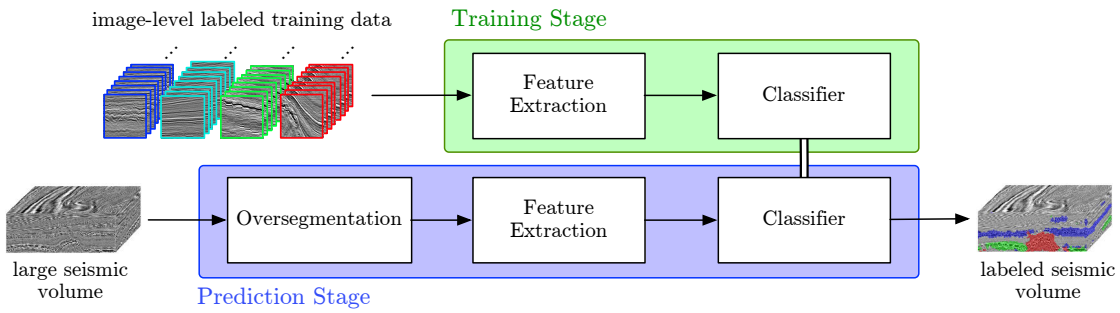


Figure 3.1: A block diagram illustrating the weakly-supervised subsurface structure labeling framework described in this chapter.

In the seismic interpretation literature, there has been a lot of interest in the semantic labeling of subsurface structures such as salt domes [51, 52, 53, 54, 55, 56, 57, 58] or faults [59, 60, 61, 62, 63]. These techniques either use classical computation seismic interpretation techniques that are not data-driven [64, 53, 54, 55, 56, 57, 60], or fully-supervised machine learning models that require strong labels [62, 65] or synthetic models [66, 67]. Other than our proposed work [48], no other work uses a weakly-supervised approach for labeling seismic structures. Although many weakly-supervised semantic segmentation methods have been proposed in the natural image domain, we review these methods in detail in Chapter 4.

The framework we propose for labeling subsurface structures using image-level labels can be divided into two main stages; the **training stage** where features ex-

tracted from the image-level labeled training data are used to train a classifier, and the **prediction stage** where the classifier is used to predict the labels of subsurface structures in seismic sections. The training stage is performed once; then once the classifier is trained, it is used in the prediction stage to predict the labels of every seismic section of interest. The overall process of labeling subsurface structures using image-level labels is illustrated in Figure 3.1

3.2.1 Training Stage

Our training process involves two steps. The first step is to extract characteristic features from each image in the training set to form a feature vector. Before extracting these features from an image $\mathbf{x}_i$, we first calculate the Hadamard product of this image with a two-dimensional Gaussian kernel of the same size, $\mathbf{g}$. This kernel gives more weight to the structures at the center of the image and less to those on the periphery, thus emphasizing local spatial correlations in seismic data. The procedure can be expressed as follow:

$$\widetilde{\mathbf{x}}_i = \mathbf{x}_i \odot \mathbf{g}, \tag{3.1}$$

where $\odot$ is the Hadamard product, and the Gaussian kernel is defined as

$$\mathbf{g}[x, y] = e^{\frac{(x-\mu_x)^2+(y-\mu_y)^2}{2\sigma^2}}, \tag{3.2}$$

where μ_x and μ_y are the x- and y- coordinates of the center of $\mathbf{x}_i$, respectively. The value of σ was set to 25 in our experiments so that pixels in the corners of the image have weights of less than 1%.

After this pre-processing step, one of many different texture and multiresolution techniques is applied to the image to generate the feature vector. The texture features include the grey level co-occurrence matrix (GLCM)[68], local radius index (LRI)[69], local binary patterns (LBP)[70] and many of its variants (completed LBP

(CLBP)[71], multiscale CLBP (M-CLBP), extended LBP (ELBP)[72], and completed local derivative pattern (CLDP)[73]). The multiresolution features include features from non-directional transforms such as discrete and stationary wavelets, in addition to directional multiresolution features from Gabor filters[74], steerable pyramids[17], contourlets[75], non-subsampled contourlets[76], and curvelets[13]. These different techniques are listed in Section 3.3.

In the second step of the training process, the feature vectors extracted from the training images are used to train an image classifier. We use a support vector machine (SVM) [77] as the classifier, which is a powerful binary classification algorithm. It seeks to find the optimal separating hyperplane between two classes by identifying the one with the maximum margin. Since we have a multi-class classification problem, we train four hard-margin SVMs with linear kernels using the one-versus-all (OVA) approach.

3.2.2 Prediction Stage

The prediction stage consists of three main steps. First, for every seismic section to be labeled, an image over-segmentation is performed such that the section is automatically divided into small segments that align with the local structures within the section. To achieve this, we use a superpixel-based over-segmentation approach that groups neighboring pixels of similar appearance into a single cluster. Oversegmentation algorithms, like normalized cuts [78], are sometimes used in computational seismic interpretation to extract subsurface structures [79, 80, 57]. However, oversegmentation here is used as a preprocessing step to enforce local spatial correlation by grouping pixels in the volume that are similar and close to each other. Here, the lack of clearly-defined boundaries between subsurface structures is inconsequential, and instead, each small segment is classified based on its texture content. This step also significantly reduces the computational cost of the labeling, since each segment

will be classified once, and the resulting label will then be propagated to all the pixels within that segment.

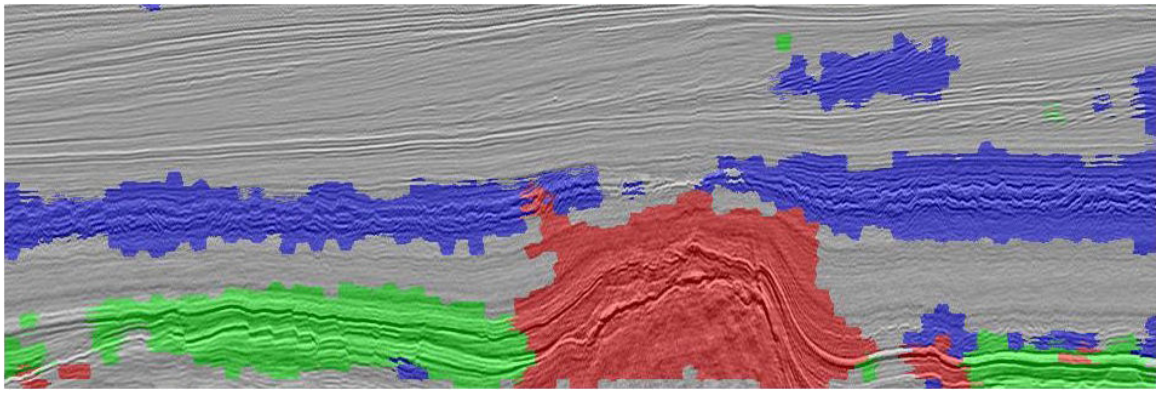
In this work, we use a superpixel segmentation algorithm based on the simple linear iterative clustering (SLIC) algorithm [81]. Graph-based over-segmentation techniques (such as graph cuts [82] and normalized cuts [78]) or gradient descent based approaches (such as the watershed algorithm [83] and turbo pixels [84]) are computationally expensive, and therefore not suitable for large seismic volumes. Therefore, simpler and more computationally efficient SLIC is a more appropriate choice for this application. In the original SLIC algorithm, vectors in the form of $[l, a, b, x, y]$ are generated for each pixel in an image to be segmented, where l, a, and b are the three components of the *Lab* color model, and x, y are the coordinates for each pixel. Then clustering is performed in a space formed by these vectors to obtain the superpixels. Because seismic images are grayscale, we compute vectors for the pixels in the form $[l, g_x, g_y, x, y]$, in which g_x and g_y refer to the gradient along the x- and y- directions, respectively. This helps align the SLIC superpixels to the small edges in the seismic section.

In the second step of the prediction stage, similar to the training process, texture attributes are extracted for each superpixel. Typically, the size of a superpixel is smaller than that of the images in the training dataset. To make sure that the attribute extraction is consistent between the training and the prediction stages, we extract a neighborhood of the same size as that of the training images centered around the centroid of the superpixel. These extracted neighborhoods are then multiplied with the Gaussian kernel $\mathbf{g}$ as in Equation 3.1. Then features extracted from these images are used to represent the superpixel at their center.

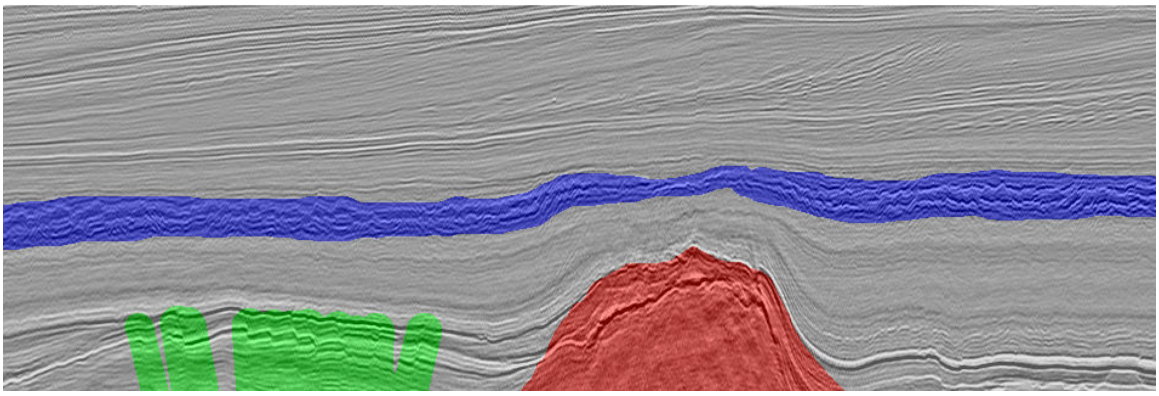
Finally, in the last step of the prediction stage, the feature vectors generated for each superpixel are fed into the SVM classifier. The classifier then classifies these superpixels into the different subsurface structures. This process is done for all the

superpixels in all the sections of the seismic volume. This leads to labeled seismic sections such as the one in Figure 3.2.

Figure 3.2 shows inline #380 from the Netherlands North Sea F3 block labeled using our workflow with curvelet features, in addition to the manually annotated result. One of the disadvantages of relying solely on image-level labels is evident in the figure. The `faults` exemplar image contains strong seismic reflectors in addition to multiple faults, and since we limited ourselves to a single exemplar image for the `faults` class, our classifier labeled most strong reflectors as belonging to the `faults` class.



(a)



(b)

Figure 3.2: (a) A labeled seismic section from the Netherlands North Sea F3 block database using curvelet features. (b) A manually annotated seismic section. The `chaotic` class is in blue, `faults` is in green, and `salt dome` is in red.

3.3 Comparison with Various Texture and Multiresolution Features

In the feature extraction stage in Figure 3.1, any feature representation can be used to test this workflow. Naturally, we would like to investigate the most suitable feature representation for this application. We conducted two comparative studies to examine various texture and multiresolution feature representations in the context of subsurface structure labeling [49, 50]. Seven feature representations were tested in each study.

In the first study[49], our focus was on a group of spatial attributes and local descriptors, including the local binary pattern (LBP), a few of its variants, and the local radius index (LRI) [85]. These attributes have been widely used for texture representation in the literature. For comparison purposes, the study also included a traditional seismic attribute computed in the spatial domain, namely, the GLCM. In the second study[50], we examined multiresolution attributes in the frequency domain for subsurface structure labeling. We examined the discrete wavelet transform and its nonsubsampled version, Gabor filters, the steerable pyramid, the contourlet transform and its nonsubsampled version, and finally, the curvelet transform.

The fourteen texture and multiresolution feature representations that we tested are briefly described below.

1. **Texture features**

    1.1. **LBP:** The local binary pattern (LBP) is a simple and efficient texture feature representation, which has become a standard local texture descriptor in the spatial domain [70]. It describes the intensity difference between a pixel and its local circular neighborhood, denoted by (P, R), where P defines the number of pixels evenly distributed on the circular neighborhood with radius R. To ensure robustness against intensity changes, LBP employs the signs of the differences instead of the exact values to form

unique binary codes for the description of local texture patterns.

1.2. **CLBP:** The completed local binary pattern (CLBP) is an LBP variant that adds the intensity of the neighbors and the center pixel to the standard LBP feature representation.

1.3. **M-CLBP:** The multi-scale CLBP (M-CLBP) combines CLBP features computed at different radii, making it a multi-scale version of the CLBP.

1.4. **ELBP:** Unlike the LBP that computes features based on neighbor-center difference, the extended LBP (ELBP) computed features based on center intensity, neighbor intensity, and radial difference. Unlike the M-CLBP that considers each scale separately, ELBP incorporates the correlation of the features at different scales.

1.5. **CLDP:** The completed local derivative pattern (CLDP) adds cross-scale correlation to the CLBP implementation through radial *sign* difference.

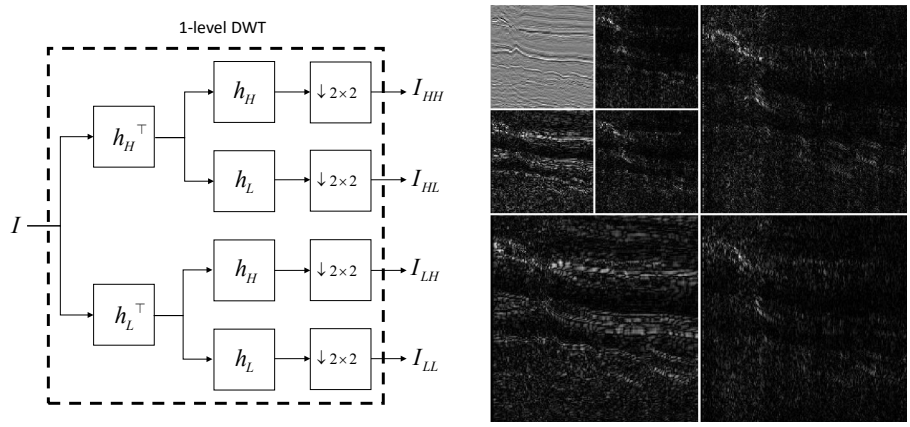
1.6. **LRI:** Although LBP and its varients implicitly capture the edge information, the local radius index (LRI) [69] provides a more explicit description of the spatial distribution of edges. It achieves this by characterizing texture patterns using the local distribution of distances between adjacent edges along a particular angle.

1.7. **GLCM:** Attributes based on the grey level co-occurrence matrix (GLCM) have been widely accepted as useful tools for texture analysis since they were proposed four decades ago [68]. The have also been widely adopted in seismic data processing and interpretation. The GLCM is a matrix that describes the co-occurrence pattern between gray levels of two neighboring pixels along a particular direction in an image. In essence, it represents a two-dimensional histogram that approximates the joint probability distribution of the adjacent gray values. It can capture textural patterns for the

selected neighborhood along the prescribed direction. For example, high values away from the diagonal in a GLCM reveal sharp changes in gray level, whereas high values close to the diagonal indicate small variations.

2. **Multiresolution features**

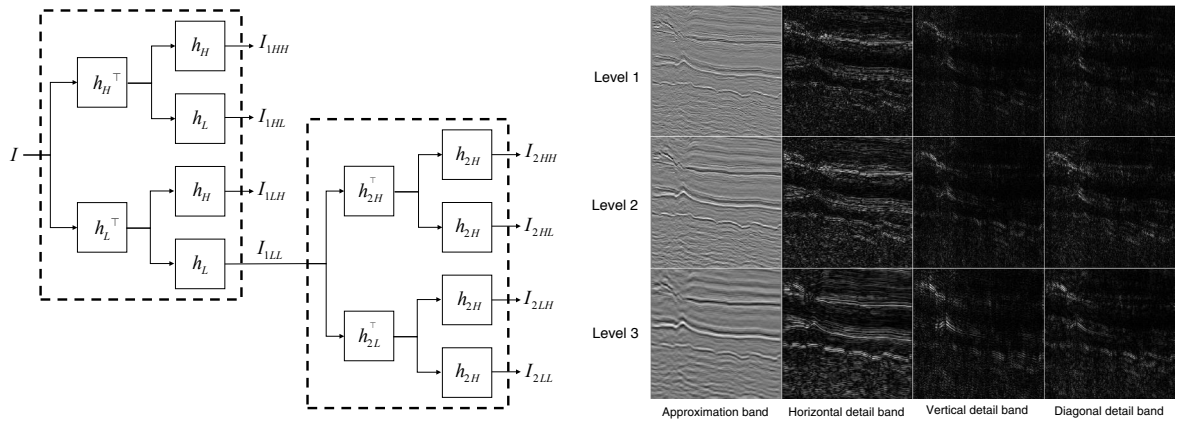
   2.1. **Discrete wavelet:** The discrete wavelet transform (DWT) is an orthonormal transform that represents an image using a dyadic dilation and translation of a function called the mother wavelet. The mother wavelet is localized in both the spatial and frequency domains. Different wavelets have been proposed and studied extensively such as Haar, Daubechies, symlet, Mexican hat, coiflet wavelets and many others [86]. The first level discrete wavelet coefficients of an image I are obtained by filtering along the horizontal direction with low pass and high pass filters to obtain I_L and I_H, respectively. Then, $\hat{I}_L$ and $\hat{I}_H$ are filtered along the vertical direction with the same filters and decimated by a factor of 2 to obtain detail images I_{HH}, I_{LH}, and I_{HL}, and an approximation image I_{LL} as shown in Figure 3.3a. For more levels, the same process is repeated on the approximation image I_{LL}. An example of a 2-level DWT of a seismic image is shown in Figure3.3b.



(a) (b)

Figure 3.3: (a) Block diagram of a 2D 1-level DWT decomposition. (b) A 2-level DWT of a seismic image.

2.2. **Stationary wavelet:** The stationary wavelet transform (SWT) is an improved version of DWT that overcomes its lack of shift-invariance. The SWT can be found in the literature with different names like the shift-invariant DWT (SIDWT)[87], the undecimated DWT (UDWT)[88], the overcomplete DWT (ODWT)[89], and the redundant DWT (RDWT)[90]. One way to implement the SWT is to remove the downsampling step from the DWT and, instead, upsample the filters in each step as shown in Figure 3.4a where h_{2H} and h_{2L} are upsampled versions of h_H and h_L, respectively. This slight modification makes SWT a shift-invariant, but redundant, transform. An example of a 3-level SWT of a seismic section is shown in Figure 3.4b.



(a) (b)

Figure 3.4: (a) 2-level decomposition for 2D SWT. (b) 3-level SWT of a seismic image.

2.3. **Gabor filter:** The Gabor filter is a linear filter whose impulse response is the product of a plane wave with a Gaussian kernel. The Gabor filter is frequently used to model the simple cell receptive fields in the human visual system [74]. Thus, it has been utilized to characterize natural and texture images especially for applications such as edge detection [91] and segmentation[92]. The impulse response of a Gabor filter centered at the origin with orientation θ and a radial frequency ω is given by:

$$H(x, y, \omega, \theta) = \exp\left\{-\frac{x^2 + y^2}{2\sigma^2}\right\} \exp\left\{i2\pi\omega\left(x\cos\theta + y\sin\theta\right)\right\} \tag{3.3}$$

Figure 3.5 shows a 2-scale and 4-orientation Gabor filter bank.

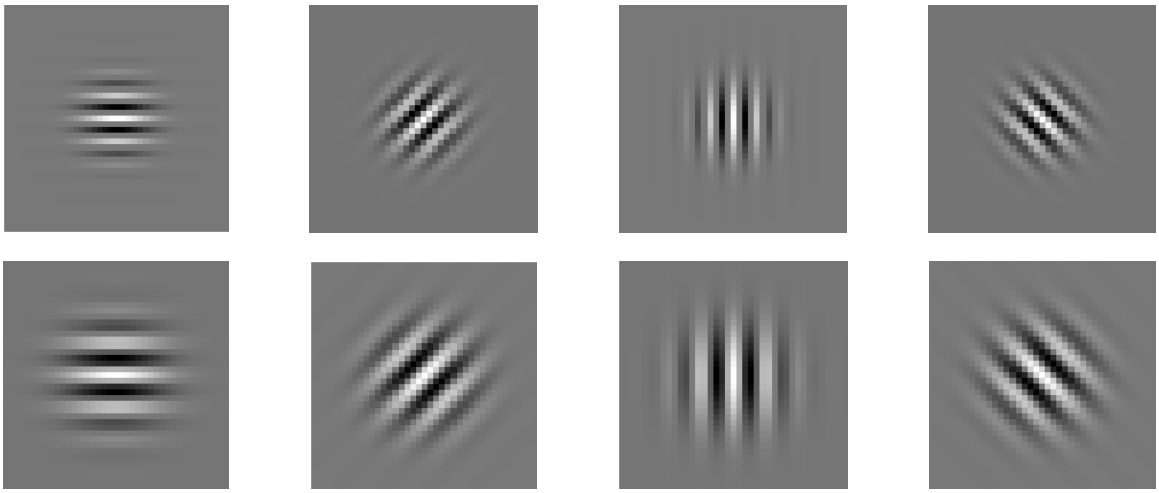


Figure 3.5: Gabor filters at two different scales and four different orientations.

2.4. **Steerable pyramid:** The steerable pyramid is a multiscale image decomposition developed by Simoncelli *et al.* [17]. As shown in Figure 3.6, the image is first decomposed into highpass and lowpass subbands and then the lowpass band is further decomposed into bandpass subbands of different orientations and a lowpass subband. The lowpass subband is then subsampled and passed as an input to a similar decomposition to obtain details at other scales. The bandpass filters capture details at different orientations and the subsampling allows it to capture details of different scales.

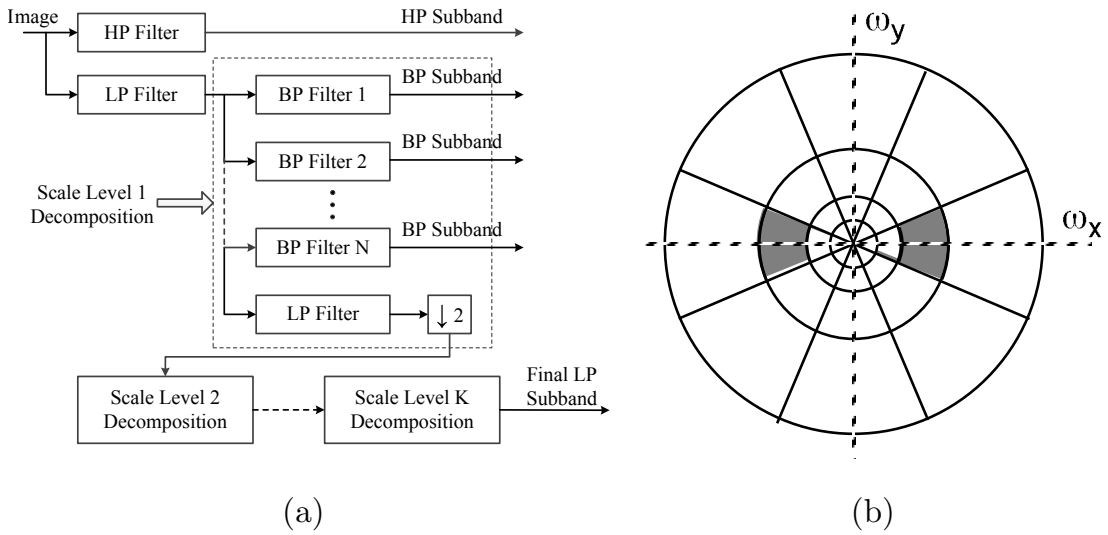


Figure 3.6: (a) Steerable pyramid filter bank and (b) Steerable pyramid spectral decomposition with 4 orientations and 4 scales. Figure adapted from [17] with permission. ©(1995) IEEE.

2.5. **Contourlets:** The contourlet transform [75] is a separable multiscale directional transform that employs iterated filter banks. The separability property makes the contourlet transform more efficient and faster to compute than other non-separable transforms. The contourlet transform is constructed based on the Laplacian pyramid [93]. The lowpass output of the pyramid is further decomposed with a biorthogonal wavelet. A directional filter bank initially proposed by [94] is then applied to each image output of the Laplacian pyramid. Figure 3.7 shows filter bank used in the contourlet transform.

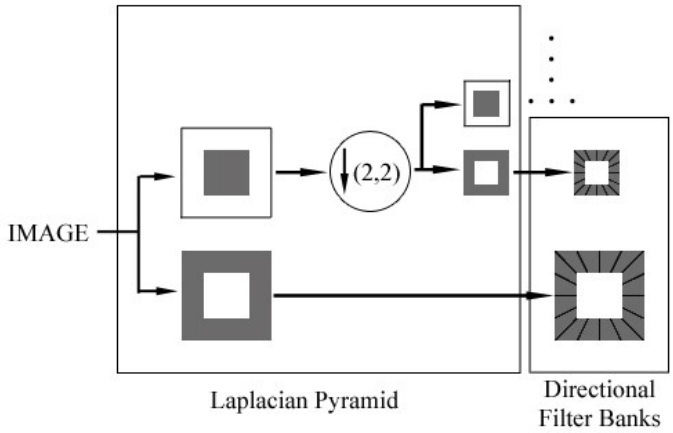


Figure 3.7: The contourlet filter bank (adapted from [75].)

2.6. **Non-subsampled contourlets:** The nonsubsampled contourlet transform is a translation-invariant version of the contourlet transform that was developed by [76]. The decomposition has two components. The first component is the nonsubsampled pyramid that performs a multiscale decomposition. The second component is the directional filter bank that performs directional decomposition. The nonsubsampled directional filter banks lead to better frequency localization when compared to the standard contourlet transform. Similar to the improvement of the SWT on the DWT, the nonsubsampled contourlet transform improves on the contourlet transform by being fully shift-invariant, at the cost of increased redundancy. Figure 3.8 illustrates the non-subsampled contourlet transform.

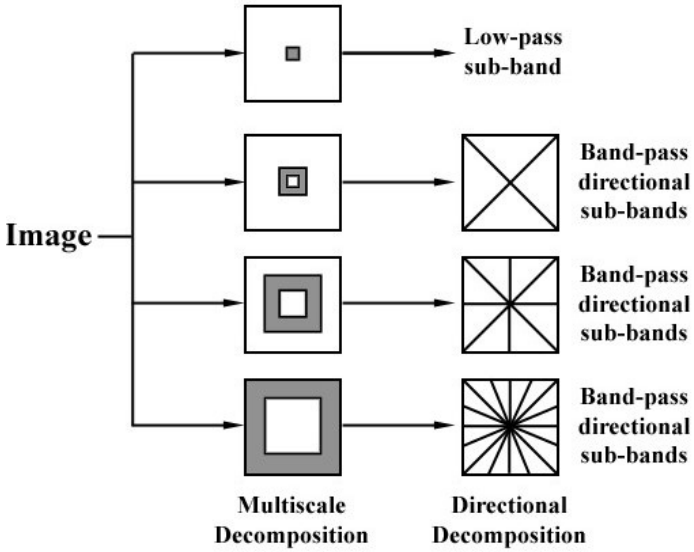


Figure 3.8: The nonsubsampled countourlet filter bank. (adapter from [95])

2.7. **Curvelets:** The curvelet transform is a directional multiscale decomposition. Curvelet frames have been shown to represent images with geometrically regular edges (such as seismic images) more compactly than other traditional multiscale representations [32]. It has also been widely used in various seismic processing and interpretation problems. Therefore would be a good multiscale feature representation to investigate for this application. For a more detailed description of the curvelet transform, see Section 2.3.

3.4 Results

We use our proposed framework to perform a labeling experiment to compare the structural interpretation results of the various texture and multiresolution features highlighted in the previous section. For this experiment, we use a subset of the publicly available LANDMASS-1 dataset [46], to form the training dataset. LANDMASS-1 consists of more than 17,000 seismic images of size 99×99 pixels extracted from the Netherlands F3 block dataset [3]. These images contain various subsurface structures

such as horizons, chaotic regions, faults, and salt domes. Since not all the images in LANDMASS-1 are suitable for our labeling task, we use the similarity-based retrieval method explained in the previous chapter to retrieve images that have the most similarity to each of the exemplar images shown in Figure 2.9. We assign these images to four classes: `chaotic`, `faults`, `salt domes`, and an `other` class that would contain all the other images that contain structures not in the first three, such as clear horizons and sigmoidal structures. The `other` class serves the purpose of showing negative examples of structures that do not belong to the first three classes. Although the dataset contains these specific structures, our proposed labeling framework can be extended to other seismic structures as well. Overall, we retrieve 1000 images for the `other` class, 1500 for `salt domes`, and 500 each for `chaotic`, and `faults`.

To objectively compare the results, we use a manually annotated seismic inline from the Netherlands F3 block dataset. Namely, we use seismic inline #380 labeled with regions that contain the `chaotic`, `faults`, and `salt dome` structures. These structures correspond to the blue, green, and red regions shown in Figure 3.2(b). The quantitative labeling results using our framework for various feature representations is summarized in Table 3.1. The evaluation metrics that we use are detailed in Appendix A.3. In addition, Figure 3.9 shows the labeling results for the various texture feature representations, while Figure 3.10 shows the results for the various multiresolution feature representation.

The results in Table 3.1 show that the GLCM and the ELBP features are the best performing texture features. It is not very surprising that the GLCM performs well since it has been widely used in seismic interpretation [98, 99, 100]. However, the ELBP shows more promise as it has slightly outperformed the GLCM. By observing the results in Figure 3.9, we note that many feature representations label the entire horizons where the faults occur as belonging to the `faults` class. The ELBP and GLCM are notable exceptions. However, the model trained with GLCM features

	Feature	PA	MIU	FWIU
Texture	LBP	0.649	0.360	0.540
	CLBP	0.768	0.483	0.668
	M-CLBP	0.765	0.486	0.663
	ELBP	0.792	0.454	0.692
	CLDP	0.724	0.457	0.615
	LRI	0.712	0.466	0.614
	GLCM	0.779	0.483	0.688
	Mean texture features	0.757	0.472	0.657
Multiresolution	Discrete wavelet *(non dir.)*	0.599	0.362	0.486
	Stationary wavelet *(non dir.)*	0.570	0.348	0.452
	Gabor filters	0.759	0.478	0.664
	Steerable pyramid	0.789	0.498	0.691
	Contourlet	0.769	0.492	0.667
	Non-subsampled countourlet	0.738	0.455	0.635
	Curvelet	**0.820**	**0.550**	**0.725**
	Mean directional multiresolution features	0.775	0.4946	0.6764

Table 3.1: Evaluation of the labeling performance for various texture and multiresolution features.

makes a similar mistake with the `salt dome` class, where the strong seismic reflections around the salt dome boundary are mistaken for the salt body.

As for the multiresolution feature representations, Table 3.1 shows the superiority of the directional multiresolution features over the non-directional ones (discrete and stationary wavelet) by a significant margin. This is expected since seismic images contain highly directional features, that cannot be captured by non-directional representations. The curvelet features outperform all the other feature representations on all the metrics we measured. This is not unexpected since the curvelet transform has been used very successfully in a wide range of seismic interpretation and processing applications [26, 34, 36, 35, 101, 102, 103, 104]. The remaining texture and directional multiresolution features perform comparably well, but not to the level of the curvelet features. On average, directional multiresolution features perform better

Table 3.2: The percentage of pixels from each class in the manually labeled inline # 380.

other	chaotic	faults	salt dome
79.74 %	9.38 %	4.49 %	6.39 %

than the texture features achieving slightly more than 2% FWIU score higher than their texture counterparts.

In terms of pixel accuracy, which gives no regard to individual classes, the curvelet features outperform all the other features with a significant margin. Table 3.2 shows the percentage of pixels that belong to every class in inline # 380. The MIU metric which normalizes the results of every class by its size shows an even more substantial advantage for the curvelets over other feature representations. One reason why it performs very well can be the curvelet transform's effectiveness in representing curve-like features which constitutes a large portion of the seismic section.

It is important to note that these results were obtained on a single seismic inline, and therefore, these results will differ if other seismic sections were used. However, it is not difficult to conclude that given the computational advantages of multiresolution feature representations, they are the preferred to the computationally demanding texture feature representations. This computational advantage is especially evident in the case of the curvelet transform, where the fast discrete curvelet transform (FDCT) is computationally efficient and seems to outperform all the other texture and multiresolution feature representations that we have tested.

3.5 Summary

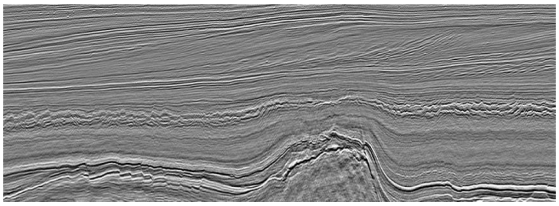
In summary, this chapter investigates the use of image-level labels to train a weakly-supervised classifier for semantically labeling seismic structures. We introduced a framework that extracts features from images that were assigned image-level labels, and then use these features to train a classifier. During the prediction stage, each

seismic section is over-segmented into a large number of superpixels that contain similar textures. Features from these superpixels are then extracted and classified. The resulting label is then propagated to all the pixels in the superpixel. This process is done for all the superpixels in every section and repeated for every section in the seismic volume. We studied various feature representations, both texture features that are computed in the spatial domain, and multiresolution features that are computed in the frequency domain. We have compared the results of labeling a seismic inline using different feature representations, and we have concluded that the curvelet features seem to outperform the other representations by a substantial margin.

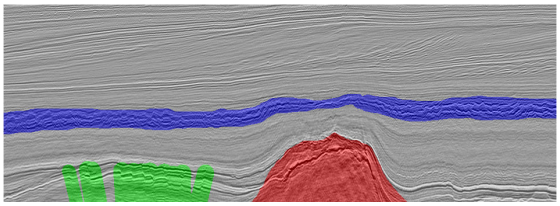
We have shown that image-level labels can indeed be used to obtain a pixel-level classification of seismic structures. However, this approach does have several disadvantages that limit its potential. First, the framework we proposed assigns class labels to superpixels, not pixels. This allows the results to be more spatially coherent and reduces false classifications; however, this also reduces the resolution of the model output to the scale of the individual superpixels. This might be acceptable based on the application, but ideally, we would like to achieve true pixel-level labeling. Second, and more importantly, since image-level labels do not encode the location of the target class, the classifier might not correctly learn features that are associated with a given structure. This is especially true for structures that are small in scale, or are subtle in nature such as the `faults` structures. For example, a large number of fault images in the training dataset have strong seismic reflections (e.g., see Figure 2.10). These strong reflections are a much more dominant feature than the faults themselves, making the classifier confuse images with strong seismic reflections as belonging to the `faults` class. This confusion is not only a shortcoming of the feature representations but, more importantly, due to the nature of the labels that were used in the training. Every image was assigned an image-level label which in turn leads the classifier to associate the strong features present in each image with

its image-level label. This is one of the main drawbacks of using image-level labels to train a semantic segmentation model. While we do not have access to pixel-level training labels, this does drive the question of whether these image-level labels can be projected somehow to pixel-level labels that encode the locations of the target classes in the training images, and wether models trained using these projected pixel-level labels would perform better compared to those trained using image-level labels.

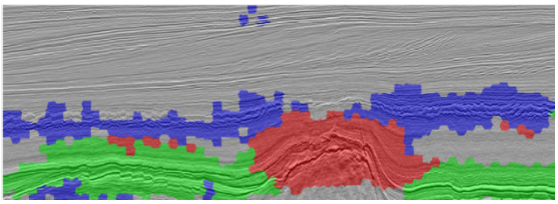
In the next chapter, we present a novel label-mapping algorithm that projects image-level labels into pixel-level labels. Furthermore, in Chapter 5 we show how a deep learning model can be trained using these projected pixel-level labels and we compare the resulting pixel-level annotations with the ones we achieved in this chapter.



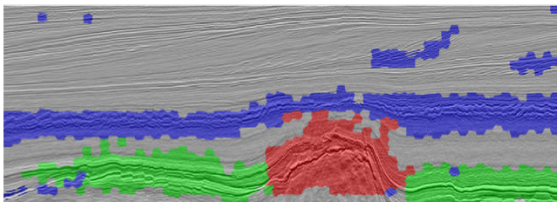
(a) Original seismic section



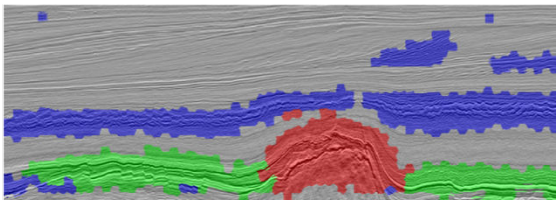
(b) Manually annotated



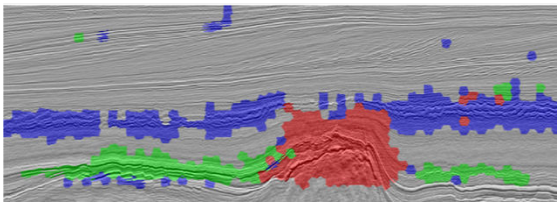
(c) LBP [70]



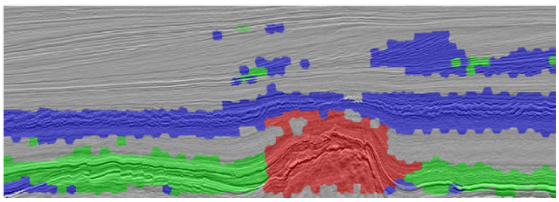
(d) CLBP [71]



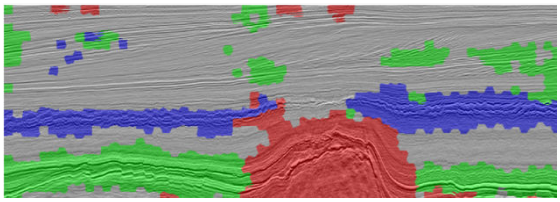
(e) MCLBP [70]



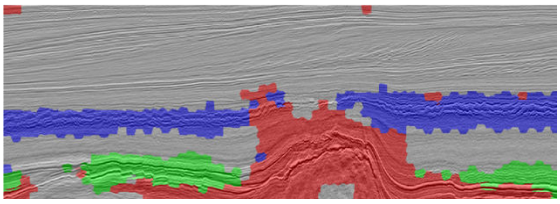
(f) ELBP [72]



(g) CLDP[96]

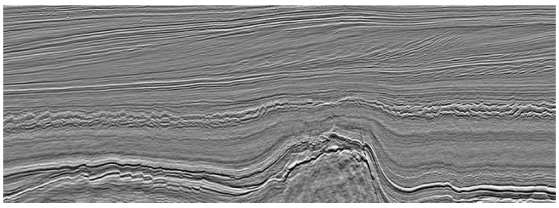


(h) LRI [69]

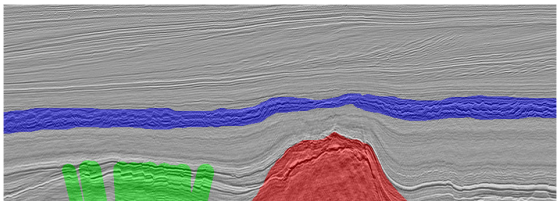


(i) GLCM [97]

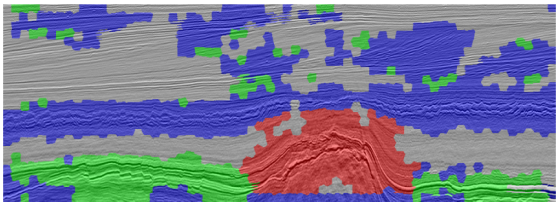
Figure 3.9: Results of our image-level labeling framework on inline #380 of the Netherlands F3 block using texture features. The colors blue, green, and red correspond to the `chaotic`, `faults` and `salt dome` classes respectively.



