Imperial College London

Department of Earth Science and Engineering

MSc in Applied Computational Science and Engineering

Independent Research Project

Final Report

Weakly-supervised Learning of Pixel-level Labelling for Seismic Structures

by

Ping-Chen Tsai

Ping-chen.tsai19@imperial.ac.uk

GitHub login: acse-pt919

August 2020

First Supervisor: Dr. Erik Barrow

Second Supervisor: Navjot Kukreja

Ovation Data Ltd

Unit10, Crayfields Industrial Park

Main Road, St Paul's Cray

Kent, BR53H

GitHub repository: https://github.com/acse-2019/irp-acse-pt919

Acknowledgement

I would like to thank my supervisors, Dr. Erik Barrow and Navjot Kukreja for their great support on project direction and code development advice throughout the research project. I would like to express my appreciation towards Ovation Data especially Haythem Zidi, Mark Wildig, Sam Wildig and Jeff Clifford for providing me technical support on interpreting seismic data at the company. I would also like to thank Jeff Clifford for hosting the weekly meeting and maintaining effective communication between us interns and Ovation Data.

Table of Contents

Abstract

The process of interpreting large quantity of seismic data require high labour input and deep understanding of domain knowledge. Ovation have vast inventory of historic data where much of them contains oil and gas deposits untapped. The growing need of automatic seismic interpretation brings supervised machine learning approaches upfront to reduce work for manual interpretation. To prepare the dataset for supervised learning, outline the precise seismic structures boundaries inside a labelled bounding box is a need for fine-grained task. In this research project, we applied Non-negative Matrix Factorization (NMF) with sparseness constraints to segment seismic structures at pixel-level from a labelled bounding box. Given a bounding box label to extract pixel labels, the approach is therefore a weakly-supervised one. We took seismic data from Netherlands North Sea F3 Block as a case study to test the software developed. Experiments on the data shows that NMF with sparsity control scheme can effectively improves the pixel-level labelling result on fault and salt, but not beneficial to chaotic structure. The result eventually is saved as a pack of classified masks, and it could potentially assist model training for various supervised learning techniques at Ovation.

Keywords: Machine learning, weakly-supervised learning, dimensional reduction, seismic interpretation

1 Introduction

Seismic structures and traps can provide effective information about potential oil and gas reserves. In the petroleum industry, the interpretation of seismic structure has long been an interest for detection of new hydrocarbon reserves, as accurate detection prevents needless and costly drilling onsite harming the environment. Ovation works with huge volumes of present and historic seismic data in exploration industry. Due to limitation of people power at Ovation, manual seismic structures detection is a labour-intensive and time-consuming task. The task is also error-prone due to subsurface complexity. Automatic detection of seismic structures becomes a need to avoid significant labour input and unnecessary cost on drilling. Recently many machine learning methods have been developed to assist automatic seismic interpretation *(Kumar & Sain, 2018; Shafiq et al., 2015).*

A popular approach to illuminate structures and traps automatically is to use image segmentation techniques based on supervised learning method. To obtain the training dataset, the interpreter needs to put a bounding box around the identified structures, and create a segmentation map for the box. Using bounding box with the segmentation map as CNN model training data has been widely attempted by researchers *(Milosavljević, 2020; Chevitarese et al., 2018)*. The trained model can perform semantic segmentation to label other seismic data at pixel level.

However, full supervision for training seismic structure segmentation relies on enormous labelled data. It would be a very costly task for Ovation to prepare thousands of segmented data for training a neural network. *Papandreou et al. (2015)* suggested that labelling by bounding box is 15 times faster/cheaper than labelling at pixel-level for a seismic interpreter. Automate the process of generating pixel-level label within a bounding box therefore become an approach to enhance efficiency of seismic interpretation.

1.1 Goal and Objectives

To relax the training dependency on huge volumes of pixel-level labels, the goal of this project is to automatically provide precise pixel position of a seismic structure within a bounding box. Faults and salt domes are selected as target labelling structures due to their obvious features. We aim to reproduce the work done by *Alaudah & AlRegib (2017)* to obtain pixel-level annotations from labelled bounding boxes. Therefore, the project work is a weakly-supervised learning approach to alleviate one of the key problems Ovation would encounters: labelling significant dataset precisely.

The proposed approach uses non-negative matrix factorization (NMF) with additional sparseness constraints to highlight the seismic features in an image. NMF is a dimensional reduction method that has recently received much attention for interpreting images using low-rank factor matrices *(Kim, He and Park, 2013).* Sparseness constraints are applied to the NMF model based on the sparsity control scheme that *Hoyer (2004)* proposed. Due to limited amount of time, the project only utilised an open source dataset *LANDMASS* (CeGP, 2015) sampled from the Netherlands North Sea F3 block (dGB Earth Sciences, 1987) as experimental input for software development. We tested pixel-level labelling effect on a stack of fault, salt, and other two existed classes data in LANDMASS: Horizon and chaotic. By the end of this project, the code can automatically generate pixel-level masks for the bounding boxes. Appendix A gives the sample result of the pixel-level labels in bounding box for LANDMASS. The resulting masks could be used to train powerful semantic segmentation models for bounding boxes. Thus the project could be an economical tool for Ovation to prepare bounding box mask data for training supervised model. Eventually precise structure detection results could not only benefit Ovation economically but also the environment.

2 Related Work

Manual interpretation of seismic data is not an efficient workflow. There are existing solutions to address this issue. In addition to supervised learning method, one possible approach is unsupervised learning by Self Organizing Map (SOM) network *(Zhang, Quieren & Schuelke, 2001)* to classify track position of geological features. The unsupervised approach does not require ground truth for training a classification model. Others have proposed to train a deep convolutional neural network for seismic traps classification *(Qian et al., 2018)*.

Apart from unsupervised learning, weakly-supervised approaches use coarsely labelled and less-informative (weakly-labelled) seismic slices to perform even more promising results in structure classification, detection, and segmentation at pixel-level *(Alaudah et al., 2018)*. *Ao et al. (2019)* proposed a semi-supervised method that combine unsupervised isolation forest with a supervised feature selection scheme to identify channels in western Bohai Sea. Other method such as semi-supervised least square support vector machine *(Luo et al., 2016)* has confirmed that a liable machine learning can effectively assist seismic interpretation.

3 Methodology and Software Description

The pixel-level labelling software developed in this project is for segmenting muli-classes seismic structures from labelled bounding boxes. The following section present the theories of the approach.

3.1 Principle of Non-negative Matrix Factorization (NMF)

Seismic image analysis usually requires non-negativity constraints because pixels in digital images are always non-negative. The non-negativity constraints allow the representation to be additive combination only, which is efficient for processing images and recognize image features.

Non-negative matrix factorization (NMF) is a dimensionality reduction problem that was firstly introduced by *Paatero and Tapper (1994)* as a better fit to the image data than conventional factor analysis, i.e. Principal component analysis (PCA). *Lee and Seung (1999)* then popularized the NMF algorithm by demonstrating how the non-negativity constraints of the algorithm lead to a parts-based representation of the data. Based on his study, we adapt a sparsity control scheme *Hoyer (2004)* proposed for NMF model to learn more local features for qualitatively better representation of seismic data (See section 3.2).

