

GEOPHYSICS®

SEISMIC FAULT DETECTION WITH CONVOLUTIONAL NEURAL NETWORK

Journal:	<i>Geophysics</i>
Manuscript ID	GEO-2017-0666.R2
Manuscript Type:	Technical Paper
Keywords:	seismic attributes, artificial intelligence, faults, interpretation
Area of Expertise:	Seismic Attributes and Pattern Recognition, Interpretation Methods

SCHOLARONE™
Manuscripts

4 SEISMIC FAULT DETECTION WITH CONVOLUTIONAL NEURAL NETWORK
51 Wei Xiong¹, Xu Ji², Yue Ma¹, Yuxiang Wang¹, Nasher M. BenHassan², Mustafa N. Ali²
2 and Yi Luo²3
4 ¹Aramco Beijing Research Center, Aramco Asia, Beijing, China. Email:
5 xiong.wei@aramcoasia.com, yue.ma@aramcoasia.com, yuxiang.wang@aramcoasia.com;6
7 ²EXPEC Advanced Research Center, Saudi Aramco, Dhahran, Saudi Arabia. Email:
8 xu.ji@aramco.com; nasher.benhasan@aramco.com; mustafa.ali.2@aramco.com;
9 yi.luo@aramco.com.

10 Running Head: Seismic fault detection with CNN

11 Original paper date of submission: October 9th, 201712 Resubmitted: May 5th, 2018

ABSTRACT

Mapping fault planes using seismic images is a crucial and time-consuming step in hydrocarbon prospecting. Conventionally, this requires significant manual efforts that normally go through several iterations to optimize how the different fault planes connect with each other. A broad range of techniques have been developed to automate this process like seismic coherence estimation, edge detection and ant-tracking, to name a few. However, these techniques do not take advantage of the valuable experience accumulated by the interpreters. We have developed a method that uses convolutional neural network (CNN) to automatically detect and map fault zones using 3D seismic images in a similar fashion to the way done by interpreters. This new technique is implemented in two steps: training and prediction. In the training step, a CNN model is trained with annotated seismic image cubes of field data, where every point in the seismic image is labeled as fault or non-fault. In the prediction step, the trained model is applied to compute fault probabilities at every location in other seismic image cubes. Unlike reported methods in the literature, our technique does not require pre-computed attributes to predict the faults. We verified our approach on both synthetic and field datasets. We clearly demonstrated that CNN-computed fault probability outperforms that obtained using the coherence technique in terms of exhibiting clearer discontinuities. With the capability of emulating human experience and evolving through training using new field datasets, deep learning tools manifest huge potential in automating and advancing seismic fault mapping.

INTRODUCTION

Seismic fault detection and characterization is a crucial and time-consuming task in seismic interpretation (Bahorich and Farmer, 1995; Aqrawi and Boe, 2011). The discontinuities need to be hand-picked from raw seismic images, which may take weeks to months on a typical-size seismic volume by an experienced interpreter. The picking results are highly dependent on the experience of interpreters, and so human-bias is inevitable. Many tools have been developed to assist interpreters for this labor-intensive task, such as coherence feature extraction from the image (Bahorich and Farmer, 1995; Luo et al., 1996; Dorn, 1998; Wu 2017), automatic tracking of the discontinuities (Admasu et al., 2006) and automatic fault extraction (Dorn et al., 2005). Many of them are deterministic methods, which suffer two types of limitations. First, such methods cannot adapt easily to various kinds of discontinuities in different seismic images. For instance, coherence technology is less sensitive to faults that are changing gradually, i.e., not sharp. Second, deterministic methods themselves cannot learn or evolve based on interpreters' experience in a systematic manner. Therefore, the picking quality will always be affected by human bias. Picking results are relying heavily on the best interpreters' knowledge and experience, which disappear when they retire. New interpreters therefore need to be continually trained to learn from the experienced ones.