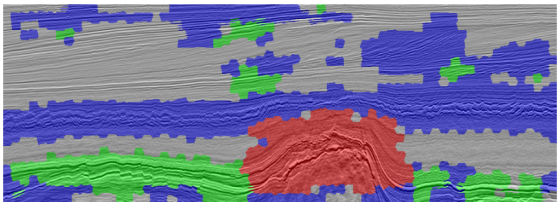
(a) Original seismic section



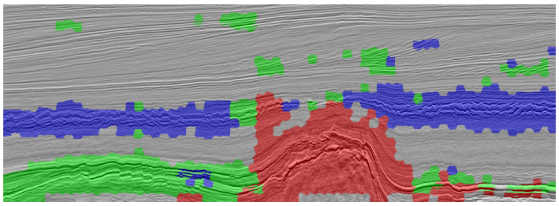
(b) Manually annotated



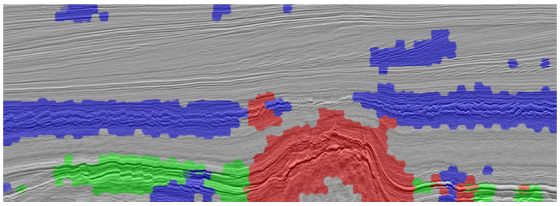
(c) Discrete Wavelet [86]



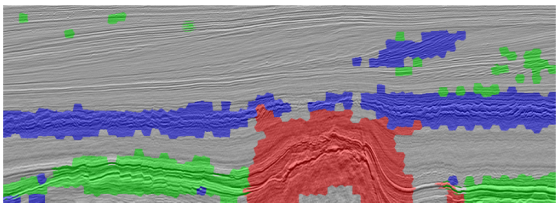
(d) Stationary Wavelet [87]



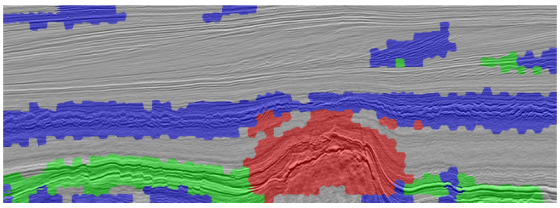
(e) Gabor Filters [74]



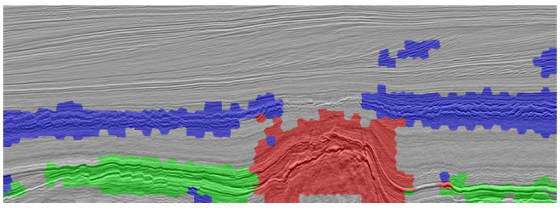
(f) Steerable Pyramid [105]



(g) Contourlet[75]



(h) Non-subsampled Contourlet [95]



(i) Curvelet [32]

Figure 3.10: Results of our image-level labeling framework on inline #380 of the Netherlands F3 block using multiresolution features. The colors blue, green, and red correspond to the `chaotic`, `faults` and `salt dome` classes respectively.

CHAPTER 4
WEAKLY-SUPERVISED LABEL MAPPING

4.1 Overview

While image-level labels are easy to obtain, learning to label structures in seismic volumes using only image-level labels is very challenging as we have seen in Chapter 3. One of the main challenges when training semantic segmentation models using image-level labels is the inherent tradeoff between localization and classification accuracy. Models that were trained to classify on the pixel-level (high localization accuracy) will invariably have poor classification accuracy, and the only way to obtain higher classification accuracy is to sacrifice the localization accuracy (i.e., the final resolution of the output predictions). The method we proposed in Chapter 3 attempted to navigate this tradeoff by having a somewhat moderate localization accuracy (superpixel-level) and a similarly moderate classification accuracy. However, it is fair to assume that if we decouple this localization/classification problem into two separate problems, we can achieve both higher localization and higher classification accuracy.

In this chapter, we address the localization accuracy problem by presenting a weakly-supervised label mapping algorithm that transforms image-level labels into pixel-level labels that encode the locations of the various target classes within each image in our training set. The classification accuracy problem is addressed in Chapter 5.

Once image-level labels are obtained, we can learn features that are common between images from each class. These class-specific features are then used to identify the pixel-level labels for each pixel in the training images. Our weakly-supervised label mapping approach is based on non-negative matrix factorization. This mapping is a

weakly supervised one since every image $\mathbf{x}_i \in \mathbb{R}_+^{n \times m}$ has only one label as opposed to $n \times m$ labels. The remainder of this chapter is organized as follows. Section 4.2 reviews the relevant literature and summarizes the main approaches and their advantages and disadvantages. Section 4.3 introduces non-negative matrix factorization and the notation that will be used throughout the remainder of this chapter. Section 4.4 introduces the constraints that we impose on our NMF formulation, and section 4.5 explains how our formulated optimization problem is solved. Section 4.6 then explains how the final results are obtained, and section 4.7 explores the results of our label mapping algorithm. Finally, section 4.8 summarizes and concludes this chapter.

4.2 Background

The problem of assigning semantic class labels to pixels is known as semantic segmentation. This problem, sometimes known as scene labeling, scene parsing, or semantic labeling, is a major computer vision research problem. The importance of semantic segmentation is that it paves the way towards higher-level image understanding, an essential computer vision task, with wide-ranging applications from image search engines [106] to autonomous cars [10] and augmented reality [107]. Figure 4.1 demonstrates the goal of semantic segmentation compared to other common computer vision tasks such as image classification, and object detection. Mapping image-level labels to pixel-level labels is a form of semantic segmentation, and therefore, for the sake of generality, we do not make any distinction between the two in this literature review.

The literature on weakly-supervised semantic segmentation in recent years can be divided into two main approaches. The first approach is based on the use of convolutional neural networks (CNNs). These techniques are mainly driven by the widely successful application of CNNs to many computer vision tasks. CNNs typically have a very large number of trainable free parameters. Therefore the main challenge with weakly-supervised CNN based techniques is to find ways to enlarge the training set,

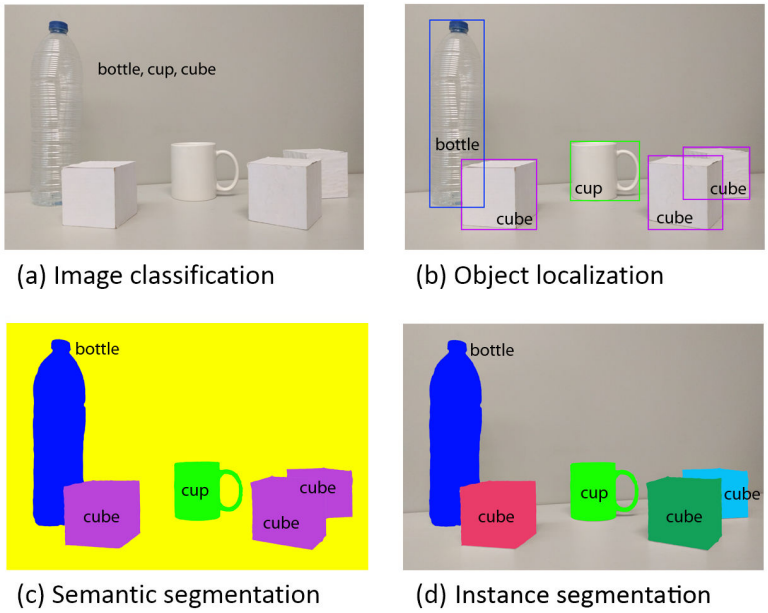


Figure 4.1: A comparison of various scene understanding tasks in computer vision, from coarse (image classification) to fine (instance segmentation). Figure adapted from [108] with permission. All rights reserved ©2017 Elsevier.

find auxiliary sources of supervision, or restrict the parameter space of the network. The second approach attempts to overcome this challenge by using matrix completion or factorization techniques that do not require large amounts of data. Methods that currently use this approach are rather limited in number and typically do not provide good spatial resolution.

In the following two subsections, we review the literature on weakly-supervised semantic segmentation using CNNs (Section 4.2.1) and matrix completion and factorization (Section 4.2.2). Then in Section 4.2.3, we summarize the two approaches and analyse their main advantages and disadvantages.

4.2.1 Convolutional Neural Networks

In recent years, there has been a considerable amount of research on using CNNs for semantic segmentation [108]. Most of the proposed methods take advantage of

large-scale fully-annotated image datasets and therefore allow for powerful machine learning algorithms to be trained. In many cases, however, fully-annotated data is not available and can be challenging to obtain. In this case, weakly-supervised techniques for semantic segmentation can be used. These techniques use weak labels that are less costly and much easier to obtain. Recently, many weakly-supervised labeling methods based on CNNs have been proposed. Some of these methods achieved comparable performance to fully-supervised methods on standard benchmarks. In this subsection, we review both strongly- and weakly-supervised techniques that are based on CNNs.

A major hurdle for the successful end-to-end application of *fully-supervised* CNNs to semantic segmentation was what seemed like a trade-off between classification and localization accuracy. Deeper networks that have multiple pooling layers have proven to be the most successful models in image classification tasks. However, their large receptive fields and increased spatial invariance (due to pooling and convolutional layers) make it difficult to infer the locations of various objects within the image using the output scores at the upper layers of the network. Many researchers attempted to overcome this hurdle by using various pre- or post-processing techniques. For example, Mostajabi *et al.* proposed representing small image superpixels using a combination of local, regional, and global features obtained from a CNN and then classifying them using a shallow neural network. Others proposed using probabilistic graphical models such as fully-connected conditional random fields (CRFs)[109] to process the coarse score maps from CNNs to finer more accurate ones [110, 111]. In addition, Yu and Koltun [112] have investigated using methods such as dilated convolutions that aggregate multi-scale contextual information while preventing the network from losing spatial resolution as it got deeper, while others provide multiple downsampled versions of the image as input to the network and then combine the multiscale predictions into a single output map [113, 114, 115].

One of the early milestones towards *fully-supervised* end-to-end semantic segmen-

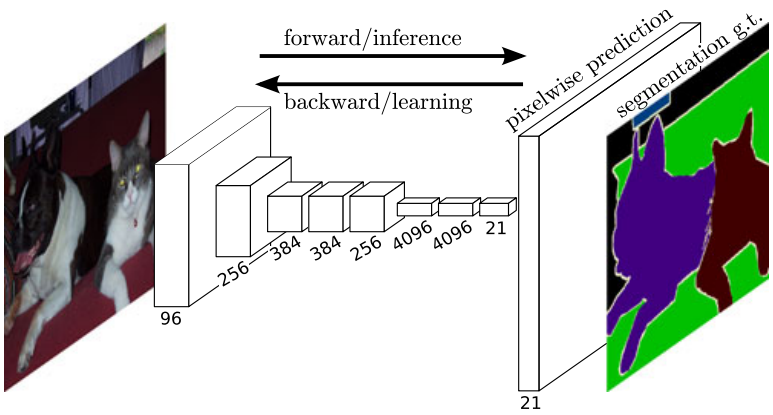


Figure 4.2: An illustration of the fully convolutional network (FCN) architecture. Reprinted, with permission from [116]. All rights reserved ©2016 IEEE.

tation using CNNs was the fully convolutional network (FCN) proposed by Long *et al.* [116]. This work showed that it is possible to achieve good semantic segmentation results using a fully convolutional network (with no fully-connected layers), and no pre- or post-processing steps. FCNs achieve this by replacing the fully-connected layers of the CNN with convolutional layers that produce coarse feature maps. These coarse feature maps are then upsampled, and concatenated with the scores from intermediate feature maps to form a more detailed output. Due to their fully-convolutional architecture, FCNs can be applied to arbitrary sized inputs, and therefore do not require the input image to be resized as in other methods. Figure 4.2 illustrates the FCN architecture.

Another influential *fully-supervised* end-to-end method was DeconvNet proposed by Noh *et al.*[117, 118]. As opposed to the upsampling layers in FCN, this method uses a symmetric encoder-decoder style network composed of stacks of convolutional and pooling layers in the encoder, and stacks of deconvolutional and unpooling layers in the decoder that mirror the encoders architecture. The role of the encoder can be seen as doing object detection and classification, while the decoder is used for accurate localization of these objects. Several encoder-decoder style CNNs for

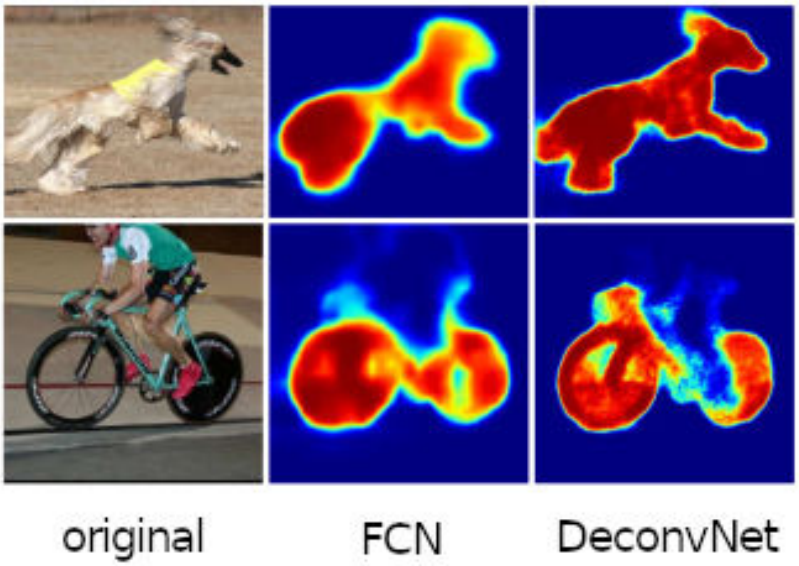


Figure 4.3: An illustration of the difference between the outputs of a) FCN and b) DeconvNet. The DeconvNet activation maps are more precise and contain far more fine details compared to FCN. Reprinted, with permission from [117]. All rights reserved ©2015 IEEE.

semantic segmentation were proposed around the same time. While DeconvNet [117] was designed to be applied on region proposals, SegNet [119] has an almost identical architecture but is applied directly on the input image. U-Net [120] improves on the base encoder-decoder architecture by concatenating the encoder feature maps with the corresponding decoder feature maps. This greatly improves how the network is trained since the backpropagated gradients can 'skip' many convolution layers using the concatenated features. Such encoder-decoder style networks can achieve finer and more accurate results than those of the FCN. Figure 4.3 shows examples of output feature maps from FCN and DeconvNet for two different classes. It is clearly evident that encoder-decoder style networks, such as DeconvNet, lead to better semantic segmentation results than FCN-like networks. However, both require vast amounts of *fully-labeled*[1] images to fully train the network.

[1]We use the terms 'strongly-labeled' and 'fully-labeled' interchangeably.

Yet, obtaining vast amounts of fully-labeled data is a very costly task. Motivated by this problem, there has been considerable interest in CNN-based techniques that only require *weak supervision*. We briefly survey the main approaches in this area; however, we limit our scope to methods that use image-level labels as their primary form of weak supervision. Using image-level labels for semantic segmentation is challenging due to the missing spatial information for each label. Some researchers formulate this problem as a multiple instance learning (MIL) problem. MIL is a formulation of weakly-supervised learning where training instances are arranged in sets, called "bags", and training labels are provided for entire bags and not instances. For semantic segmentation, the instances are often pixels or superpixels, and the bags are images. Towards this end, several MIL approaches have been proposed [121, 122, 123, 124]. However, some of these methods are not trained end-to-end and require multiple forward passes to localize single objects, making it difficult to scale them to large datasets, such as large seismic volumes. Alternatively, other methods have been proposed that use a recursive refinement procedure based on expectation-maximization, where the pixel-level labels are predicted, and then used as new ground truth annotations to update the model [125, 126, 127]. Also, others have proposed using superpixels to constrain the pixel-level labels to neighborhoods of pixels that are visually similar [122, 128], or exploit side-information such as saliency maps that highlight salient objects in images [126, 129, 130]. Oquab *et al.* [131] proposed an approach for *discriminative localization* using CNNs based on global max pooling that can localize objects in a single forward pass, by inspecting the contribution of each hidden unit in the network to the final output class. Zhou *et al.* [132] proposed a similar approach only using global average pooling, and showed that while max pooling performs better for classification tasks, average pooling leads to better results for localization. Furthermore, others have proposed masking regions within the image, and studying the response of the network to identify which regions

cause the maximum activation [133, 134]. However, these methods were proposed for *object localization* where the goal is to locate the coordinates of an object within an image. These methods do not perform well on the more difficult problem of semantic segmentation, where the goal is to classify each pixel to the class it belongs to, even if the class was not provided as an image-level label.

Despite the success of CNN-based methods—both fully- and weakly-supervised—in semantic segmentation tasks, the CNN-based methods we reviewed above have several limitations that prevent them from being readily applied to seismic data. First, all of these methods initialize their models with the weights of fully-supervised networks trained on the ImageNet dataset [122, 123, 125, 126, 129]. While ImageNet [135] has more than 14 million natural images, no such datasets exist for seismic interpretation. In addition, many of these methods require large amounts of training data; some methods use web crawled images from sites like `flickr.com` to augment their training data [129], while others augment their data by using image-level labels from massive datasets such as ImageNet [122]. Finally, despite their success, CNNs are rarely used alone in weakly-supervised settings. Other techniques are typically used to introduce an auxiliary source of supervision. For example, some methods use smoothing priors [122, 128], region proposal [136], or saliency maps [126, 127, 129, 130]. Others [125, 129, 137, 130, 138] use fully-supervised CRFs to fine-tune their output labels or use pixel-level labels for validation [121], something that would not be possible in the absence of pixel-level labels. This can give these methods an unfair advantage over other truly weakly-supervised techniques.

4.2.2 Matrix Completion and Factorization

For our seismic interpretation application, we do not have access to any pixel-level annotations; therefore, using CNNs directly to obtain pixel-level labels is a rather unpractical approach. In addition, all the existing pretrained networks for image classi-

fication, saliency detection, region proposals, and the fully-connected CRFs that the previous techniques would use are pretrained on *natural* images. The features learned by these models would not transfer well to a completely different visual domain such as seismic interpretation. Other alternative techniques based on matrix completion or factorization typically do not require large amounts of data and therefore are better suited for our specific problem. In this subsection, we review the main matrix completion or factorization based techniques that are related to our method.

Matrix completion is the task of predicting missing entries in partially-observed matrices, while matrix factorization is the process of decomposing a matrix into the product of two or more matrices. Matrix completion or factorization based techniques perform very well in multi-label *image classification* tasks. This is the problem where images, or superpixels, have many noisy labels assigned to them. Given enough images with such labels, the goal is then to remove the noisy labels, and keep the correct ones. For example, Cabral *et al.* [139] formulated the weakly-supervised multi-label image classification problem as a convex low-rank matrix completion problem, and devised algorithms to solve it.

Non-negative matrix factorization (NMF) [140, 141] is a widely used matrix factorization technique that decomposes a non-negative matrix into two lower-rank non-negative matrices. While NMF has been used for many wide-ranging applications, its use in semantic segmentation has been very limited. Hong *et al.* [142] proposed a framework for *clustering images* retrieved from a reference dataset using sparse and orthogonal NMF. The images are initially represented by noisy labels based on their similarity to labeled reference images; then NMF with sparsity and orthogonality constraints is used to refine these noisy labels. Others have proposed using graph-regularized matrix factorization based methods [143, 144] to infer the labels of superpixels that were extracted using other techniques. They propose propagating the noisy image-level labels to the various image segments, and then by solving a

matrix factorization problem, they infer the correct label for each image segment. Furthermore, [145] proposed a non-negative matrix co-factorization based approach that jointly learns a discriminative dictionary and a linear classifier that classifies features from superpixels into different classes. The previous three methods [143, 144, 145] can be viewed as weakly-supervised semantic segmentation techniques, but their use of features extracted from superpixels limits the resolution of the resulting labels to the size of the superpixels in the image. Yuan *et al.* [146] proposed an unsupervised texture image segmentation algorithm using an NMF formulation on the singular value decomposition (SVD) of features extracted from texture images.

All the matrix completion or factorization based techniques we have reviewed attempt to do image (or superpixel) classification and not dense labeling on the pixel level. To the best of our knowledge, no other work (besides ours) attempts to map image-level labels to pixel-level labels using a matrix factorization formulation. Our proposed weakly-supervised method for mapping our image-level labels into pixel-level labels is based on non-negative matrix factorization (NMF). This method does not require any additional training data nor does it require any pre- or post-processing techniques, or auxiliary sources of supervision. And unlike some of the weakly-supervised methods we reviewed previously, our method applies directly on the pixel-level.

4.2.3 Summary

In summary, recent approaches to weakly-supervised semantic segmentation have widely relied on CNNs. CNNs provide a powerful and robust feature representation that has been driving the success of fully-supervised semantic segmentation models. For the weakly-supervised setting, the primary challenge is to train these large CNN models and to enable them to localize the various target classes. In the natural image domain, the availability of large annotated datasets such as ImageNet, have

greatly helped in this regard. All the weakly-supervised methods proposed in the literature use models pretrained on ImageNet (usually for image classification, but in some cases pretrained *semantic segmentation* models are used as well [124]). In addition, many methods proposed in the literature use various techniques to enhance the localization ability of these weakly-supervised models. A closer look reveals that these techniques either rely on higher forms of weak supervision (such as bounding boxes in the case of object proposals) or rely on pixel-level labels (such as in the case of fully-connected CRFs). Also, some of these methods [125, 122, 128, 129] use weak labels from other datasets or even from online image search engines such as `flickr.com` to augment their training set. Table 4.1 shows a summary of the main CNN-based techniques that have been proposed in the literature for weakly-supervised semantic segmentation. The highlighted region of Table 4.1 shows various techniques that these methods use to overcome weak supervision.

In the case of seismic interpretation, we do not have access to large annotated datasets such as ImageNet, and cannot use websites such as `flickr.com` to augment our training data. Similarly, all the auxiliary sources of supervision that the CNN-based methods use cannot be directly applied to a completely different application domain such as seismic images, and many of them rely on pixel-level labels. We also do not have access to CNN models pretrained on seismic data, and therefore training these large networks from scratch using a limited amount of annotated data will be a challenging task and these models will be highly prone to over-fitting. Therefore we opted for a matrix factorization based approach that works well for limited amounts of annotated data.

There has been a few matrix completion or factorization based techniques proposed in the literature for image classification and clustering. For weakly-supervised semantic segmentation, a few methods were proposed, but they are all limited to inferring the labels of regions or superpixels within the image, not actual pixels. In

addition, several of these techniques use pretrained CNN models to extract features from these regions or superpixels and require access to ImageNet. Table 4.2 summarizes these techniques. Unlike the other matrix completion or factorization based techniques, our proposed method infers true pixel-level labels and does not require any pretrained models, or access to large annotated datasets such as ImageNet or any other form of data augmentation or auxiliary supervision. In the next few sections, we introduce this method in detail.

Table 4.1: A summary of the main CNN-based techniques, and the methods they use to overcome weak-supervision. MIL, EM, and SP refer to multiple instance learning, expectation maximization, and superpixels, respectively. Auxiliary sources of supervision annotated with (F) and (W) indicate fully-supervised and weakly-supervised techniques respectively.

	MIL	EM	SP	Uses weak labels from other datasets	Pretrained model	Auxiliary sources of supervision
Pathak *et al.* (2014) [121]	✓				✓	-
Papandreou *et al.* (2015) [125]		✓		✓	✓	Dense CRF (F) + pixel-level labels for validation (F)
Pinheiro *et al.* (2015) [122]	✓		✓	✓	✓	Smoothing prior (F) + object proposals (W)
Hou *et al.* (2016) [126]		✓			✓	Saliency maps (F) + attention model (W)
Kim *et al.* (2016) [123]	✓				✓	-
Kwak *et al.* (2017) [128]		✓	✓	✓	✓	Smoothing priors (F)
Wei *et al.* (2017) [129]		✓		✓	✓	Saliency maps (F) + Dense CRF (F)
Durand *et al.* (2017) [124]	✓				✓	Uses a pretrained semantic segmentation model (FCN ResNet-101) without the last layer (F)
Hou *et al.* (2017) [127]		✓			✓	Saliency maps (F) + attenion maps (W)

Table 4.2: A summary of related matrix completion or factorization based techniques proposed in the literature.

	Inference Level					
	Image	Region	Superpixel	Pixel	Requires ImageNet	Pretrained CNN model
Cabral *et al.* (2015) [139]	✓	✓			✓	✓
Niu *et al.* (2015) [143]		✓				
Hong *et al.* (2016) [142]	✓				✓	✓
Zhang and Gong *et al.* (2016) [145]			✓			
Cabral *et al.* (2017) [144]		✓			✓	✓
Ours (2017) [147, 148]				✓		

4.3 Non-Negative Matrix Factorization

NMF [140, 141] is a commonly used matrix factorization technique that is closely related to many unsupervised machine learning techniques such as k-means [149] and spectral clustering [150]. NMF decomposes a non-negative matrix $\mathbf{X} \in \mathbb{R}_+^{N_p \times N_s}$ into the product of two lower-rank matrices $\mathbf{W} \in \mathbb{R}_+^{N_p \times N_f}$, and $\mathbf{H} \in \mathbb{R}_+^{N_f \times N_s}$ such that both $\mathbf{W}$ and $\mathbf{H}$ are non-negative, and $N_f < \min(N_p, N_s)$. In other words we have,

$$\mathbf{X} \approx \mathbf{WH}. \tag{4.1}$$

In our work, given the image-level labeled images, $\{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_{N_s}\}$, we vectorize them to construct the data matrix $\mathbf{X}$ such that each image is a column in $\mathbf{X}$. The data matrix $\mathbf{X}$ has N_s such images, each of which is a vector of length N_p. Here, we use N_p, N_s, and N_f to denote the number of pixels, the number of samples, and the number of features (or the *rank* of $\mathbf{X}$) respectively. NMF factorizes the data matrix $\mathbf{X}$ into two non-negative matrices, a basis matrix $\mathbf{W}$ and a coefficient matrix $\mathbf{H}$. In clustering terms, the columns of $\mathbf{W}$ represent N_f number of clusters in the data, whereas the columns of $\mathbf{H}$ represent the memberships of each of the images to the different clusters in the data. Here, the clusters represent features extracted from different seismic structures like salt domes, faults, or horizons.

The regular NMF problem does not have a closed-form solution and is typically solved by minimizing the following objective function

$$\arg\min_{\mathbf{W},\mathbf{H}} ||\mathbf{X} - \mathbf{WH}||_F^2 \quad \text{s.t.} \mathbf{W}, \mathbf{H} \geqslant 0, \tag{4.2}$$

where, $|| \cdot ||_F$ is the Frobenius norm and $\geqslant$ is used to indicate element-wise inequality. Lee and Seung [141] proposed an efficient method of solving this problem using multiplicative update rules and proved that they converge to a local minima.

4.4 Sparsity and Orthogonality Constraints

Our label mapping algorithm is based on NMF partly because it can be used to learn a parts-based representation [140], where each feature would represent a localized "part" of the data. This allows us to learn class-specific features for our entire dataset, and then use NMF to assign labels to each pixel in the dataset based on which features were used to represent that pixel. To learn the class-specific features, we initialize the feature matrix $\mathbf{W}^0$ using k-means applied separately on the different classes in the data matrix $\mathbf{X}$. This ensures that each feature $\mathbf{w}_i$ in the matrix $\mathbf{W}$ corresponds to a single class. As a byproduct of applying k-means on each class, we obtain a binary cluster membership matrix, $\mathbf{Q} \in \{0,1\}^{N_f \times N_\ell}$ where N_ℓ is the number of classes, that encodes which feature in $\mathbf{W}$ came from which class. Specifically, the element $\mathbf{Q}(i,j) = 1$ if the feature $\mathbf{w}_i$ belongs to class j.

In practice however, this parts-based representation is rarely achieved using the formulation in Equation 4.2. To remedy this, we impose a sparsity constraint on the feature matrix $\mathbf{W}$ such that the sparsity of every feature $\mathbf{w}_i$ satisfies

$$\rho(\mathbf{w}_i) = \frac{\sqrt{N_p} - \frac{||\mathbf{w}_i||_1}{||\mathbf{w}_i||_2}}{\sqrt{N_p} - 1}, \tag{4.3}$$

where $\rho(\cdot)$ indicates the sparsity of a vector. This value is always between zero and one, with higher values indicating higher sparsity. To enforce this sparsity constraint, we follow the algorithm proposed by Hoyer [151]. While other measures of sparsity can be used, such as the ℓ_1 norm, the measure in Equation 4.3 is scale-invariant and normalized to the range $[0,1]$. Furthermore, it has been successfully used in wide-ranging NMF applications, e.g., [142, 152].

The sparsity constraint on the features matrix $\mathbf{W}$ greatly helps in learning a parts-based representation. However, we would also expect that each feature in $\mathbf{W}$ to only represent a few images. In other words, in our setup it is very unlikely that the same

feature will be present in a large number of the images. This is because NMF is not scale-, rotation-, or translation-invariant, and therefore, each sparse feature is limited in its ability to represent a large number of images. To enforce this expectation, that each feature only represents a few images, we impose an orthogonality constraint on the coefficients matrix $\mathbf{H}$. To finalize our formulation, we add two regularization terms on $\mathbf{W}$ and $\mathbf{H}$. Our problem then becomes

$$\begin{aligned} \arg\min_{\mathbf{W},\mathbf{H}} \; ||\mathbf{X} - \mathbf{W}\mathbf{H}||_F^2 + \gamma||\mathbf{H}\mathbf{H}^T - \mathbf{I}||_F^2 + \lambda_1||\mathbf{W}||_F^2 \\ +\lambda_2||\mathbf{H}||_F^2 \;\; \text{s.t. } \mathbf{W}, \mathbf{H} \geqslant 0 \text{ and } \rho(\mathbf{w}_i) = \rho_w, \end{aligned} \tag{4.4}$$

where matrix $\mathbf{I}$ is an identity matrix. The values γ_1, λ_1, and λ_2 are constants, and ρ_w is the desired sparsity level.

4.5 Multiplicative Update Rules

There are several approaches to solving the problem in Equation 4.3. Lee and Seung [141] proposed an efficient method of solving the base NMF problem using multiplicative update rules and proved that they converge to a local minima. We adopt a similar approach to solving the problem in Equation 4.3. The detailed derivation is shown in Appendix B.

Instead of solving the problem in Equation 4.4 for both $\mathbf{W}$ and $\mathbf{H}$, we decouple this problem into two separate sub-problems. The first,

$$\arg\min_{\mathbf{W}} \; ||\mathbf{X} - \mathbf{W}\mathbf{H}||_F^2 + \lambda_1||\mathbf{W}||_F^2 \;\; \text{s.t. } \mathbf{W} \geqslant 0, \rho(\mathbf{w}_i) = \rho_w, \tag{4.5}$$

is solved for $\mathbf{W}$ while $\mathbf{H}$ is held constant. Then the second,

$$\arg\min_{\mathbf{H}} \; ||\mathbf{X} - \mathbf{W}\mathbf{H}||_F^2 + \gamma||\mathbf{H}\mathbf{H}^T - \mathbf{I}||_F^2 + \lambda_2||\mathbf{H}||_F^2 \;\; \text{s.t. } \mathbf{H} \geqslant 0, \tag{4.6}$$

is solved for $\mathbf{H}$ while $\mathbf{W}$ is held constant. We use gradient descent to derive the following multiplicative update rules for $\mathbf{W}$:

$$\mathbf{W}^{t+1} = \frac{\mathbf{W}^t \odot (\mathbf{X}\mathbf{H}^{t^T})_{ij}}{(\mathbf{W}^t\mathbf{H}^t\mathbf{H}^{t^T} + \lambda_1\mathbf{W}^t)_{ij}}, \tag{4.7}$$

and for $\mathbf{H}$:

$$\mathbf{H}^{t+1} = \frac{\mathbf{H}^t \odot \left(\mathbf{W}^{t+1^T}\mathbf{X} + \gamma\mathbf{H}^t\right)_{ij}}{(\mathbf{W}^{t+1^T}\mathbf{W}^{t+1}\mathbf{H}^t + \lambda_2\mathbf{H}^t + \gamma\mathbf{H}^t\mathbf{H}^{t^T}\mathbf{H}^t)_{ij}}. \tag{4.8}$$

Here, $\odot$ represents element-wise multiplication, the division operation is performed in an element-wise fashion as well, while the superscript of each matrix indicates the iteration number in which it was computed. To increase the stability of the convergence, it is possible to renormalize the columns of $\mathbf{W}$ and the rows of $\mathbf{H}$ at every iteration to have constant energy. The multiplicative update rules (MURs) in Equations 4.7 and 4.8 are applied successively until both $\mathbf{W}$ and $\mathbf{H}$ converge.

As we show in Appendix B, these multiplicative update rules are a special case of gradient descent with an automatic step size selection. Choosing to solve this problem using MURs instead of other gradient descent based techniques has many advantages. One advantage is the guaranteed non-negativity of $\mathbf{W}$ and $\mathbf{H}$ when they are initialized with non-negative values. Another significant advantage is that MURs preserve the initial sparsity values of $\mathbf{W}$ and $\mathbf{H}$. This means that we can apply the sparsity constraint only once on the initial feature matrix $\mathbf{W}^0$, and not have to apply it in all the remaining iterations. This greatly improves the computational efficiency of this approach. Furthermore, MUR computations can be highly efficient when done on a GPU, rather than a CPU. This is because MURs only rely on elementary matrix operations that GPUs are optimized for. We have achieved more than two orders of magnitude speedup when this problem is solved on a GPU, using the PyTorch deep learning library, rather than a traditional CPU-based implementation. A detailed derivation of these MURs is shown in Appendix B, and summary of our proposed

Algorithm 1: Weakly Supervised Label Mapping

Input: Data matrix $\mathbf{X} \in \mathbb{R}^{N_p \times N_s}$, image-level labels $\mathbf{y} \in \mathbb{Z}^{N_\ell}$, number of classes N_ℓ, feature sparsity value ρ_w, and number of features per class k.

Output: Cluster membership matrix $\mathbf{Q} \in \mathbb{R}^{N_f \times N_\ell}$, final features matrix $\mathbf{W}^{\mathsf{final}} \in \mathbb{R}^{N_p \times N_f}$, and final coefficients matrix $\mathbf{H}^{\mathsf{final}} \in \mathbb{R}^{N_f \times N_s}$.

$\mathbf{W}^0, \mathbf{Q} = \mathsf{kMeansOnEachClass}(\mathbf{X}, k, \mathbf{y})$

$\mathbf{W}^0 = \mathsf{applySparsityConstraint}(\mathbf{W}^0, \rho_w)$

$\mathbf{H}^0 \sim \mathsf{Uniform}(0, 1)$

while *not converged* **do**

  $\mathbf{W}^{t+1} = \frac{\mathbf{W}^t \odot (\mathbf{X}\mathbf{H}^{t^T})_{ij}}{(\mathbf{W}^t\mathbf{H}^t\mathbf{H}^{t^T} + \lambda_1 \mathbf{W}^t)_{ij}}$

  $\mathbf{H}^{t+1} = \frac{\mathbf{H}^t \odot (\mathbf{W}^{t+1^T}\mathbf{X} + \gamma_1(\mathbf{B} + \mathbf{B}^T)\mathbf{H}^t)_{ij}}{(\mathbf{W}^{t+1^T}\mathbf{W}^{t+1}\mathbf{H}^t + \gamma_1 \mathbf{H}^t\mathbf{H}^{t^T}\mathbf{H}^t + \lambda_2 \mathbf{H}^t)_{ij}}$

  $t = t + 1$

end

algorithm is shown in algorithm 1.

4.6 Extracting the Labels

Once $\mathbf{W}$ and $\mathbf{H}$ have converged, each column of $\mathbf{H}$, $\mathbf{h}_n$, indicates the features used to construct the nth image. Since every feature in $\mathbf{W}$ should correspond to a single class, we can predict the label of each pixel in the image by knowing which features are used to represent it. In other words, we can map the coefficients in $\mathbf{h}_n$ to the seismic structures that make up the image. Thus for image $\mathbf{x}_n$ we can obtain

$$\mathbf{L}_n = \mathbf{W}(\mathbf{Q} \odot (\mathbf{h}_n \mathbf{1}^T)) \qquad \forall n = [1, \cdots, N_s], \tag{4.9}$$

where $\mathbf{1}$ is a column vector of ones of length N_ℓ. The matrix $\mathbf{Q}$ is used to encode our knowledge of the image-level labels, and how the matrix $\mathbf{W}$ was initialized. The resulting matrix, $\mathbf{L}_n \in \mathbb{R}_+^{N_p \times N_\ell}$ shows the likelihood of each seismic structure for each pixel in the image. Then, the pixel-level labels for image $\mathbf{x}_n$ correspond to the seismic

Algorithm 2: Extracting the Pixel-Level Labels

Input: Cluster membership matrix $\mathbf{Q} \in \mathbb{R}^{N_f \times N_\ell}$, final features matrix $\mathbf{W}^{\mathsf{final}} \in \mathbb{R}^{N_p \times N_f}$, final coefficients matrix $\mathbf{H}^{\mathsf{final}} \in \mathbb{R}^{N_f \times N_s}$, number of classes N_ℓ, and confidence threshold τ.