An image data matrix can be regarded as $X \in \mathbb{R}^{N_p \times N_s}$ with non-negative element. N_p is the total number of pixels, or the dimension, of a single image. N_s is the number of samples in the data. Each column x_i is an 1D flatten image array of grayscale pixels. X consists of multi-class data. Figure 1 shows how to build a 3-classes data matrix.

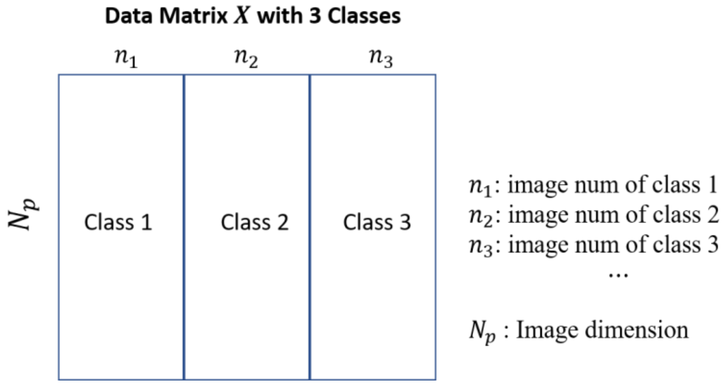


Fig. 1 Creating a data matrix X for processing. Row: single image dimension. Column: total number of images in the dataset. In this case $n_1 + n_2 + n_3 = N_s$.

NMF decomposes X into two low-dimension factors: a basis matrix $W \in \mathbb{R}^{N_p \times N_f}$ and coefficient matrix $H \in \mathbb{R}^{N_f \times N_s}$. N_f is the number of feature components in the lower dimensional space that satisfies the condition $N_f < \min\{N_p, N_s\}$. The relationship of data matrix and its factors is thus given as:

$$X \approx WH \text{ subject to constraints } W, H \geq 0. \quad (1)$$

The NMF algorithm minimizes the distance between X and the product WH by the squared Frobenius norm. This turns the algorithm to minimize a non-convex optimization problem containing variables W and H:

$$\arg\min_{W,H} \frac{1}{2} \|X - WH\|_F^{\ 2} \quad (2)$$

The frobenius norm is represented as $\|\cdots\|_F$ in equation (2). *Ding, He & Simon (2005)* studied NMF from clustering point of view. They demonstrated that factor W is equivalent to centroids of K-means clustering. Based on their findings, this project initialises W using Kmeans centroids. Figure 2 illustrates the creation of W once cluster centroids number k is determined.

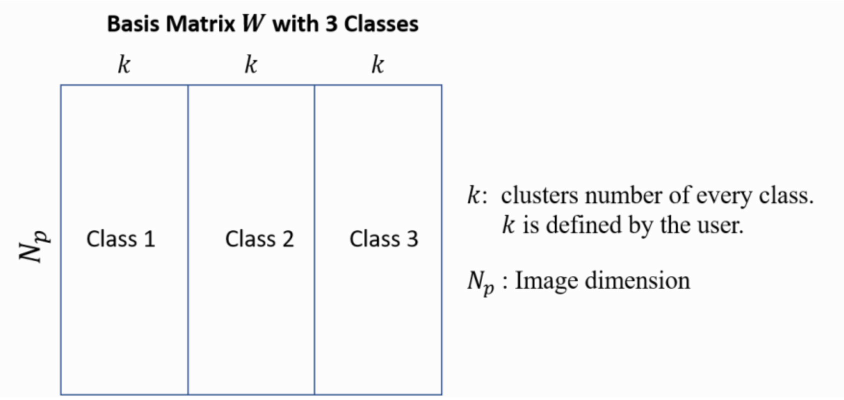


Fig. 2 Basis matrix W initialisation. Row: single image dimension. Column: total number of features defined. In this case, $k + k + k = N_f$.

The optimal value of k is determined by elbows method in this project which is further discussed in section 5.2. To simplify the software, H uses random initialisation with uniform distribution between 0 and 1, as H must be non-negative. Other initialisation method for W and H such as Nonnegative Double Singular Value Decomposition (NNDSVD) *(Boutsidis, C. & Gallopoulos, 2008)* is used in Python NMF package.

3.2 Sparseness Constraint

A seismic image is a big collection of pixels. If the image size is $n \times n$, the image exists in a n^2 dimensional space $\mathbb{R}^{n^2}$. NMF has the property of producing sparse representation of images to reduce data dimensionality, but sometimes the standard sparseness does not bring intuitive representation. Explicitly applying the target sparseness to a seismic image is to map each column of basis matrix from high dimensional space $\mathbb{R}^a$ to a lower dimensional space $\mathbb{R}^{a-1}$. Sparseness constraint keeps only few significant non-zero features and turns unwanted features to zeros by explicitly setting target L1 and L2 norm for every matrix column. This project chooses to sparsify basis matrix columns w_i. *Hoyer (2004)* has proposed a sparsity ρ relationship with the L1 norm and L2 norm of matrix W:

$$\rho(w) = \frac{\sqrt{N_p} - \|w\|_1/\|w\|_2}{\sqrt{N_p} - 1} \tag{3}$$

From equation (3), we can compute L1 to L2 ratio φ according to the desired sparsity set by user. A sparsity equals to 1 indicate complete sparseness, and 0 indicate complete dense. The sparseness constraint is enforced by applying projection operator algorithm introduced by *Hoyer (2004)*. The algorithm takes the vector w_i as input and find a vector v that has target L1 and L2 norms that achieve the pre-defined sparseness $\rho(w)$.

Ratio φ can also be used for model regularization. The regularized objective function of NMF algorithm in Python sklearn package (Pedregosa *et al, 2011*) contains a regularization intensity parameter α. As a result, the objective function for the optimization problem is:

$$\arg\min_{W,H} \frac{1}{2} \|X - WH\|_F^{\ 2} + \alpha\varphi\|W\|_1 + \alpha\varphi\|H\|_1$$

$$+ \frac{\alpha(1-\varphi)}{2} \|W\|_F^{\ 2} + \frac{\alpha(1-\varphi)}{2} \|H\|_F^{\ 2} \tag{4}$$

We choose multiplicative update rule to update W and H successively until they converge. The updating process is totally done by the NMF function from Sklearn. The outputs from the function eventually are the updated W and H. The representation is the product of the two factors. The images can thus be reconstructed by the updated product WH. Experiment on sparsity control to generate different representation is introduced in section 5.3.

3.3 Obtaining Pixel-level Labels from NMF Representation

Every element in the updated W represent a single seismic structure. Each column of the updated coefficient matrix h_n is the weight of features for reconstruction of the n_{th} image. By mapping h_n with the n_{th} row in W, we can calculate the likelihood of every image pixel to be the corresponding structure presented in W. According to *Alaudah& AlRegib (2017)*, a likelihood matrix Y_n can be computed for N_s number of image samples along with a cluster membership matrix $Q \in \{0,1\}^{N_f \times N_l}$, where N_l is the total number of class labels given by the data. Element $Q_{ij} = 1$ if the structure in w_i is label j. The likelihood matrix is hence computed as equation (5):

$$Y_n = W(Q \cdot (h_n * 1_{1\times N_l})) \text{ subjects to } n \in [1, N_s], \tag{5}$$

Where $1_{1\times N_l}$ is a 1D array of N_l number of ones. After constructing Y_n for all the images from the representation, the likelihood matrix Y becomes a 4D array with dimension $N_s \times$ single image rows×single image columns× N_l. Constructing a matrix Y is illustrated in figure 3.