Machine learning (especially deep learning) technologies are powerful for mining features or relationships from data, which makes them quite suitable for learning from human experience (Fulkerson et al., 1995; LeCun et al., 2015; Schmidhuber, 2015). Many novel data-driven technologies have been developed in the field of deep learning to perform big-data analytics automatically. One of the most popular deep learning technologies is the convolutional neural network (CNN), with successful applications in image recognition and classification (LeCun and Bengio, 1998; Krizhevsky et al., 2012). CNNs have been intensively and broadly used in

identifying faces, objects, traffic signs and recognizing speeches, besides powering vision in robots and self-driving cars (Lawrence et al., 1997; Abdel-Hamid et al., 2014; LeCun et al., 2010; Bojarski et al., 2016). For example, they can take an image as an input and tell us what is in the image. The state-of-the-art CNNs can recognize images even better than humans. Deep learning technologies are showing great potential in many scientific fields other than computer science. There is also increasing interest in applying machine learning or deep learning technologies to seismic data processing and interpretation, such as seismic phase classification (Ramirez Jr and Meyer, 2011), data interpolation (Jia and Ma, 2017), as well as geophysical feature extraction (Huang et al., 2017; Chehrazi et al., 2013). Some recent work has proposed new approaches based on deep learning technologies for automatic geophysical feature extraction such as fault detection from pre-migrated data with 2D and 3D synthetic tests (Zhang et al., 2014; Chehrazi et al., 2013; Araya-Polo et al., 2017). This work shows promising results from applying deep learning technologies in geophysics.

In this paper, we introduce the CNN for detecting geologic faults from 3D seismic images for practical applications. A CNN model is trained to detect faults from raw seismic images sampled on 3D grids or voxels in a supervised learning approach. Real data from 7 annotated seismic cubes are used to generate the training dataset with one seismic cube used for validation. We first demonstrate the capability of the trained model for fault detection by applying it to a synthetic seismic image cube with artificial faults. For real data tests, the trained network is then applied to a different seismic cube that the network has never seen in the training process. The fault probability derived from the CNN shows similar but better highlighting of seismic discontinuities than seismic coherence.

METHODS AND RESULTS

Problem formulation and training dataset

We formulate the seismic fault detection problem as an image classification task, which takes as input the seismic image patches around a certain central point in the seismic image cube. The output is the binary fault or non-fault classification result for the central point. For a central point O in the 3D seismic cube, we extract 2D slices with dimension 24×24 along in-line, cross-line and time directions respectively. Those three slices form a $24 \times 24 \times 3$ input data sample x for the network, as depicted in Figure 1. Note that the input of the network has three channels, which is similar as conventional image-classification problems with RGB colored images as inputs. Meanwhile, the central point O is associated with a label y denoted by fault (1) or non-fault (0) indicating whether a picked fault is present or not. The input data x associated with a true label y is called a sample denoted as (x, y) in the training dataset. The ground-truth labels can be obtained from interpreters picking or auto-picking.

The network takes an input x and outputs the probability of the center point being a fault, which is denoted as fault probability for simplicity. The term fault probability is widely used to highlight seismic faults although the calculation methods of probability might be different from our paper (Wu and Hale, 2016; Wu, 2017; Neff et al., 2000). It can be used as a seismic attribute to highlight discontinuities in the images, which will be discussed later. To be more specific, the network classifies a point O (center of the input x) to be label y with a probability of $p(Y = y|x)$. For a batch of n samples $\{(x_i, y_i), i = 1, \dots, n\}$ from the training dataset, the network is trained to minimize the cross-entropy objective function

$$L = -\frac{1}{n} \sum_{i=1}^n \log p(Y = y_i|x_i) \quad (1)$$

The training dataset can be generated based on faults picked by interpreters or auto-picking algorithms. In this paper, we classified eight different 3D seismic image cubes of real data using a skeletonized-seismic-coherence-based auto-picking method (Qi et al., 2017). After annotation, every point in the training cubes is labeled as fault or non-fault using a threshold strategy: the points with skeletonized coherence value larger than a predefined threshold is classified as fault, others are non-fault. One of the training cubes and its annotation result is shown in Figure 2. The cube is from the GeoFrameTM reservoir characterization software where it is used for demonstrations of seismic attribute calculations on real data (Schlumberger, 2017). Other real data cubes are from Saudi Aramco. All the cubes share the same spatial grid sizes of 25 meters for inline and cross-line axes, while the vertical time sampling interval is either 0.002 or 0.004 seconds. We keep the time sampling interval as it is for real data to show the adaptability and robustness of the neural network. This is similar to the classification of dogs and cats using images with varying sizes and resolutions. As long as the training dataset covers a reasonable distribution of the samples, a successfully trained network should be capable of recognizing samples from the same distribution. Despite this, it is possible that the network would fail for seismic cubes with the spatial or time sampling intervals far from those in the training datasets.