Output: Pixel-level labels matrix $\mathbf{Y} \in \mathbb{Z}^{N_p \times N_s}$.

```
for n ← 1 to N_s do
    h_n = H^final(:, n)
    L_n = W^final(Q ⊙ (h_n 1_{1×N_ℓ}))
    y_n(i) = arg max_j  L_n(i, j)                    ∀i = [1, ··· , N_p]
    q_n(i) = max_j  L_n(i, j)                        ∀i = [1, ··· , N_p]
    y_{n(q_n<τ)} = 0
end
Y = [y_1, y_2, ··· , y_{N_s}]
```

structure given by

$$\mathbf{y}_n(i) = \arg\max_j \mathbf{L}_n(i,j) \quad \forall i = [1, \cdots, N_p], \tag{4.10}$$

where $\mathbf{L}_n(i,j)$ denotes the element in the ith row and j^{th} column of matrix $\mathbf{L}_n$. However, due to the nature of the weakly-supervised mapping of the labels, there is an element of uncertainty in the mapping. Since the features $\mathbf{w}_i$ are sparse, some pixels in an image $\mathbf{x}_n$ may not have a feature that accurately represents all the pixels within it. These pixels typically end up being represented as a weighted sum of a large number of different features, often from different classes and having small coefficients. This leads to noisy labeling results. To remedy this, we introduce a new `uncertain` class that contains pixels with uncertain labels. We define our confidence, $\mathbf{q}_n \in \mathbb{R}^{N_p}$, in the predicted label of every pixel in the image $\mathbf{x}_n$ as

$$\mathbf{q}_n(i) = \max_j \mathbf{L}_n(i,j) \quad \forall i = [1, \cdots, N_p]. \tag{4.11}$$

We can then assign any pixel whose confidence is less than a threshold τ to the

`uncertain` class, denoted as class 0

$$\mathbf{y}_{n(\mathbf{q}_n<\tau)} = 0. \tag{4.12}$$

Once we obtain the pixel-level labels $\mathbf{y}$ for each image, we apply a 3×3 median filter to clear any noisy labels and get the final labeling result for that image. We do this for all N_s images and concatenate the results to construct the pixel-level labels matrix $\mathbf{Y} \in \mathbb{Z}^{N_p \times N_s}$ that contains the final pixel-level labels for all the images in the data matrix

$$\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_{N_s}]. \tag{4.13}$$

This process is summarized in algorithm 2.

4.7 Results

To apply the label mapping algorithm we have presented in this chapter, we vectorize the images with image-level labels to construct the data matrix $\mathbf{X}$. The order of the images in $\mathbf{X}$ is encoded in vector $\mathbf{y}$ that stores the image-level labels of each image in $\mathbf{X}$. We then apply the k-means clustering algorithm on each class separately and use the results to initialize matrix $\mathbf{W}^0$ after we impose the sparsity constraint in Equation 4.3, we also obtain the binary cluster membership matrix $\mathbf{Q}$ based on where the features from each class were stored in $\mathbf{W}^0$. The coefficients matrix $\mathbf{H}^0$ is initialized with uniform random numbers in the range $[0, 1]$. The values of λ_1, λ_2 and γ are chosen empirically as 0.1, 0.5, and 5 respectively. The sparsity of the initial features ρ_w is set to 0.4. Additionally, the confidence threshold τ is set to 0.001.

We then apply the MURs in Equation 4.7 and 4.8 successively until both $\mathbf{W}$ and $\mathbf{H}$ converge. Figure 4.4 shows the convergence curves for the $\mathbf{W}$ objective function in Equation 4.5, the $\mathbf{H}$ objective function in Equation 4.6, and the overall objective function defined in Equation 4.4. We see that although we did not attempt to solve

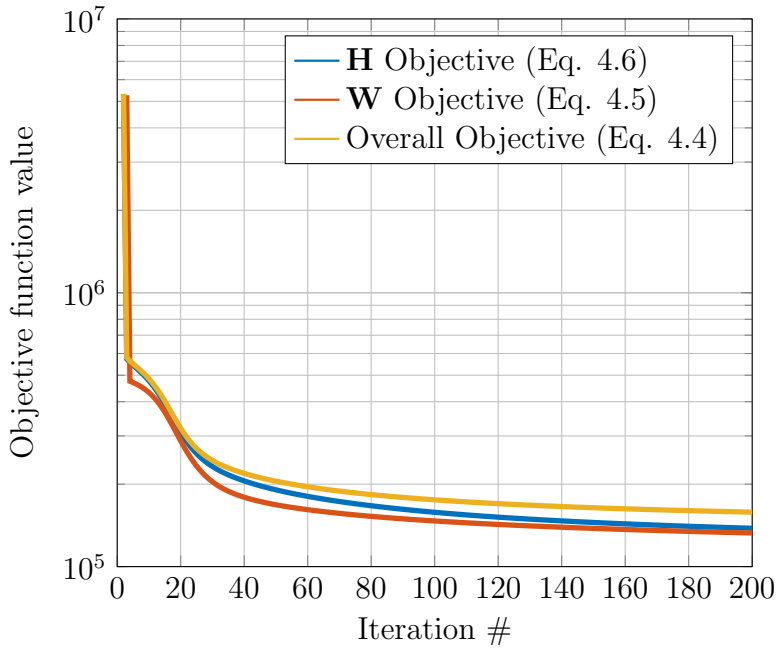


Figure 4.4: A plot showing the convergence curves for the multiplicative update rules for **W** and **H** as well as the overall objective function in equation 4.4.

Equation 4.4 directly, the two MURs in Equation 4.7 and 4.8 effectively minimize the overall objective function. We terminate our optimization at the 200th iteration. Performed on a GPU, this process takes around 3 seconds.

Figure 4.5 shows the effect of the orthogonality constraint on the final coefficients matrix $\mathbf{H}^{\mathsf{final}}$. On the left, $\mathbf{H}^{\mathsf{final}}$ is shown when the orthogonality constraint is not used. One can notice that there are many long horizontal lines. These lines indicate that many features (from different classes) are used the represent all the images in **X**. In other words, our features in **W** that were generated with the explicit assumption that each features belongs to a single class, are used to represent images from *all* classes. This is problematic and leads to wrong results. However, when the orthogonality constraint is applied (see Figure 4.5 on the right), we see that many of these lines disappear and we are left with a few features representing each class. The remaining long horizontal lines mostly belong to the `other` class, that we expect to be represented across all images.

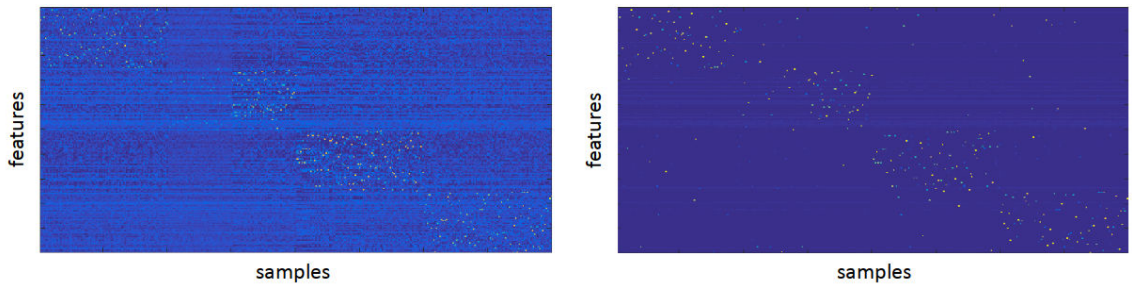


Figure 4.5: The effect of the orthogonality term in Equation 4.4 on the final cofficient matrix $\mathbf{H}^{\mathsf{final}}$. On the left, $\mathbf{H}^{\mathsf{final}}$ without the orthogonality term, and on the right matrix $\mathbf{H}^{\mathsf{final}}$ with the orthogonality term.

Figure 4.6 shows the initial labels for four different images from the four different classes, as well as the generated labels for various iterations in the optimization process. We note that since the coefficient matrix $\mathbf{H}$ was initialized with uniform random values in the range $[0, 1]$, our "initial" confidence computed using Equation 4.11 is very high, and consequently, very few pixels in the initial labels had coefficient smaller than τ and therefore belonged to the `uncertain` class. However, immediately after the MURs are applied the confidence values of the labels drastically drops, to the degree that all the pixels in the images during the first iteration are labeled as `uncertain`. However, as the optimization progresses, confidence in various predicted labels gradually increases. Towards the end of the optimization process, the orthogonality term in Equation 4.4 plays a more prominent role in ensuring that most features in $\mathbf{W}$ represent only a few images in $\mathbf{X}$, this significantly reduces the number of noisy labels.

Since our similarity-based retrieval workflow might produce a few images that do not belong to the same class as the reference image, we might end up with a few wrong image-level labels. However, the k-means initialization step of $\mathbf{W}^0$ greatly enhances the robustness of our label mapping algorithm to mislabeled images. To validate this claim, we examine the effect of wrongly retrieved images on the final pixel-level labels and analyze the robustness of our label mapping algorithm. We achieve this by

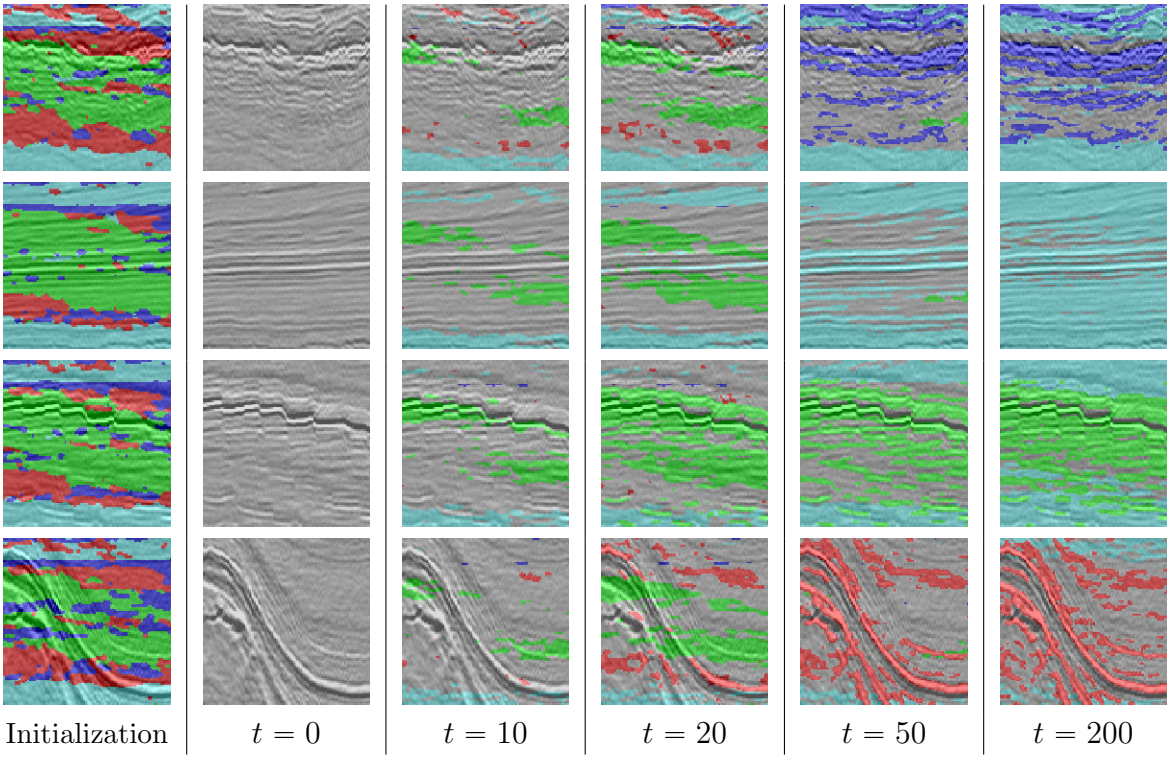


Figure 4.6: Results of our weakly-supervised label mapping approach for sample images from each class during different iterations. The initial labels (i.e., with randomly initialized coefficients) are also shown in the first column. The blue, green, and red colors correspond to the `chaotic`, `fault`, and `salt dome` classes respectively. The gray color represents areas of low confidence.

artificially replacing images in $\mathbf{X}$ with wrongly-retrieved images, and then computing the final pixel-level labels and comparing the performance of our label mapping algorithm relative to the base case where no images are replaced. The performance is evaluated using the *relative pixel accuracy* that measures the percentage of pixels that are classified identically to the case where no wrongly retrieved images are injected in $\mathbf{X}$. Pixels with low confidence in the base case are ignored. Fig. 4.7 shows the drop in relative pixel accuracy as the percentage of wrongly-labeled images in $\mathbf{X}$ increases for varying numbers of feature clusters per class, k. Fig. 4.8 shows a similar plot for different values of the feature sparsity ρ_w. Overall, the larger the number of clusters, and the higher the sparsity, the more robust the label mapping algorithm

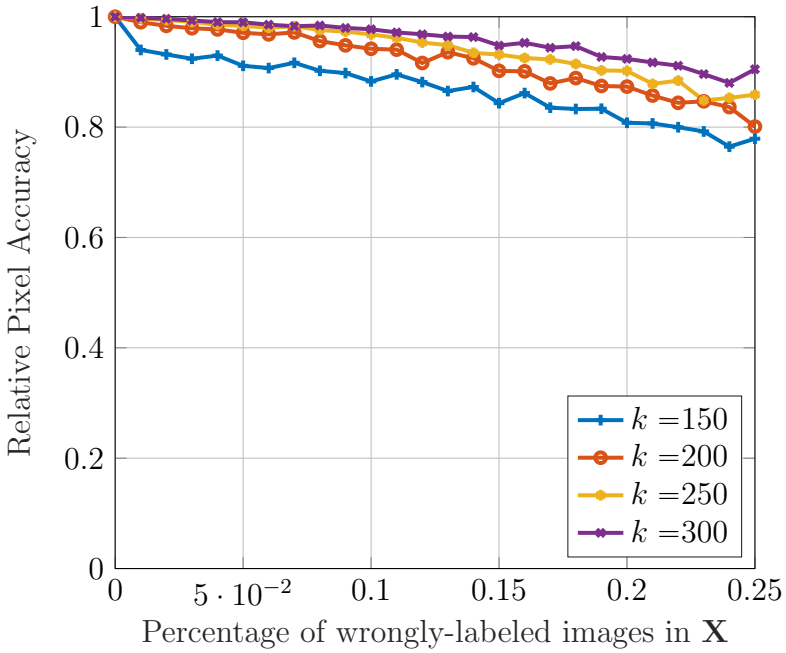


Figure 4.7: The robustness of our label mapping algorithm to mislabeled images for various numbers of feature clusters per class, k.

is. In addition, we note that even when 20% of the images are wrongly retrieved, the relative pixel accuracy only drops by around 10%.

Figure 4.11 displays several examples selected at random from the final results that we obtain for various classes of subsurface structures. Since we do not have ground-truth pixel-level annotations, the evaluation of the results is subjective. However, it is rather easy to observe the main differences in the results between our proposed method, and the two other baselines we implement. For our proposed approach, we notice that it maps the pixel-level labels into the correct locations that correspond to the various subsurface structures present within the image. To compare, the results using regular NMF (as defined in Equation 4.2) are shown in Figure 4.9. We observe that these results are extremely noisy, and do not capture the subsurface structures correctly. Figure 4.10 shows the results for when we use sparse initial features in $\mathbf{W}^0$ with the regular NMF problem in 4.2. These results are better than those in Figure 4.9, but they contain bands of misclassified pixels, typically in the center of the

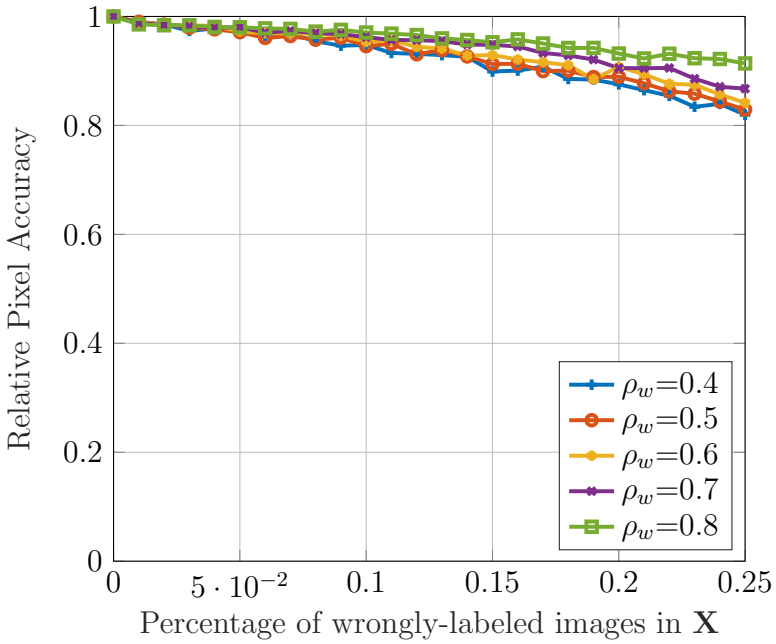


Figure 4.8: The robustness of our label mapping algorithm to mislabeled images for various feature sparsity levels, ρ_w.

image. The results of our proposed approach are far better at localizing the different subsurface structures, and they do not exhibit the same bands of misclassified pixels. Furthermore, it is important to note that the proposed approach is not limited to these particular classes of subsurface structures and can be easily applied to any other structure as long as a sufficient number of similar images are retrieved for each class.

4.8 Summary

In summary, we have introduced a novel weakly-supervised label mapping algorithm that maps image-level labels to pixel-level labels. We have shown how none of the techniques in the literature can be used to address this problem in the context of seismic interpretation where labeled data is scarce, and there are no large annotated datasets that can be used. We have introduced an efficient algorithm based on multiplicative update rules to solve the problem that have formulated, and we

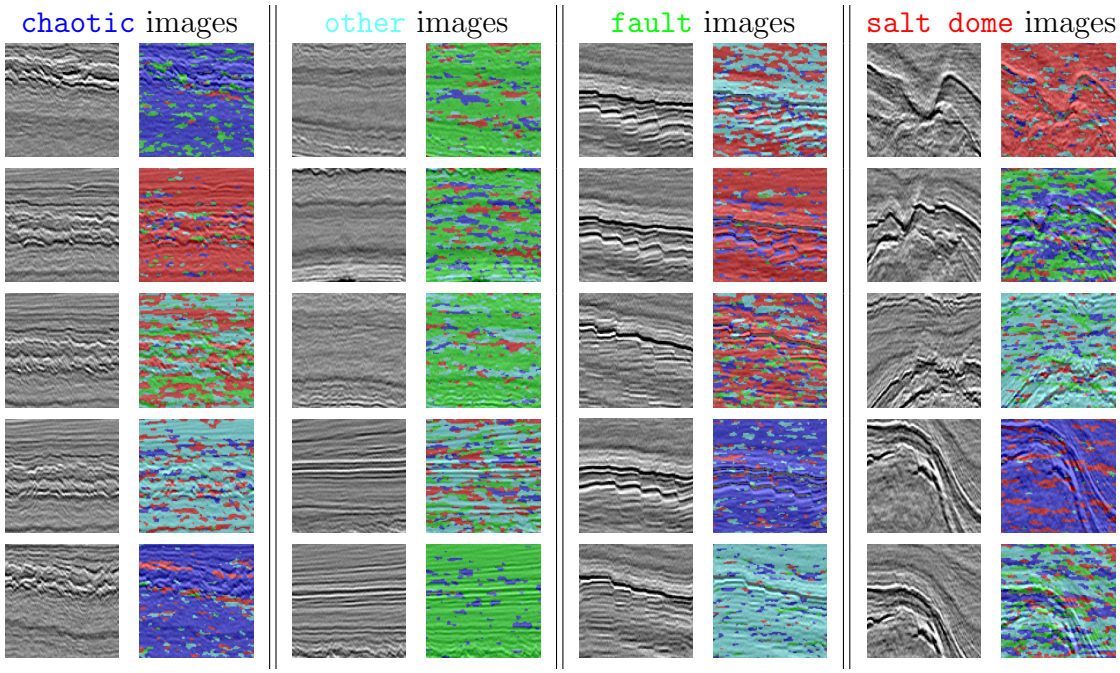


Figure 4.9: Results of our weakly-supervised label mapping approach for various subsurface structures. The first two columns show images containing `chaotic` structures and the corresponding `chaotic` pixel-level labels generated by our method in blue. The middle two columns show images that contain `fault` structures, and `fault` pixel-level labels in green. The last two columns show images that contain `salt dome` bodies or boundaries, and `salt dome` pixel-level labels in red.

have presented a detailed analysis of the proposed method. In addition, we showed sample results of our method compared to baseline methods that use non-negative matrix factorization. Overall, our method has proven to be very effective in inferring the pixel-level labels of thousands of images in our retrieved dataset, and we show in the next two chapters how these weak pixel-level labels can be used to train deep networks to semantically label structural and stratigraphic features in seismic data.

There are several areas where this approach can be improved, however. First, different classes of subsurface structures can often have different scales, whereas the method we have currently proposed uses a fixed size image for every class. It is worth investigating methods to alleviate this issue. Also, the final pixel-level labels are sensitive to the initial features, $\mathbf{W}^0$. While we have shown that k-means can easily

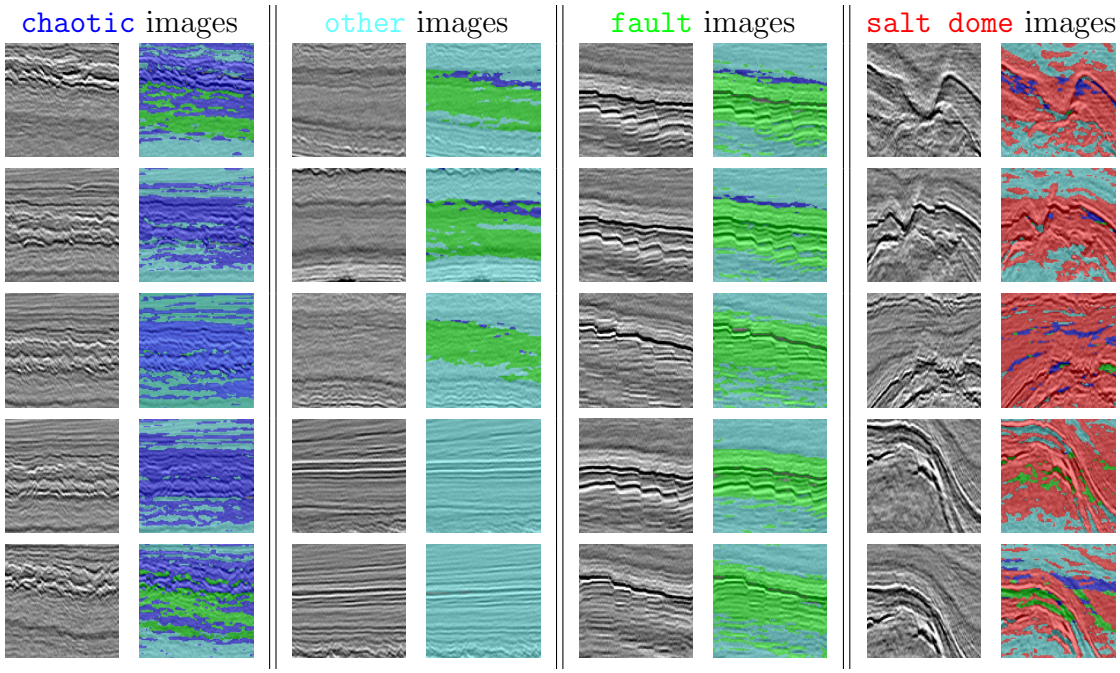


Figure 4.10: Results of our weakly-supervised label mapping approach for various subsurface structures. The first two columns show images containing `chaotic` structures and the corresponding `chaotic` pixel-level labels generated by our method in blue. The middle two columns show images that contain `fault` structures, and `fault` pixel-level labels in green. The last two columns show images that contain `salt dome` bodies or boundaries, and `salt dome` pixel-level labels in red.

be used to initialize $\mathbf{W}^0$, it is worth investigating other more promising methods for initializing $\mathbf{W}^0$ such as a convolutional autoencoder (CAE). Also, if the data matrix $\mathbf{X}$ has a wrong sparsity structure, applying the sparsity constraint in Equation 4.3 to form the feature matrix $\mathbf{W}$ might not lead to representative features of the different classes in $\mathbf{X}$. In that case, other techniques should be used to initialize $\mathbf{W}$. Finally, there are a few parameters such as the sparsity level ρ_w, the number of retrieved images per class M, and the regularization constants such as γ that need to be set by the interpreter based on their empirical assessment of the results.

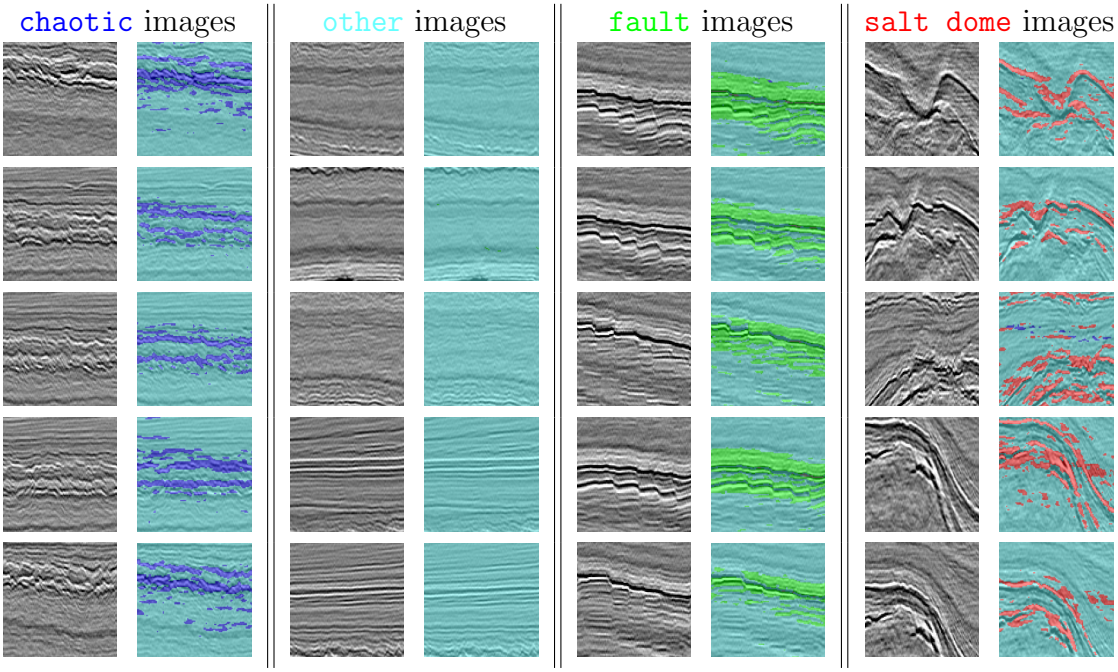


Figure 4.11: Results of our weakly-supervised label mapping approach for various subsurface structures. The first two columns show images containing `chaotic` structures and the corresponding `chaotic` pixel-level labels generated by our method in blue. The middle two columns show images that contain `fault` structures, and `fault` pixel-level labels in green. The last two columns show images that contain `salt dome` bodies or boundaries, and `salt dome` pixel-level labels in red.

CHAPTER 5

STRUCTURAL INTERPRETATION WITH WEAK PIXEL-LEVEL LABELS

5.1 Overview

In the previous chapters, we have presented various elements of our weakly-supervised framework for automatically generating large numbers of pixel-level training samples using only a few exemplar images that are annotated on the image-level. Our goal in this chapter is to demonstrate how these weak training labels can be used for the semantic labeling of subsurface structures. Using automatically-generated weak training data comes with many challenges. The labels are of a lesser quality than manually obtained labels, and each label has an associated confidence value that expressed the label mapping algorithm's confidence in the accuracy of the generated labels. We can denote our weak training data as $\mathcal{D} = \{\mathbf{x}_i, \mathbf{y}_i, \mathbf{q}_i\}_{i=1}^{N}$, where $\mathbf{x}_i$ are the images, $\mathbf{y}_i$ are the generated pixel-wise labels, $\mathbf{q}_i$ are the associated confidence values for each of the labels, and N is the number of images in our training data. Our goal in this chapter is not only to use the set of images and weak labels $\{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^{N}$ to train deep models for the semantic labeling of subsurface structures, but to also explore methods to exploit the confidence values, $\{\mathbf{q}_i\}_{i=1}^{N}$, to improve the performance of these models.

In the next section, we will explore methods in the literature that use deep learning for structural interpretation. Then in Section 5.3, we introduce our proposed method, including introducing the network architecture, and how we modify the loss function to learn from weak labels. Section 5.4 shows sample structural interpretation results that we have obtained on the Netherlands F3 block. We also compare the results to

those that we obtained in Chapter 3. Finally, Section 5.5 presents a summary of this chapter.

5.2 Background

An important step in structural interpretation is to identify and label important subsurface structures that can potentially trap hydrocarbon reservoirs such as faults and salt dome structures. In addition, the identification of these structures greatly helps geophysicists create accurate geological models for the seismic survey, further enhancing their ability to identify possible locations for hydrocarbon reservoirs. There are many methods proposed in the literature that use classical techniques such as edge detection [153, 154, 155], phase congruency [156, 157], and various seismic and texture attributes [55, 54, 158, 159] for detecting seismic structures. AlRegib *et al.* [4] provide a good overview of these methods, and more recent techniques based on machine learning. Unfortunately, many of the proposed classical techniques require manual processing of the data and are not robust enough to easily generalize to large seismic volumes. Some are very computationally expensive, and are typically designed as aides to a seismic interpreter, rather than standalone techniques.

In recent years, many deep learning based methods have been proposed for various structural seismic interpretation tasks. Waldeland and Solberg [51, 160] first proposed using a CNN for classifying salt bodies. While they used a simple image classification architecture, their results illustrated the great potential of using deep networks compared to hand-engineered features. Shi *et al.* [161] proposed improving on this by using an encoder-decoder style architecture that is better suited for semantic segmentation tasks such as classifying salt domes. Di *et al.* [162] proposed a technique for detecting faults using a traditional multilayer perceptron (MLP). Guo *et al.* [163] proposed using a fully-convolutional CNN for fault prediction, while Huang *et al.* [164] proposed combining both CNN and traditional machine learning models

with a variety of seismic attributes for identifying faults. Wu *et al.* [67] proposed to train a CNN to predict fault orientations on synthetic data and then showed that it could also generalize well to real seismic data.

All *these methods require "strong" labels*[1] that are obtained by manual labeling from an interpreter. Manually labeling data for training deep learning models can be as laborious and time-consuming as manual interpretation workflows. Furthermore, over-training a network on a relatively small amount of manually annotated data can lead to overfitting, and therefore poor generalization performance. Additionally, all of the techniques introduced in the literature for structural seismic interpretation so far are limited to extracting a single class of subsurface structures. This not only means that training these models and using them for inference would take much more time for multiple classes of seismic structures, but more importantly, the various machine learning models would not learn a joint feature space to accurately discriminate the various classes of subsurface structures. This can lead to false classifications and requiring much more training data and compute power than if a single CNN was used for the classification of multiple seismic structures. In the next section, we will introduce our proposed method which is the only approach that both addresses the issue of lack of sufficient data (through weakly-supervised learning), and uses a single CNN to semantically label multiple classes of seismic structures.

5.3 Proposed Method

In our work, we use a weakly-supervised learning approach to address the problem of lack of sufficient training data. In Chapter 2 we have introduced our proposed method to retrieve large numbers of seismic images that contain similar subsurface structures to examplar images selected by an interpreter. Then, we have shown how these images

[1]Here, "strong" labels mean high-quality labels generated by a domain expert. This is opposed to automatically-obtained "weak" labels that convey far less information than strong ones, and are usually far less accurate, but are much easier to obtain.

can be assigned image-level labels, and how these labels can be mapped to pixel-level labels using the algorithm we presented in Chapter 4. In this chapter, we show how these weak labels can be used to effectively train a deep encoder-decoder style CNN for semantically labeling not one, but *multiple* classes of seismic structures. Furthermore, we propose a modification of the loss function that exploits the confidence values of the weak labels to improve the overall robustness of the model. In this section, we introduce two main aspects of our proposed method. First, we introduce the network architecture that we adopt in this work for semantically labeling subsurface structures. Then, we introduce how we adapt the network loss function to train the network more effectively with weak labels.

5.3.1 Network architecture

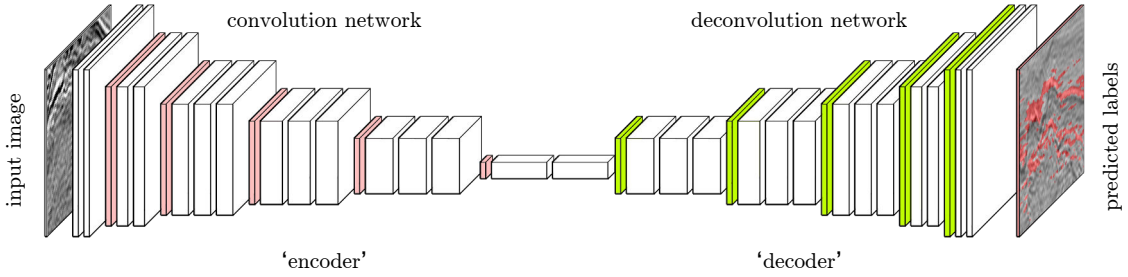


Figure 5.1: The architecture of the deconvolution network used in this work. White layers are convolution or deconvolution layers. Red layers are max-pooling layers, while green layers are unpooling layers.

Deconvolution networks [117] were previously introduced in Section 4.2.1. A deconvolution network has a symmetric encoder-decoder style architecture composed of stacks of convolution and pooling layers in the encoder, and stacks of deconvolution and unpooling layers in the decoder that mirror the encoders architecture. The role of the encoder can be seen as doing object detection and classification, while the decoder is used for accurate localization of these objects within the image. This

encoder-decoder style architecture can achieve finer and more accurate results than those of earlier networks such as FCN, and therefore is adopted in our work.

Figure 5.1 outlines the architecture of the deconvolution network used in our work. This architecture has 30 convolution and deconvolution layers (shown in white). These layers are typically followed by a rectified linear unit (ReLU) non-linearity. Five maxpoling layers (shown in red) perform 2×2 max pooling to select the maximum filter response within small windows. The indices of the maximum responses for every pooling layer are then shared with their respective unpooling layers (shown in green) to undo this pooling operation and get a higher resolution image.

5.3.2 Adapting the loss function for weak labels

Since our weak labels are generated automatically, they are not of the same quality as labels obtained from an expert interpreter. However, since obtaining such labels does not require any manual labor nor expensive computational resources, we can use these labels to train our model and modify our network loss function not to trust these weak labels too much. The loss function has a significant impact on the features that the network learns during training, and therefore must be adjusted to incorporate the different confidence values we have in our training data. To achieve this, we modify a recently introduced loss function called the focal loss (FL) [165] that was proposed for dense detection of objects in computer vision tasks. Our modification of the focal loss allows it to take the pixel confidence values into account when computing the loss. We call the resulting loss function the weak focal loss (WFL). The WFL can be viewed as a generalization of the FL as the FL, and the commonly used multiclass cross entropy loss are special cases of the WFL.

If we denote the output predictions of our model as $\tilde{p}(x)$ where x is the pixel index, it is common to normalize these predictions (often referred to as *logits*) using

the softmax function

$$p_n(x) = \frac{e^{\tilde{p}_n(x)}}{\sum_{j=1}^{N_\ell} e^{\tilde{p}_j(x)}}, \tag{5.1}$$

where $p_n(x)$ is the normalized prediction for the nth class at pixel x. One of the advantages of applying softmax to the output predictions of the model is that it maps the output of the network to probability values (a.k.a. $p_n(x) \in (0, 1]$ and $\sum_n p_n(x) = 1$). Another advantage of the softmax function is that it prevents the normalized model outputs from having a 0 probability for any class. This helps stabilize the learning process and prevents the network loss function from becoming infinite.

Further, in multiclass classification problems, it is common to encode the ground truth labels in a "one-hot vector" format. This means that the ground truth for a pixel is represented by a binary vector of length N_ℓ, where N_ℓ is the number of classes, instead of a single integer in the range $[1, N_\ell]$. We refer to the one-hot encoded ground truth labels as $\ddot{q}(x)$ where the symbol $\ddot{}$ refers to the binary nature of these labels. Now, we can write the widely-used multiclass cross-entropy loss as

$$\mathsf{CE}(\ddot{q}(x), p(x)) = -\sum_{n=1}^{N_\ell} \ddot{q}_n(x) \log p_n(x). \tag{5.2}$$

Given this definition of the CE loss, the focal loss can be written as

$$\mathsf{FL}(\ddot{q}(x), p(x)) = -\sum_{n=1}^{N_\ell} (1 - p_n(x))^\gamma \ddot{q}_n(x) \log p_n(x), \tag{5.3}$$

where the term $(1 - p_n(x))^\gamma$ reduces the loss for relatively well-classified examples and lets the network focus more on harder misclassified examples. The parameter γ controls how much weight is given to regions with low *predicted* confidence. Our

WFL loss can then be defined as

$$\mathsf{WFL}(q(x), p(x)) = -\sum_{n=1}^{N_\ell} (1 - p_n(x))^\gamma q_n(x)^\alpha \log p_n(x), \tag{5.4}$$

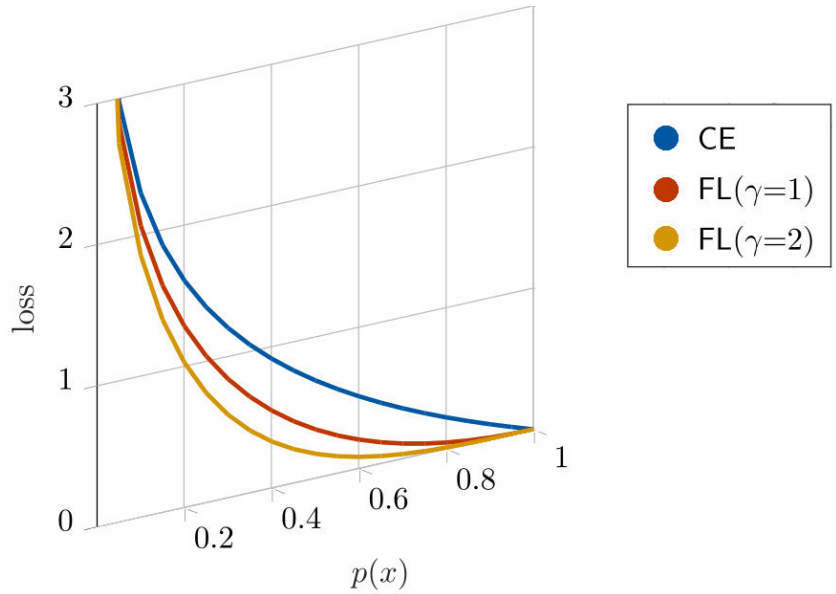
where now we have replaced the one-hot encoded ground truth labels, $\ddot{q}(x)$, with the confidence values of the weak labels, $q(x)$. Further, the parameter α governs the relationship between the confidence values and the loss function. This loss function allows us to put more weight on misclassified regions in the images and not trust our weak labels as much, especially if the model is particularly confident in a certain classification. Furthermore, the lesser the confidence value in a weak label, the lesser that label contributes to the overall loss of the image.