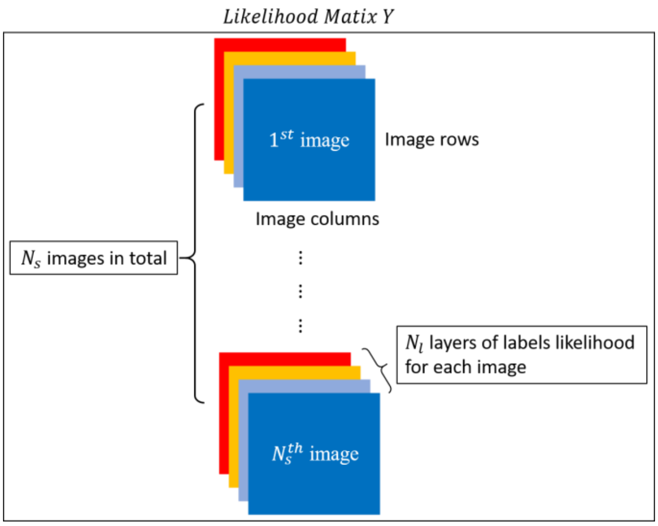


Fig.3 Construction of likelihood matrix Y. Each image are quantified by N_l number of likelihood matrices in coloured layer. Each layer represents a class label. Each layer indicates the likelihood of every pixel being in this class.

Theoretically in an image, for every pixel-level label j, the value of likelihood is the maximum among layers of labels matrices. Therefore *Alaudah& AlRegib (2017)* proposed equation (6) to obtain label i for each pixel of the NMF representation.

$$label_i = \arg\max_j Y_{nj} \tag{6}$$

The following pseudo code demonstrates the details of extracting pixel-level labels. The final output will be a pack of pixel-level label masks.

Algorithm: Label coordination extraction
1. Apply equation (5) to obtain the likelihood matrix Y
2. **for** n = 1: Number of sample images **do**
    Y[n] = gaussian_filter(*Y[n]*)
  end for
3. **for** each pixel in *Y[n]* **do**
    Find maximum value *max_val* in label layers corresponding to position
  end for
4. **for** each pixel in Y[n] **do**
    Find index, *Indices*, of the present layer for the *max_val* found in step 3.
    (Note: Index is the label of the pixel)
    Reshape the *Indices* to image dimension
  end for
5. **for** every *Indices* found in step 4 **do**
    filtered_Indices = median_filter(*Indices*)
  end for
6. Set a *threshold* for labelling
7. **for** each pixel in *Y[n]* **do**
    confidence = sum(Y[n])/*max_val* ∈ *Y[n]*
    if *confidence* < *threshold* **do**:
      filtered_Indices [*current pixel*] = *background_label*
    end if
  end for
8. Return *filtered_Indices*

To accomplish the pixel label extraction scheme, multi-classes (at least two) input are recommended. The algorithm presented can detect degree of confidence of a pixel being a specific structure.

3.4 Software Architecture

Figure 4 illustrates the software workflow. The whole pixel-level label prediction process is a bounding box segmentation task with no feedback loop in the software system. LANDMASS dataset contains four classes of labelled bounding boxes. More details of LANDMASS are presented in section 5.1. All the input data should be preprocessed to have the same dimension. Images are normalised to grayscale between 0 and 1. User is required to define the input number of classes, corresponding integer labels, number of images for each class and target sparsity at the beginning of in `mainTest.py`. Also, user is required to explicitly create a data matrix X with N_p columns at `mainTest.py` before input to `NMF_Sparse.py` for factors initialisation. For X, column x_i is a single flatten image (see figure 1 for data matrix illustration). To save the final output mask into .png, set the paths and call function `label_pixel.save_toPath(…)`.

To be noted that the proposed approach could be taken when the given image set has highly similar texture content within each class. *Alaudah& AlRegib (2017)* thus suggested that data cleaning and content similarity measurement are crucial steps before applying the NMF algorithms for labelling, otherwise the pixel labels would not be accurate.

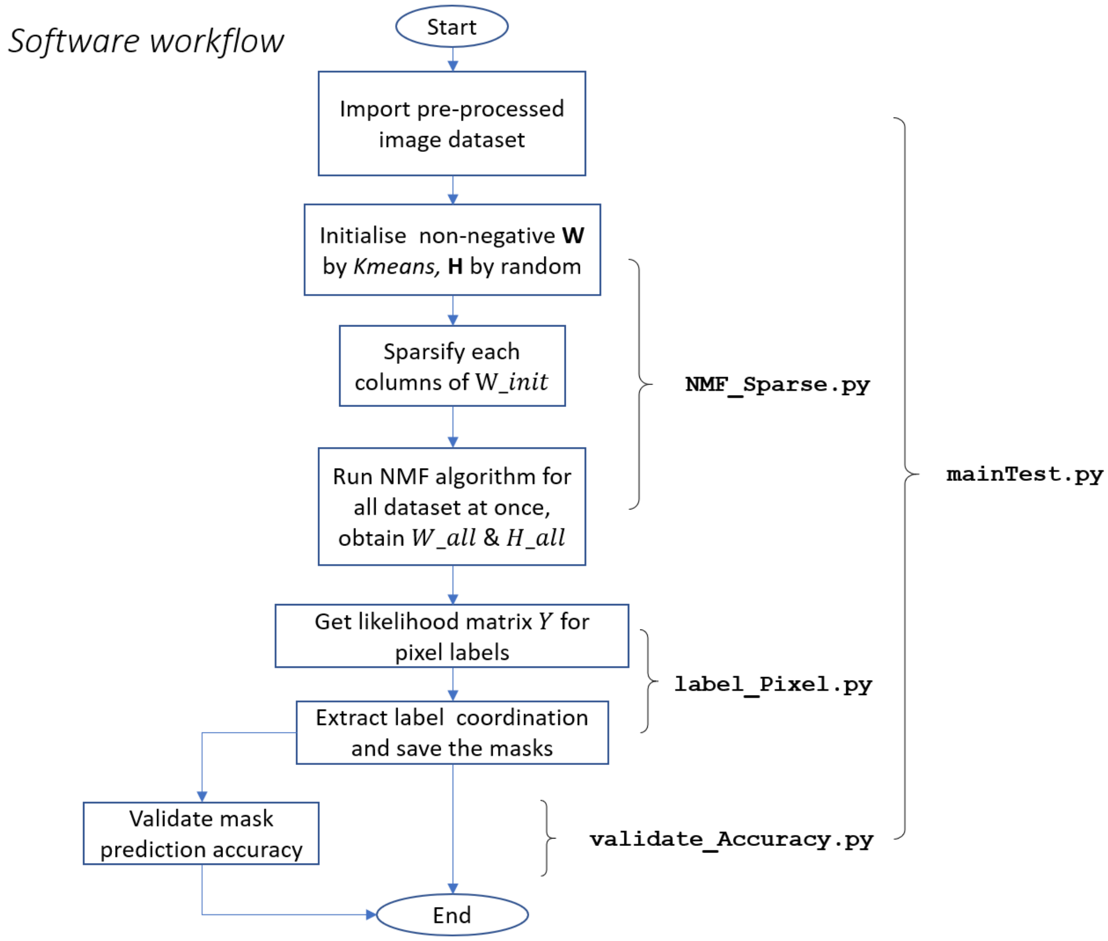


Fig 4. Workflow Pixel-level labels extraction using NMF method: Import course labelled box data, factors initialisation, apply sparseness constraints, fit NMF model for all classes at once, extract pixel labels and validate accuracy(optional) in each image. The responsible modules are labelled beside the chart.