One of the eight cubes is randomly selected for validation and will not be seen by the network during the training process, while the other seven cubes are used to generate the training dataset. The training dataset is constructed by randomly selecting 50000 points from each cube labeled as faults (which is approximately 0.2% of all fault points in the cubes), and another 50000 points labeled non-fault. Due to the nature of the seismic image, the number of non-fault points is much larger than fault points in the cubes. To make the likelihood of observing both

1
2
3
4 classes equal, we intentionally select the same number of fault and non-fault training samples.
5
6 Overall, there are 0.7 million samples in the training dataset. Since it does not really matter
7 which space axis is defined as inline or cross-line, we also use data augmentation during training
8 to exchange inline and cross-line axes for every sample.
9
10
11
12
13 Figure 3 shows some representative fault and non-fault samples randomly selected from the
14 training dataset. While it is not easy for human eyes to distinguish every single fault and non-
15 fault sample, we can see a systematic difference between the two classes. Non-fault samples
16 show more continuous seismic events. A validation dataset of the same size is constructed in the
17 same way as the training dataset, which is used to monitor the training process and determine the
18 termination of training.
19
20
21
22
23
24
25
26
27
28
29
30 **CNN model and Training**

31 We use a convolutional neural network for the image classification task as described in the
32 last section. The high-level architecture of the network is shown in Figure 4. The network takes
33 three seismic image slices of size 24×24 as an input and outputs a label prediction for the center
34 point. Before the input is fed into the network, it is normalized by subtracting the mean and
35 dividing by the standard deviation. Similar to the classical CIFAR-10 classification problem
36 (Krizhevsky and Hinton, 2009; Google, 2017), the network consists mainly of two convolutional
37 (Conv) layers and two fully-connected (FC) layers followed by a softmax classifier, which gives
38 the label prediction output (Bishop, 2006). The Conv layers both have 64 filters with sizes of
39 5×5. There are 384 and 192 feature maps in the two FC layers, respectively. After every Conv
40 layer and FC layer, we applied rectified linear activation (ReLU). Max-pooling with both size
41 and stride being 2×2 and local response normalization (LRN) are used after both Conv layers.
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60

The final softmax classifier produces a probability indicating the likelihood of a fault being presented in the center point of the input. The prediction of the label is then generated by applying a threshold (0.5) to the output likelihood values, such that the one above the threshold would be considered to be a fault location, while those below would not.

We train the network from scratch by initializing the weights of all the layers as in the CIFAR-10 tutorial of Tensorflow (Google, 2017; Abadi et al., 2016). We use the gradient decent optimizer implemented in Tensorflow with the default parameters and the initial learning rate being 0.1. The training samples are randomly shuffled before they are fed into the network for every epoch, which is a critical step to obtain good network performance. The model is saved when the error (or loss) function [equation (1)] reaches a plateau during the optimization process, and it is then validated with the validation dataset. The saved model achieved classification accuracies of about 73% on both the training and validation datasets.

CNN results vs. Coherence

After training, we first apply the trained model for predicting faults in a simple 3D synthetic seismic cube. The synthetic cube is constructed by extending a single trace from a training cube horizontally along the inline axis, with artificial discontinuities along three “perfect” fault lines with different slopes; the 2D section (cf. Figure 5a) is further extended along the cross-line with 5 degrees rotation along the time axis. The testing data for the CNN model is then extracted from the synthetic cube.

The model obtains 99% classification accuracy for this synthetic cube. The fault probability result for an inline section is shown in Figure 5b. It is clear that the three faults are successfully

detected by the network although it has never seen this kind of artificial fault during training. There are also some horizon footprints in the fault probability map (values below 0.4). If we look carefully, the footprints are weak while the seismic events are strong in the image. Our tests show that such artifacts can be removed by training the network with learning samples from the synthetic cube. We have chosen to accept some of the structural footprint after consulting with interpreters who noted that such “imperfections” are indeed helpful for them to distinguish true and false fault probabilities. We may need to address this issue for automatic fault picking using machine learning, which is a natural extension of the current project. The successful detection of the “perfect” fault indicates that the CNN model learned to distinguish fault from non-fault zones.