Finally, the loss for the entire image is the sum of the individual pixel-wise losses

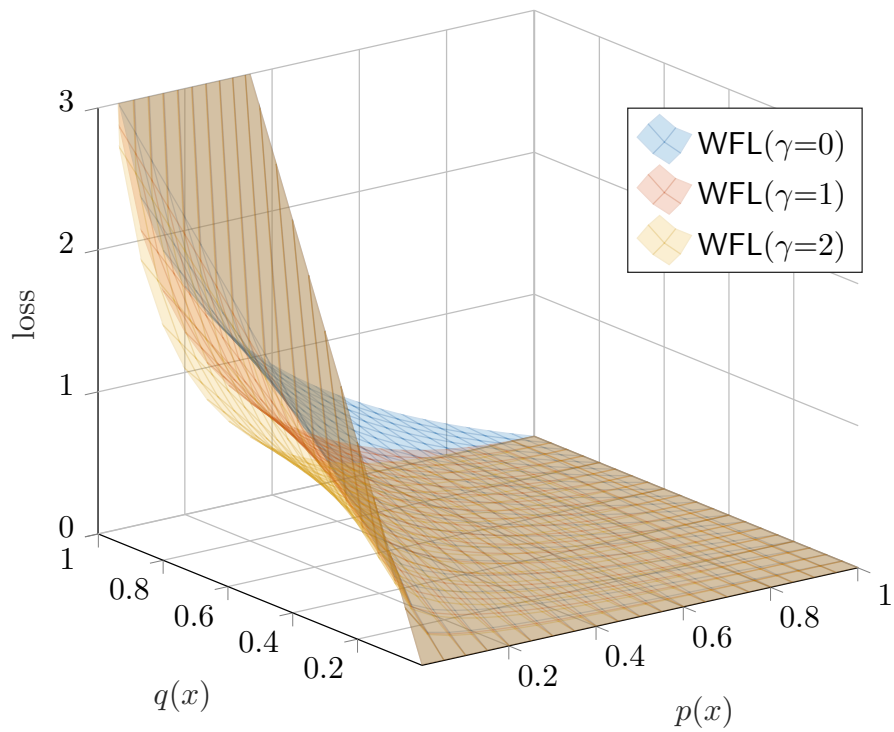
$$\mathsf{WFL}(q(\mathbf{x}_i), p(\mathbf{x}_i)) = \sum_{x \in \mathcal{X}} \mathsf{WFL}(q(x), p(x)), \tag{5.5}$$

where $\mathcal{X}$ is the set of the pixels that have confidence values greater than the confidence threshold τ as defined in Chapter 4.

Figure 5.2 shows a comparison between the CE, FL, and WFL losses for different values of γ and using $\alpha = 1$ in the case of WFL. As the value of γ increases, less emphasis is put on regions where the network has learned relatively well and is fairly confident in its predictions. Instead, more emphasis is put on regions where the network has not learned to classify the underlying structure effectively. Figure 5.2 also allows us to see how CE and FL are special cases of the WFL. The FL loss is a special case of the WFL loss with binary one-hot labels, and the CE loss is a special case of the FL loss when $\gamma = 0$. Later in the results section, we show the results of the WFL loss and compare it with the CE equivalent for non-binary labels (i.e., WFL with $\gamma = 0$). More results on this are presented in Chapter 6.



(a)



(b)

Figure 5.2: An illustration of the difference between cross entropy loss (CE), focal loss (FL), and weak focal loss (WFL) for different values of γ and using $\alpha = 1$.

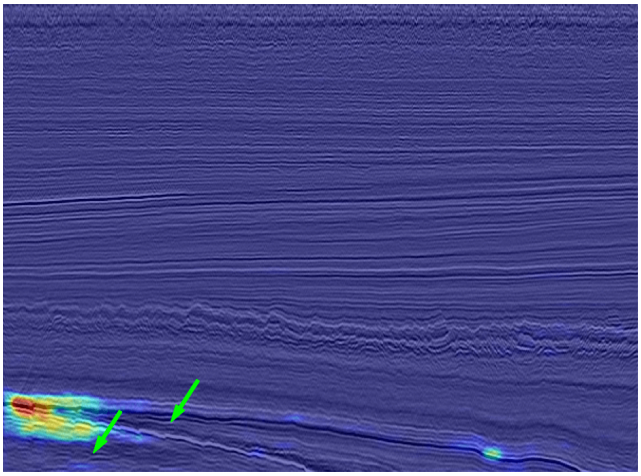
5.4 Results

We train our deconvolution network, shown in Figure 5.1, on thousands of automatically generated weak labels similar to those shown in Figure 4.11. To artificially increase the size of our training dataset, we use several data augmentation techniques. These techniques include random horizontal flipping, random rotations of up to $\pm 15^\circ$ of all the images in our dataset and their corresponding labels, and adding random Gaussian noise. These data augmentation techniques help to artificially increase the size of our training set. Throughout our training, we set aside 25% of the training data for model selection and validation purposes. Once our model's parameters are selected, we retrain our network on the entirety of the training data. We use the AdaDelta optimizer, a batch size of 32, and we use $\gamma = 1$ with the WFL loss. In our work, we empirically found that a linear relationship between the confidence values and the loss function (a.k.a. $\alpha = 1$) to work sufficiently well in most cases. Once our deconvolution network is trained, we apply it to the Netherlands F3 block [3] in a sliding window fashion to label the various subsurface structures in the data. This is done both in the inline and the crossline directions; then the final results are obtained by taking the element-wise product of the two. This step helps reduce any false-positive classifications.

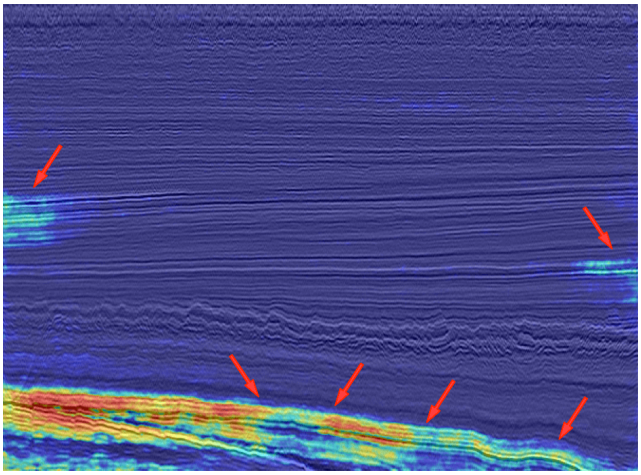
Figure 5.4 shows the class heat map highlighting `chaotic`, `faults`, and `salt dome` structures in inline #250 of the F3 block. It can be seen that the output of the model clearly extracts the details of the various subsurface structures with very few false positives. Additionally, Figure 5.5 shows a 3D cross-section of the F3 block with the boundaries of several salt domes highlighted with great accuracy. We note that our model highlights only the salt dome boundaries and that there are hardly any false positives present in the entire volume.

To compare the results of our deconvolution network and the FCN, we show in

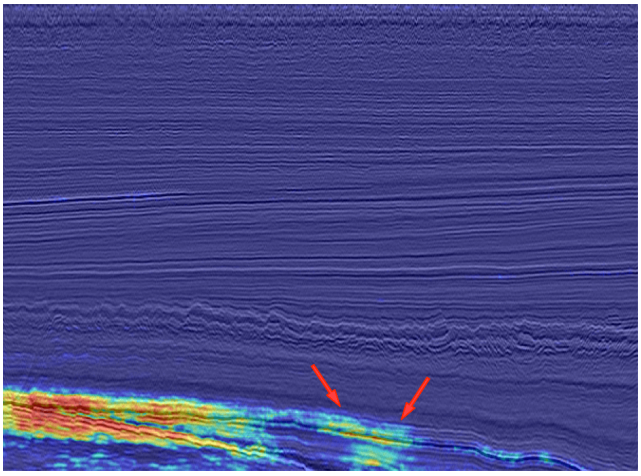
Figure 5.3(a) and 5.3(b) the results of labeling the `faults` class in crossline #1 of the F3 block using an FCN-8s network and our deconvolution network respectively. FCN-8s is the best performing variant of the FCN architecture proposed by [116]. Both FCN-8s and our deconvolution network were trained using the weak CE loss (WFL with $\gamma = 0$). We notice that due to the upsampling operations in FCN, several faults in the crossline where not labeled. These false negatives are shown as green arrows in Figure 5.3. On the other hand, the deconvolution network result using the weak CE loss was overly confident in the existence of faults in regions that had strong reflections, even though they did not have any fault structures. These false positives are shown as red arrows in Figure 5.3. To observe the effect of using the WFL with $\gamma = 1$ instead of $\gamma = 0$ (a.k.a. the weak CE loss), we show in Figure 5.3(c) the result of labeling the same crossline with a deconvolution network that used the WFL with $\gamma = 1$. By comparing Figure 5.3(b) and 5.3(c), We notice that the $(1 - p_n(x))^\gamma$ term in the WFL loss helps reduce false positives by not putting too much trust in the weak training data.



(a) FCN-8s using weak CE (WFL with $\gamma = 0$)

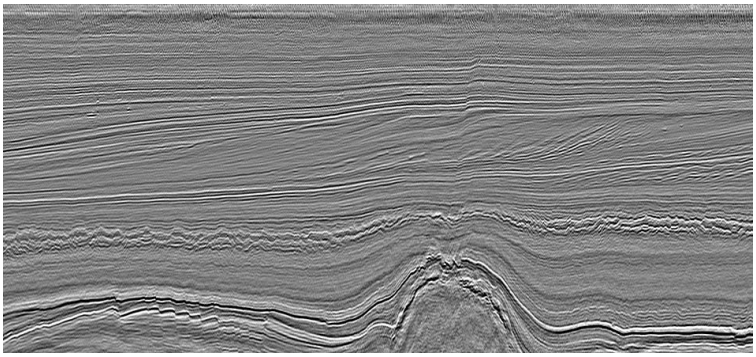


(b) Deconvolution network using weak CE (WFL with $\gamma = 0$)

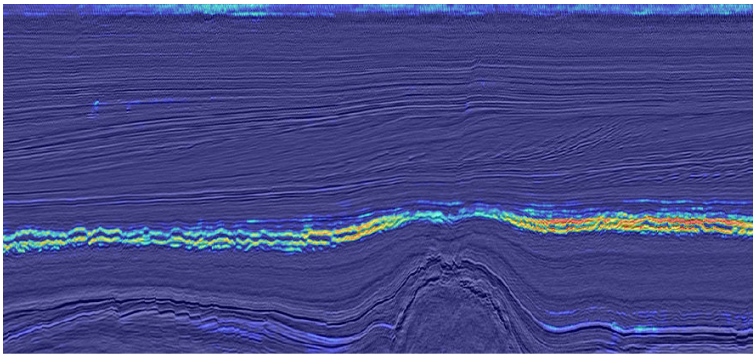


(c) Deconvolution network using WFL (ours)

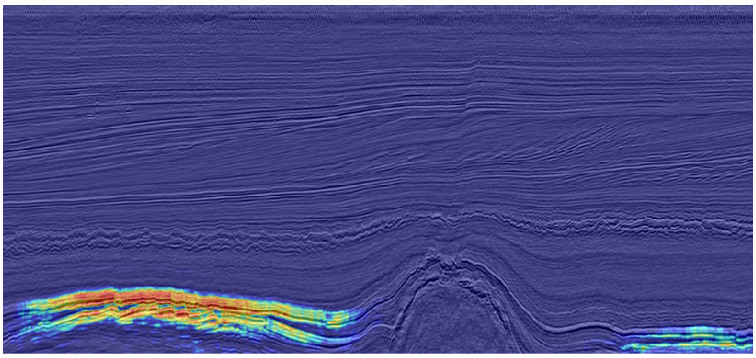
Figure 5.3: Fault structures in crossline #1 highlighted using either deconvolution network or FCN-8s, and using either the cross entropy loss (CE) or the weak focal loss (WFL). Green arrows indicate false negatives, while red arrows indicate false positives.



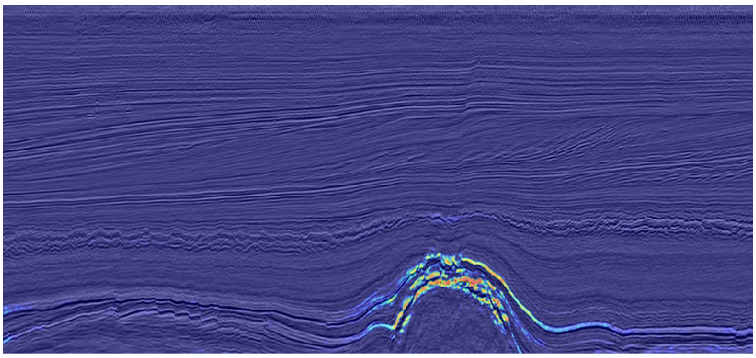
(a) original seismic



(b) `chaotic` class highlighted



(c) `faults` class highlighted



(d) `salt dome` class highlighted

Figure 5.4: Results using our model to highlight various subsurface structures in inline #350 of the Netherlands F3 block.

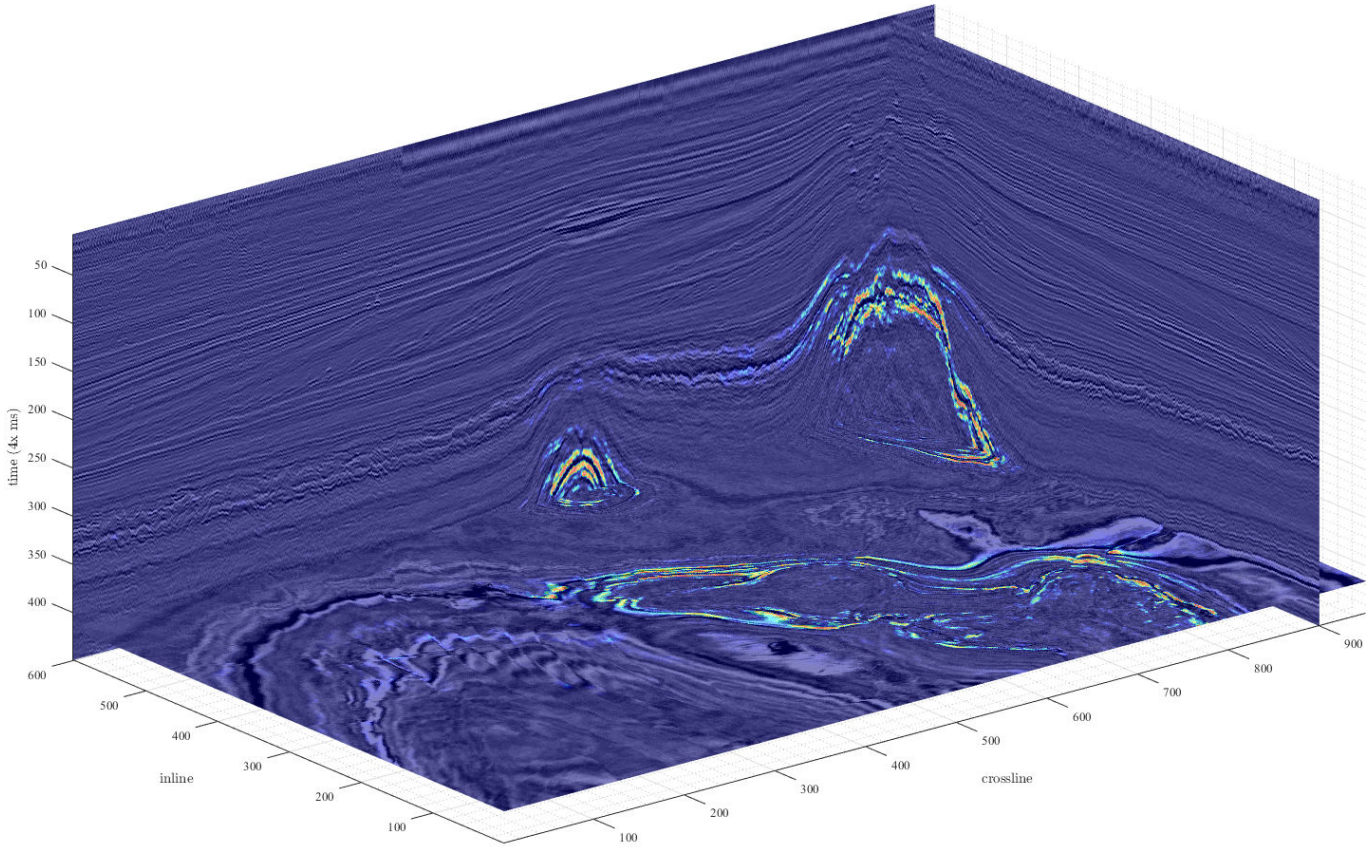


Figure 5.5: A 3D view of the Netherlands F3 block, with our model highlighting the three salt dome structures in the F3 block.

Finally, it is important to compare the results that were obtained in this chapter with those obtained in Chapter 3 that only relied on image-level labels. While the models that were trained to obtain the results using image-level or pixel-level labels are completely different, it is still worthwhile to compare the final results. Figure 5.6(a) shows the manually labeled inline #380 of the Netherlands F3 block, with `chaotic` pixels labeled in blue, `faults` pixels in green, and `salt dome` in red. Figure 5.6(b) shows the best labeling result obtained in Chapter 3 that used curvelet features. We note that there are many false positives in all three classes, and the boundaries between the different classes do not conform to the actual seismic structures. Figure 5.6(c) shows the result of our deconvolution network trained using our WFL loss function with $\gamma = 1$ after post-processing [2]. We note that other than a small region in the `faults` class, there are hardly any false positives. Also, the resulting labels are visibly much more similar to the manually labeled section that those in Figure 5.6(b). Table 5.1 summarizes the objective results of the two results in Figure 5.6(b) and (c). As we expect, the method we propose in this chapter significantly outperforms the technique that was presented in Chapter 3 that relies only on image-level labels.

Table 5.1: A comparison of the labeling results for the method presented in this chapter versus the method presented in Chapter 3 that only uses image-level labels.

Method	PA	MIU	FWIU
SVM (curvelet features) using image-level labels	0.820	0.550	0.725
Deconvolution networks with our WFL loss, $\gamma = 1$	**0.893**	**0.643**	**0.823**

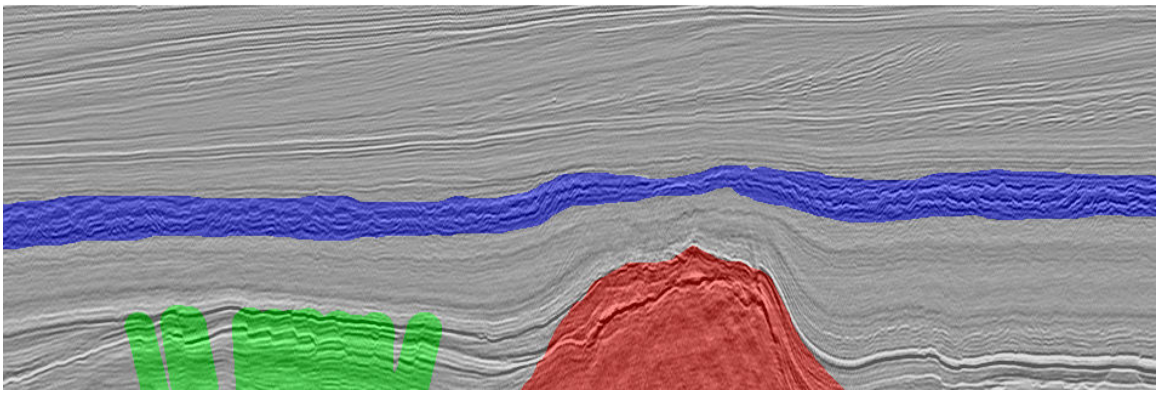
5.5 Summary

In summary, in this chapter, we demonstrated how our weak labels could successfully train a deep deconvolution network for the semantic labeling of subsurface structures.

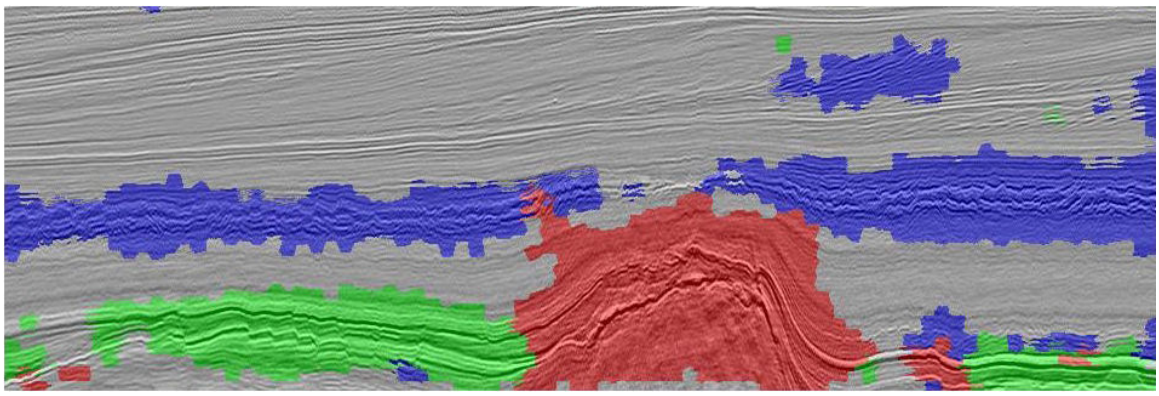
[2] Binary thresholding using Otsu's method, automatic removal of small objects, then morphological closing and dilation.

We proposed an adaptation of the focal loss that allows for the use of continuous confidence values for training deep networks with weak automatically-generated labels. We have shown results on the Netherlands F3 block using both our proposed method and a baseline model that uses the FCN architecture and an equivalent to the CE loss. The results show that mapping image-level labels to pixel-level labels, and then using these labels to train a deep network to do the classification outperforms the technique that was proposed in Chapter 3 that only relied on image-level labels. Our objective results on the Netherlands F3 block show a 10% increase in the FWIU metric. In addition, by observing the resulting labeled section, we notice that the technique presented in this chapter helps reduce false positive classifications.

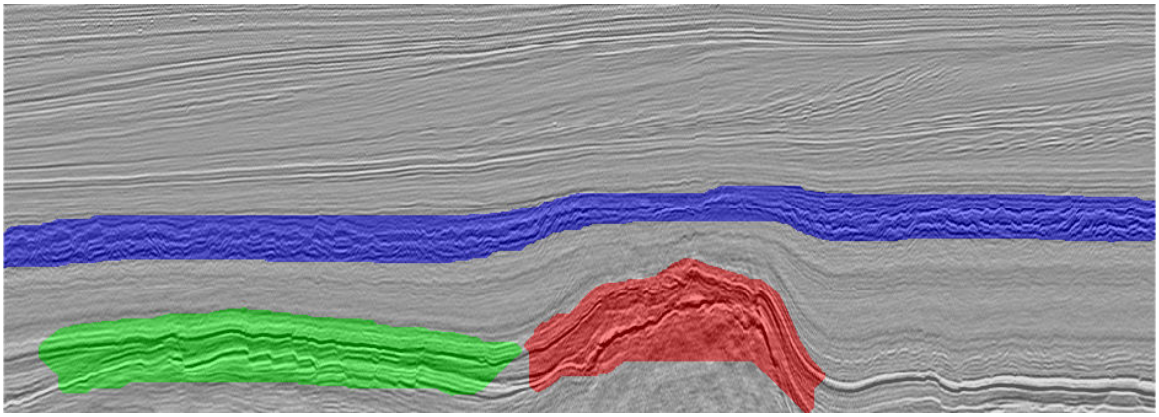
The results in this section are extremely promising, and show that only a few images annotated on the image level are sufficient for our proposed weakly-supervised framework to successfully label various subsurface structures in a large seismic survey such as the Netherlands F3 block. However, given the limited manually annotated data for structural interpretation, it is difficult to objectively analyze our results in more detail. In addition, we would like to study wether our proposed weakly-supervised labeling framework can easily be extended to other problems within the seismic interpretation domain. In the next chapter, we introduce a large fully-labeled seismic stratigraphic interpretation dataset, and apply our weakly-supervised framework to the more difficult problem of facies classification. This allows us to study the robustness of our proposed approach, and allows us to accurately compare the performance of our model trained with either strong or weak labels.



(a)



(b)



(c)

Figure 5.6: (a) Manually labeled inline 380 of the Netherlands North Sea F3 block. (b) Labeling result of the best-performing model in Chapter **??** using curvelet features. (c) Labeling result of the model presented in this chapter that uses mapped pixel-level labels. The `chaotic` class is colored in blue, `faults` is in green, and `salt dome` is in red.

CHAPTER 6
STRATIGRAPHIC INTERPRETATION WITH WEAK PIXEL-LEVEL LABELS

6.1 Overview

In the previous chapters, we have introduced our framework for generating weakly-labeled training data and then training deep networks with these weak labels to semantically label subsurface structures. In this chapter, we apply our weakly-supervised semantic labeling approach to the more challenging problem of seismic stratigraphic interpretation.

Stratigraphy is a branch of geology that studies rock layers. Stratigraphic interpretation uses observations from seismic data, well logs, and core data to study sedimentary facies and depositional processes. Stratigraphic interpretation is one of the major components in a seismic interpretation workflow and can be a great tool to help identify hydrocarbon system elements such as reservoirs, seals, and source rocks [166]. An important task in stratigraphic interpretation is seismic facies classification where the goal is to predict overall rock types from seismic data. When only seismic data is used, only large-scale lithostratigraphic features such as lithostratigraphic groups or formations and regional (or global) sequences can be resolved [167]. As Figure 6.1 shows, the resolution of seismic data limits the size of stratigraphic units that can be resolved. Smaller stratigraphic subdivisions are typically resolved through well logs and core data. Figure 6.2 shows examples of various exposed lithostratigraphic units. Our goal in this chapter is to classify every pixel in a 3D seismic volume according to their main lithostratigraphic unit.

In our effort to extend our weakly-supervised semantic labeling approach to strati-

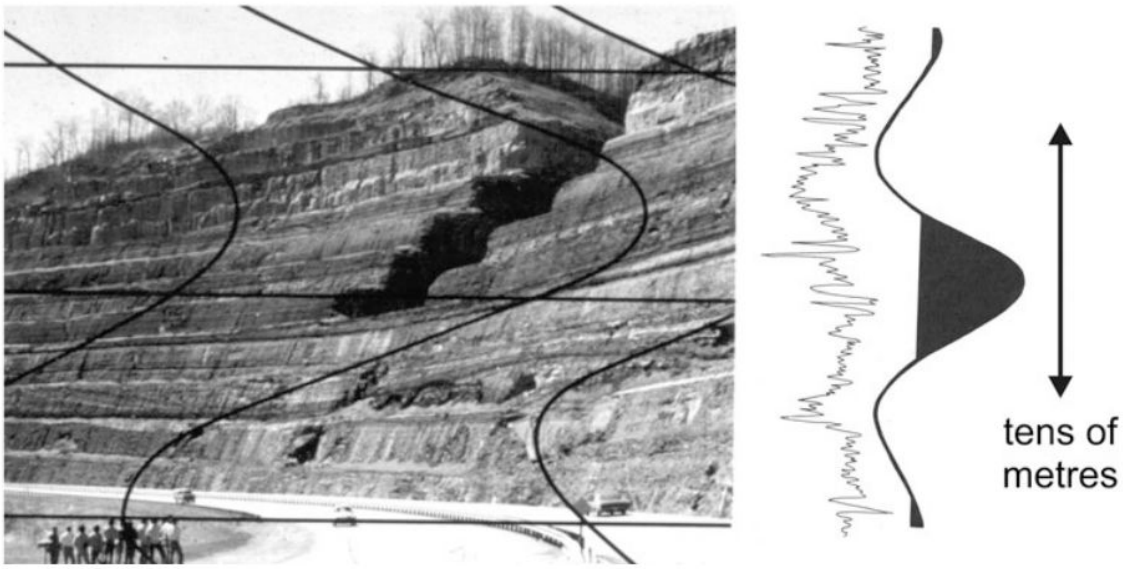


Figure 6.1: The scale of a typical seismic waveform compared to an outcrop (left), and compared to a wireline log (right). The frequencies used in seismic exploration (10–60 Hz) have long wavelengths, and therefore, the resolution of seismic data is limited to large-scale stratigraphic features. Figure adapted from [167] with permission. ©(2016) Springer.

graphic interpretation, and as a result of a collaboration with an experienced geoscientist, we released the largest fully-annotated seismic facies classification dataset currently available [168]. We also proposed two fully-supervised baseline models for facies classification based on a deconvolution network architecture. The first baseline is a patch-based model that is trained using a large number of small patches extracted from all the inlines and crosslines in the training set. The second baseline is a section-based model that was trained directly on entire inlines and crosslines of the data. The results of these baseline models are later compared to the results of our weakly-supervised models. In addition, we have open-sourced all the codes that were used to train and test our baseline models using the PyTorch deep learning library[1]. The goal of making the dataset and the code publically available is to help advance the progress of machine learning research in this domain by providing a large high-

[1]Code and data are available from: `www.github.com/olivesgatech/facies_classification_benchmark`

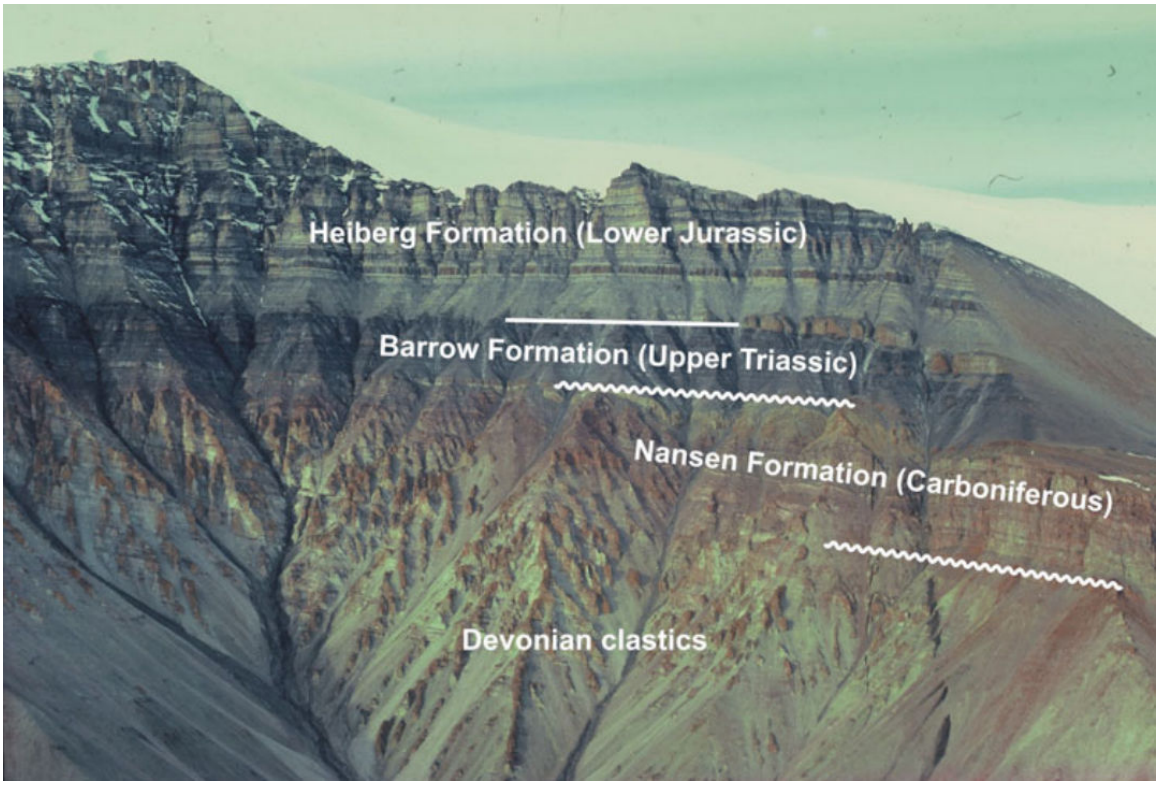


Figure 6.2: An example of exposed lithostratigraphic formations in the mountains of northern Ellesmere Island, Canada. Figure adapted from [167] with permission. ©(2016) Springer. Photo credit A. F. Embry.

quality annotated dataset with a specific training and testing scheme for researchers to train and benchmark their various models and to allow other researchers to further build on previous advances in this domain. We introduce this dataset in detail in Section 6.3.

While our weakly-supervised framework was designed for structural interpretation tasks, applying it to stratigraphic interpretation would allow us to analyze our approach further, and study its applicability to other problems in seismic interpretation. Furthermore, given the amount of annotated data that we have released in our dataset, we can better analyze the performance of our weakly-supervised approach compared to conventional fully-supervised techniques.

In the next section, we review the relevant facies classification literature; then in

Section 6.3, we introduce our dataset in detail including background knowledge about the geology of the Netherlands F3 block, and how our dataset was created. Then, in Section 6.4, we introduce two baseline models for facies classification. In Section 6.5 we describe the experimental setup and show how we obtained weak labels for facies classification using our weakly-supervised framework. We show the results in Section 6.6 and compare the results of our fully- and weakly-supervised models. Finally, we summarize this chapter in Section 6.7.

6.2 Background

Facies classification is a commonly studied problem in the seismic interpretation literature. There is a very rich literature on traditional supervised and unsupervised methods for facies classification, e.g., [169, 170, 171]. These include methods that are based on SVMs and artificial neural nets. Also, many unsupervised facies classification (or more accurately, clustering) methods have been proposed in the literature. K-means, principal component analysis (PCA), and self-organizing maps (SOM) are some of the most popular unsupervised approaches for facies clustering. Zhao *et al.* [172] provides a review of some of the most commonly used traditional techniques.

In recent years, facies classification methods based on deep learning have shown great promise. In 2017, Rutherford Ildstad and Bormann [65] proposed a basic 5-layer convolutional neural network (CNN) for facies classification, and made the annotated data publically available. Rutherford Ildstad and Bormann only partially annotated a single inline from the Netherlands F3 block (shown in Figure 6.3) for their model. Dramch and Lüthje [173] used this partially annotated inline to fine-tune CNN models pretrained on ImageNet, and compared the performance of different CNN architectures such as VGG16 and ResNet50 at facies classification. Zhao [174] trained two CNN architectures for facies classification, an image-classification model, and an encoder-decoder style architecture, and compared their results. Zhao annotated 40

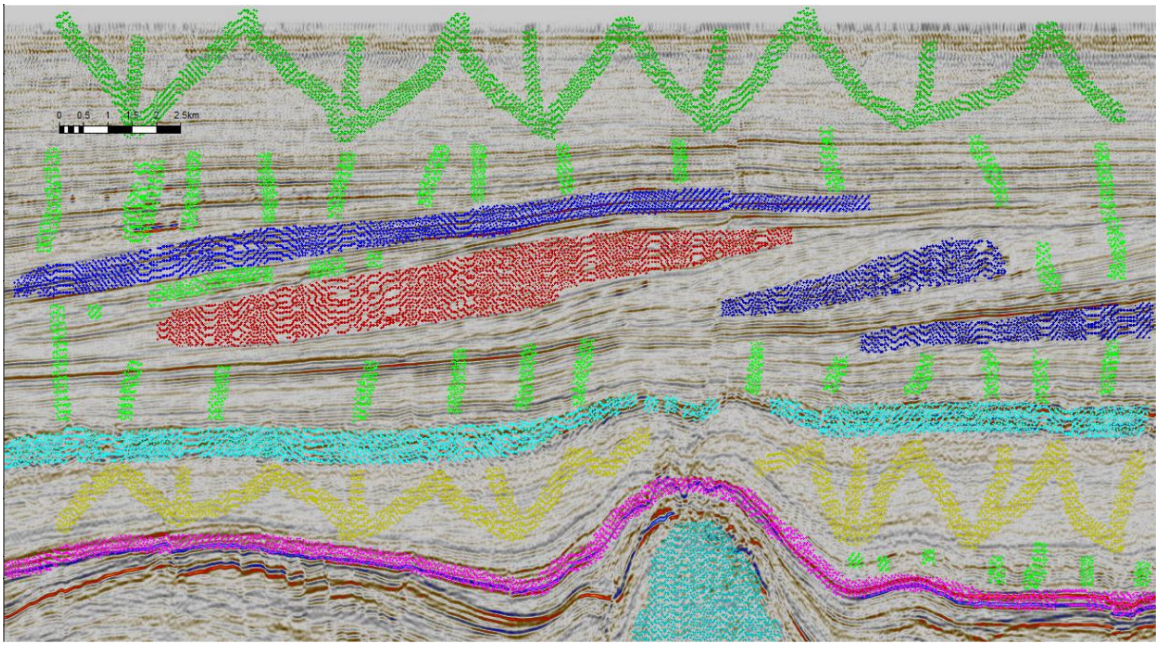


Figure 6.3: The publically-available annotated inline of the Netherlands F3 block from Rutherford Ildstad and Bormann [65]. The inline contains partial annotations for nine classes of seismic facies.

additional inlines of the Netherlands F3 block using similar classes of facies as those used by Rutherford Ildstad and Bormann [65]. However, the annotated data is not publically available. Furthermore, the results in [174] suffer from severe overfitting, and the testing scheme (10% of the training data randomly selected) does not accurately reflect real-life scenarios where the testing data will not be highly correlated with the training data. Similarly, Di *et al.* [175] expanded on the annotations of Rutherford Ildstad and Bormann by manually annotating 12 inlines. He then trained a 6 layer deconvolution network on his annotated data. These inlines were not made publically available. The method proposed by Di *et al.* forces all the images in the seismic volume to be resized to a fixed 256×256 grid, therefore losing many of the details in the seismic data. Furthermore, the results on the training data are not very accurate, and no quantitative results were given.