3.4.1 Input Constraints for Ovation's Data

Unfortunately, the software has not been tested on Ovation's data due to time limitation. Ovation's data is given as links to png files. Each png contains numerous labelled bounding boxes with coordinations. We assume the content inside the bounding boxes could be retrieved by python package 'request'. Box content will be classified by labels: One label per box. Corresponding .png information will be stored for result projection. At the preprocessing stage, if a box content has low texture similarity to other boxes in the same class, i.e. uncommon facies direction or cropped structure, then the box should be removed from the input data. Measuring content similarity and thus constructive input selection would be extra work in addition to pixel-level labelling. One suggested approach would be applying the non-parametric texture attribute measurement *Alfarraj et al.(2016)* proposed. Each bounding box should be normalized and resized to the same dimension using Python Imaging Library.

Once the masks for bounding boxes are produced by `label_pixel.py`, we could project the mask contents onto the original png using coordination information.

To validate result accuracy on Ovation's data, it requires true segmentation map created by domain experts in binary form. `validate_Accuracy.py` is responsible for checking accuracies.

3.5 Accuracy Checking

`validate_Accuracy.py` is responsible for accuracy checking. This project uses multilabel classification score to assess the accuracy of predicted labels, but it is not a mandatory step for producing mask. The score computes the percentage of pixels of predicted label that match exactly to the ground truth.

Another method of checking segmentation accuracy is by Intersection over Union (IoU) using Jaccard Index.

$$IoU = \frac{Area\ Overlapped}{Area\ of\ Union} \tag{7}$$

Equation (7) takes area overlapped between the predicted labels and ground-truth samples as numerator. The denominator $Area\ of\ Union$ is the area encompassed by both the prediction and ground truth.

4 Code metadata

Platform

The code is developed using Python version 3.7.6 on the open-source web application Jupyter notebook version 6.0.3. Visual Studio Code (v 1.48) is used for cleaning-up and final compiling.

The link to the developed code: https://github.com/acse-2019/irp-acse-pt919

Libraries

The following libraries are utilised for code development.

1. NumPy (v 1.18.1)
   NumPy consists of various numerical computing routines for multidimensional array.
2. SciPy (v 1.4.1)
   SciPy is a free Python fundamental library for scientific computing. It offers user-friendly tools for numerical computation.
3. Scikit-learn (v 0.22.1)
   Scikit-learn is built on NumPy, SciPy and matplotlib. It contains tools for classification, clustering, dimensionality reduction and pre-processing *(Pedregosa* et al, *2011).*
4. Scikit-image (v 0.16.2)
   Scikit-image contains free collection of image processing algorithms for code developer *(Stéfan et. al, 2014).*
5. Imageio (v 2.6.1)
   Imageio provide tools for reading and writing image files. It runs on Python 2.7 and 3.4+ *(imageio contributors, 2019)*.

External software for Image Annotation

Computer Vision Annotation Tool (CVAT) is a free and powerful tool for ground truth generation.

The link to CVAT is https://github.com/opencv/cvat.

5 Implementation and Code Description

5.1 Experimental Dataset: LANDMASS

LANDMASS contains 4 categories of seismic structures: chaotic, horizons, fault and salt. It has 1000 image in each class, and we import 500 images of each class for experiment. In this project, we named the horizons as class 'unknown' because this project is not interested in checking accuracy of segmenting horizon. Given the label of each image data, the code is specifically tuned to fit LANDMASS for classifying fault and salt at pixel-level. Chaotic and unknown class are two additional samples for tool capability experiment.

The chaotic, salt and fault ground truth are generated by CVAT for LANDMASS dataset. Each class has 30 ground truths for accuracy checking purpose. A ground truth is a segmentation map in binary form.

As high textural content similarity is crucial for each input class, otherwise the predicted labels will not be accurate. Fortunately, the LANDMASS is cleaned, and the texture content shows high similarity. Figure 5 displays 5 of image data in every imported class in LANDMASS.

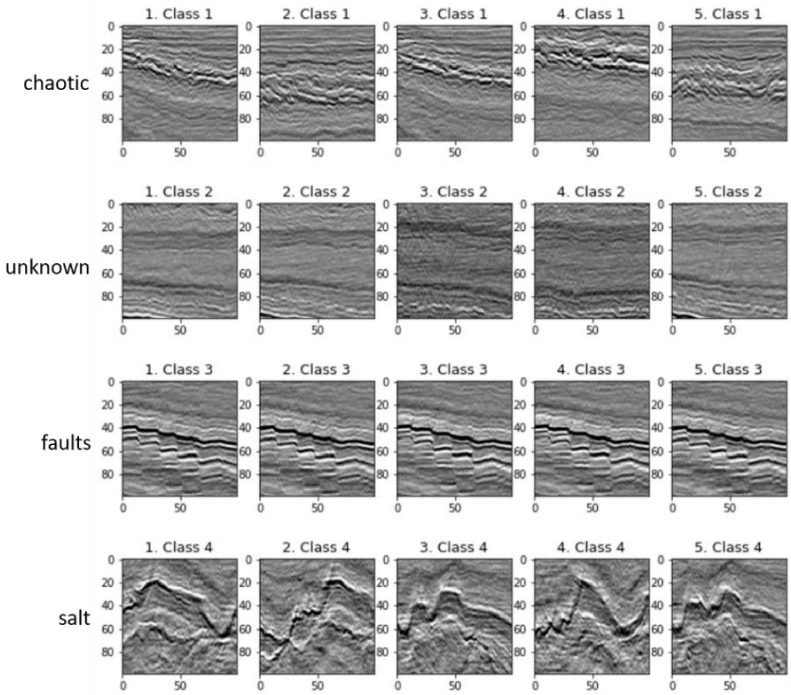


Fig 5. Five randomly selected images in each class of the LANDMASS Dataset. Classes in row orders are chaotic, unknown, faults, and salt.

5.2 Initialisation of basis matrix W and coefficient matrix H

After importing the data, perform Kmeans clustering on individual class of images in `mainTest.py`.