The trained model is next applied to the testing cube comprising a 3D seismic image volume of real data as shown in Figure 6. The cube size is 1000×655×1083. The top, front and side panels are showing time section ($T=312$), inline section ($Y=423$) and cross-line section ($X=417$), respectively. Locations of the sections are marked by the black lines in the figure. For this real data cube, the CNN obtains about 74% classification accuracy; close to that obtained for training and validation datasets.

The performance of the trained CNN model can be further examined by comparing the output fault probability with seismic coherence. Fault probability cube output by the CNN is shown in Figure 7a, along with results from a seismic coherence cube (Figure 7b). Seismic coherence is a well-known and widely used attribute to highlight discontinuities in seismic image (Bahorich and Farmer, 1995). As we can see in Figure 7 and Figure 8, the CNN results show a higher resolution compared to the coherence volume with seismic faults as well as channels being clearer in the CNN results. Recalling that the fault probability calculation is made

independently for different points, the clear continuous outlines of discontinuities in the fault probability images show that the trained network performs robustly in the presence of noise.

The computational time of our method is relatively high because we need to calculate the fault probability of each location in 3D image cube. For one calculation, the trained model is fed with the sample of a certain location, and output the fault probability. In our implementation, one such calculation took approximately 10 milliseconds with one CPU core. Since the calculations of different location are independent, they can be implemented in parallel. For the 3D image cube of size $1000 \times 655 \times 1083$ as mentioned in our field data test, it took about 2.5 hours to obtain the fault probability result using a computer cluster with 20 nodes (40 CPU cores in each node). So our method is still practical even without possible improvements by reducing duplicate calculations in adjacent locations.

DISCUSSIONS

Automating seismic interpretation using artificial intelligence technologies has long been one of the most interesting topics in oil and gas industry (Henderson, 2016). Some recent works have shown good performance of neural networks for identifying faults with a variety of seismic attributes on synthetic data (Meldahl et al., 2001; Huang et al., 2017; Chehrazi et al., 2013) such as local dip, curvature and fault likelihood, and using these attribute as inputs to the network for training. Our strategy of using raw seismic images is more straightforward, and has been demonstrated with both synthetic and real data.

To classify a certain point in a seismic cube as fault/non-fault, we construct the sample with three orthogonal slices of seismic images as the input to the network. It may be more natural to

use 3D blocks as the representative sample in seismic cubes. This approach will be tested with a re-designed neural network using 3D convolutional layers. The 3D convolution requires several orders of magnitude more computation power, therefore, the current proposed approach can be chosen as a practical technique for the next several years.

The strategy of transferable learning (Pan and Yang, 2010; Yosinski et al., 2014) can also be used to further enhance the performance of the network on different seismic cubes. A trained network can be adapted to new seismic cubes by further training with samples from the new cube. Interpreters just need to pick and annotate a small amount of samples from the new cube. As we have noted earlier, the range of spatial sampling sizes and time steps in our tests are limited. Although transferable learning might help to enlarge the capability of CNN on different seismic volumes, it may not succeed on all kinds of cubes. For seismic cubes with quite different spatial and time sampling rates from the training dataset, adaptive resampling might be necessary. Additional work needs to be done to investigate the applicability of a trained CNN model to different input samples.

CONCLUSION

We developed a CNN-based methodology for seismic fault detection using 3D seismic volumes. The fault probability result on real data outperformed the widely used seismic coherence for highlighting discontinuities in the migrated images. With less than one million training samples (0.2% of available Fault samples in the training cubes), the network learned to detect synthetic faults perfectly and faults in real data with quite good performance.

In this paper, we generated the training samples via auto-picking. This method may introduce bias into the network training. It would be ideal to use faults picked from the best

interpreters; this is what we are going to do as our next step. With human picks as input, the network will learn from many experienced interpreters and this should reduce human bias and help capture their experience. Over time, the network can evolve with more data/training samples from other interpreters. Indeed, deep learning technologies have shown to be able to approach and often exceed human judgments in image classification, speech recognition and healthcare applications. These advantages and the capability of the CNN we have shown in this work make it very attractive for deployment in future automatic seismic interpretation systems.

ACKNOWLEDGMENTS

The authors would like to thank Saudi Arabian Oil Company (Saudi Aramco) for its support and permission to publish this paper.