The limited number of annotated sections used by recent papers [65, 174, 175] is

understandable given that the annotation process is time-consuming, requires subject matter expertise, and can be fairly subjective. However, this limited quantity of annotated data undermines the mass potential machine learning could have when deployed in such a field. To overcome this problem, some researchers have used unsupervised deep learning techniques such as deep convolutional autoencoders [176, 177, 178], while Peters *et al.* [179] proposed a weakly-supervised method that uses partial annotations and extends them along class boundary lines.

Whether researchers annotate their own training data or use other techniques, there still remains a lack of large publicly-available annotated datasets for seismic stratigraphic interpretation that can be used for training and comparing the performance of different models. Furthermore, it is common for papers that apply deep learning for facies classification, or other seismic interpretation tasks, to not contain quantitative results, but rather rely solely on subjective visual inspection of the results (e.g., [175]). All of this leads to highly subjective results and greatly hinders the ability of researchers to compare different approaches against each other and understand the advantages and disadvantages of each approach.

To address these issues, and to help make machine learning research in seismic interpretation more reproducible, we open-source a fully-annotated 3D geological model containing multiple classes of lithostratigraphic units in the Netherlands F3 Block. This model is grounded in the geology of the region and based on the study of both the 3D seismic data and 26 different well logs in the Netherlands F3 block or its vicinity. The data also includes fault planes that we have extracted from the F3 block. The next section provides a brief overview of the geology of the Netherlands F3 block and introduces our annotated model and how it was obtained.

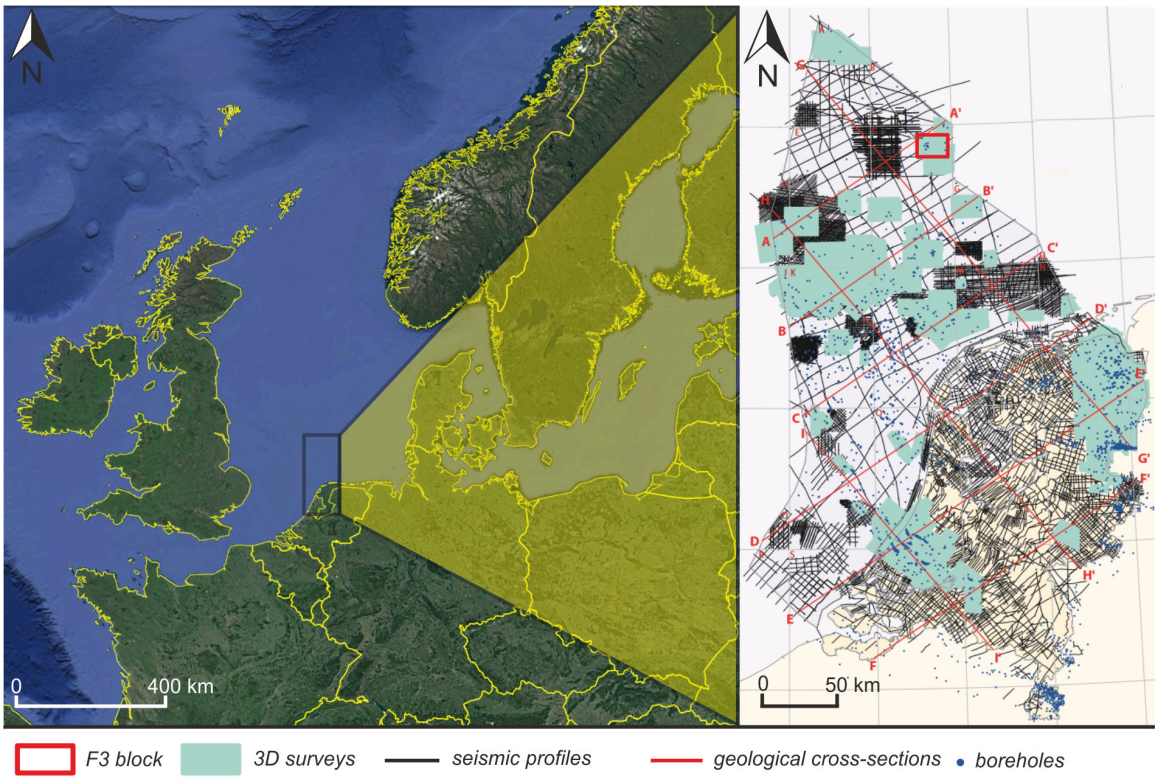


Figure 6.4: The location of the F3 block. Adapted from [180].

6.3 A 3D Geological Model of the Netherlands F3 Block

The North Sea is rich in hydrocarbon deposits, which is why this area is very well studied in the literature [181]. The North Sea continental shelf, located off the shores of the Netherlands, is divided into geographical zones described by different letters of the alphabet; within these zones are smaller areas marked with numbers. One of these areas is a rectangle of dimensions 16 km x 24 km known as the F3 block, see Figure 6.4. In 1987, the F3 block 3D seismic survey was conducted to identify the geological structures of this area and to search for hydrocarbon reservoirs. In addition, many boreholes were drilled within the F3 block throughout the years. The F3 block became one of the most widely known and studied seismic surveys after dGB Earth Sciences made the data obtained from the survey publicly available.

This section aims to briefly describe the geology of the survey area and introduce

the 3D geological model that we have developed and how it was obtained.

6.3.1 The geology of the F3 block

Within the continental shelf of the North Sea, ten groups of lithostratigraphic units have been identified in the literature [182, 183, 184, 180]. These groups and their main lithostratigraphic features are listed below from newest to oldest:

1. Upper North Sea group: claystones and sandstones from Miocene to Quaternary.
2. **Lower and Middle North Sea groups**: sands, sandstones, and claystones from Paleocene to Miocene.
3. **Chalk group**: carbonates of Upper Cretaceous and Paleocene.
4. **Rijnland group**: clay formations with sandstones of Upper Cretaceous.
5. **Schieland, Scruff and Niedersachsen groups**: claystones of Upper Jurassic and Lower Cretaceous.
6. **Altena group**: claystones and carbonates of Lower and Middle Jurassic.
7. **Lower and Upper Germanic Trias groups**: sandstones and claystones of Triassic.
8. **Zechstein group**: evaporites and carbonates of Zechstein.
9. **Upper and Lower Rotliegend groups**: siliceous rocks and basalts of the Lower Zechstein.
10. **Limburg group**: Upper carboniferous siliceous rock, which are the bedrock for hydrocarbons.

The F3 block is located on the border of two tectonic structures: the Step Graben and the Dutch Central Graben (see Figure 6.5). These tectonic structures are characterized by different lithostratigraphic units of varying thickness. This varying thickness is a result of tectonic activity [185, 186], which was started in the Variscan orogeny [187]. The area within the Step Graben is strongly disturbed by salt diapirs, which were active several times, from the Zechstein to the Paleogene period [188]. On the other hand, and as a result of subsiding Jurassic rocks, the Altena, Scruff, Schieland and Niedersachsen groups are observed only within the Dutch Central Graben [180].

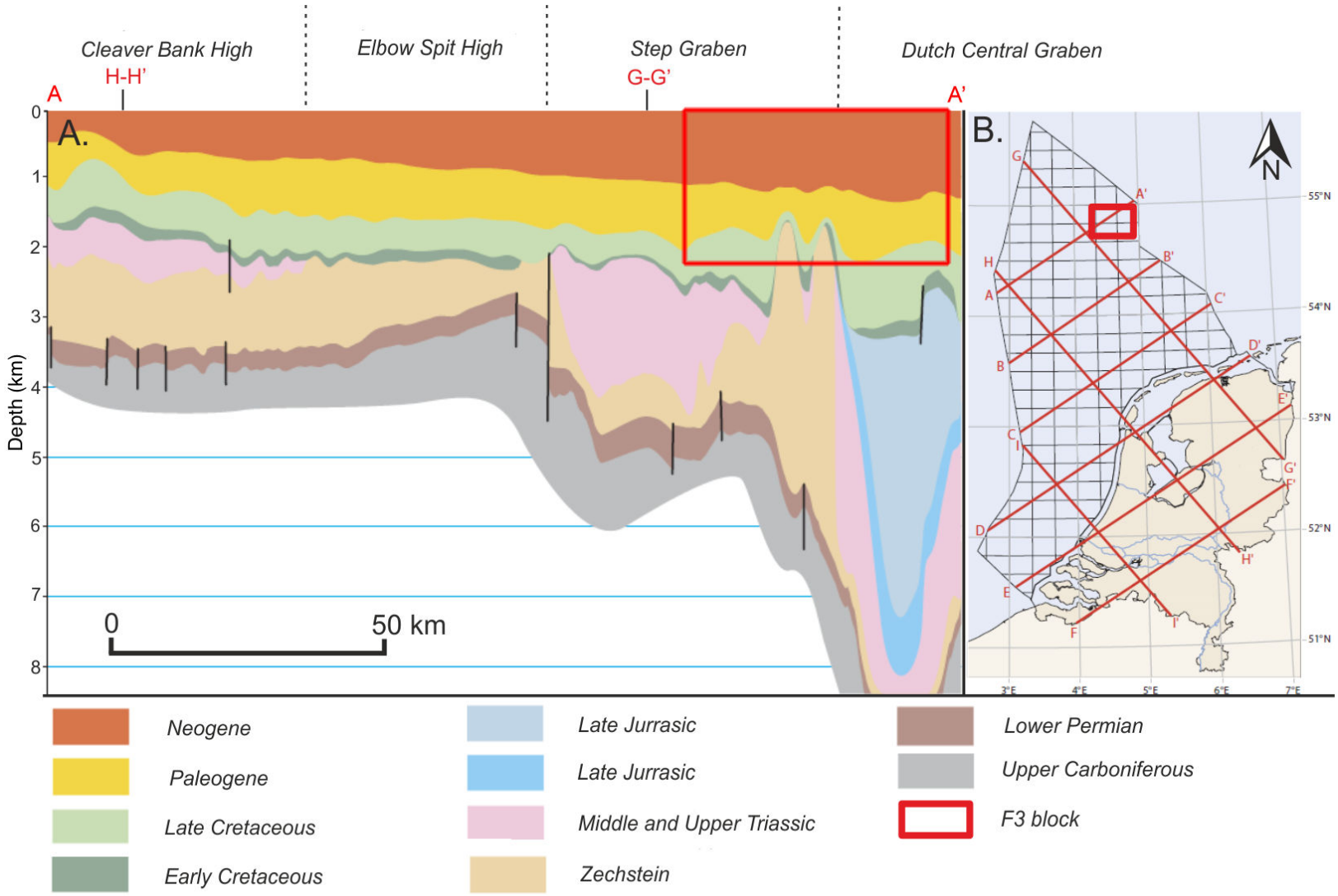


Figure 6.5: A) A geological cross-section of the North Sea continental shelf along axis A-A'; B) A map of the location of the cross-section. Adapted from [180].

6.3.2 The modeling process

To prepare our 3D geological model of the F3 block, we relied on both well logs and 3D seismic data. The next two subsections describe this process.

3D model building using well logs data

The well log data were obtained from a website managed by the Geological Survey of the Netherlands (`www.nlog.nl`). The data (including information related to coordinates, true vertical depth, measured depth along the curvature, inclinations, and individual horizons) were collected for 26 boreholes located within the F3 block or its vicinity. The exact locations of these wells are visualized in Figure 6.6.

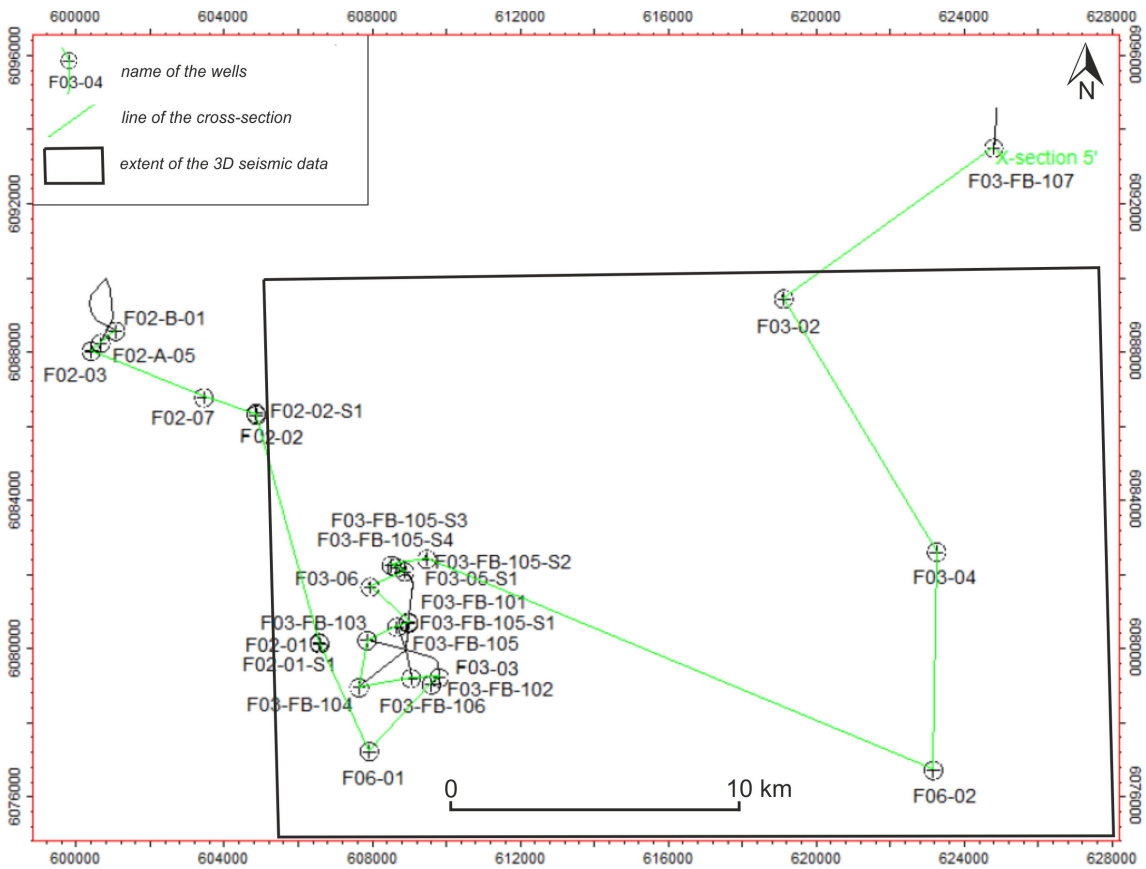


Figure 6.6: Locations of the boreholes that were used to create the geological model.

Originally, the 26 wells contained 40 different horizons, so it was necessary to as-

sign these different horizons to the various lithostratigraphic units that were adopted in literature and were presented in the previous subsection. The next step was correlating wells with each other. After that, it was possible to create a preliminary 3D model based on the well log data by using Petrel's *make/edit surface* tool. This process facilitated the preliminary visualization of the range of individual horizons, which was very helpful in the further interpretation of the 3D seismic data.

3D model building using seismic data

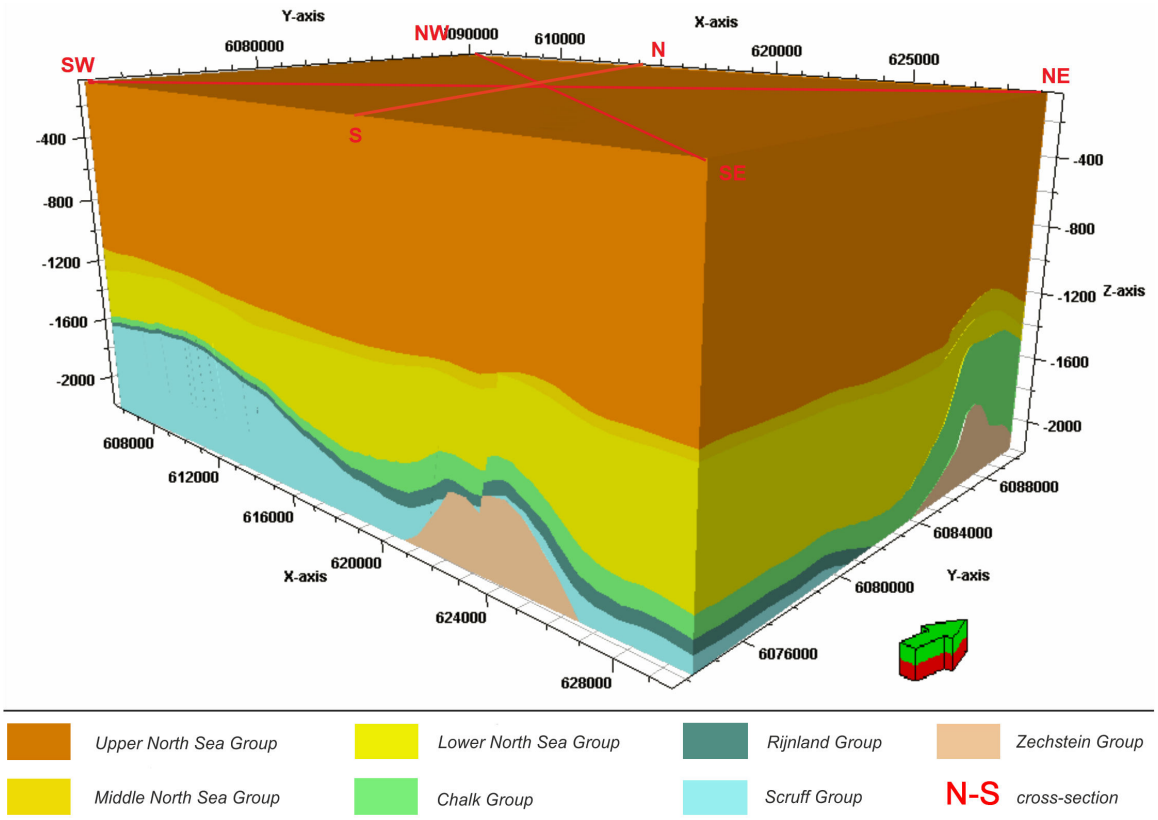


Figure 6.7: A 3D view of our geological model of the F3 block.

The F3 block data was migrated in time, not depth, so it was necessary to do time-depth conversion since the structural model must be prepared in the depth domain. OpendTect 5.0 was used to perform the time-depth conversion using a velocity model that was provided with the F3 block data.

The next step in creating the model was faults-surface interpretation. Using Petrel's *polygon editing* tool, we interpreted the main fault surfaces and the fault networks were created by using the *fault framework modeling* tool. Horizons were interpreted in a similar fashion, but by using the *seeded 3D autotracking* tool, which interpolated data automatically and took into account the faults networks modeled previously.

Based on the interpreted horizons and faults, preliminary modeling was conducted. This was done using the *horizon modeling* tool with *volume-based modeling*, which is an advanced method of isochronous geological space modeling. The preliminary model included several imperfections in the interpretations of horizons and faults, so it was necessary to re-model several faults and conduct small corrections in the interpreted horizons.

After this, it was possible to create the final three-dimensional model which highlights the regions between individual horizons. Here, Petrel's *structural modeling* module in the *horizon modeling* tool was used in addition to the *create zone model* function. The final 3D geological model is shown in Figure 6.7.

6.3.3 The 3D geological model

Within our 3D geological model of the F3 Block, we identified seven groups of lithostratigraphic units (see Figure 6.7). These are (from newest to oldest): the Upper North Sea group, the Middle North Sea group, the Lower North Sea group, the Chalk group, the Rijnland group, the Scruff group, and the Zechstein group.

These groups can be divided into three structural levels: **Cenozoic** (Lower, Middle, and Upper North Sea groups), **Mesozoic** (Scruff, Rijnland, and Chalk groups), and **Permian** (Zechstein group).

As is evident in Figure 6.7, the F3 Block is characterized by highly variable geological structures, both in the horizontal and vertical range, which is manifested by

the differential thicknesses of individual units and by the expanded faults network related to salt tectonics. The area of the F3 Block can be divided into two regions: Eastern and Western. The Eastern region is disturbed by the occurrence of Zechstein diapirs and irregular faults network. The Western region is characterized by regular fault networks and a more uniform thickness of lithostratigraphic units.

The **Upper North Sea group** is the youngest and the flattest lithostratigraphic unit within our model. The top of the Upper North Sea group is the bottom of the North Sea at the same time, which is about -40 meters above sea level (m a.s.l). Differences in the depth of the ocean floor are small, and they are maximally 6 meters within the whole F3 Block. It can be noted that the depth of this top decreases from SW to NE. The thickness of the Upper North Sea group varies from about 1000 m (in places deformed by Permian diapirs) to about 1320 m in the northern part of the research area (see Figure 6.7).

Below the Upper North Sea group lies the **Middle North Sea group**. The depth of the top of this unit ranges from -1000 m a.s.l. within the diapir in NE part of the F3 Block to about -1360 m a.s.l. in the northern part of this area, between diapirs. The thickness of the Middle North Sea group is from 20 to 150 m. As in the case of the Upper North Sea group, there is a clear relationship between the occurrence of Zechstein salts and the depth and thickness of this unit. Differences in the thickness of this unit between both sites of faults are also visible.

The next unit is the **Lower North Sea group**. This unit contains similar lithostratigraphic units to the Middle North Sea group, but is visually distinct in the seismic data. The top is at a depth from -1100 m a.s.l., while the thickness is from about 180 to 750 m.

The top of the **Chalk group** is at depth from -1300 m a.s.l. (above the diapirs in the NE part of the survey) to -2100 m a.s.l. (in the Eastern part of the survey, which is undisturbed by diapirs). The minimum thickness of this unit is 25 m, while above

the salt diapirs in NE part of the F3 Block, this substantially increases to 525 m.

The **Rijnland group** is submerged in the NNE direction, while it is the shallowest in the SW part of the F3 Block and above some Zechstein diapirs at the center of the survey (see Figure 6.9). The maximum thickness of the Rijnland group is about 200 m (above some diapirs), while in the other parts of the F3 block it can be less than 20 m or does not occur at all.

The **Scruff group**, similar to the Rijnland group, is thinned out in NNE direction, more or less in the middle of the F3 Block, where the top of this layer has a depth of -2180 m a.s.l. This layer is shallowest (-1500 m a.s.l.) in the SW part of the F3 block and above the Zechstein diapirs in the Southern part of the survey. The thickness of the Scruff group within our model boundaries ranges from 100 m to almost 700 m, but is much larger in reality and can reach several kilometers [180].

The **Zechstein group** occurs only in the eastern part of the survey, as irregularly-shaped salt diapirs. The shallowest part of the Zechstein group is at a depth of -1500 m a.s.l. while the maximum thickness of the Zechstein group within the research area is about 700 m. However, as in the case of the Scruff group, the depth is much bigger. According to the literature, it can reach several kilometers [180].

In addition to the identified groups of lithostratigraphic units mentioned above, we have also identified three generations of **faults**. The first generation are reverse, oblique-slip, sinistral faults with an SSW-NNE orientation. This direction is connected with the course of the tectonic axis of the Dutch Central Graben, which (similar to the whole Graben) has an SSW-NNE orientation. The second generation of faults are normal, oblique-slip, dextral faults with a W-E orientation. Finally, the third generation are faults that are genetically linked with faults from the first and second generations, but were disturbed by the Permian halokinesis. Figure 6.8 shows an overhead view of the three generations of faults that we have identified. Also, Figure 6.9 shows two diagonal cross sections along the SW-NE and NW-SE axis in

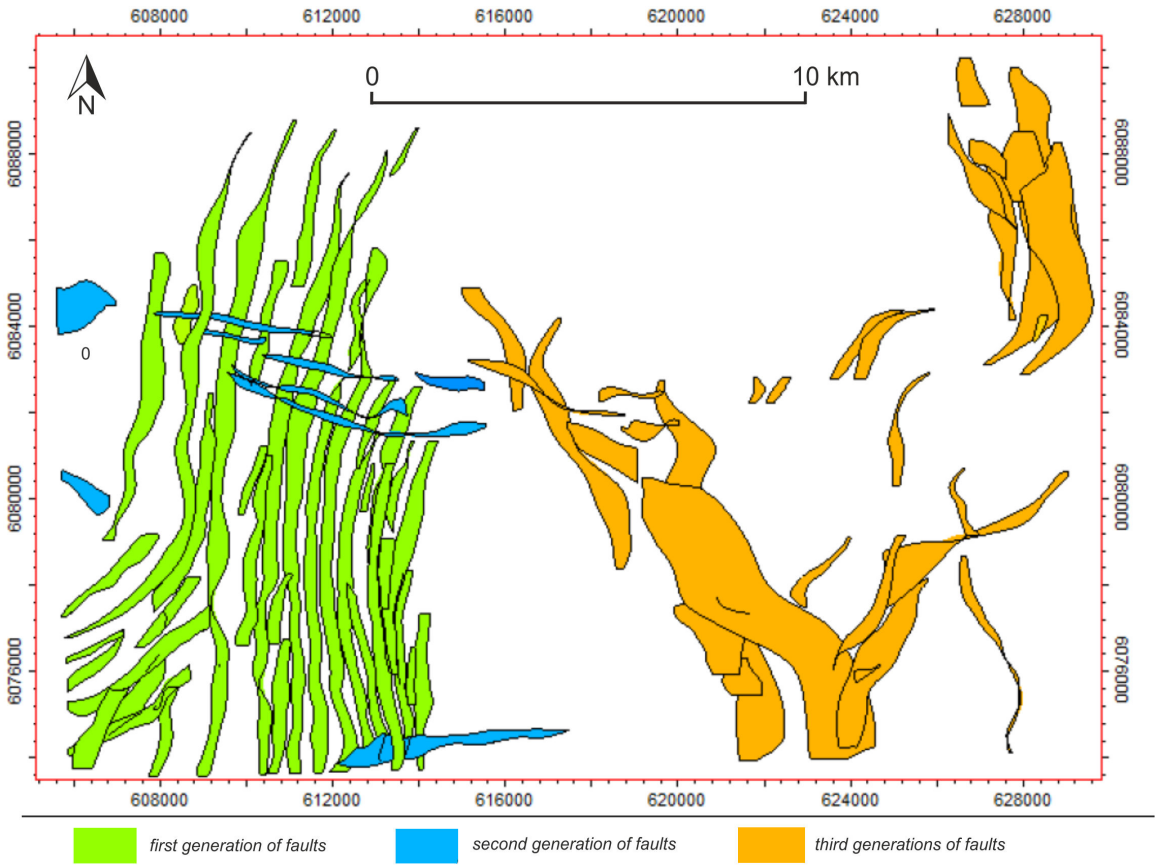


Figure 6.8: An overhead view of 3D fault planes from three different generations of faults that we have identified in the F3 block.

our 3D model shown in Figure 6.7.

6.4 Deconvolution Network Baseline

To benchmark our weakly-supervised models, we propose two fully-supervised baseline models for facies classification based on the deconvolution network architecture we used previously in Chapter 5. The two baseline models are a section-based and a patch-based model. These two models use the same architecture and almost identical hyperparameters but differ in the way they are trained and the way they are used to label the seismic volume.

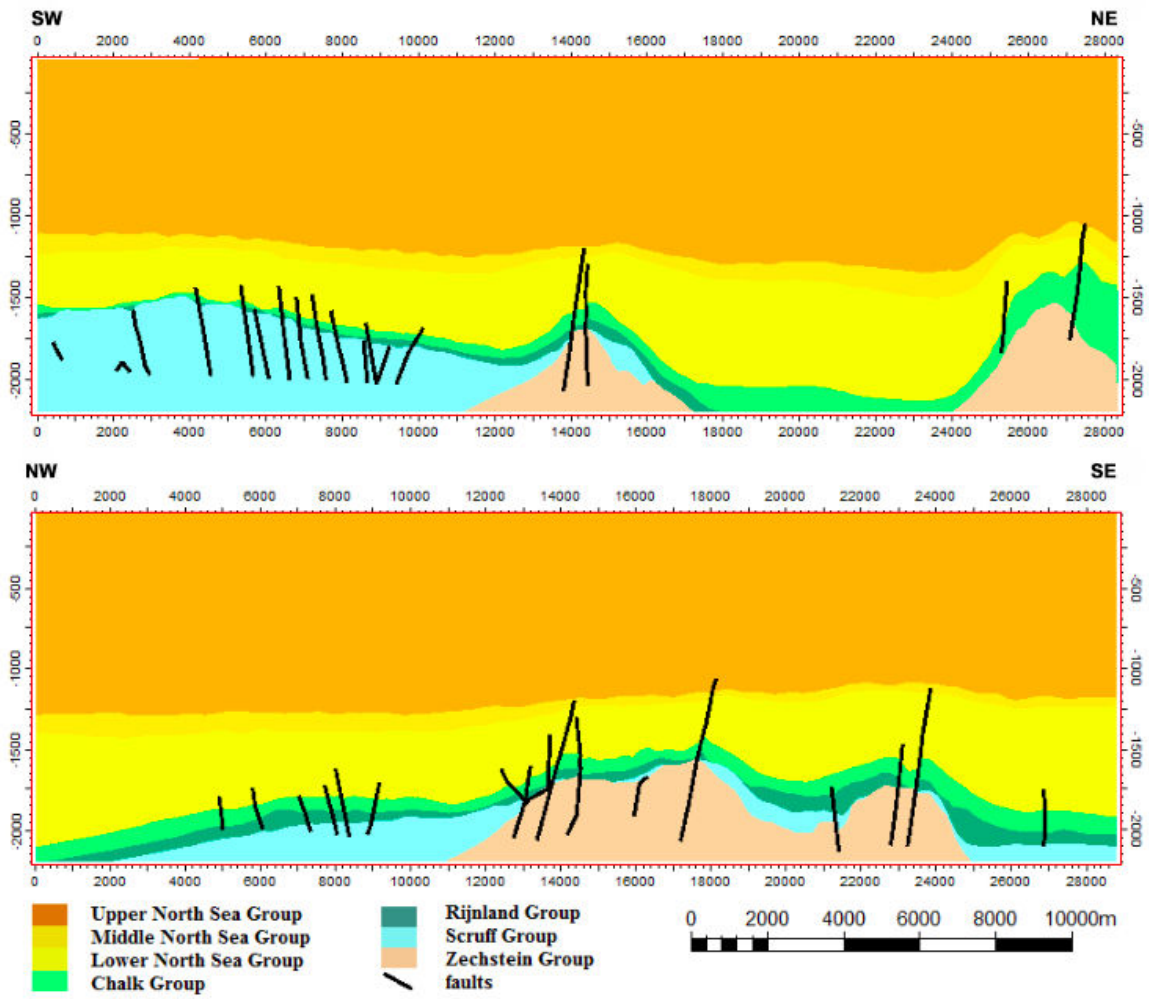


Figure 6.9: Two diagonal cross sections of our 3D geological model in Figure 6.7.

Patch-based model:

The patch-based model is trained on small patches extracted from the inlines and crosslines of the training data. For very large seismic volumes, this approach can be more feasible than using entire sections for training. At training time, the patches of seismic data and their associated labels are sampled randomly from the inlines and crosslines of the training set. During test time, the model samples overlapping patches in the inline and crossline direction and averages the results to generate a 2D labeled version of the test inline or crossline. This is done for all inlines and crosslines

in the test sets. Since our weak labels are generated on the patch-level, we can only train a weakly-supervised patch-based model.

Section-based model:

The section-based model is trained on entire inline and crossline sections. The advantage of this approach is two-fold. First, since the network is fed an entire section, it can easily learn the relationships between different lithostratigraphic units and can take the depth information into account when labeling the section. The second advantage is more practical. Training and testing entire sections at once means the network can be trained or tested very quickly since there are only a relatively small number of seismic inlines and crosslines[2]. One advantage of using a fully convolutional architecture (such as the one we are using) is that the size of the network input does not have to be fixed. The size of the output of the network changes as the size of its input change. Therefore, the different size of the inline and crossline sections does not pose any problem to the training of this network[3].

Other variations:

In addition to the baseline section- and patch-based models, we have trained other variations of these models to test how they can be improved. We have tested the following variations:

In addition to the baseline patch- and section-based models, we have trained other variations of these models to test how they can be improved. We have tested the following variations:

- *Baseline + data augmentation:* data augmentation applies different label-preserving

[2]This is assuming the GPU memory is large enough to handle the size of the seismic sections. On our NVIDIA Titan X GPU, we trained the baseline section-based network – eight sections at a time – in about 70 minutes.

[3]While the sizes of the inlines and crosslines do not need to match, their resolutions (in terms of meters/pixel) should. In our case, pixels in the inline and crossline directions are both 25m×25m.

transformations to the training data such as small rotations, random horizontal flipping, and the addition of Gaussian noise. This can help increase the training sample size, and help the network generalize better to the test data.

- *Baseline + data augmentation + skip connections:* we further improve on the previous model by adding skip connections. In a deep neural network, the output of a layer is typically passed on as the input to the next layer in the network. Skip connections allow the output of a layer to be also passed as an input to a layer farther up the network, skipping intermediate layers in the process. These connections are implemented by directly adding the outputs of various layers in the encoder part of the deconvolution network to the outputs of the corresponding layers in the decoder. Skip connections help networks overcome the vanishing gradient problem [189] by providing "shortcuts" for the computed gradients to propagate to the lower layers of the network.

6.5 Experimental Setup

In this section, we introduce the main elements of the experimental setup, including how the final geological model was created, how the model is split into training and testing sets, and the results of applying similarity-based retrieval and weakly-supervised label mapping to obtain weak labels from our training set.

6.5.1 The geological model

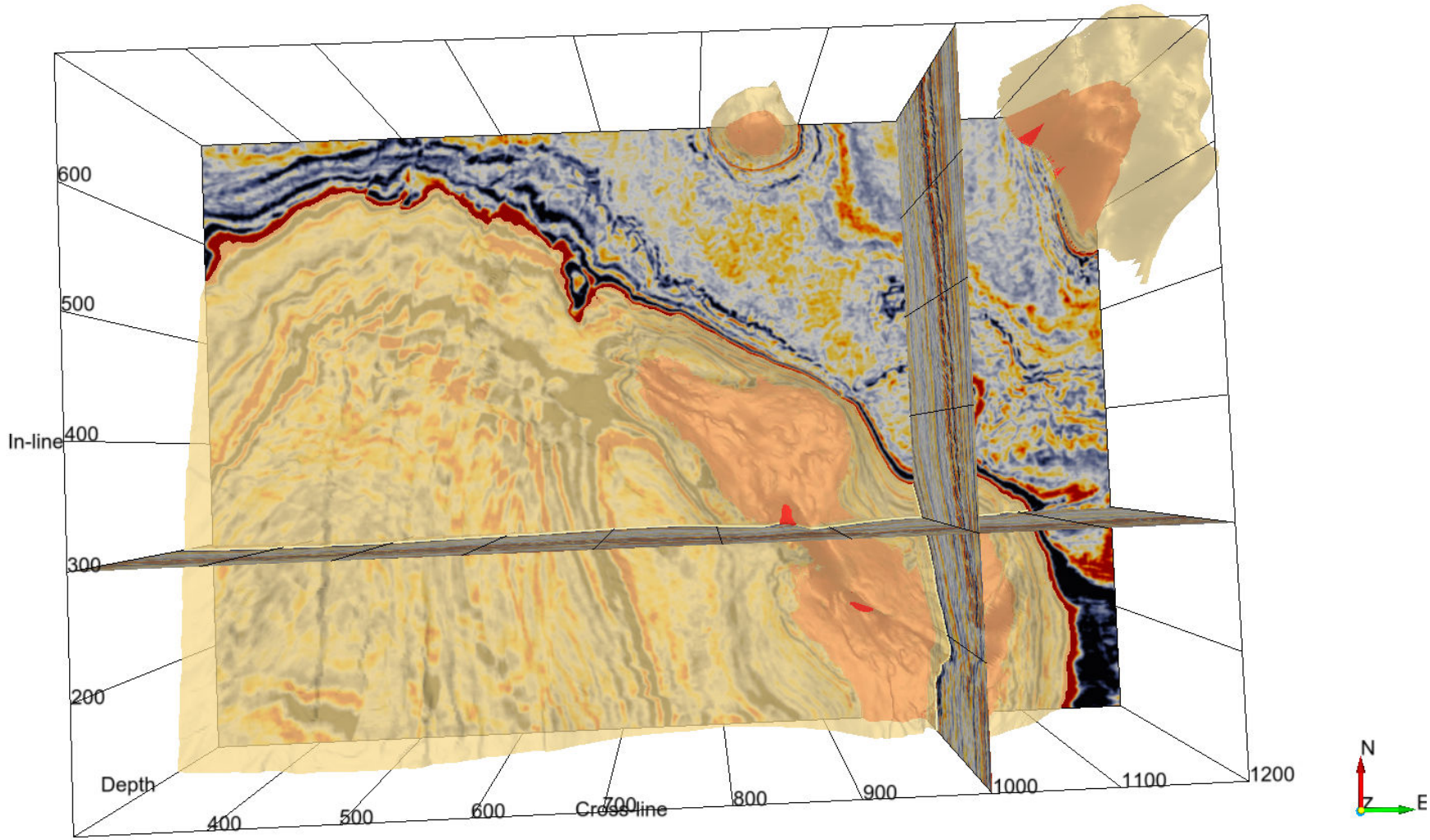


Figure 6.10: A 3D view of the F3 block from above with the Zechstein Group shown in red, while the Chalk Group is shown in a semi-transparent beige color. Inline 300 and crossline 1000 divide the survey into four regions. The NW region of the survey is used for training, while the SW region constitutes the first test set. The remaining region East of crossline 1000 constitutes the second test set.