The clustering centroids k can be determined by user by various methods. We used elbow method to select k. The idea of elbow method is to iterate Kmeans with various values of k and compute the sum of squared errors (SSE) for each cluster. Plot the computed errors with respect to value of k and choose the point at the 'elbow' of the plotted line. Sadly the curve for each trial illustrated in figure 6 does not indicate an obvious elbow with even higher number of k. To reduce the NMF software runtime, we choose a point (in yellow) where the SSE starts to diminish at a reasonable low number of clusters for LANDMASS.

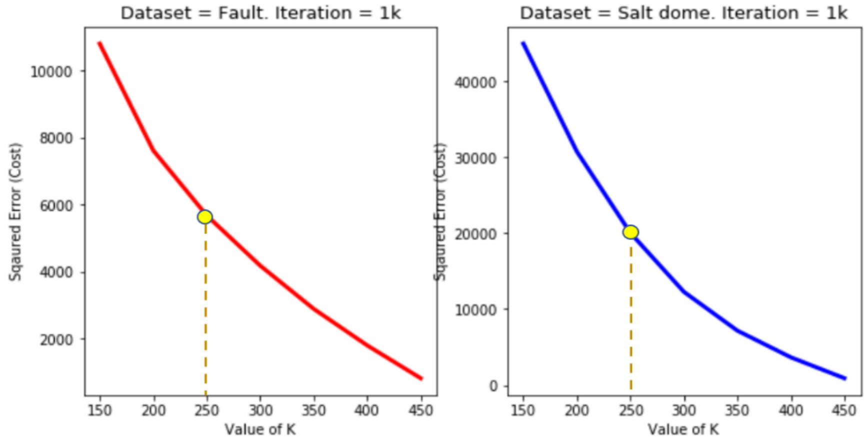


Fig. 6 Choosing number of feature components using elbow method using fault (left) and salt (right) from k=15 to k=450.

The centroids of Kmeans are stacked horizontally to initialise our basis matrix W.

5.3 Experiments with sparsity control

We initialised a W by Kmeans and test the approach without sparseness constraints by solely applying NMF package in sklearn. After updating, plot the product WH for LANDMASS shown in figure 7 (right image). No effective features are highlighted in the right image.

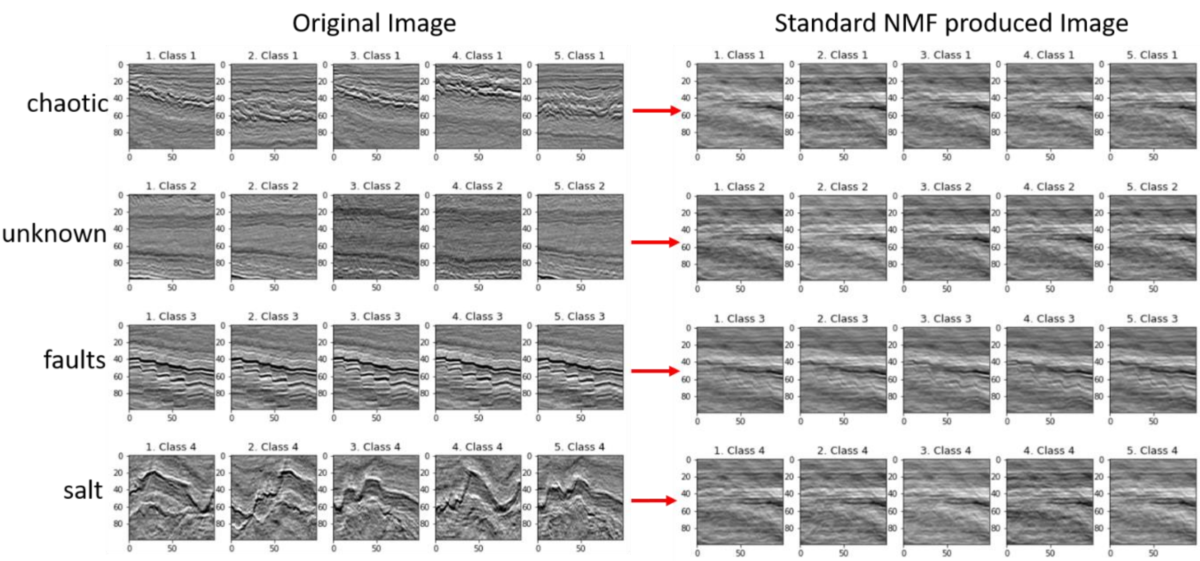


Fig 7. From original image(left) to standard NMF result without sparsity control(right). Without sparsity control, product WH fails in highlighting structure feature (right).

Hoyer (2004) found out that standard NMF sometimes decomposes the data to a highly global representation. Figure 7 confirms the statement and shows that standard NMF cannot produce intuitive features for LANDMASS image set.

To investigate why blurry result are shown without sparseness constraints, histograms in figure 8 plots the updated basis matrix W magnitude with and without sparseness constraint, respectively. The histograms illustrate the concept of sparseness and demonstrate why sparseness benefits NMF image representation. Each element in W is a seismic structural feature. An element with low magnitude means the feature is idle and vice versa. A good image representation is expected to 'turn' a few significant features active. The active feature will be highlighted in the NMF representation.

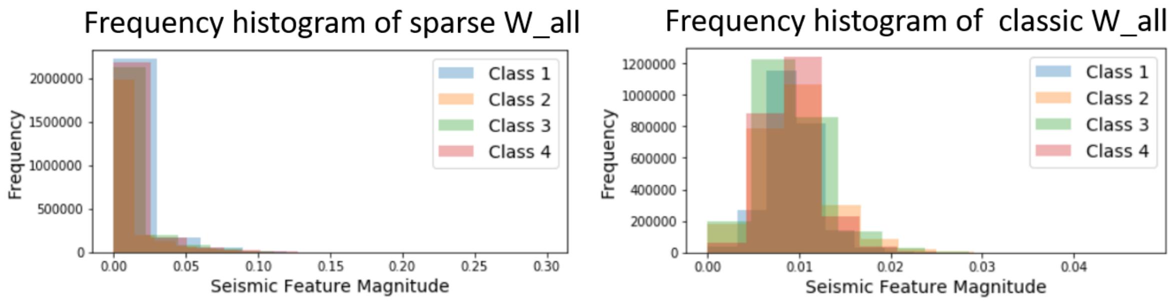


Fig. 8 Comparing sparse W(left) and standard W(right) for four classes. Sparsed W plot has most feature magnitudes closed to 0. Standard W has wider distribution regarding feature magnitude: Less components stand out at large magnitude. Large number of active feature components in standard W distracts the result.

From the right plot in figure 8, more structural components are active without sparseness constraints. Those dispensable features with relatively large magnitude distract WH and result in poor seismic representation shown in figure 7(right image). Users can adjust variable `set_sparsity` in `mainTest.py` to tune the model and eventually generate more distinguishable seismic representation for specific data input.

Figure 9 takes the salt class as example to show how degree of sparsity affects representation.