REFERENCES

- Abadi, M., A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, and M. Devin, 2016, Tensorflow: Large-scale machine learning on heterogeneous distributed systems: arXiv preprint arXiv:1603.04467.
- Abdel-Hamid, O., A.-r. Mohamed, H. Jiang, L. Deng, G. Penn, and D. Yu, 2014, Convolutional neural networks for speech recognition: IEEE/ACM Transactions on audio, speech, and language processing, **22**, no. 10, 1533–1545. doi: 10.1109/TASLP.2014.2339736.
- Admasu, F., S. Back, and K. Toennies, 2006, Autotracking of faults on 3D seismic data: Geophysics, **71**, no. 6, A49–A53. doi: 10.1190/1.2358399.

Aqrawi, A. A., and T. H. Boe. 2011, Improved fault segmentation using a dip guided and modified 3D Sobel filter: 81th Annual International Meeting, SEG, Expanded Abstracts, 999-1003. doi: 10.1190/1.3628241.

Araya-Polo, M., T. Dahlke, C. Frogner, C. Zhang, T. Poggio, and D. Hohl, 2017, Automated fault detection without seismic processing: *The Leading Edge*, **36**, no. 3, 208-214. doi: 10.1190/tle36030208.1.

Bahorich, M., and S. Farmer, 1995, 3-D seismic discontinuity for faults and stratigraphic features: The coherence cube: *The Leading Edge*, **14**, no. 10, 1053-1058. doi: 10.1190/1.1437077.

Bishop, C. M, 2006, Pattern recognition and machine learning: Springer.

Bojarski, M., D. Del Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. D. Jackel, M. Monfort, U. Muller, and J. Zhang, 2016, End to end learning for self-driving cars: arXiv preprint arXiv:1604.07316.

Chehrazi, A., H. Rahimpour-Bonab, and M. R. Rezaee, 2013, Seismic data conditioning and neural network-based attribute selection for enhanced fault detection: *Petroleum Geoscience*, **19**, no. 2, 169-183. doi: 10.1144/petgeo2011-001.

Dorn, Geoffrey, 1998, Modern 3-D seismic interpretation: *The Leading Edge*, **17**, no. 9, 1262-1262. doi: 10.1190/1.1438121.

Dorn, Geoffrey, James, H., Dopkin, D., Payne, B., 2005, Automatic fault extraction in 3D seismic interpretation: 67th Conference and Exhibition, EAGE, Extended Abstract.

Fulkerson, B., D. Michie, D. J. Spiegelhalter, and C. C. Taylor, 1995, Machine learning, neural and statistical classification: *Technometrics*, **37**, no. 4, 459. doi: 10.2307/1269742.

Google, 2017, Tensorflow tutorials: convolutional neural networks,

https://www.tensorflow.org/tutorials/deep_cnn, accessed 22 August 2017.

-
- Hale, D, 2012, Fault surfaces and fault throws from 3D seismic images: 82th Annual International Meeting, SEG, Expanded Abstracts, 1-6. doi: 10.1190/segam2012-0734.1.
- Henderson, J., 2016, An intelligent approach to seismic interpretation: GEO ExPro, **13**, no. 3.
- Huang, L., X. Dong, and T. E. Clee, 2017, A scalable deep learning platform for identifying geologic features from seismic attributes: The Leading Edge, **36**, no. 3, 249-256. doi: 10.1190/tle36030249.1.
- Jia, Y., and J. Ma, 2017, What can machine learning do for seismic data processing? An interpolation application: Geophysics, **82**, no. 3, V163-V177. doi: 10.1190/geo2016-0300.1.
- Krizhevsky, A., and G. Hinton, 2009, Learning multiple layers of features from tiny images: M.S. thesis, University of Toronto.
- Krizhevsky, A., I. Sutskever, and G. E. Hinton, 2012, Imagenet classification with deep convolutional neural networks: Advances in neural information processing systems, **25**, no. 2, 1097-1105.
- Lawrence, S., C. L. Giles, A. C. Tsoi, and A. D. Back, 1997, Face recognition: a convolutional neural-network approach: IEEE Transactions on Neural Networks, **8**, no. 1, 98-113. doi: 10.1109/72.554195.
- LeCun, Y., and Y. Bengio, 1998, Convolutional networks for images, speech, and time series, *in* Michael A. Arbib, ed., The handbook of brain theory and neural networks: MIT Press. 255-258.
- LeCun, Y., Y. Bengio, and G. Hinton, 2015, Deep learning: Nature, **521**, no. 7553, 436-444. doi: 10.1038/nature14539.