Table 6.1: The percentage of pixels from different classes in the training set.

Zechstein	Scruff	Rijnland/Chalk	Lower N. S.	Middle N. S.	Upper N. S.
1.48%	3.17%	6.53%	48.44%	11.89%	28.49%

The final geological model that we use to train and test our models is not the entire volume shown in Figure 6.7. The time-depth conversion process of the seismic data resulted in some artifacts. These artifacts were concentrated along the sigmoidal structure in the Upper North Sea group. Due to these artifacts, and missing data on the sides of the survey, we only use the data between inlines 100 and 701, crosslines 300 and 1201, and depth between 1005 and 1877 meters. Furthermore, we combine the Rijnland and the Chalk groups in our final model to a single class due to various issues with processing the Rijnland/Chalk boundary when generating the final model. Table 6.1 shows the percentage of different classes in our training set.

In addition to the final model labels and seismic data, we also release the original horizons for all the lithostratigraphic units, in addition to the extracted fault planes from all three generations.

6.5.2 The train/test split

Careful selection of the training and testing sets is crucial in any machine learning application. This is especially important in seismic data, where neighboring sections are highly correlated. Selecting the training and testing sections randomly will lead to artificially good test results, that are not representative of the actual generalization performance of the tested models. Therefore, it is essential to minimize the correlation between the training and testing sets as much as possible. It is also important to ensure that both the training and testing sets have adequate representation of all the classes in the dataset.

Therefore, we decide to split the data as shown in Figure 6.10. Namely, the data is split into the following three sets:

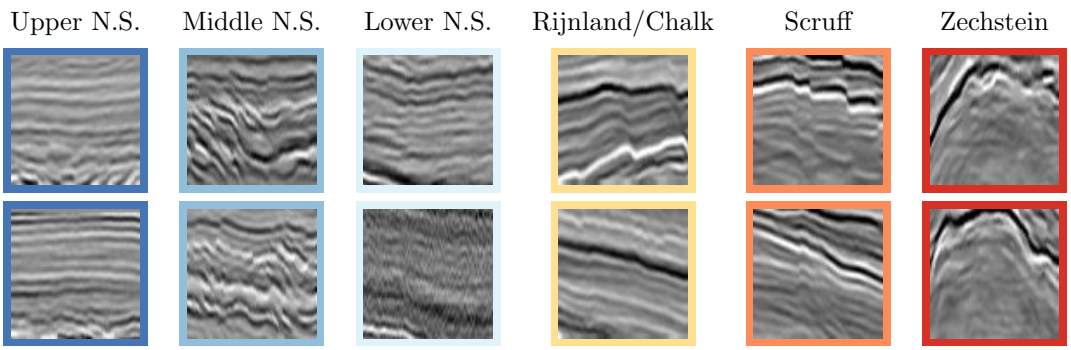


Figure 6.11: The exemplar images of each class of lithostratigraphic units that were used to retrieve the images from the seismic volume. Each class has two exemplar images; one in the inline direction (top row), and another in the crossline direction (bottom row). These images are 75×75 pixels.

1. **Training set:** This includes all the data in the range of inlines [300,700] and crosslines [300,1000].

2. **Test set #1:** This set includes all the data in the range of inlines [100,299] and crosslines [300,1000].

3. **Test set #2:** This sets includes all the data in the ranges of inlines [100,700] and crosslines [1001,1200]. This set includes a large Zechstein diapir in the NE of the survey that is never seen in the training set.

6.5.3 Obtaining weak labels

To obtain weak labels to train our weakly-supervised models, we first select exemplar images from each class of lithostratigraphic units in our dataset. We select two 75×75 exemplar images for every class. One exemplar is selected in the inline direction, while the other is in the crossline direction. Figure 6.11 shows the examplar images that we have chosen. We then apply the similarity-based retrieval method described in Chapter 2 to retrieve $M = 1000$ images for every class of lithostratigraphic units. Example retrieved images from every class are shown in Figure 6.12.

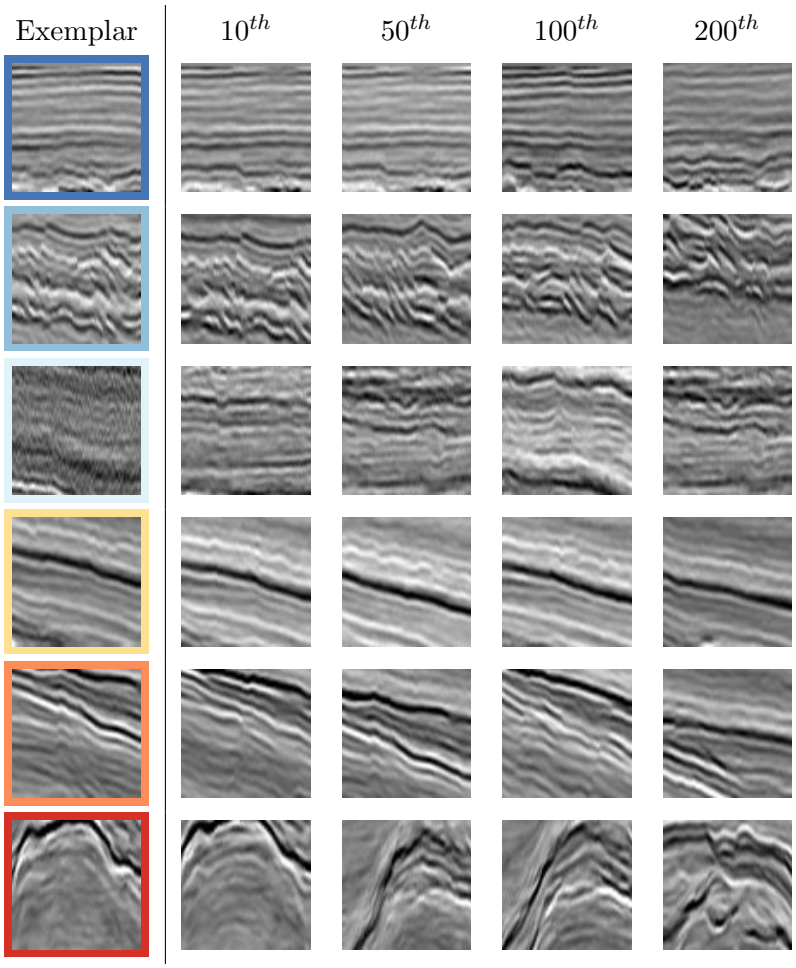


Figure 6.12: Sample retrieved images from each class. The first column shows the exemplar image. The remaining columns show the 10th, 50th, 100th, and 200th retrieved images from each class respectively.

Next, we apply our label mapping algorithm that was described in Chapter 4 to map the image-level labels to pixel-level labels. In the structural interpretation case, there is one structure in each image, and therefore, there is at most two classes in each image: the structure itself and pixels assigned to the `other` class. However, in the stratigraphic interpretation case, the label mapping is far more challenging since each image can contain pixels from many other classes. The larger number of classes certainly affects the results of the label mapping. Figure 6.13 shows examples of images with their mapped labels, along with the ground truth labels for comparison.

In our weakly-supervised models, we exploit our knowledge of the depositional history of the various lithostratigraphic units in the dataset. This is achieved by weighing the confidence maps of each lithostratigraphic units by how far removed they are from the lithostratigraphic unit that is indicated in the image-level label of every image. For example, if an image was assigned a Zechstein image-level label, indicating it was likely extracted from the deeper end of the survey, then confidence maps of shallower classes such as Upper and Middle North Sea will be given less weight than deeper classes such as Rijnland/Chalk or Scruff. Selecting these weights should be a part of the hyperparameter tuning for our weakly-supervised models; however, for the sake of simplicity, we limit the models in our results section to binary weights, such that non-neighboring classes are assigned zero confidence values.

Since we have access to the ground truth labels, we can objectively evaluate the accuracy of our label mapping. These results are summarized in Table 6.2 for various values of the normalized confidence threshold $\tilde{\tau}$. This normalized confidence threshold is applied after confidence values are normalized to the range $[0, 1]$. Any pixel with a confidence value less than $\tilde{\tau}$ is ignored. The first row shows the results for the mapped pixel-level labels when $\tilde{\tau} = 0$. The second row shows the results for when we assign zero confidence values for non-neighboring lithostratigraphic units. We note that this simple adjustment of the confidence values increases the MCA score by 10%. Next, we show the results for when the normalized confidence threshold, $\tilde{\tau}$, is set to 0.167. This value corresponds to $1/N_c$, and indicates what the normalized confidence value would be if the weakly-mapped labels were randomly assigned. As Table 6.2 shows, there is a significant increase in all metrics over the baseline case when these pixels are ignored. This indicates that labels with low confidence are likely to be incorrect, and therefore should not be trusted in the training process. The final row shows the best results for when $\tilde{\tau}$ is optimal (in our case, this is when $\tilde{\tau} = 0.78$). In a real scenario, we do not have access to the ground truth labels, and therefore we are not

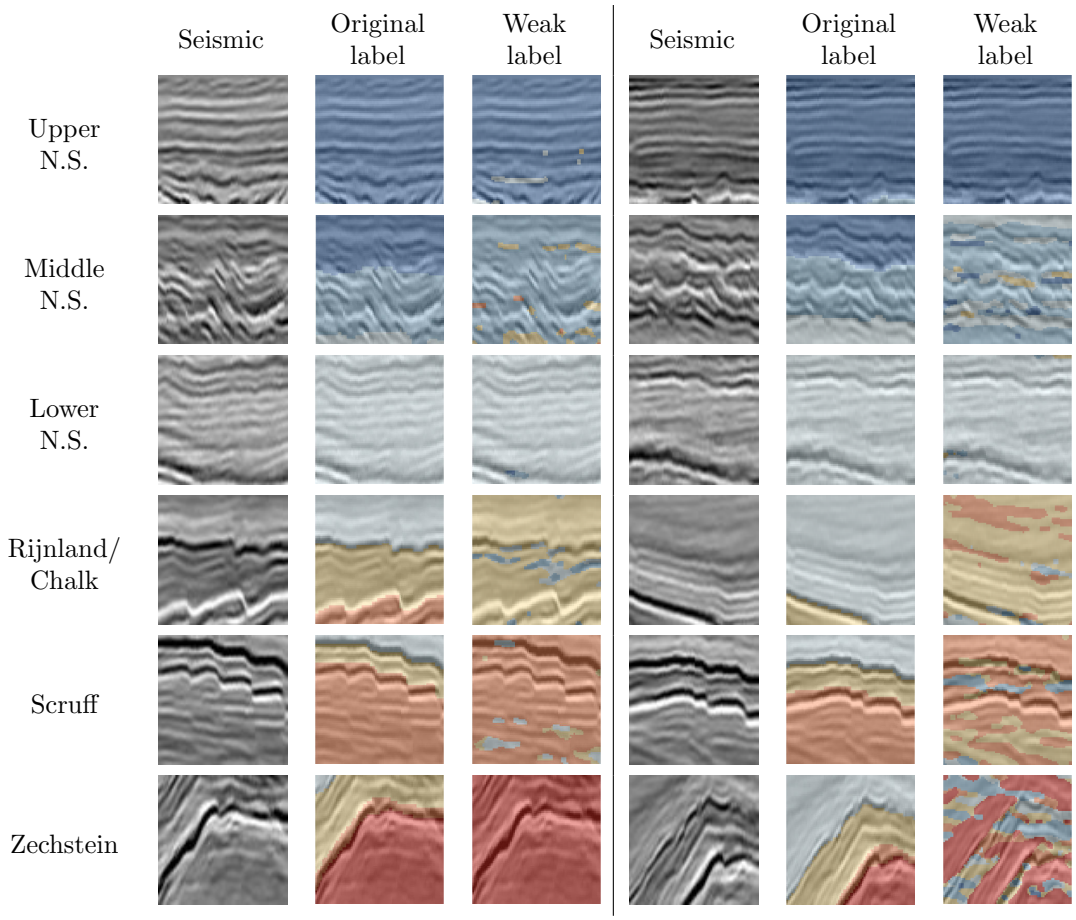


Figure 6.13: Results of the label mapping for each class of lithostratigraphic units. Each row contains two examples from the same class showing the original seismic image, the ground truth pixel-level labels, and the pixel-level labels obtained from our label mapping algorithm.

able to select a value for $\tilde{\tau}$ that maximizes the accuracy of the resulting weak labels. For our weakly-supervised models in the results section, we use $\tilde{\tau} = 0$ while assigning zero confidence values to non-neighboring classes.

6.5.4 Training the models

In the fully-supervised case, we trained both the section- and the patch-based models on the data extracted from the entire training set. The section-based model is trained on the entire set 400 inlines and 700 crosslines that constitute the training set shown in Figure 6.10. The fully-supervised patch-based model is trained on 75×75 patches

Table 6.2: The accuracy of the weak labels used in this chapter compared to the ground truth labels. The accuracy is computed using different metrics for different values of the normalized confidence threshold $\tilde{\tau}$.

Normalized Confidence Threshold $\tilde{\tau}$	PA	MCA	FWIU
0 (*no threshold*)	0.525	0.576	0.394
0 (*no threshold, zeroing out non-neighboring units*)	0.589	0.674	0.454
0.167 (*1/number of classes*)	0.603	0.681	0.474
0.167 (*1/number of classes, zeroing out non-neighboring units*)	0.626	0.717	0.498
0.78 (*optimal*)	0.707	0.738	0.546

extracted at regular overlapping intervals from the training set. The overlap between each patch and the next one is 33%. In the weakly-supervised case, we only have access to weakly-labeled patches, and therefore we only train a weakly-supervised patch-based model. This model is trained only on 6000 weakly-labeled patches. Table 6.3 summarizes the size of the training set for each model.

Table 6.3: The size and amount of training data for various models.

Model	Supervision	Training data size
Patch-based	Fully-supervised	56,000 patches of size 75×75
Section-based	Fully-supervised	400 inlines and 700 crosslines
Patch-based	Weakly-supervised	6,000 patches of size 75×75

All these fully- and weakly-supervised models and their variations are trained on their corresponding training data until their validation loss converges. The Adam optimizer is used for all the models. The cross entropy loss is used for the fully-supervised models, while our weak focal loss (WFL) is used for the weakly-supervised models. Furthermore, we experiment with weighing the WFL for every image by its similarity to the exemplar image that was used to retrieve it. We call the resulting loss function the similarity-weighted WFL or (SW-WFL). For an image $\mathbf{x}_i$, the SW-WFL

can be written as:

$$\mathsf{SW{-}WFL}(q(\mathbf{x}_i), p(\mathbf{x}_i)) = s_i \mathsf{WFL}(q(\mathbf{x}_i), p(\mathbf{x}_i)), \tag{6.1}$$

where s_i is the similarity between image $\mathbf{x}_i$ and the exemplar image that was used to retreive it, and WFL is the weak focal loss that we introduced in Section 5.3.2. The similarity values in the SW-WFL are computed using Method 2 that was proposed in Chapter 2.

6.6 Results

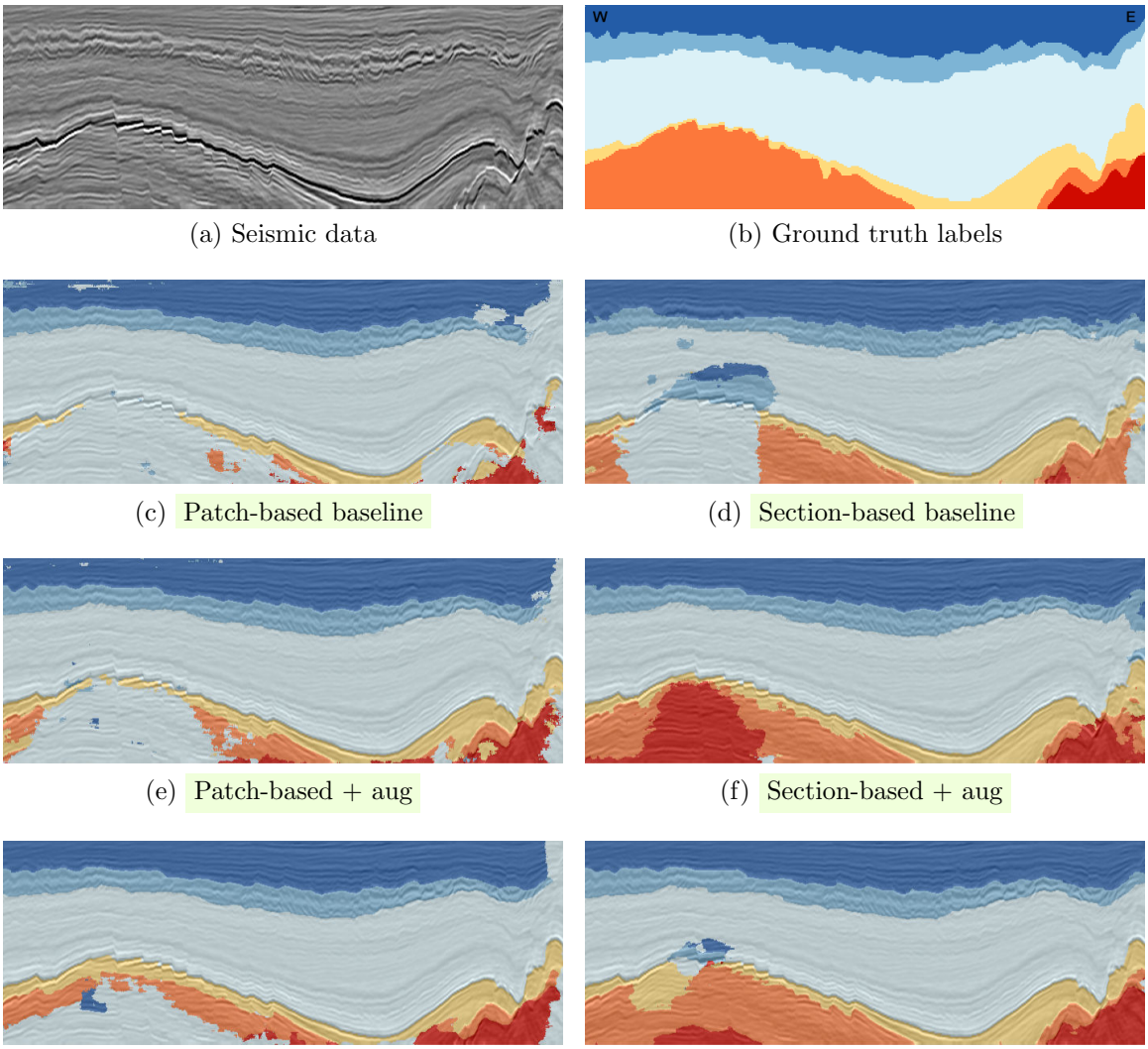
We divide the results section into two subsections. The first describes the results of the fully-supervised baseline models on our facies classification dataset that we have introduced in the previous sections. The second subsection describes the results of our weakly-supervised models and compares them to the fully-supervised ones.

6.6.1 Fully-Supervised Results

We train our fully supervised patch- and section-based models on the data specified in Table 6.3 until they converge on the validation set. We use a 10% hold-out validation set for all our models. After the models have finished training, we test them by using them to predict the labels for all inlines and crosslines in both test sets. We then compute the performance metrics on the final results. Table 6.4 summarizes the objective results for all the fully-supervised models that we have tested on both test sets. Also, Figure 6.14 shows the results for inline 200 in test set #1 for all the models in Table 6.4. We specifically choose inline 200 since it is exactly in the middle of test set #1 and therefore it should give us a better idea of how these models perform on *average* on test set #1. In the remainder of this section, we will discuss these results in more detail.

Table 6.4: Results of various strongly-supervised models when tested on both test splits of our dataset. All metrics are in the range $[0, 1]$, with larger values being better. The best performing model for every metric is highlighted in bold.

Model \ Metric	PA	Class Accuracy						MCA	FWIU
		Zechstein	Scruff	Rijnland/Chalk	Lower N. S.	Middle N. S.	Upper N. S.		
Patch-based baseline	0.788	0.264	0.074	0.499	**0.992**	0.804	0.754	0.565	0.640
Patch-based + aug	0.852	0.434	0.221	0.707	0.974	0.884	0.916	0.689	0.743
Patch-based + aug + skip	0.862	0.458	0.286	0.673	0.974	0.912	0.926	0.705	0.757
Section-based baseline	0.879	0.219	0.539	0.744	0.951	0.872	0.973	0.716	0.789
Section-based + aug	0.901	**0.714**	0.423	**0.812**	0.979	**0.940**	0.956	0.804	**0.844**
Section-based + aug + skip	**0.905**	0.602	**0.674**	0.772	0.941	0.938	**0.974**	**0.817**	0.832



(a) Seismic data

(b) Ground truth labels

(c) Patch-based baseline

(d) Section-based baseline

(e) Patch-based + aug

(f) Section-based + aug

(g) Patch-based + aug + skip

(h) Section-based + aug + skip

Figure 6.14: The results of the different fully-supervised models on inline 200 from test set #1.

Patch-based vs section-based models:

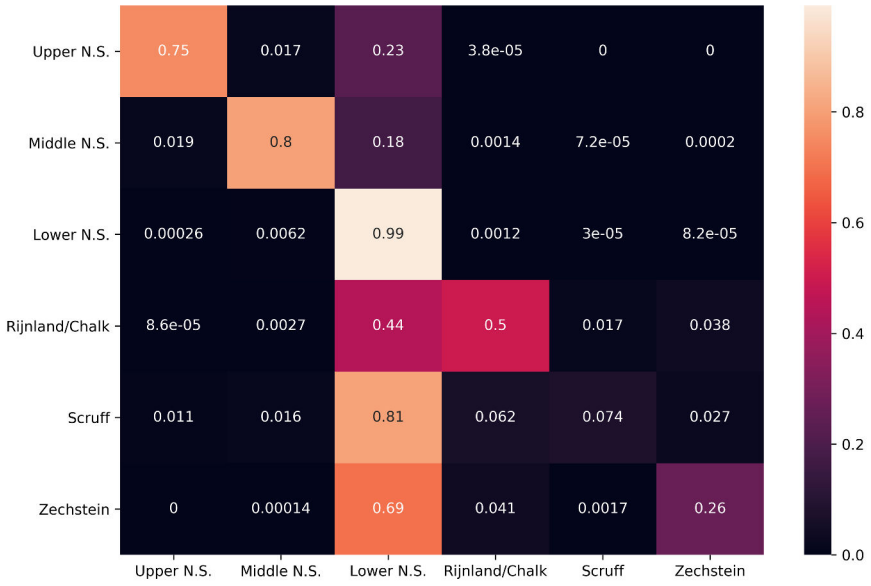
Since the patch-based models are trained on patches from different depths in the data, they can easily confuse various classes that typically exist at different depths. For example, the patch-based models in Figure 6.14 often confuse the Scruff group in the bottom left of the image with the Lower North Sea group, while the section-based models do not suffer from these problems as often. Figure 6.15 shows the confusion

matrices for the baseline patch and section models. It shows how the patch-based *baseline* model confuses many classes in our test sets with the Lower North Sea group. The *baseline* section-based model is better at classifying the other classes as Figure 6.15(b) shows. Additionally, since patch-based models are applied in a sliding window fashion, they typically perform slightly worse at the boundaries of the images where not many model outputs are averaged to create the final labeled section. This sliding window technique also makes the test-time performance of patch-based models extremely slow compared to section-based models.

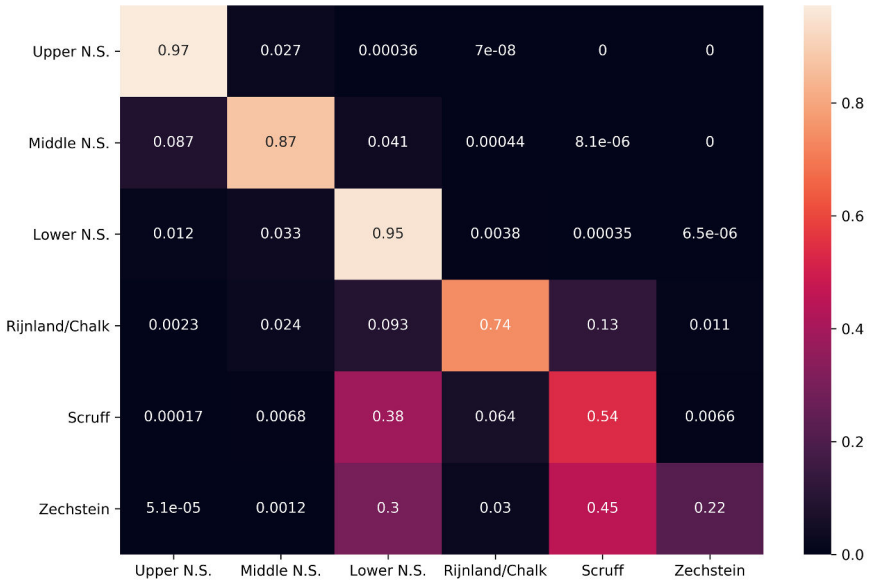
Table 6.4 shows that both patch and section-based models perform reasonably well on the North Sea groups, with the section-based models performing better. However, for smaller classes such as the Scruff and Zechstein groups, the section-based models show a clear advantage. The MCA score shows a 15% improvement of the section-based baseline model vs. the patch-based model. Overall, section-based models are superior to patch-based models due to their ability to incorporate spatial and contextual information within each seismic section. They also have the advantage of being faster to train and test. In our case, our label mapping algorithm is performed on patches, and therefore, all our weakly-supervised models are patch-based models.

Data augmentation and skip connections:

Data augmentation is a technique to artificially increase the size of the training set. This is quite useful when training a large network with a limited amount of training data. We use simple augmentation operations including randomly rotating the patch or the section by up to $\pm 15^\circ$, adding random Gaussian noise, and randomly flipping the patches or the sections horizontally. Using data augmentation significantly improved the results for both baseline models, but especially the patch-based model. The FWIU and MCA scores increased by more than 10% in the patch-based model, and significantly improved the results for smaller classes such as the Zechstein and



(a) Patch-based baseline model



(b) Section-based baseline model

Figure 6.15: Confusion matrices for our two fully-supervised *baseline* models on both test set #1 and #2. Each row shows the distribution of the model output for each class.

Scruff groups. The results of the section-based model were also enhanced by using data augmentation, although to a lesser degree.

For both patch- and section-based models, adding skip connections can improve the results, and speed up the training. This is especially noticeable in the patch-based model where adding skip connections to the *baseline + aug* model improved the results by about 1% in the PA metric and approximately 1.5% in the MCA and FWIU metrics. The improvement in the results of the section-based models is more subtle, as adding skip connections only improved the PA result by 0.1%. Interestingly, the Scruff group which is the worst performing class in both the patch and section-based model seemed to benefit the most from the addition of skip connections. The class accuracy score for the Scruff group increased by 6.5% and 25% in the patch and section based models respectively. Overall, adding skip connections can improve the results and speed up to the training process. In the case of the patch-based model, the skip connection model converged four times faster than the baseline.

6.6.2 Weak vs. Strong Supervision

Since we have concluded in the previous subsection that the *baseline + aug + skip* model seems to perform best for the patch-based model, we use this variant for all the models we train in this subsection. Here, we compare the results of two fully-supervised models that both use the CE loss. The first uses the entire 56,000 patches of size 75×75 for training. The next model uses the same 6000 training patches that our weakly-supervised models use, only it uses the ground truth labels from our dataset, and not the weakly-mapped labels. This second model would be a better model to compare our weakly-supervised results as they both use the same amount of data, and the only difference is the form of the supervision (strong/weak) and the loss function that is used. For our weakly-supervised models, we train three models. The first uses cross entropy loss and therefore does not use the confidence values in

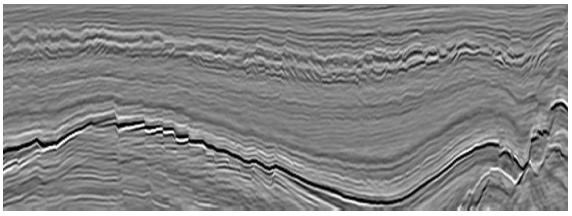
any way during the training process. This is a rather naïve approach, but we train this model to highlight the advantage of using our proposed WFL loss function. The second model uses our WFL loss function with $\gamma = 2$. Finally, the last model uses our similarity-weighted SW-WFL loss function with $\gamma = 2$ as well. Table 6.5 summarizes the objective results for both our strongly- and weakly-supervised models, and Figure 6.16 shows sample results from inline #200 in test set #1 for all the models listed in the table. Also, Figure 6.17 shows sample results from inline #400 in test set #2.

Using the smaller set of 6,000 training patches compared to the complete training set only reduced the performance of our fully-supervised model by 1% in the FWIU score and about 1.2 % in the PA score. This indicates that while more training data certainly helps improve the results, a smaller subset of the training data is sufficient to achieve very competitive results. Our baseline weakly-supervised model that uses cross entropy performs rather poorly with a pixel accuracy of 33.7%. Using the WFL loss helps significantly improve the results, reaching a PA score of almost 51%. Also, by using the similarity-weighted weak focal loss (SW-WFL), this result increases to nearly 53%. Figure 6.18 shows the confusion matrices for our fully-supervised model (using the same 6,000 patches) and the weakly-supervised model that uses the SW-WFL loss. We note that both models perform very poorly on the Scruff class and Zechstein classes, with the weakly-supervised model performing surprisingly better than the fully-supervised model in the Zechstein class. However, the weakly-supervised model commonly confuses neighboring classes, such as confusing the Rijnland/Chalk class with Zechstein and confusing the Upper North Sea class with the Middle North Sea class.

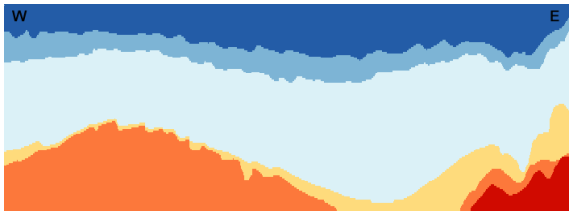
Figure 6.17 shows sample results of our weakly-supervised models compared the fully-supervised model that uses the same amount of training data. The results are shown for inline 400 from test split #2. Inline 400 is rather simple, lacking any pixels from the lower three classes in our model. We notice that the weakly-supervised

models perform rather well, but their poor localization performance results in smaller classes, such as the Middle North Sea class, being disproportionately large compared to the ground truth. This effect can also be observed clearly with the Middle North Sea and the Rijnland/Chalk classes in Figure 6.16 as well.

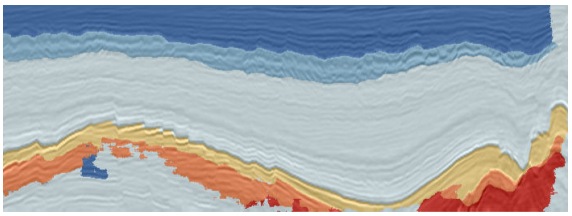
There are several ways where these weakly-supervised results can be improved; this includes using more exemplar images and retrieving more images for every exemplar. Also, using more data augmentation during training and testing, and adding regularization to the network to prevent it from overfitting to the weak labels can help. Finally, better exploitation of side information such as confidence and similarity values can potentially further improve the results, and a more robust label mapping algorithm can undoubtedly improve the localization accuracy of the resulting weakly-mapped labels.



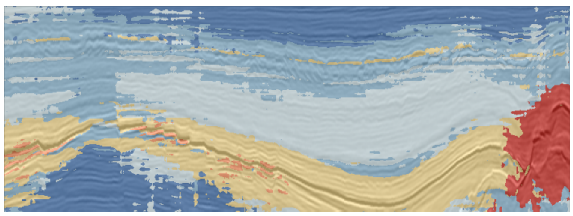
(a) Seismic data



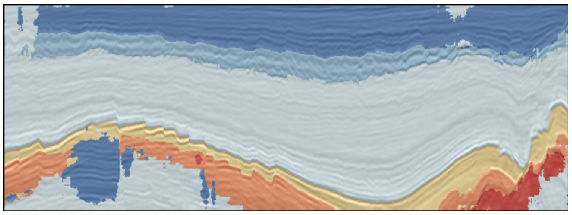
(b) Ground truth labels



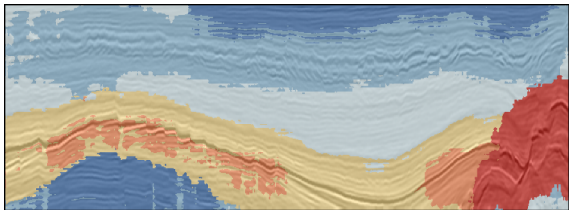
(c) CE (entire training set)



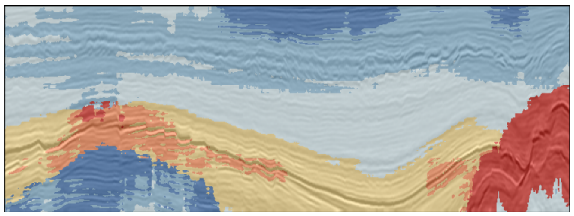
(d) CE



(e) CE (6,000 training patches)



(f) WFL with $\gamma = 2$

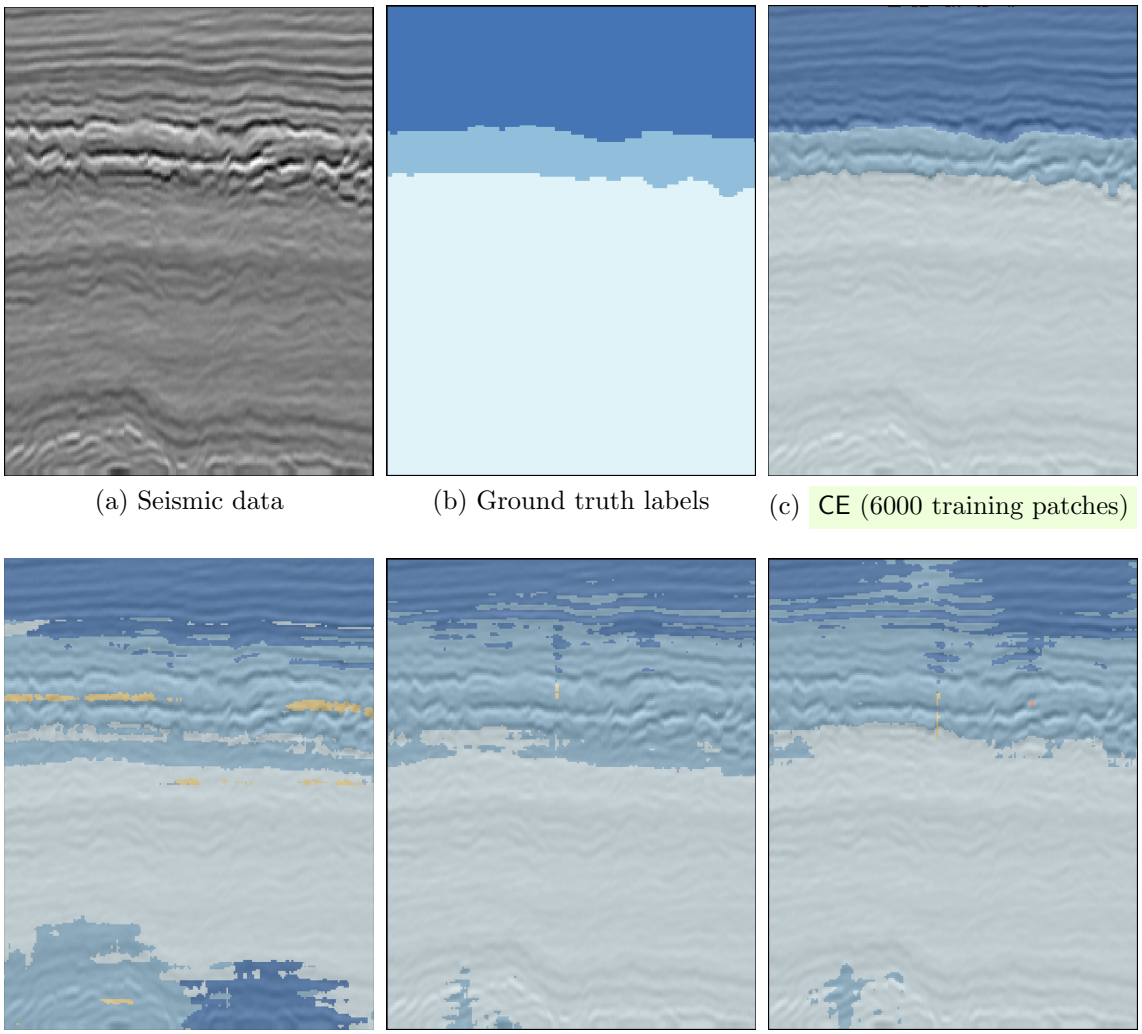


(g) SW-WFL with $\gamma = 2$

Figure 6.16: The results of the different weakly- and strongly- supervised models on inline 200 from test set #1. These results correspond to the models listed in Table 6.5 and the colors correspond to the colored classes in the same table.

Table 6.5: A comparison of various weakly- and strongly- supervised models when tested on both test splits of our dataset. All models listed in this table use the *patch-based + aug + skip* variant. All metrics are in the range $[0, 1]$, with larger values being better. The best performing model for every metric is highlighted in bold.