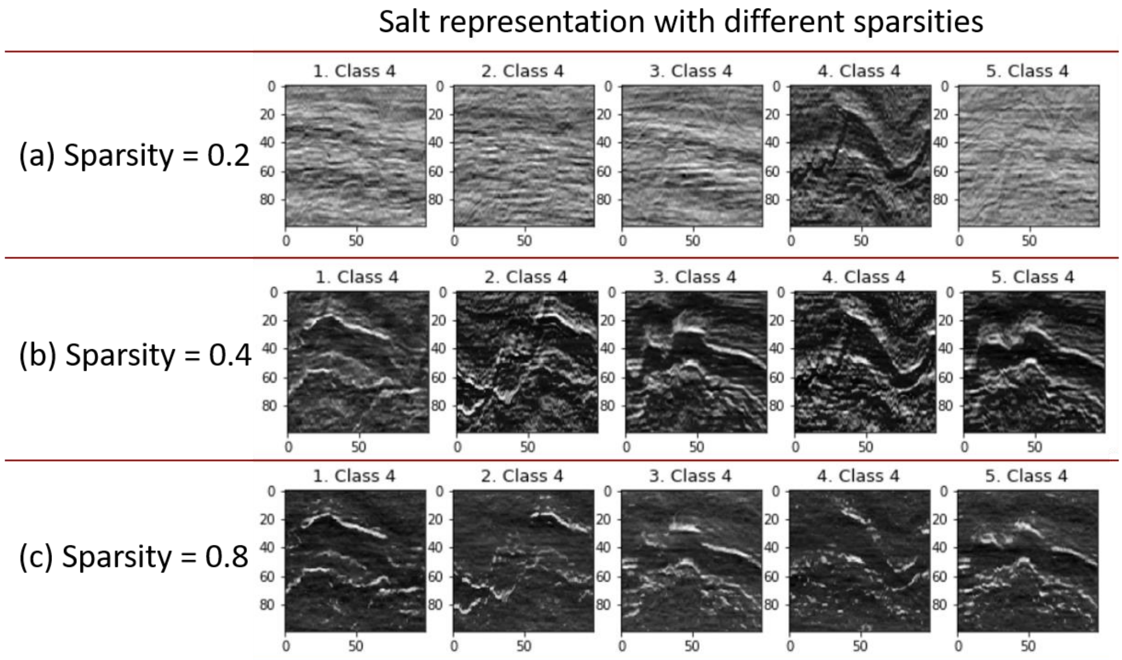


Fig. 9 Salt features learning from LANDMASS dataset with various degree of sparsity. Each column is transformed from the same image. (a) Sparsity = 0.2, salt features are ambiguous. (b) Sparsity = 0.4, salt features are more distinguishable. (c) Sparsity = 0.8, salt features are highlighted in white with background blurred.

When the sparseness is increased, the output product changes from global representation to local representation: From (a) to (c). Higher sparsity (close to 1) highlighted more local features than global ones. By adjusting the sparsity level, Ovation can test out the sparsity that best fit their data.

5.4 Label coordination extraction

Constructing a likelihood matrix for label position extraction. This can be done by the function `label_Pixel.extract_coordination(…)` and plotted by `label_Pixel.plotLabels(…)`. The pixels in a bounding box are classified into $N_s + 1$ number of colours for illustration. Each colour represents a structure label. Figure 8 shows chaotic, fault and salt prediction samples along with corresponding ground truth.

By observation we found that although our input are bounding-box labels, the resulting pixels labels matched most of the structure texture direction in the original seismic image. In figure 10, some salt(red) and fault(blue) was detected in chaotic image (first row). One possible explanation for this phenomenon is that the texture features of chaotic are similar to that of salt and fault. The mismatch of pixel labels implies the significance of using content similarity measurement to select appropriate input images for Ovation due to its data complexity.

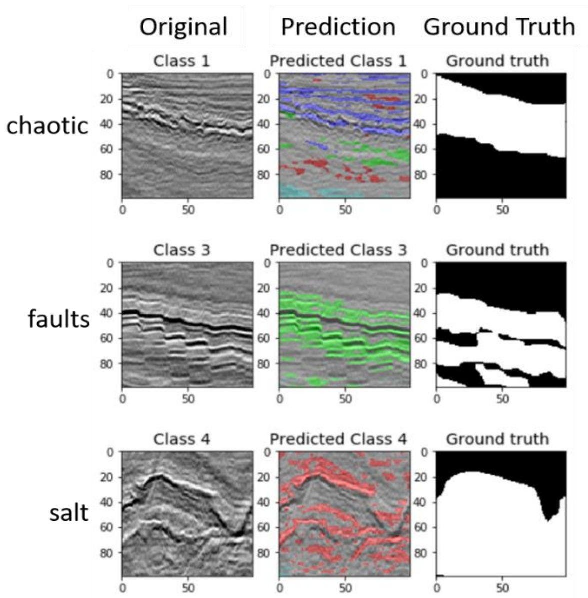


Fig. 10. NMF with sparseness constraints classification results versus ground truth for chaotic, faults and salt. For LANDMASS, $N_s + 1$ = 5 and thus 5 colors. Blue: chaotic. Green: faults. Red: salt domes. Light blue: unknown. Gray: background.

5.5 Result

To fit the LANDMASS data, setting a threshold can greatly improve the pixel label position prediction. Threshold is a function of mean of confidence matrix $\overline{conf}$, standard deviation of confidence matrix σ and a constant c. Users can modify c during tuning stage in `label_Pixel.extract_coordination(…)`. The threshold is calculated as:

$$Threshold = \overline{conf} + \sigma / c \quad (8)$$

In addition to threshold, other parameters also have impact on the label extraction results. Two of the critical factors are sparsity and the regularization intensity alpha. Jaccard Index from equation (7) and standard accuracy score are calculated for tuning purpose. We built `tuneModel.py` to plot accuracy trends when changing sparsity and alpha. Table 1 and Table 2 list parameters for tuning sparsity and alpha respectively. Figure 11 and 12 shows trends of scores for class 1 chaotic, class 3 fault and class 4 salt.

NMF Parameter Table 1				
Iteration	Sparsity	Alpha	Threshold	Sigma
1	0.4	0.057	$\overline{conf} + \sigma/2.7$	0.45
2	0.5125	0.057	$\overline{conf} + \sigma/2.7$	0.45
3	0.625	0.057	$\overline{conf} + \sigma/2.7$	0.45
4	0.7375	0.057	$\overline{conf} + \sigma/2.7$	0.45
5	0.85	0.057	$\overline{conf} + \sigma/2.7$	0.45

Table 1. Changing sparsity level

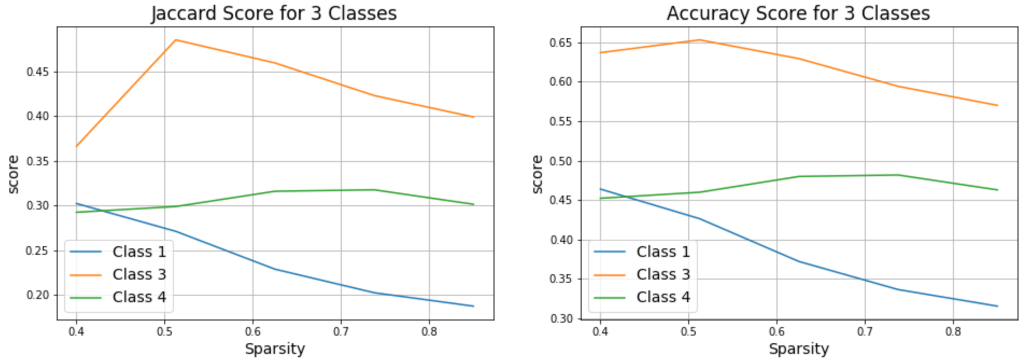


Fig. 11 Accuracy checking by Jaccard Score and Accuracy score with various degree of sparsity.