1
2
3 LeCun, Y., K. Kavukcuoglu, and C. Farabet, 2010, Convolutional networks and applications in
4 vision: IEEE International Symposium on Circuits and Systems: Nano-Bio Circuit
5
6 Fabrics and Systems, 253-256.

7
8
9
10 Luo, Y., W. G. Higgs, and W. S. Kowalik, 1996, Edge detection and stratigraphic analysis using
11
12 3D seismic data: 76th Annual International Meeting, SEG, Expanded Abstracts, 324-327.
13
14 doi: 10.1190/1.1826632.

15
16
17 Meldahl, P., R. Heggland, B. Bril, and P. d. Groot, 2001, Identifying faults and gas chimneys
18 using multiatributes and neural networks: The Leading Edge, **20**, no. 5, 474-482. doi:
19
20 10.1190/1.1438976.

21
22
23 Neff, D. B., J. R. Grismore, and W. A. Lucas, 2000, Automated seismic fault detection and
24 picking: U. S. Patent 6018498 A.

25
26
27 Pan, S. J., and Q. Yang, 2010, A survey on transfer learning: IEEE Transactions on knowledge
28 and data engineering, **22**, no. 10, 1345-1359. doi: 10.1109/TKDE.2009.191.

29
30
31 Qi, J., G. Machado, and K. Marfurt, 2017, A workflow to skeletonize faults and stratigraphic
32 features: Geophysics, **82**, no. 4, O57-O70. doi: 10.1190/geo2016-0641.1.

33
34
35 Ramirez Jr, J., and F. G. Meyer, 2011, Machine learning for seismic signal processing: Phase
36 classification on a manifold: 10th International Conference on Machine Learning and
37 Applications, 382-388.

38
39 Schlumberger, 2017, GeoFrame Seismic Attribute ToolKit Bundle,
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
<https://www.software.slb.com/products/geoframe/gf-geophysics/seismic-attribute-toolkit>,
accessed 22 August 2017.

Schmidhuber, J., 2015, Deep learning in neural networks: An overview: Neural networks, **61**,
85-117. doi: 10.1016/j.neunet.2014.09.003.

Wu, X., 2017, Directional structure-tensor-based coherence to detect seismic faults and channels:

Geophysics, **82**, no. 2, A13-A17. doi: 10.1190/geo2016-0473.1.

Wu, X., and D. Hale, 2016, Automatically interpreting all faults, unconformities, and horizons from 3D seismic images: Interpretation, **4**, no. 2, T227-T237. doi: 10.1190/INT-2015-0160.1.

Yosinski, J., J. Clune, Y. Bengio, and H. Lipson, 2014, How transferable are features in deep neural networks?: NIPS'14 Proceedings of the 27th International Conference on Neural Information Processing Systems, 3320-3328.

Zhang, C., C. Frogner, M. Araya-Polo, and D. Hohl, 2014, Machine-learning based automated fault detection in seismic traces: 76th Conference and Exhibition, EAGE, Extended Abstract. doi: 10.3997/2214-4609.20141500.

LIST OF FIGURES

Figure 1. A schematic of the CNN input which consists of inline, cross-line (xline) and time slices centered on the point *O*. For better visualization, the size of the slices in this figure is 100×100 which is larger than that used in the training and calculation process (24×24). The spatial grid size is 25 meters while the vertical time sampling interval is either 0.002 or 0.004 seconds for different cubes.

Figure 2. One of the training cubes and its annotation, (a) the seismic image cube with (b) every point annotated as fault (1, black) or non-fault (0, white).

Figure 3. Samples labeled as (a) fault and (b) non-fault. All samples are randomly selected from the training dataset, except the first row showing synthetic samples. Each sample is composed of front, side and top panels which are inline, cross-line and time slices, respectively. The size of each slice is 24×24 . The spatial grid size is 25 meters while the vertical time sampling interval is either 0.002 or 0.004 seconds for different cubes.

Figure 4. A schematic of the network architecture. Overall, the network contains two convolution layers (Conv) and two fully-connected (FC) layers followed by a softmax classifier, which gives the label prediction with probabilities. The squares and dashed lines on the images indicate the convolution and pool operations in the network, the circles are neurons. Red circle indicates the output unit for fault probability while blue indicates non-fault probability. Rectified linear activation unit (ReLU), max-pooling (Pool) and local response normalization (LRN) are also used in the network.