Model \ Metric	PA	Class Accuracy						MCA	FWIU
		Zechstein	Scruff	Rijnland/Chalk	Lower N. S.	Middle N. S.	Upper N. S.		
CE (entire training set)	**0.862**	**0.458**	0.286	0.673	**0.974**	**0.912**	**0.926**	**0.705**	**0.757**
CE (6,000 training patches)	0.850	0.222	**0.351**	**0.739**	0.959	0.853	0.923	0.675	0.747
CE	0.337	0.281	0.028	0.282	0.554	0.165	0.416	0.288	0.234
WFL with $\gamma = 2$	0.509	0.383	0.130	0.306	0.585	0.905	0.387	0.449	0.401
SW-WFL with $\gamma = 2$	0.529	0.356	0.111	0.379	0.667	0.860	0.294	0.445	0.416



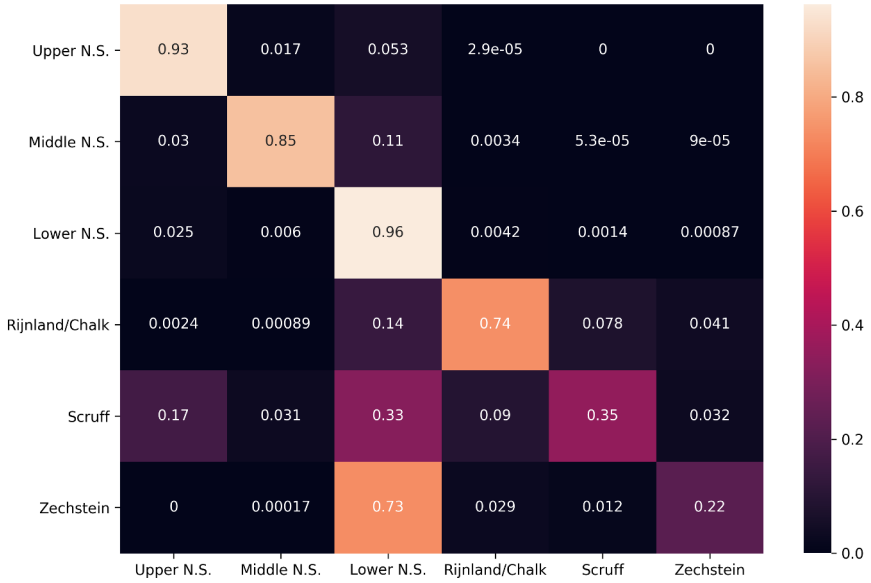
(a) Seismic data (b) Ground truth labels (c) CE (6000 training patches)

(d) CE (e) WFL with $\gamma = 2$ (f) SW-WFL with $\gamma = 2$

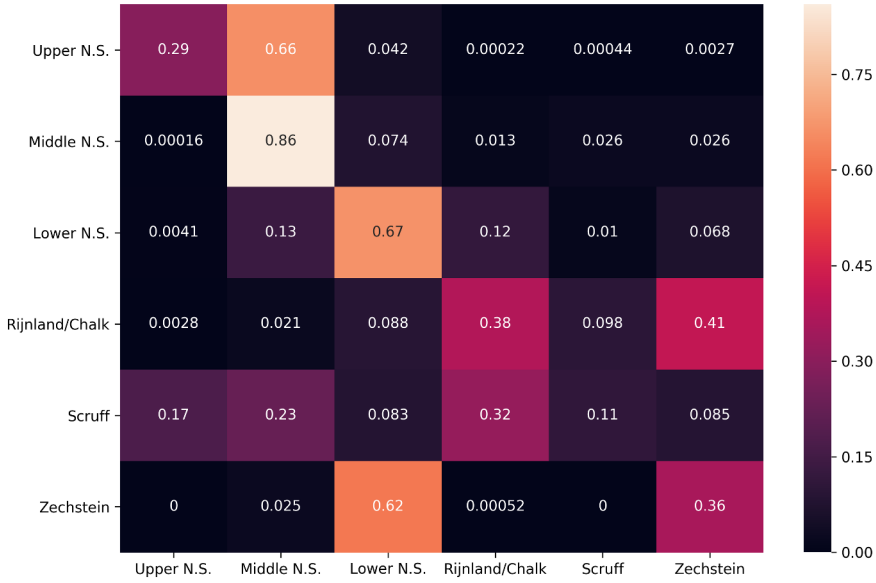
Figure 6.17: The results of the different weakly- and strongly- supervised models on inline 400 from test set #2. These results correspond to the models listed in Table 6.5 and the colors correspond to the colored classes in the same table.

6.7 Summary

In this chapter, we first introduce a new fully-annotated dataset for facies classification. This dataset includes six different lithostratigraphic classes based on the underlying geology of the Netherlands F3 block. The dataset also includes fault planes from three different generations that we have identified in the F3 block. In ad-



(a) CE (6,000 training patches)



(b) SW-WFL with $\gamma = 2$

Figure 6.18: Confusion matrices for a fully-supervised and a weakly-supervised model using the same architecture and trained on the same amount of data and tested on both test set #1 and #2. Each row shows the distribution of the model output for each class.

dition, this dataset was used to train two fully-supervised deep learning models—and a few variants that use data augmentation and skip connections—for facies classification. The two models are a patch- and a section-based model, both based on a deconvolution network architecture. We then compute objective results to evaluate the performance of these models on our test set.

Furthermore, we apply the weakly-supervised framework that was introduced in the previous chapters on this dataset. We use similarity-based retrieval to retrieve thousands of images based on their similarity to two exemplar images for each class of lithostratigraphic units. We assign these images image-level labels and then use our weakly-supervised label mapping algorithm to map the image-level labels to weak pixel-level labels. We then use our proposed weak focal loss (WFL), as well as a newly-introduced similarity-weighted weak focal loss (SW-WFL) to train our deep patch-based network on these weak labels.

We compute objective results for the accuracy of the label mapping, and we also contrast the performance of the fully-supervised models to the weakly-supervised models. We show that our weakly-supervised framework is not limited to seismic structural interpretation and that our weakly-supervised models can learn to semantically label seismic lithostratigraphic units, and can therefore be applied to stratigraphic seismic interpretation. However, the results of weakly-supervised models will always be less accurate than fully-supervised models, and more research is required in this domain to attempt to minimize the gap between fully- and weakly-supervised approaches. By releasing this dataset, we can start the journey of benchmarking and testing various strategies that can help to bring the results of weakly-supervised methods closer to fully-supervised ones.

CHAPTER 7
CONCLUSION

While deep learning has revolutionized the fields of machine learning and computer vision, the availability of annotated data to train deep neural networks is one of the main bottlenecks to the successful application of deep learning to many problem domains. This is especially true in the case of seismic interpretation where annotated data is very scarce, and where oil and gas exploration and production companies seldom share their data. Seismic interpretation is an excellent application domain where weakly-supervised learning can play a significant role in enabling state-of-the-art deep learning models to automate the most time-consuming and laborious interpretation tasks. In this dissertation, we have presented a weakly-supervised framework for the semantic labeling of large seismic volumes using state-of-the-art deep learning models. We have focused specifically on problems related to structural and stratigraphic interpretation as they heavily rely on the analysis of visual data (i.e., 3D seismic volumes) rather than well logs or core data.

Our weakly-supervised framework can be summarized in the following steps:

1. **Similarity-based retrieval:** An interpreter first selects images that exemplify the visual features of each class of interest. These can be seismic structures as in the case of structural interpretation, or seismic facies as in the case of stratigraphic interpretation. Based on the provided exemplar images, we use a state-of-the-art texture similarity measure that we have proposed to retrieve a large number of images that contain similar visual features. Within a certain similarity-threshold, these images can be assigned image-level labels. This process is detailed in Chapter 2.

2. **Weakly-supervised label mapping:** Given a large number of images with image-level labels, we developed a novel algorithm based on non-negative matrix factorization that maps these image-level labels to pixel-level labels that can encode the location of the target classes within each image. This label-mapping algorithm also produces confidence values for each of the predicted labels. We refer to these mapped pixel-level labels and their confidence values as "weak labels". The process of obtaining these weak labels is detailed in Chapter 4.

3. **Training deep CNNs using weak labels:** Given the weak labels from the previous step, we can train deep convolutional neural networks for structural or stratigraphic interpretation tasks. Training deep networks using noisy labels is challenging, and therefore we have proposed a few steps to make the training process more effective. This includes introducing a loss function specifically designed for weak labels. We call this loss function the weak focal loss (WFL). The WFL adjusts the training loss of the network to prevent it from putting too much trust in the weak labels. We have discussed this step in the context of seismic structural interpretation in Chapter 5, and in the context of stratigraphic interpretation in Chapter 6.

In addition to the main steps mentioned above, we have investigated the semantic labeling of seismic volumes using image-levels labels exclusively in Chapter 3. We have shown that while it is indeed possible to rely exclusively on image-level labels, the results are often lacking when compared to the results obtained when models are trained with weak pixel-level labels.

The results of our weakly-supervised framework show that incredibly, using as little as one or two exemplar images per class, we can generate a large dataset of images with weak pixel-level labels and use these labels effectively to train deep CNN models for various interpretation tasks. While the results will never be as accurate as fully-supervised models (as we have shown in Chapter 6), this is a very promising

approach to dramatically lower the costs associated with obtaining annotated data for application domains where such annotated data is not available publically or is expensive to obtain. In addition, given the ease of obtaining weak labels, we believe weakly-supervised learning approaches will become more mainstream in the future as more research is conducted in this field. We believe that the work we have presented here will allow for more advances in the field of weakly-supervised learning, and more specifically, in machine learning-enabled seismic interpretation.

7.1 Contributions

We can summarize the main contributions of this dissertation as follows:

1. Proposed a novel weakly-supervised framework for the semantic labeling of visual data using a limited number of exemplar images for every class. We have applied this framework on applications related to seismic structural and stratigraphic interpretation and showed promising results. In the case of stratigraphic interpretation, we have also compared the results of weakly-supervised and strongly-supervised models.

2. Proposed a state-of-the-art texture similarity measure. We showed that this measure is computationally efficient and significantly outperforms other measures in the literature in both retrieval and clustering accuracy, in addition to it being more consistent across different classes compared to other measures in the literature.

3. Proposed a method for semantic labeling of visual data using only image-level labels. In addition, we used this method to investigate various texture and multiresolution feature representations in their ability to accurately represent seismic data.

4. Proposed a novel weakly supervised label mapping algorithm, based on orthogonal non-negative matrix factorization. We demonstrated how this algorithm can learn common features within each class, and then use these features to map image-level labels to pixel-level labels. We have also shown how this algorithm computes confidence values in the mapped pixel-level labels. Additionally, we have shown how this algorithm is robust to noisy data, and we quantified the accuracy of this algorithm in mapping the labels of lithostratigraphic units.

5. Proposed a new loss function that allows deep networks to use the confidence values of weak labels to guide the training of the network. The weak focal loss allows us to put more weight on misclassified regions in the images and not trust the weak training labels as much, especially if the model is particularly confident in a particular classification. In addition, we have introduced a variant of this loss function that improves on it by weighing the loss for each image by its similarity to the exemplar image used to retrieve it.

6. Finally, we have prepared and released the single largest annotated dataset in the field of seismic stratigraphic interpretation. Our dataset is fully-annotated and is based on the careful study of the geology of the survey area, the seismic data, *and* various well logs within and around the seismic survey. We also established a standard benchmark for training and testing various models on this dataset, and made publically available all the codes to train and test models on this benchmark. In addition, we have also released two smaller datasets, Landmass-1 and Landmass-2, that contain various images of seismic structures that were used in our structural interpretation work.

7.2 Future Research Suggestions

Weakly-supervised methods for semantic segmentation, object detection, and other machine learning tasks will continue to be one of the most important areas of research in machine learning. Increasingly, the main limitation to the successful application of machine learning to many new application domains is the limited amount of annotated data. Given what we presented in this thesis, several promising research directions can help extend and improve the proposed weakly-supervised semantic labeling framework. These promising future research directions include the following:

- Throughout this thesis, we have mainly considered two-dimensional images; however, seismic data is 3D in nature. To increase the robustness of the overall framework, it is essential to exploit the 3D nature of seismic volumes. This involves using the 3D curvelet transform for the similarity-based retrieval, extracting 3D features and using them to perform the label mapping, and using 3D data to train CNN models for semantic segmentation.

- In Chapter 2, we have shown the excellent results of our similarity-based retrieval method. However, when we have more than one exemplar image for each class, the images are retrieved independently for each exemplar image. It would be promising to investigate methods to use all the exemplar images to jointly define the distribution of each class, and then retrieve images based on this class distribution, rather than independently for each exemplar image.

- The label mapping algorithm we presented in Chapter 4 can be made more robust by using a deep convolutional autoencoder (CAE). A CAE can learn a compact feature representation for an image without needing any annotations. By training a CAE to represent seismic images, it can be used to 'encode' thousands of seismic images that have image-level labels. We can then apply the label mapping algorithm we proposed directly on these encoded features.

Once the mapping is done, the results can be decoded to the pixel-space again to obtain pixel-level labels. There are many advantages to such an approach. First, this approach would be much more robust to variations in the training data including translations, rotations, and scale. Also, this technique would not be limited to seismic images and can be easily applied to other visual domains where such a label mapping would be useful.

- In Chapter 5, we have shown how using the weak focal loss can improve the training of a model with weak labels. However, the topic of learning with weak labels has not been explored fully in our work, nor in the literature at large. It is a promising research direction to investigate methods for learning using weak labels with their associated confidence values. One direction is using the confidence values to guide what the model focuses on during training. Another possible direction is to investigate how can the weak labels themselves (or the models that were trained using them) be improved using the associated confidence values in the weak labels (or the outputs of the models) in an iterative way. This can possibly be done through an expectation-maximization (EM) framework where a model would predict the weak labels, and then those weak labels are used to improve the model and so on until the system converges. These are only two suggested directions, but there is a lot to explore from both a theoretical and an application-oriented standpoint.

- Finally, our weakly-supervised semantic labeling framework was focused on seismic interpretation. However, our framework is not limited to this application and can be extended to similar applications that do not have sufficient annotated data. These applications include biomedical imaging and remote sensing where there is a massive amount of raw data, but limited annotations available.

CHAPTER 8
THESIS PRODUCTS

Below is a summary of the some of the products that resulted from this thesis. Many of the products including the codes and preprints are publically available[1].

8.1 Invention Disclosures

1. G. AlRegib and **Y. Alaudah**,"A method for transforming image-level labels to pixel-level labels," invention disclosure (GTRC ID: 7837) filed with Georgia Tech in February 2018.

2. G. AlRegib, Z. Wang, A. Girdhar, H. Di, M. Alfarraj, and **Y. Alaudah**,"CLoud Visual ExploreR (CLeVER): A Cloud-based Interpretation Platform for Rapid Seismic Data Analysis," invention disclosure filed with Georgia Tech in February 2018.

8.2 Magazine Articles

1. G. AlRegib, M. Deriche, Z. Long, H. Di, Z. Wang, **Y. Alaudah**, M. A. Shafiq, M. Alfarraj, "Subsurface Structure Analysis using Computational Interpretation and Learning", *IEEE Signal Processing Magazine*, March 2018.

8.3 Datasets

1. LANDMASS-1 and LANDMASS-2: `https://ghassanalregib.com/landmass/`

2. Facies Classification Benchmark: `https://github.com/olivesgatech/facies_classification_benchmark`

[1] www.ghassanalregib.com/publications

8.4 Journal Articles

1. **Y. Alaudah**, P. Michłowicz, M. Alfarraj, G. AlRegib "A Machine Learning Benchmark for Facies Classification," *Interpretation*, submitted Dec. 2018.

2. Z. Wang, H. Di, M. A. Shafiq, **Y. Alaudah**, and G. AlRegib, "Successful Leveraging of Image Processing and Machine Learning in Seismic Structural Interpretation: A Review", The Leading Edge, 37(6), 451-461.

3. **Y. Alaudah**, M. Alfarraj, and G. AlRegib, "Structure Label Prediction Using Similarity-Based Retrieval and Weakly-Supervised Label Mapping", *Geophysics*, August 2018.

4. M. Alfarraj, **Y. Alaudah**, Z. Long, and G. AlRegib, "Multiresolution Analysis and Learning for Computational Seismic Interpretation," *The Leading Edge*, February 2018.

5. Z. Long, **Y. Alaudah**, M. A. Qureshi, Y. Hu, Z. Wang, M. Alfarraj, G. AlRegib, A. Amin, M. Deriche, and S. Al-Dharrab, "A comparative study of texture attributes for characterizing subsurface structures in migrated seismic volumes," *Interpretation*, 2017.

8.5 Conference Papers

1. **Y. Alaudah**, M. Soliman, and G. AlRegib, "Facies Classification with Weak and Strong Supervision – A Comparative Study", *submitted to SEG 89th Annual Meeting*, San Antonio, Texas, Sep. 15-20, 2018.

2. **Y. Alaudah** and G. AlRegib, "Weakly-Supervised Subsurface Structure Labeling," *SBGf/SEG Machine Learning Workshop*, Rio De Janeiro, Brazil, May 2018.

3. **Y. Alaudah**, S. Gao, and G. AlRegib, "Learning to label seismic structures with deconvolution networks and weak labels", *SEG 88th Annual Meeting*, Anaheim, California, Oct. 14-19, 2018.

4. **Y. Alaudah** and G. AlRegib, "A weakly-supervised approach to seismic structure labeling," *87th Annual SEG Meeting Extended Abstracts*, Houston, Texas, 2017.

5. **Y. Alaudah** and G. AlRegib, "A directional coherence attribute for seismic interpretation," *87th Annual SEG Meeting Extended Abstracts*, Houston, Texas, 2017.

6. **Y. Alaudah**, H. Di, and G. AlRegib, "Weakly Supervised Seismic Structure Labeling via Orthogonal Non-Negative Matrix Factorization", *European Association of Geoscientists & Engineers, 79th Annual Conference & Exhibition (EAGE)*, Paris, France, June 12-15, 2017.

7. M. Shafiq, **Y. Alaudah**, and G. AlRegib, "Salt dome delineation using edge- and texture-based attributes,", *European Association of Geoscientists & Engineers, 79th Annual Conference & Exhibition (EAGE)*, Paris, France, June 12-15, 2017.

8. M. Shafiq, H. Di, **Y. Alaudah**, and G. AlRegib, "Interpreter-Assisted Interactive Delineation of Salt Domes using Phase Congruency and Gradient of Texture Attributes," *American Association of Petroleum Geologists, Annual Convention and Exhibition (ACE)*, 2-5 April, 2017.

9. M. Shafiq, **Y. Alaudah**, G. AlRegib, and M. Deriche, "Phase Congruency for Image Understanding with Applications in Computational Seismic Interpretation," *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, New Orleans, USA, March 5-9, 2017

10. **Y. Alaudah** and G. AlRegib, "Weakly-Supervised Labeling of Seismic Volumes Using Reference Exemplars," *IEEE Intl. Conference on Image Processing (ICIP)*, Phoenix, Arizona, USA, Sep. 25-28, 2016.

11. M. Alfarraj, **Y. Alaudah**, and G. AlRegib, "Content-adaptive Non-parametric Texture Similarity Measure," *IEEE workshop on Multimedia Signal Processing (MMSP)*, Montreal, Canada, Sep. 21-23, 2016.

12. M. Shafiq, **Y. Alaudah**, and G. AlRegib, "A hybrid approach for salt dome delineation within migrated seismic volumes," *78th EAGE Annual Conference & Exhibition*, Vienna, Austria, May 30-June 2, 2016.

13. **Y. Alaudah** and G. AlRegib, "A Generalized Tensor-Based Coherence Attribute," *European Association of Geoscientists & Engineers, 78th Annual Conference & Exhibition (EAGE)*, Vienna, Austria, May 30-June 2, 2016.

14. **Y. Alaudah**, M. Shafiq, and G. AlRegib, "A Hybrid Spatio-Frequency Approach for Delineating Subsurface Structures in Seismic Volumes," *SIAM conference on Imaging Science*, New Mexico, USA, May 23-26, 2016.

15. Z. Long, **Y. Alaudah**, M. Qureshi, M. Farraj, Z. Wang, A. Amin, M. Deriche, and G. AlRegib, "Characterization of migrated seismic volumes using texture attributes: a comparative study," *SEG Annual Meeting*, New Orleans, Louisiana, Oct. 18-23, 2015.

16. **Y. Alaudah** and G. AlRegib, "A Curvelet-Based Distance Measure for Seismic Images," *IEEE Intl. Conf. on Image Processing (ICIP)*, Quebec City, Canada, Sept. 27-30, 2015.

Appendices

APPENDIX A
EVALUATION METRICS

A.1 Retrieval

The performance of a similarity measure is quantified using information retrieval metrics. To present these metrics, let us first define the following sets:

- $\mathcal{R}_i^{(j)} = \left\{\mathbf{r}_i^{(1)}, \mathbf{r}_i^{(2)}, \ldots, \mathbf{r}_i^{(j)}\right\}$ is the set of the first j retrieved images for $\mathbf{x}_i$. Note that the elements of $\mathcal{R}_i^{(j)}$ are sorted according to their similarity to $\mathbf{x}_i$ such that $\mathsf{Similarity}\left(\mathbf{x}_i, \mathbf{r}_i^{(k)}\right) \geqslant \mathsf{Similarity}\left(\mathbf{x}_i, \mathbf{r}_i^{(k+1)}\right)$.

- $\mathcal{C}_i$ is the set of all images that are of the same class as $\mathbf{x}_i$; excluding $\mathbf{x}_i$ itself.

- $\mathcal{R}_i^{(j)} \cap \mathcal{C}_i$ is the intersection set of $\mathcal{R}_i^{(j)}$ and $\mathcal{C}_i$. It contains images that are of the same class as the query image $\mathbf{x}_i$ in the set of retrieved images $\mathcal{R}_i^{(j)}$.

Next, we define information retrieval metrics that were used to assess the performance of the similarity measures.

- **Precision at M (P@M)** is the average percentage of the correctly retrieved images when M images are retrieved. Formally,

$$\mathrm{P@}M = \frac{1}{N_s} \sum_{i=1}^{N_s} \frac{\left|\mathcal{R}_i^{(M)} \cap \mathcal{C}_i\right|}{\left|\mathcal{R}_i^{(M)}\right|}, \tag{A.1}$$

  where $|\cdot|$ is the number of elements in the set.

- **Retrieval Accuracy (RA)** is the P@M when M is equal to the number of

elements that are of the same class the query images, i.e. $M = |\mathcal{C}_i|$.

$$\text{RA} = \frac{1}{N_s} \sum_{i=1}^{N_s} \frac{\left| \mathcal{R}_i^{(|\mathcal{C}_i|)} \cap \mathcal{C}_i \right|}{\left| \mathcal{R}_i^{(|\mathcal{C}_i|)} \right|}. \tag{A.2}$$

- **Average Precision (AP)** for query image $\mathbf{x}_i$ is a measure of precision that takes into account the order of which the correct images are retrieved. It is defined as:

$$\text{AP}_i = \frac{1}{|\mathcal{C}_i|} \sum_{j=1}^{N_s-1} \frac{\left| \mathcal{R}_i^{(j)} \cap \mathcal{C}_i \right|}{\left| \mathcal{R}_i^{(j)} \right|} \times \mathbf{1}_{\{r_i^{(j)} \in \mathcal{C}_i\}}, \tag{A.3}$$

  where $\mathbf{1}_{\{r_i^{(j)} \in \mathcal{C}_i\}}$ is the indicator function and it is equal to 1 if and only if $r_i^{(j)} \in \mathcal{C}_i$, and 0 otherwise. **Mean Average Precision (MAP)** is the mean value of AP for all images in the dataset.

- **Receiver Operating Characteristics (ROC)** is a plot of the true positive rate (TPR) versus False Positive Rate (FPR) for different similarity thresholds. TPR is the percentage of pairs of images that are correctly identified as similar by the similarity measure. FPR is the percentage of pairs of images that are not similar but were identified as similar by the similarity measure. The area under the ROC curve, denoted as **AUC**, is used as a measure of the discriminative power of a similarity measure. The ideal ROC curve would have perfect TPR (TPR=1) for all values of FPR, and in this case, the area under the curve would be maximum AUC = 1.

All of the metrics defined above are in the range $[0, 1]$ with 1 being a perfect score.

A.2 Clustering

We use the **rand index** to evaluate our performance in the clustering experiments. The rand index is defined as follows. For each pair of images, $\mathbf{x}_i$ and $\mathbf{x}_j$, in the dataset,

we compare the results obtained by k-means clustering with the ground truth which are the image labels. Then we count the number of correctly clustered pairs. A pair is said to be correctly clustered if:

- $\mathbf{x}_i$ and $\mathbf{x}_j$ are of the same class and are in the same cluster in the similarity-based clustering.
- $\mathbf{x}_i$ and $\mathbf{x}_j$ are of different classes and are in different clusters in the similarity-based clustering.

If p_{correct} is the total number of correctly clustered pairs and $p_{\text{total}} = \binom{N_s}{2}$ is the total number of possible pairs in the dataset, The Rand Index is defined as the ratio of the two numbers,

$$\textbf{Rand Index} = \frac{p_{\text{correct}}}{p_{\text{all}}} = \frac{2p_{\text{correct}}}{N_s(N_s-1)}. \tag{A.4}$$

The rand index results in values in the range $[0, 1]$ with 1 being a perfect score.

A.3 Semantic Segmentation

If we denote $\mathcal{G}_i$ as the set of pixels manually labeled as i (i.e. belonging to the ith class), $\mathcal{F}_i$ as the set of pixels classified by our classifier as i, and N_ℓ as the number of classes, then the set of correctly classified pixels is the intersection set $\mathcal{F}_i \cap \mathcal{G}_i$. If we use $|\cdot|$ to denote the number of elements in a set, then we can define the following metrics:

- **Pixel Accuracy (PA)** is the percentage of pixels over all classes that are correctly classified,

$$\text{PA} = \frac{\sum_i |\mathcal{F}_i \cap \mathcal{G}_i|}{\sum_i |\mathcal{G}_i|}. \tag{A.5}$$

- **Class Accuracy for class y_i ($\mathbf{CA}_i$)** is the percentage of pixels that are correctly classified in a class y_i.

$$\mathrm{CA}_i = \frac{|\mathcal{F}_i \cap \mathcal{G}_i|}{|\mathcal{G}_i|}. \tag{A.6}$$

We can also define the Mean Class Accuracy (MCA) as the average of CA over all classes,

$$\mathrm{MCA} = \frac{1}{N_\ell} \sum_i \mathrm{CA}_i = \frac{1}{N_\ell} \sum_i \frac{|\mathcal{F}_i \cap \mathcal{G}_i|}{|\mathcal{G}_i|} \tag{A.7}$$

- **Intersection over Union ($\mathbf{IU}_i$)** is defined as the number of elements of the intersection of $\mathcal{G}_i$ and $\mathcal{F}_i$ over the number of elements of their union set,

$$\mathrm{IU}_i = \frac{|\mathcal{F}_i \cap \mathcal{G}_i|}{|\mathcal{F}_i \cup \mathcal{G}_i|}. \tag{A.8}$$

This metric measures the overlap between the two sets and it should be 1 if and only if all pixels were correctly classified. Averaging the IU over all classes results in the **Mean Intersection over Union (MIU)** metric:

$$\mathrm{MIU} = \frac{1}{N_\ell} \sum_i \mathrm{IU}_i = \frac{1}{N_\ell} \sum_i \frac{|\mathcal{F}_i \cap \mathcal{G}_i|}{|\mathcal{F}_i \cup \mathcal{G}_i|}. \tag{A.9}$$

- **Frequency-Weighted Intersection over Union (FWIU)** is a weighted average of IU over all classes such that classes with higher frequency are given more weight:

$$\mathrm{FWIU} = \frac{1}{\sum_i |\mathcal{G}_i|} \cdot \sum_i |\mathcal{G}_i| \cdot \mathrm{IU}_i = \frac{1}{\sum_i |\mathcal{G}_i|} \cdot \sum_i |\mathcal{G}_i| \cdot \frac{|\mathcal{F}_i \cup \mathcal{G}_i|}{|\mathcal{F}_i \cap \mathcal{G}_i|} \tag{A.10}$$

.

APPENDIX B

DERIVATION OF MULTIPLICATIVE UPDATE RULES

We would like to derive the multiplicative update rules shown in equations 4.7 and 4.8. These multiplicative update rules solve the optimization problems introduced in equations 4.5 and 4.6. To derive the multiplicative update rules, we adopt an approach similar to that proposed by [141]. We will derive the gradient descent updates to solve the problem for $\mathbf{W}$ and $\mathbf{H}$ separately. Then, we solve the problem in equation 4.4 by alternately updating $\mathbf{W}$ and $\mathbf{H}$ successively until they converge. We will derive the multiplicative update rules using the objective functions in equations 4.5 and 4.6. For the sake of the simplicity of the derivation, we drop all the constraints in equations 4.5 and 4.6, and later show that the derived multiplicative update rules for the non-constrained problem also solve the constrained optimization problem under the conditions that we have. Additionally, we have shown in Figure 4.4 that solving these two problems iteratively also solves the problem in equation 4.4. Therefore, for matrix $\mathbf{W}$ we have

$$\arg\min_{\mathbf{W}} ||\mathbf{X} - \mathbf{W}\mathbf{H}||_F^2 + \lambda_1 ||\mathbf{W}||_F^2, \tag{B.1}$$

and for $\mathbf{H}$, we have

$$\arg\min_{\mathbf{H}} ||\mathbf{X} - \mathbf{W}\mathbf{H}||_F^2 + \gamma_1 ||\mathbf{H}\mathbf{H}^T - \mathbf{B}||_F^2 + \lambda_2 ||\mathbf{H}||_F^2. \tag{B.2}$$

We derive the multiplicative update rules for $\mathbf{W}$ and $\mathbf{H}$ respectively in the following two subsections.

B.1 Multiplicative Update Rule for W

If we denote the objective function defined in B.1 as $\mathcal{F}_{\mathbf{W}}$, we can rewrite $\mathcal{F}_{\mathbf{W}}$ as

$$\mathcal{F}_{\mathbf{W}} = \mathrm{Tr}\big((\mathbf{X} - \mathbf{WH})^T(\mathbf{X} - \mathbf{WH})\big) + \lambda_1 \mathrm{Tr}(\mathbf{W}^T\mathbf{W}), \tag{B.3}$$

where $\mathrm{Tr}(\cdot)$ denotes the trace of a matrix. This is a well known property of the Frobenius norm. Simplifying the expression further, and employing the property that $\mathrm{Tr}(\mathbf{A} + \mathbf{B}) = \mathrm{Tr}(\mathbf{A}) + \mathrm{Tr}(\mathbf{B})$, we obtain

$$\begin{aligned} \mathcal{F}_{\mathbf{W}} = \mathrm{Tr}(\mathbf{X}^T\mathbf{X}) - \mathrm{Tr}(\mathbf{X}^T\mathbf{WH}) - \mathrm{Tr}(\mathbf{H}^T\mathbf{W}^T\mathbf{X}) \\ + \mathrm{Tr}(\mathbf{H}^T\mathbf{W}^T\mathbf{WH}) + \lambda_1 \mathrm{Tr}(\mathbf{W}^T\mathbf{W}). \end{aligned} \tag{B.4}$$

Taking the partial derivative of $\mathcal{F}_{\mathbf{W}}$ with respect to $\mathbf{W}$ we get

$$\begin{aligned} \frac{\partial \mathcal{F}_{\mathbf{W}}}{\partial \mathbf{W}} &= -2(\mathbf{XH}^T) + 2(\mathbf{WHH}^T) + 2\lambda_1\mathbf{W} \\ &\propto \ -\mathbf{XH}^T + \mathbf{WHH}^T + \lambda_1\mathbf{W} \end{aligned} \tag{B.5}$$

The gradient descent update for $\mathbf{W}$ will then be a step in the direction of the negative gradient. In other words,

$$\mathbf{W}^{t+1} = \mathbf{W}^t + \eta\big(\mathbf{XH}^{t^T} - \mathbf{W}^t\mathbf{H}^t\mathbf{H}^{t^T} - \lambda_1\mathbf{W}^t\big), \tag{B.6}$$

where η is the step size. Note that this is an additive update rule. The negative signs indicate that even if the values in $\mathbf{X}$, $\mathbf{W}^0$ and $\mathbf{H}^0$ are non-negative, we are not guaranteed to arrive at a non-negative final solution. However, by selecting our step size as

$$\eta = \frac{\mathbf{W}^t}{\mathbf{W}^t\mathbf{H}^t\mathbf{H}^{t^T} + \lambda_1\mathbf{W}^t}, \tag{B.7}$$

and substituting in the gradient descent update in equation B.6, the additive rule

becomes a multiplicative update rule:

$$\mathbf{W}^{t+1} = \mathbf{W}^t \odot \frac{\big(\mathbf{X}\mathbf{H}^{t^T} + \epsilon\big)_{ij}}{\big(\mathbf{W}^t\mathbf{H}^t\mathbf{H}^{t^T} + \lambda_1\mathbf{W}^t + \epsilon\big)_{ij}}. \tag{B.8}$$

We add a small positive real number ϵ to avoid division by zero. This result is identical to the result in equation 4.7.

B.2 Multiplicative Update Rule for H

Similarly for $\mathbf{H}$, we write the objective function in equation B.2 as

$$\begin{aligned}\mathcal{F}_{\mathbf{H}} = \mathrm{Tr}\big((\mathbf{X} - \mathbf{W}\mathbf{H})^T(\mathbf{X} - \mathbf{W}\mathbf{H})\big) + \lambda_2\mathrm{Tr}(\mathbf{H}^T\mathbf{H}) \\ + \gamma_1\mathrm{Tr}\big((\mathbf{H}\mathbf{H}^T - \mathbf{B})^T(\mathbf{H}\mathbf{H}^T - \mathbf{B})\big).\end{aligned} \tag{B.9}$$

Taking the partial derivative of $\mathcal{F}_{\mathbf{H}}$ with respect to $\mathbf{H}$, and simplifying the expression further,

$$\begin{aligned}\frac{\partial\mathcal{F}_{\mathbf{H}}}{\partial\mathbf{H}} &= -2\mathbf{W}^T\mathbf{X} + 2(\mathbf{W}^T\mathbf{W}\mathbf{H}) + 2\lambda_2\mathbf{H} \\ &\quad + 4\gamma_1\mathbf{H}^T\mathbf{H}\mathbf{H}^T - 2\gamma_1(\mathbf{B} + \mathbf{B}^T)\mathbf{H} \\ &\propto (\mathbf{W}^T\mathbf{W}\mathbf{H}) + \lambda_2\mathbf{H} + \gamma_1\mathbf{H}^T\mathbf{H}\mathbf{H}^T \\ &\quad - \big(\mathbf{W}^T\mathbf{X} + \gamma_1(\mathbf{B} + \mathbf{B}^T)\mathbf{H}\big).\end{aligned} \tag{B.10}$$

The gradient descent update step then becomes

$$\begin{aligned}\mathbf{H}^{t+1} = \mathbf{H}^t + \eta\Big(\mathbf{W}^{t+1^T}\mathbf{X} + \gamma_1(\mathbf{B} + \mathbf{B}^T)\mathbf{H}^t \\ - (\mathbf{W}^{t+1^T}\mathbf{W}^{t+1}\mathbf{H}^t + \lambda_2\mathbf{H}^t + \gamma_1\mathbf{H}^{t^T}\mathbf{H}^t\mathbf{H}^{t^T})\Big).\end{aligned} \tag{B.11}$$

If we select the step size to be

$$\eta = \frac{\mathbf{H}^t}{\mathbf{W}^{\mathbf{t+1}^T}\mathbf{W}^{t+1}\mathbf{H}^t + \lambda_2\mathbf{H}^t + \gamma_1\mathbf{H}^{t^T}\mathbf{H}^t\mathbf{H}^{t^T}}, \tag{B.12}$$

and substitute this value in equation B.11 and simplify, we arrive at the multiplicative update rule for $\mathbf{H}$

$$\mathbf{H}^{t+1} = \frac{\mathbf{H}^t \odot \left(\mathbf{W}^{t+1^T}\mathbf{X} + \gamma_1(\mathbf{B} + \mathbf{B}^T)\mathbf{H}^t + \epsilon\right)_{ij}}{\left(\mathbf{W}^{t+1^T}\mathbf{W}^{t+1}\mathbf{H}^t + \lambda_2\mathbf{H}^t + \gamma_1\mathbf{H}^{t^T}\mathbf{H}^t\mathbf{H}^{t^T} + \epsilon\right)_{ij}} \tag{B.13}$$

This is identical to the update rule shown in equation 4.8.

B.3 Constrained Optimization

The update rules shown in equations B.8 and B.13 solve the non-constrained problems in equation B.1 and B.2. However, our original problem in equation 4.4 is a constrained one. Since we initialize the matrices $\mathbf{W}$ and $\mathbf{H}$ with non-negative values, it is trivial to see that the multiplicative update rules in equations B.8 and B.13 will always give non-negative results, thus satisfying the non-negativity constraint. Furthermore, since the sparsity constraint on the features $\mathbf{w}_i$ is applied to the initial features, $\mathbf{W}^0$, any further application of the update rule in equation B.8 will not modify the zero elements in the matrix $\mathbf{W}$, and hence, the initial feature sparsity is preserved. Therefore, although we solved for the non-constrained problem in equations B.1 and B.2, our solution is still valid for the constrained problem in equation 4.4.

REFERENCES

[1] K. Boman, *Big data growth continues in seismic surveys*, 2015.

[2] Ö. Yilmaz, *Seismic data analysis*. Society of Exploration Geophysicists Tulsa, 2001, vol. 1.

[3] dGB Earth Sciences, "The Netherlands offshore, the north sea, F3 block - complete," 1987.

[4] G. AlRegib, M. Deriche, Z. Long, H. Di, Z. Wang, Y. Alaudah, M. A. Shafiq, and M. Alfarraj, "Subsurface structure analysis using computational interpretation and learning: A visual signal processing perspective," *IEEE Signal Processing Magazine*, vol. 35, no. 2, pp. 82–98, Mar. 2018.

[5] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, *et al.*, "Imagenet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.

[6] M. Guillaumin, D. Küttel, and V. Ferrari, "Imagenet auto-annotation with segmentation propagation," *International Journal of Computer Vision*, vol. 110, no. 3, pp. 328–348, 2014.

[7] J. Xu, A. G. Schwing, and R. Urtasun, "Learning to segment under various forms of weak supervision," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3781–3790.

[8] N. K. Lioudis, *How do average costs compare among various oil drilling rigs?* 2018.

[9] M. Everingham, S. A. Eslami, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The Pascal visual object classes challenge: A retrospective," *International journal of computer vision*, vol. 111, no. 1, pp. 98–136, 2015.

[10] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 3213–3223.

[11] L. Torresani, "Weakly supervised learning," in *Computer Vision: A Reference Guide*, K. Ikeuchi, Ed. Boston, MA: Springer US, 2014, pp. 883–885, ISBN: 978-0-387-31439-6.