NMF Parameter Table 2				
Iteration	Sparsity	Alpha	Threshold	Sigma
1	0.44	0.01	$\overline{conf} + \sigma/2.7$	0.45
2	0.44	0.0575	$\overline{conf} + \sigma/2.7$	0.45
3	0.44	0.105	$\overline{conf} + \sigma/2.7$	0.45
4	0.44	0.1525	$\overline{conf} + \sigma/2.7$	0.45
5	0.44	0.2	$\overline{conf} + \sigma/2.7$	0.45

Table 2. Changing regularization intensity alpha.

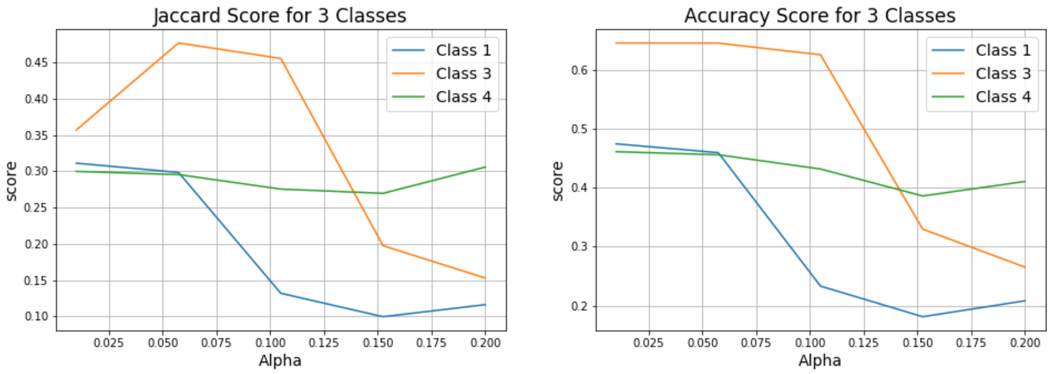


Fig. 12 Accuracy checking by Jaccard Score and Accuracy score with various degree of regularization intensity α.

We observe that even structure edge and texture direction by prediction are well highlighted by observation, the accuracy score plotted in figure 11 and 12 are low. This could be explained by figure 10 that ground truths are filled but the predicted labels are scattered. By tuning, we found that the combination of sparsity= 0.5125, alpha = 0.0575 can produce a model with the highest accuracy scores 65%(fault) and 47%(salt) to fit LANDMASS. Sample predicted results for the whole LANDMASS using the same model are attached in Appendix A, while Appendix B shows corresponding binary masks output by `label_pixel.createBinaryMask(…)`.An inference is drawn from the results that the model must be tuned again to fit Ovation's data due to high degree of texture complexity in their data.

6 Discussion

The software was tested on LANDMASS only, so there are uncertainties towards testing on Ovation dataset. It shows higher accuracy with fault and salt, but lower with chaotic. Using brute force search can find us optimal sparsity and regularization intensity to fit the data.

A key strength of the approach is that it can process a stack of data with different classes at the same time. The sparsity control scheme is implemented to better highlight the seismic structural features. However, the label prediction at pixel level performed poorly when the observed directional texture content is complicated, i.e. chaotic.

We have not investigated reliability of the approach with other structures due to limited preprocessed data. The potential difficulties of using Ovation's data as input is to prepare bounding boxes with similar texture content and validate the accuracy (constraints see section 3.4.1). A well-preprocessed dataset is the key for our model to generate effective masks and saved for supervised learning. Therefore, the portability of the tool has to be improved to be directly applicable for a new dataset. If we overcome those difficulties, the software could be a viable tool for Ovation to detect seismic structures in a bounding box efficiently.

7 Conclusion

In summary, we applied a weakly-supervised approach to highlight multi-class seismic features and output the pixel-level label masks from coarsely labelled bounding boxes. The approach was developed based on Non-negative Matrix Factorization with sparseness constraints proposed by *Alaudah & AlRegib (2017).* We demonstrated that with a clean dataset, LANDMASS from the Netherlands North Sea F3 block, a good match of pixel-level label prediction on fault and salt could be produced. The constraints to apply Ovation's dataset are similar texture selection and generating ground truths for validation. The project outcomes could be saved to png masks and ought to help the company reduce the effort and time put on manual structure labelling.

7.1 Future Work

Accuracy improvement

The accuracy of the work result should be improved. After applying NMF, variable `classified_all` contains all the labelled pixels' position. One possible approach to increase the overlapped region between predicted and the ground truth using `classified_all` is:

1. Use Canny Edge or Sobel filter to detect seismic structure edges for form a polygon
2. Exclude any coordination that is outside the polygon
3. Filled the polygon

Other possible method could be using a density-sensitive kernel to find out the area with the most highlighted features. Then dilate the highlighted area using `binary_dialtion()` function in SciPy package to expand the label shape and fill the holes.

Code Efficiency

Slow convergence rate of sparsifying W by `NMF_Sparse.sparcify_columns(…)` and `NMF_Sparse.projection(…)` is a huge problem for the algorithm. More efficient code structure with stronger computation property could be introduced to save runtimes.

Apply output masks to train CNN model

The output binary masks are segmentation map for bounding boxes. No additional positional information will be output. If the output results are satisfactory on Ovation's data, we can use them for training purpose. `label_pixel.save_toPath(…)` saves masks following the order of box samples in data matrix X. A possible approach for using the masks on Ovation's data is:

1. Import original image from urls. Create a same-size blank image for each imported image.
2. Re-assign the 'True' mask value. If 3 classes are imported, then Class 1 mask should contain 0 and 1; Class 2 mask should contain 0 and 2; Class 3 mask should contain 0 and 3, etc...
3. Project the mask onto the blank image at corresponding bounding box position using coordination given by JSON files. The blank image then becomes a big mask with many classes of pixel-level labels.
4. Use the original image along with the mask as labels to train the CNN model.

Bibliography

Alaudah, Y. & AlRegib, G. (2017) A weakly supervised approach to seismic structure labeling. *In: SEG Technical Program Expanded Abstracts 2017*. 17 August 2017 Society of Exploration Geophysicists. p.

Alaudah, Y., Gao, S. & AlRegib, G. (2018) Learning to label seismic structures with deconvolution networks and weak labels. *In: SEG Technical Program Expanded Abstracts 2018. 27 August 2018 Society of Exploration Geophysicists*. p.

Alfarraj, M., Alaudah, Y. & AlRegib, G. (2016) Content-adaptive non-parametric texture similarity measure. In: 2016 IEEE 18th International Workshop on Multimedia Signal Processing (MMSP). [Online]. September 2016 IEEE. p. Available from: doi:10.1109/mmsp.2016.7813338.

Ao, Y., Li, H., Zhu, L. & Yang, Z. (2019) A SCiForest based semi-supervised learning method for the seismic interpretation of channel sand-body. *Journal of Applied Geophysics.* [Online] 167, 51–62. Available from: doi:10.1016/j.jappgeo.2019.04.019.

Boutsidis, C. & Gallopoulos, E. (2008) SVD based initialization: A head start for nonnegative matrix factorization. Pattern Recognition. [Online] 41 (4), 1350–1362. Available from: doi:10.1016/j.patcog.2007.09.010.

CeGP. (2015) LANDMASS: large north-sea dataset of migrated aggregated seismic structures. [Online]. Available from: http://cegp.ece.gatech.edu/codedata/LANDMASS/default.htm.

Chevitarese D., Szwarcman D., Silva R. M. D, & Vital Brazil E. (2018) Seismic facies segmentation using deep learning. *In American Association of Petroleum Geologists* (AAPG).

Ding, C., He, X. & Simon, H.D. (2005) On the Equivalence of Nonnegative Matrix Factorization and Spectral Clustering. *In: Proceedings of the 2005 SIAM International Conference on Data Mining*. [Online]. 21 April 2005 Society for Industrial and Applied Mathematics. p. Available from: doi:10.1137/1.9781611972757.70.

Imageio contributors. (2019) *Imageio*. [Online]. Available from: https://pypi.org/project/imageio/2.6.1/

Kim, J., He, Y. & Park, H. (2013) Algorithms for nonnegative matrix and tensor factorizations: a unified view based on block coordinate descent framework. *Journal of Global Optimization*. [Online] 58 (2), 285–319. Available from: doi:10.1007/s10898-013-0035-4.

Kumar, P.C. & Sain, K. (2018) Attribute amalgamation-aiding interpretation of faults from seismic data: An example from Waitara 3D prospect in Taranaki basin off New Zealand. *Journal of Applied Geophysics.* 159, 52–68.

Luo, W.-P., Li, H.-Q. & Shi, N. (2016) Semi-supervised least squares support vector machine algorithm: application to offshore oil reservoir. *Applied Geophysics*. [Online] 13 (2), 406–415. Available from: doi:10.1007/s11770-016-0564-y.

Milosavljević, A. (2020) Identification of Salt Deposits on Seismic Images Using Deep Learning Method for Semantic Segmentation. *ISPRS International Journal of Geo-Information*. 9 (1), 24.

Paatero, P., Tapper, U. (1994) Positive matrix factorization: a non-negative factor model with optimal utilization of error estimates of data values. Environmetrics **5**(1), 111–126

Papandreou G., Chen L., Murphy K., Yuille AL. (2015) Weakly and semi-supervised learning of a DCNN for semantic image segmentation. CoRR abs/1502.02734, 1502.02734

Pedregosa et al.(2011) Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research, 12, pp. 2825-2830.

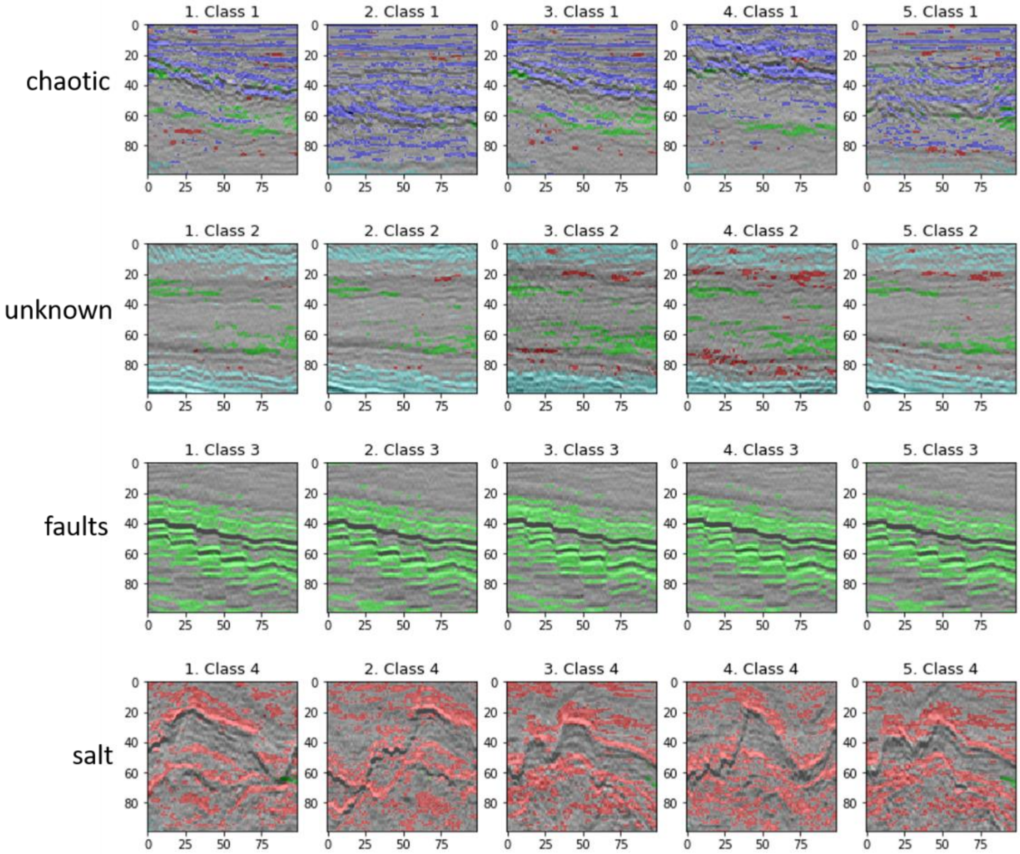
Shafiq, M.A., Wang*, Z., Amin, A., Hegazy, T., et al. (2015) Detection of Salt-dome Boundary Surfaces in Migrated Seismic Volumes Using Gradient of Textures. *In: SEG Technical Program Expanded Abstracts 2015*. 19 August 2015 Society of Exploration Geophysicists. p.

Stéfan van der Walt, Johannes L. Schönberger, Juan Nunez-Iglesias, François Boulogne, Joshua D. Warner, Neil Yager, Emmanuelle Gouillart, Tony Yu and the scikit-image contributors. (2014). *scikit-image: Image processing in Python*. [Online]. PeerJ 2: e453. Available from: https://doi.org/10.7717/peerj.453

Zhang, L., Quieren, J. & Schuelke, J. (2001) Chapter 10 Self-Organizing Map (SOM) network for tracking horizons and classifying seismic traces. *In: Computational neural networks for geophysical data processing*. [Online]. Elsevier. pp. 155–170. Available from: doi:10.1016/s0950-1401(01)80024-0.

Appendix A: Sample Classification Result in Colour Mask

Experiment on LANDMASS: Sample Classification Result in Colour Mask
Parameter: Sparsity = 0.5125, alpha = 0.0575
Blue: chaotic. Green: faults. Red: salt domes. Light blue: unknown. Gray: background.



Appendix B: To-be-saved Binary Mask Samples

Experiment on LANDMASS: Sample Output Binary Mask
Parameter: Sparsity = 0.5125, alpha = 0.0575
Row in order: chaotic, unknown, faults, salt domes