[12] W. Zhou, H. Li, and Q. Tian, "Recent advance in content-based image retrieval: A literature survey," *arXiv preprint arXiv:1706.06064*, 2017.

[13] E. Candes, L. Demanet, D. Donoho, and L. Ying, "Fast discrete curvelet transforms," *Multiscale Modelling and Simulations*, vol. 5, no. 3, pp. 861–899, 2005.

[14] Z. Wang and A. C. Bovik, "Mean squared error: Love it or leave it? a new look at signal fidelity measures," *IEEE Signal Processing Magazine*, vol. 26, no. 1, pp. 98–117, Jan. 2009.

[15] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *Image Processing, IEEE Transactions on*, vol. 13, no. 4, pp. 600–612, 2004.

[16] Z. Wang and E. P. Simoncelli, "Translation insensitive image similarity in complex wavelet domain," in *In Acoustics, Speech, and Signal Processing, 2005. Proceedings.(ICASSP'05). IEEE International Conference on*, Citeseer, 2005.

[17] E. P. Simoncelli and W. T. Freeman, "The steerable pyramid: A flexible architecture for multi-scale derivative computation," in *Image Processing, 1995. Proceedings., International Conference on*, IEEE, vol. 3, 1995, pp. 444–447.

[18] X. Zhao, M. G. Reyes, T. N. Pappas, and D. L. Neuhoff, "Structural texture similarity metrics for retrieval applications," in *Image Processing, 2008. ICIP 2008. 15th IEEE International Conference on*, IEEE, 2008, pp. 1196–1199.

[19] J. Zujovic, T. Pappas, and D. Neuhoff, "Structural texture similarity metrics for image analysis and retrieval," *Image Processing, IEEE Transactions on*, vol. 22, no. 7, pp. 2545–2558, Jul. 2013.

[20] E. J. Candes and D. L. Donoho, "New tight frames of curvelets and optimal representations of objects with piecewise c2 singularities," *Communications on Pure and Applied Mathematics*, vol. 57, no. 2, pp. 219–266, 2004.

[21] D. Zhang, M. M. Islam, G. Lu, and I. J. Sumana, "Rotation invariant curvelet features for region based image retrieval," *International journal of computer vision*, vol. 98, no. 2, pp. 187–201, 2012.

[22] S Arivazhagan, L Ganesan, and T. S. Kumar, "Texture classification using curvelet statistical and co-occurrence features," in *Pattern Recognition, 2006.*

ICPR 2006. 18th International Conference on, IEEE, vol. 2, 2006, pp. 938–941.

[23] F Gómez and E Romero, "Rotation invariant texture characterization using a curvelet based descriptor," *Pattern Recognition Letters*, vol. 32, no. 16, pp. 2178–2186, 2011.

[24] M. N. Do and M. Vetterli, "Texture similarity measurement using kullback-leibler distance on wavelet subbands," in *Image Processing, 2000. Proceedings. 2000 International Conference on*, IEEE, vol. 3, 2000, pp. 730–733.

[25] S. Selvan and S. Ramakrishnan, "SVD-based modeling for image texture classification using wavelet transformation," *Image Processing, IEEE Transactions on*, vol. 16, no. 11, pp. 2688–2696, 2007.

[26] H. Al-Marzouqi and G. AlRegib, "Similarity index for seismic data sets using adaptive curvelets," in *SEG Technical Program Expanded Abstracts 2014*. 2014, pp. 1470–1474. eprint: `http://library.seg.org/doi/pdf/10.1190/segam2014-1228.1`.

[27] ——, "Curvelet transform with learning-based tiling," *Signal Processing: Image Communication*, vol. 53, pp. 24–39, 2017.

[28] H. Al-Marzouqi and G. AlRegib, "Using the coefficient of variation to improve the sparsity of seismic data," in *Global Conference on Signal and Information Processing (GlobalSIP), 2013 IEEE*, Dec. 2013, pp. 630–630.

[29] Z Long, Z Wang, and G AlRegib, "Seisim: Structural similarity evaluation for seismic data retrieval: Presented at the 2015 ieee iccspa," IEEE, 2015.

[30] K. J. Marfurt, V Sudhaker, A. Gersztenkorn, K. D. Crawford, and S. E. Nissen, "Coherency calculations in the presence of structural dip," *Geophysics*, vol. 64, no. 1, pp. 104–111, 1999.

[31] E. J. Candès and D. L. Donoho, "Curvelets - A surprisingly effective nonadaptive representation for objects with edges," *Curves and Surfaces*, vol. C, no. 2, pp. 1–10, 2000.

[32] E. J. Candès and D. L. Donoho, "New tight frames of curvelets and optimal representations of objects with piecewise c2 singularities," *Communications on pure and applied mathematics*, vol. 57, no. 2, pp. 219–266, 2004.

[33] J. Fadili and J.-L. Starck, "Curvelets and ridgelets," in *Encyclopedia of Complexity and Systems Science*, Springer, 2009, pp. 1718–1738.

[34] G. Hennenfent and F. J. Herrmann, "Seismic denoising with nonuniformly sampled curvelets," *Computing in Science & Engineering*, vol. 8, no. 3, pp. 16–25, 2006.

[35] F. J. Herrmann, D. Wang, and D. J. Verschuur, "Adaptive curvelet-domain primary-multiple separation," *Geophysics*, vol. 73, no. 3, A17–A21, 2008.

[36] H. Chauris and T. Nguyen, "Seismic demigration/migration in the curvelet domain," *Geophysics*, vol. 73, no. 2, S35–S46, 2008.

[37] E. Candès, L. Demanet, D. Donoho, and L. Ying, "Fast Discrete Curvelet Transforms," *Multiscale Modeling & Simulation*, vol. 5, no. 3, pp. 861–899, Jan. 2006.

[38] Y. Alaudah and G. AlRegib, "A curvelet-based distance measure for seismic images," in *Image Processing (ICIP), 2015 IEEE International Conference on*, Sep. 2015, pp. 4200–4204.

[39] S.-H. Cha, "Comprehensive survey on distance/similarity measures between probability density functions," *International Journal of Mathematical Models and Methods in Applied Sciences*, vol. 1, pp. 300–307, 4 2007.

[40] M. Kokare, B. Chatterji, and P. Biswas, "Comparison of similarity metrics for texture image retrieval," in *TENCON 2003. Conference on Convergent Technologies for the Asia-Pacific Region*, IEEE, vol. 2, 2003, pp. 571–575.

[41] R. Hu, S. Ruger, D. Song, H. Liu, and Z. Huang, "Dissimilarity measures for content-based image retrieval," in *Multimedia and Expo, 2008 IEEE International Conference on*, IEEE, 2008, pp. 1365–1368.

[42] H. Liu, D. Song, S. Rüger, R. Hu, and V. Uren, "Comparing dissimilarity measures for content-based image retrieval," in *Asia Information Retrieval Symposium*, Springer, 2008, pp. 44–50.

[43] S. Patil and S. Talbar, "Content based image retrieval using various distance metrics," in *Data Engineering and Management*, Springer, 2012, pp. 154–161.

[44] M. Alfarraj, Y. Alaudah, and G. AlRegib, "Content-adaptive non-parametric texture similarity measure," in *2016 IEEE 18th International Workshop on Multimedia Signal Processing (MMSP)*, Sep. 2016, pp. 1–6.

[45] O. Roy and M. Vetterli, "The effective rank: A measure of effective dimensionality," in *European signal processing conference (EUSIPCO)*, 2007, pp. 606–610.

[46] CeGP, "LANDMASS: large north-sea dataset of migrated aggregated seismic structures," 2015.

[47] I. Borg and P. J. Groenen, *Modern multidimensional scaling: Theory and applications*. Springer Science & Business Media, 2005.

[48] Y. Alaudah and G. AlRegib, "Weakly-supervised labeling of seismic volumes using reference exemplars," in *Image Processing (ICIP), 2016 IEEE International Conference on*, IEEE, 2016, pp. 4373–4377.

[49] Z. Long, Y. Alaudah, M. Ali Qureshi, Y. Hu, Z. Wang, M. Alfarraj, G. AlRegib, A. Amin, M. Deriche, S. Al-Dharrab, *et al.*, "A comparative study of texture attributes for characterizing subsurface structures in seismic volumes," *Interpretation*, vol. 6, no. 4, T1055–T1066, 2018.

[50] M. Alfarraj, Y. Alaudah, Z. Long, and G. AlRegib, "Multiresolution analysis and learning for computational seismic interpretation," *The Leading Edge*, vol. 37, no. 6, pp. 443–450, 2018. eprint: `https://doi.org/10.1190/tle37060443.1`.

[51] A. Waldeland and A. Solberg, "Salt classification using deep learning," in *79th EAGE Conference and Exhibition 2017*, 2017.

[52] P. Guillen, G. Larrazabal, G. González, D. Boumber, and R. Vilalta, "Supervised learning to detect salt body," *Society of Exploration Geophysicists*, 2015.

[53] C. Ramirez, G. Larrazabal, and G. Gonzalez, "Salt body detection from seismic data via sparse representation," *Geophysical Prospecting*, vol. 64, no. 2, pp. 335–347, 2016.

[54] M. A. Shafiq, Z. Wang*, A. Amin, T. Hegazy, M. Deriche, and G. AlRegib, "Detection of salt-dome boundary surfaces in migrated seismic volumes using gradient of textures," in *SEG Technical Program Expanded Abstracts 2015*, Society of Exploration Geophysicists, 2015, pp. 1811–1815.

[55] T. Hegazy* and G. AlRegib, "Texture attributes for detecting salt bodies in seismic data," in *SEG Technical Program Expanded Abstracts 2014*, Society of Exploration Geophysicists, 2014, pp. 1455–1459.

[56] A. Berthelot, A. H. Solberg, and L.-J. Gelius, "Texture attributes for detection of salt," *Journal of Applied Geophysics*, vol. 88, pp. 52–69, 2013.

[57] J. Lomask, R. G. Clapp, and B. Biondi, "Application of image segmentation to tracking 3d salt boundaries," *GEOPHYSICS*, vol. 72, no. 4, P47–P56, 2007. eprint: `https://doi.org/10.1190/1.2732553`.

[58] F. Farrokhnia, A. R. Kahoo, and M. Soleimani, "Automatic salt dome detection in seismic data by combination of attribute analysis on crs images and igu map delineation," *Journal of Applied Geophysics*, vol. 159, pp. 395–407, 2018.

[59] X. Wu and D. Hale, "Automatically interpreting all faults, unconformities, and horizons from 3d seismic images," *Interpretation*, vol. 4, no. 2, T227–T237, 2016.

[60] Z. Wang and G. AlRegib, "Interactive fault extraction in 3-d seismic data using the hough transform and tracking vectors," *IEEE Transactions on Computational Imaging*, vol. 3, no. 1, pp. 99–109, 2017.

[61] M. A. Shafiq, H. Di, and G. AlRegib, "A novel approach for automated detection of listric faults within migrated seismic volumes," *Journal of Applied Geophysics*, 2018.

[62] H. Di, Z. Wang, and G. AlRegib, "Seismic fault detection from post-stack amplitude by convolutional neural networks," in *80th EAGE Conference and Exhibition 2018*, 2018.

[63] X. Wu and S. Fomel, "Automatic fault interpretation with optimal surface voting," *Geophysics*, vol. 83, no. 5, pp. 1–52, 2018.

[64] T. Coleou, M. Poupon, and K. Azbel, "Unsupervised seismic facies classification: A review and comparison of techniques and implementation," *The Leading Edge*, vol. 22, no. 10, pp. 942–953, 2003. eprint: `http://dx.doi.org/10.1190/1.1623635`.

[65] C. Rutherford Ildstad and P. Bormann, *MalenoV: Tool for training and classifying SEGY seismic facies using deep neural networks*, `https://github.com/bolgebrygg/MalenoV`, 2017.

[66] M. Araya-Polo, T. Dahlke, C. Frogner, C. Zhang, T. Poggio, and D. Hohl, "Automated fault detection without seismic processing," *The Leading Edge*, vol. 36, no. 3, pp. 208–214, 2017.

[67] X. Wu, Y. Shi, S. Fomel, and L. Liang, "Convolutional neural networks for fault interpretation in seismic images," in *SEG Technical Program Expanded Abstracts 2018*, Society of Exploration Geophysicists, 2018, pp. 1946–1950.

[68] R. M. Haralick, K. S. Shanmugam, and I. Dinstein, "Textural features for image classification.," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 3, no. 6, pp. 610–621, 1973.

[69] Y. Zhai, D. L. Neuhoff, and T. N. Pappas, "Local radius index-a new texture similarity feature," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2013, pp. 1434–1438.

[70] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 24, no. 7, pp. 971–987, 2002.

[71] Z. Guo, L. Zhang, and D. Zhang, "A completed modeling of local binary pattern operator for texture classification," *Image Processing, IEEE Transactions on*, vol. 19, no. 6, pp. 1657–1663, 2010.

[72] L. Liu, L. Zhao, Y. Long, G. Kuang, and P. Fieguth, "Extended local binary patterns for texture classification," *Image and Vision Computing*, vol. 30, no. 2, pp. 86–99, 2012.

[73] Y. Hu, Z. Long, and G. AlRegib, "Completed local derivative pattern for rotation invariant texture classification," in *Proceedings of the International Conference on Image Processing (ICIP)*, IEEE, 2016, pp. 3548–3552.

[74] J. G. Daugman, "Complete discrete 2-d gabor transforms by neural networks for image analysis and compression," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 36, no. 7, pp. 1169–1179, 1988.

[75] M. N. Do and M. Vetterli, "The Contourlet Transform: An Efficient Directional Multiresolution Image Representation," *IEEE Transactions on Image Processing*, vol. 14, no. 12, pp. 1–16, 2005.

[76] A. L. da Cunha, J. Zhou, and M. N. Do, "The nonsubsampled contourlet transform: theory, design, and applications.," *IEEE transactions on image processing : a publication of the IEEE Signal Processing Society*, vol. 15, no. 10, pp. 3089–3101, 2006.

[77] V. Vapnik, "An overview of statistical learning theory," *IEEE Transactions on Neural Networks*, vol. 10, no. 5, pp. 988–999, Sep. 1999.

[78] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 888–905, Aug. 2000.

[79] A. D. Halpert, R. G. Clapp, and B. Biondi, "Salt delineation via interpreter-guided 3d seismic image segmentation," *Interpretation*, vol. 2, no. 2, T79–T88, 2014. eprint: `http://dx.doi.org/10.1190/INT-2013-0159.1`.

[80] D. Hale and J. Emanuel, "Seismic interpretation using global image segmentation," in *SEG Technical Program Expanded Abstracts 2003*. 2005, pp. 2410–2413. eprint: `http://library.seg.org/doi/pdf/10.1190/1.1817860`.

[81] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Susstrunk, "SLIC superpixels compared to state-of-the-art superpixel methods," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 34, no. 11, pp. 2274–2282, Nov. 2012.

[82] R. Zabih and V. Kolmogorov, "Spatially coherent clustering using graph cuts," in *2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2004), with CD-ROM, 27 June - 2 July 2004, Washington, DC, USA*, 2004, pp. 437–444.

[83] L. Vincent and P. Soille, "Watersheds in digital spaces: An efficient algorithm based on immersion simulations," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 13, no. 6, pp. 583–598, Jun. 1991.

[84] A. Levinshtein, A. Stere, K. N. Kutulakos, D. J. Fleet, S. J. Dickinson, and K. Siddiqi, "Turbopixels: Fast superpixels using geometric flows," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 12, pp. 2290–2297, Dec. 2009.

[85] Z. Long, Y. Alaudah, M. A. Qureshi, Y. Hu, Z. Wang, M. Alfarraj, G. AlRegib, A. Amin, M. Deriche, and S. Al-Dharrab, "A comparative study of texture attributes for characterizing subsurface structures in migrated seismic volumes," *Submitted to Interpretation*, 2017.

[86] I. Daubechies, *Ten lectures on wavelets*. Siam, 1992, vol. 61.

[87] H Guo, M Lang, J. Odegard, and C. Burrus, "Nonlinear processing of a shift-invariant dwt for noise reduction and compression," in *Proceedings of the International Conference on Digital Signal Processing*, 1995, pp. 332–337.

[88] M. Lang, H. Guo, J. E. Odegard, C. S. Burrus, and R. Wells Jr, "Noise reduction using an undecimated discrete wavelet transform," *Signal Processing Letters, IEEE*, vol. 3, no. 1, pp. 10–12, 1996.

[89] R. Zaciu, C. Lamba, C. Burlacu, and G. Nicula, "Image compression using an overcomplete discrete wavelet transform," *Consumer Electronics, IEEE Transactions on*, vol. 42, no. 3, pp. 800–807, 1996.

[90] J. E. Fowler, "The redundant discrete wavelet transform and additive noise," *Signal Processing Letters, IEEE*, vol. 12, no. 9, pp. 629–632, 2005.

[91] R. Mehrotra, K. R. Namuduri, and N. Ranganathan, "Gabor filter-based edge detection," *Pattern recognition*, vol. 25, no. 12, pp. 1479–1494, 1992.

[92] A. K. Jain and F. Farrokhnia, "Unsupervised texture segmentation using gabor filters," *Pattern recognition*, vol. 24, no. 12, pp. 1167–1186, 1991.

[93] P. Burt and E. Adelson, "The laplacian pyramid as a compact image code," *IEEE Transactions on Communications*, vol. 31, no. 4, pp. 532–540, Apr. 1983.

[94] R. H. Bamberger and M. J. T. Smith, "A filter bank for the directional decomposition of images: Theory and design," *IEEE Transactions on Signal Processing*, vol. 40, no. 4, pp. 882–893, Apr. 1992.

[95] F. Lu, Q. Zhao, and G. Yang, "Nonsubsampled contourlet transform-based algorithm for no-reference image quality assessment," *Optical Engineering*, vol. 50, no. 6, pp. 067 010–067 010, 2011.

[96] Y. Hu, Z. Long, and G. AlRegib, "Completed local derivative pattern for rotation invariant texture classification," in *Image Processing (ICIP), 2016 IEEE International Conference on*, IEEE, 2016, pp. 3548–3552.

[97] R. M. Haralick, K. Shanmugam, *et al.*, "Textural features for image classification," *IEEE Transactions on systems, man, and cybernetics*, no. 6, pp. 610–621, 1973.

[98] A. Berthelot, A. H. S. Solberg, E. Morisbak, and L.-J. Gelius, "3d segmentation of salt using texture attributes," in *SEG Technical Program Expanded Abstracts 2012*. 2012, pp. 1–5. eprint: `https://library.seg.org/doi/pdf/10.1190/segam2012-1443.1`.

[99] Z. Wang, C. Yin, and W. Zhao, "Glcm parameters of channel texture analysis," in *SEG Technical Program Expanded Abstracts 2011*. 2011, pp. 1989–1993. eprint: `https://library.seg.org/doi/pdf/10.1190/1.3627597`.

[100] A. Amin*, M. Deriche, T. Hegazy, Z. Wang, and G. AlRegib, "A novel approach for salt dome detection using a dictionary-based classifier," in *SEG Technical Program Expanded Abstracts 2015*. 2015, pp. 1816–1820. eprint: `https://library.seg.org/doi/pdf/10.1190/segam2015-5925748.1`.

[101] G. Zhang, J. Zheng, X. Yin, and Y. Pu, "Coherence cube based on curvelet transform," in *SEG Technical Program Expanded Abstracts 2008*. 2008, pp. 924–928. eprint: `https://library.seg.org/doi/pdf/10.1190/1.3063790`.

[102] V. Kumar and F. J. Herrmann, "Deconvolution with curvelet-domain sparsity," in *SEG Technical Program Expanded Abstracts 2008*. 2008, pp. 1996–2000. eprint: `https://library.seg.org/doi/pdf/10.1190/1.3059287`.

[103] D. Donno, H. Chauris, and M. Noble, "Curvelet-based multiple prediction," *GEOPHYSICS*, vol. 75, no. 6, WB255–WB263, 2010. eprint: `https://doi.org/10.1190/1.3502663`.

[104] M. Naghizadeh and M. Sacchi, "Ground-roll attenuation using curvelet downscaling," *GEOPHYSICS*, vol. 83, no. 3, pp. V185–V195, 2018. eprint: `https://doi.org/10.1190/geo2017-0562.1`.

[105] E. P. Simoncelli, W. T. Freeman, E. H. Adelson, and D. J. Heeger, "Shiftable multiscale transforms," *IEEE Transactions on Information Theory*, vol. 38, no. 2, pp. 587–607, Mar. 1992.

[106] J. Wan, D. Wang, S. C. H. Hoi, P. Wu, J. Zhu, Y. Zhang, and J. Li, "Deep learning for content-based image retrieval: A comprehensive study," in *Proceedings of the 22nd ACM international conference on Multimedia*, ACM, 2014, pp. 157–166.

[107] M. Billinghurst, A. Clark, G. Lee, *et al.*, "A survey of augmented reality," *Foundations and Trends® in Human–Computer Interaction*, vol. 8, no. 2-3, pp. 73–272, 2015.

[108] A. Garcia-Garcia, S. Orts-Escolano, S. Oprea, V. Villena-Martinez, and J. Garcia-Rodriguez, "A review on deep learning techniques applied to semantic segmentation," *arXiv preprint arXiv:1704.06857*, 2017.

[109] P. Krähenbühl and V. Koltun, "Efficient inference in fully connected CRFs with Gaussian edge potentials," in *Advances in neural information processing systems*, 2011, pp. 109–117.

[110] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

[111] S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, and P. H. Torr, "Conditional random fields as recurrent neural networks," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1529–1537.

[112] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," *arXiv preprint arXiv:1511.07122*, 2015.

[113] C. Farabet, C. Couprie, L. Najman, and Y. LeCun, "Learning hierarchical features for scene labeling," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1915–1929, Aug. 2013.

[114] G. Lin, C. Shen, A. Van Den Hengel, and I. Reid, "Efficient piecewise training of deep structured models for semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 3194–3203.

[115] L.-C. Chen, Y. Yang, J. Wang, W. Xu, and A. L. Yuille, "Attention to scale: Scale-aware semantic image segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 3640–3649.

[116] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3431–3440.

[117] H. Noh, S. Hong, and B. Han, "Learning deconvolution network for semantic segmentation," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1520–1528.

[118] S. Hong, S. Kwak, and B. Han, "Weakly supervised learning with deep convolutional neural networks for semantic segmentation: Understanding semantic layout of images with minimum human supervision," *IEEE Signal Processing Magazine*, vol. 34, no. 6, pp. 39–49, 2017.

[119] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *arXiv preprint arXiv:1511.00561*, 2015.

[120] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, Springer, 2015, pp. 234–241.

[121] D. Pathak, E. Shelhamer, J. Long, and T. Darrell, "Fully convolutional multi-class multiple instance learning," *arXiv preprint arXiv:1412.7144*, 2014.

[122] P. O. Pinheiro and R. Collobert, "From image-level to pixel-level labeling with convolutional networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1713–1721.

[123] H.-E. Kim and S. Hwang, "Deconvolutional feature stacking for weakly-supervised semantic segmentation," *arXiv preprint arXiv:1602.04984*, 2016.

[124] T. Durand, T. Mordan, N. Thome, and M. Cord, "Wildcat: Weakly supervised learning of deep convnets for image classification, pointwise localization and segmentation," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2017)*, 2017.

[125] G. Papandreou, L.-C. Chen, K. P. Murphy, and A. L. Yuille, "Weakly-and semi-supervised learning of a deep convolutional network for semantic image segmentation," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1742–1750.

[126] Q. Hou, P. K. Dokania, D. Massiceti, Y. Wei, M.-M. Cheng, and P. Torr, "Mining pixels: Weakly supervised semantic segmentation using image labels," *arXiv preprint arXiv:1612.02101*, 2016.

[127] Q. Hou, D. Massiceti, P. K. Dokania, Y. Wei, M.-M. Cheng, and P. H. Torr, "Bottom-up top-down cues for weakly-supervised semantic segmentation," in *International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition*, Springer, 2017, pp. 263–277.

[128] S. Kwak, S. Hong, and B. Han, "Weakly supervised semantic segmentation using superpixel pooling network.," in *AAAI*, 2017, pp. 4111–4117.

[129] Y. Wei, X. Liang, Y. Chen, X. Shen, M.-M. Cheng, J. Feng, Y. Zhao, and S. Yan, "STC: A simple to complex framework for weakly-supervised semantic segmentation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 11, pp. 2314–2320, 2017.

[130] P. Tang, X. Wang, A. Wang, Y. Yan, W. Liu, J. Huang, and A. Yuille, "Weakly supervised region proposal network and object detection," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 352–368.

[131] M. Oquab, L. Bottou, I. Laptev, and J. Sivic, "Is object localization for free?-weakly-supervised learning with convolutional neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 685–694.

[132] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2921–2929.

[133] L. Bazzani, A. Bergamo, D. Anguelov, and L. Torresani, "Self-taught object localization with deep networks," in *Applications of Computer Vision (WACV), 2016 IEEE Winter Conference on*, IEEE, 2016, pp. 1–9.

[134] D. Kim, D. Yoo, I. S. Kweon, *et al.*, "Two-phase learning for weakly supervised object localization," *arXiv preprint arXiv:1708.02108*, 2017.

[135] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "Imagenet large scale visual recognition challenge," *International Journal of Computer Vision (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015.

[136] Y. Wei, X. Liang, Y. Chen, Z. Jie, Y. Xiao, Y. Zhao, and S. Yan, "Learning to segment with image-level annotations," *Pattern Recognition*, vol. 59, pp. 234–244, 2016.

[137] A. Kolesnikov and C. H. Lampert, "Seed, expand and constrain: Three principles for weakly-supervised image segmentation," in *European Conference on Computer Vision*, Springer, 2016, pp. 695–711.

[138] A. Roy and S. Todorovic, "Combining bottom-up, top-down, and smoothness cues for weakly supervised image segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 3529–3538.

[139] R. Cabral, F. De la Torre, J. P. Costeira, and A. Bernardino, "Matrix completion for weakly-supervised multi-label image classification," *IEEE transactions on pattern analysis and machine intelligence*, vol. 37, no. 1, pp. 121–135, 2015.

[140] D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization.," *Nature*, vol. 401, no. 6755, pp. 788–91, 1999. arXiv: `arXiv:1408.1149`.

[141] D. Lee and H. Seung, "Algorithms for non-negative matrix factorization," *Advances in neural information processing systems*, no. 1, pp. 556–562, 2001. arXiv: `0408058v1 [arXiv:cs]`.

[142] S. Hong, J. Choi, J. Feyereisl, B. Han, and L. S. Davis, "Joint image clustering and labeling by matrix factorization," *IEEE transactions on pattern analysis and machine intelligence*, vol. 38, no. 7, pp. 1411–1424, 2016.

[143] Y. Niu, Z. Lu, S. Huang, P. Han, and J.-R. Wen, "Weakly supervised matrix factorization for noisily tagged image parsing," in *IJCAI*, 2015, pp. 3749–3755.

[144] Z. Lu, Z. Fu, T. Xiang, P. Han, L. Wang, and X. Gao, "Learning from weak and noisy labels for semantic segmentation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 3, pp. 486–500, 2017.

[145] G. Zhang and X. Gong, "Nonnegative matrix cofactorization for weakly supervised image parsing," *IEEE Signal Processing Letters*, vol. 23, no. 11, pp. 1682–1686, Nov. 2016.

[146] J. Yuan, D. Wang, and A. M. Cheriyadat, "Factorization-based texture segmentation," *IEEE Transactions on Image Processing*, vol. 24, no. 11, pp. 3488–3497, 2015.

[147] Y. Alaudah and G. AlRegib, "A weakly supervised approach to seismic structure labeling," in *SEG Technical Program Expanded Abstracts 2017*. 2017, pp. 2158–2163. eprint: `https://library.seg.org/doi/pdf/10.1190/segam2017-17793533.1`.

[148] Y. Alaudah, M. Alfarraj, and G. AlRegib, "Structure label prediction using similarity-based retrieval and weakly supervised label mapping," *GEOPHYSICS*, vol. 84, no. 1, pp. V67–V79, 2019. eprint: `https://doi.org/10.1190/geo2018-0028.1`.

[149] A. C. Türkmen, "A review of nonnegative matrix factorization methods for clustering," pp. 1–23, 2015. arXiv: `1507.03194`.

[150] C. Ding, X. He, and H. D. Simon, "On the equivalence of nonnegative matrix factorization and spectral clustering," *Proceedings of the fifth SIAM International Conference on Data Mining (SDM)*, no. 4, pp. 606–610, 2005.

[151] P. O. Hoyer, "Non-negative matrix factorization with sparseness constraints," *The Journal of Machine Learning Research*, vol. 5, pp. 1457–1469, 2004. arXiv: `0408058 [cs]`.

[152] V. K. Potluru, J. Le Roux, B. A. Pearlmutter, J. R. Hershey, and M. E. Brand, "Coordinate descent for mixed-norm nmf," 2013.

[153] A. A. Aqrawi, T. H. Boe, and S. Barros, "Detecting salt domes using a dip guided 3d sobel seismic attribute," in *SEG Technical Program Expanded Abstracts 2011*, Society of Exploration Geophysicists, 2011, pp. 1014–1018.

[154] A. Asjad and D. Mohamed, "A new approach for salt dome detection using a 3d multidirectional edge detector," *Applied Geophysics*, vol. 12, no. 3, pp. 334–342, 2015.

[155] Z. Jing, Z. Yanqing, C. Zhigang, and L. Jianhua, "Detecting boundary of salt dome in seismic data with edge-detection technique," in *SEG Technical Program Expanded Abstracts 2007*, Society of Exploration Geophysicists, 2007, pp. 1392–1396.

[156] M. A. Shafiq, Y. Alaudah, H. Di, and G. AlRegib, "Salt dome detection within migrated seismic volumes using phase congruency," in *SEG Technical Program Expanded Abstracts 2017*, Society of Exploration Geophysicists, 2017, pp. 2360–2365.

[157] M. A. Shafiq, Y. Alaudah, G. AlRegib, and M. Deriche, "Phase congruency for image understanding with applications in computational seismic interpretation," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017, pp. 1587–1591.

[158] A. Abdelnabi, K. Liu, S. Gao, and Y. Abushalah, "Seismic attributes aided fault detection and enhancement in the sirte basin, libya," in *SEG Technical Program Expanded Abstracts 2017*, Society of Exploration Geophysicists, 2017, pp. 2340–2344.

[159] B. Zhang, Y. Liu, M. Pelissier, and N. Hemstra, "Semiautomated fault interpretation based on seismic attributes," *Interpretation*, vol. 2, no. 1, SA11–SA19, 2014.

[160] A. U. Waldeland, A. C. Jensen, L.-J. Gelius, and A. H. S. Solberg, "Convolutional neural networks for automated seismic interpretation," *The Leading Edge*, vol. 37, no. 7, pp. 529–537, 2018.

[161] Y. Shi, X. Wu, and S. Fomel, "Automatic salt-body classification using a deep convolutional neural network," in *SEG Technical Program Expanded Abstracts 2018*, Society of Exploration Geophysicists, 2018, pp. 1971–1975.

[162] H. Di, M. Shafiq, and G. AlRegib, "Patch-level mlp classification for improved fault detection," in *SEG Technical Program Expanded Abstracts 2018*, Society of Exploration Geophysicists, 2018, pp. 2211–2215.

[163] B. Guo, L. Liu, and Y. Luo, "Automatic seismic fault detection with convolutional neural network," in *International Geophysical Conference, Beijing, China, 24-27 April 2018*, Society of Exploration Geophysicists and Chinese Petroleum Society, 2018, pp. 1786–1789.

[164] L. Huang, X. Dong, and T. E. Clee, "A scalable deep learning platform for identifying geologic features from seismic attributes," *The Leading Edge*, vol. 36, no. 3, pp. 249–256, 2017.

[165] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," *arXiv preprint arXiv:1708.02002*, 2017.

[166] S. Wiki, *Seismic stratigraphy*, Online; accessed 03/19/2019, 2019.

[167] A. D. Miall, "Stratigraphy: The modern synthesis," in *Stratigraphy: A Modern Synthesis*, Springer, 2016, pp. 311–370.

[168] Y. Alaudah, P. Michalowicz, M. Alfarraj, and G. AlRegib, "A machine learning benchmark for facies classification," *arXiv preprint arXiv:1901.07659*, 2019.

[169] T. Coléou, M. Poupon, and K. Azbel, "Unsupervised seismic facies classification: A review and comparison of techniques and implementation," *The Leading Edge*, vol. 22, no. 10, pp. 942–953, 2003.

[170] M. C. de Matos, P. L. Osorio, and P. R. Johann, "Unsupervised seismic facies analysis using wavelet transform and self-organizing maps," *Geophysics*, vol. 72, no. 1, P9–P21, 2006.

[171] M. K. Dubois, G. C. Bohling, and S. Chakrabarti, "Comparison of four approaches to a rock facies classification problem," *Computers & Geosciences*, vol. 33, no. 5, pp. 599 –617, 2007.

[172] T. Zhao, V. Jayaram, A. Roy, and K. J. Marfurt, "A comparison of classification techniques for seismic facies recognition," *Interpretation*, vol. 3, no. 4, SAE29–SAE58, 2015. eprint: `https://doi.org/10.1190/INT-2015-0044.1`.

[173] J. S. Dramsch and M. Lüthje, "Deep-learning seismic facies on state-of-the-art cnn architectures," in *SEG Technical Program Expanded Abstracts 2018*. 2018, pp. 2036–2040. eprint: `https://library.seg.org/doi/pdf/10.1190/segam2018-2996783.1`.

[174] T. Zhao, "Seismic facies classification using different deep convolutional neural networks," in *SEG Technical Program Expanded Abstracts 2018*. 2018, pp. 2046–2050. eprint: `https://library.seg.org/doi/pdf/10.1190/segam2018-2997085.1`.

[175] H. Di, Z. Wang, and G. AlRegib, "Real-time seismic-image interpretation via deconvolutional neural network," in *SEG Technical Program Expanded Abstracts 2018*. 2018, pp. 2051–2055. eprint: `https://library.seg.org/doi/pdf/10.1190/segam2018-2997303.1`.

[176] F. Qian, M. Yin, X.-Y. Liu, Y.-J. Wang, C. Lu, and G.-M. Hu, "Unsupervised seismic facies analysis via deep convolutional autoencoders," *GEOPHYSICS*, vol. 83, no. 3, A39–A43, 2018. eprint: `https://doi.org/10.1190/geo2017-0524.1`.

[177] M. A. Shafiq, M. Prabhushankar, H. Di, and G. AlRegib, "Towards understanding common features between natural and seismic images," in *SEG Tech-*

nical Program Expanded Abstracts 2018. 2018, pp. 2076–2080. eprint: `https://library.seg.org/doi/pdf/10.1190/segam2018-2996501.1`.

[178] A Veillard, O Morére, M Grout, and J Gruffeille, "Fast 3d seismic interpretation with unsupervised deep learning: Application to a potash network in the north sea," in *80th EAGE Conference and Exhibition 2018*, 2018.

[179] B. Peters, J. Granek, and E. Haber, "Automatic classification of geologic units in seismic images using partially interpreted examples," *arXiv preprint arXiv:1901.03786*, 2019.

[180] E. Duin, J. Doornenbal, R. Rijkers, J. Verbeek, and T. E. Wong, "Subsurface structure of the netherlands-results of recent onshore and offshore mapping," *Netherlands Journal of Geosciences*, vol. 85, no. 4, p. 245, 2006.

[181] J. C. Doornenbal, "Kilka uwag o kartografii wglkebnej i modelowaniu geologicznym w holandii," *Przeglkad Geologiczny*, vol. 62, no. 12, p. 806, 2014.

[182] H. Van Adrichem Bogaert and W. Kouwe, "1997," *Stratigraphic nomenclature of the Netherlands, revision and update. Mededelingen Rijks Geologische Dienst*, vol. 50, 1993.

[183] H. Mijnlieff, "Top pre-permian distribution map and some thematic regional geologic maps of the netherlands," *ICCP*, 2002.

[184] M. Scheck-Wenderoth and J. Lamarche, "Crustal memory and basin evolution in the central european basin system—new insights from a 3d structural model," *Tectonophysics*, vol. 397, no. 1-2, pp. 143–165, 2005.

[185] P. A. Ziegler, "Evolution of the arctic-north atlantic and the western tethys," 1988.

[186] ——, "Geological atlas of western and central europe," Geological Society of London, 1990.

[187] B. Schroot and H. De Haan, "An improved regional structural model of the upper carboniferous of the cleaver bank high based on 3d seismic interpretation," *Geological Society, London, Special Publications*, vol. 212, no. 1, pp. 23–37, 2003.

[188] G Remmelts, "Salt tectonics in the southern north sea, the netherlands," in *Geology of Gas and Oil under the Netherlands*, Springer, 1996, pp. 143–158.

[189] S. Hochreiter, Y. Bengio, P. Frasconi, J. Schmidhuber, *et al.*, *Gradient flow in recurrent nets: The difficulty of learning long-term dependencies*, 2001.

VITA

Yazeed Alaudah received his B.Sc. degree with a first honors distinction in Electrical Engineering from King Fahd University of Petroleum & Minerals (KFUPM) in Dhahran, Saudi Arabia, in 2011. He received an M.S. degree with a minor in mathematics from Georgia Institute of Technology in 2013. He received his Ph.D. degree in electrical and computer engineering from Georgia Institute of Technology in May 2019. Yazeed joined the Multimedia & Sensors Lab (MSL), and later the Center for Energy and Geo Processing (CeGP) in 2013 and 2015 respectively. During his studies at Georgia Tech, Yazeed worked as a research assistant working on various problems related to visual data representation, computational seismic interpretation, and machine learning. He also worked as a research intern at Panasonic Automotive Innovation Center and Mitsubishi Electric Research Labs (MERL) in addition to working as a machine learning engineer with Airbus Aerial. In May 2013, Yazeed received the SACM Outstanding Student Award. He also received the CSIP Outstanding Research Award in December 2018 and later was nominated for the Roger P. Webb Outstanding Graduate Research Assistant Award. Yazeed is also a lifetime member of the IEEE Eta Kappa Nu (HKN) Honor Society. His research interests are at the intersection of machine learning, computer vision, and seismic interpretation.