Figure 5. CNN testing results on a simple synthetic seismic data, (a) an inline section showing artificial seismic faults and (b) seismic faults probability obtained from CNN model. The section size is 376×376 . The spatial grid size is 25 meters while the vertical time sampling interval is 0.002 seconds.

Figure 6. The seismic image cube of real data for CNN testing. The cube size is $1000 \times 655 \times 1083$. The spatial grid size is 25 meters while the vertical time sampling interval is 0.002 seconds.

Figure 7. Real data example showing (a) the fault probability cube from the CNN prediction, compared with (b) the corresponding coherence cube. The cube size is $1000 \times 655 \times 1083$. The spatial grid size is 25 meters while the vertical time sampling interval is 0.002 seconds.

Figure 8. Time slices of fault probability [(a) and (c)] from CNN prediction, compared with seismic coherence [(b) and (d)], at two different time slices, T=312 (top row) and T=387 (bottom row), respectively. The section size is 655×1083 . The spatial grid size is 25 meters.

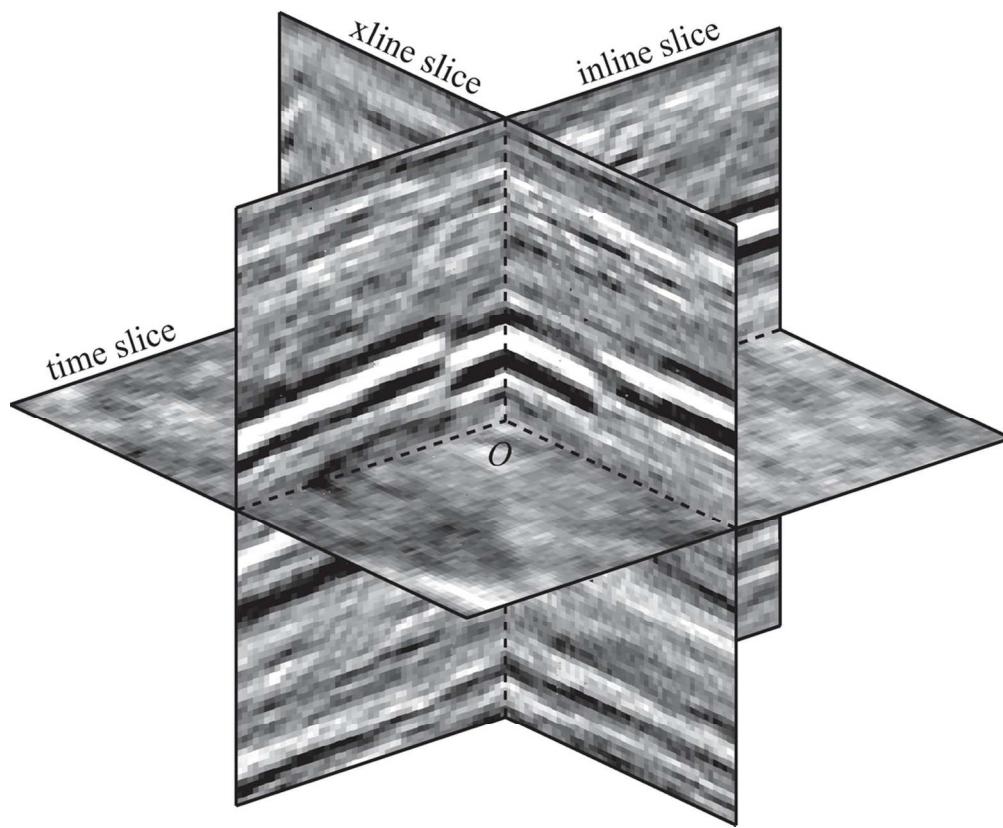


Figure 1. A schematic of the CNN input which consists of **inline, cross-line (xline) and time slices** centered on the point O. For better visualization, the size of the slices in this figure is 100×100 which is larger than that used in the training and calculation process (24×24). The spatial grid size is 25 meters while the vertical time sampling interval is either 0.002 or 0.004 seconds for different cubes.

99x81mm (300 x 300 DPI)

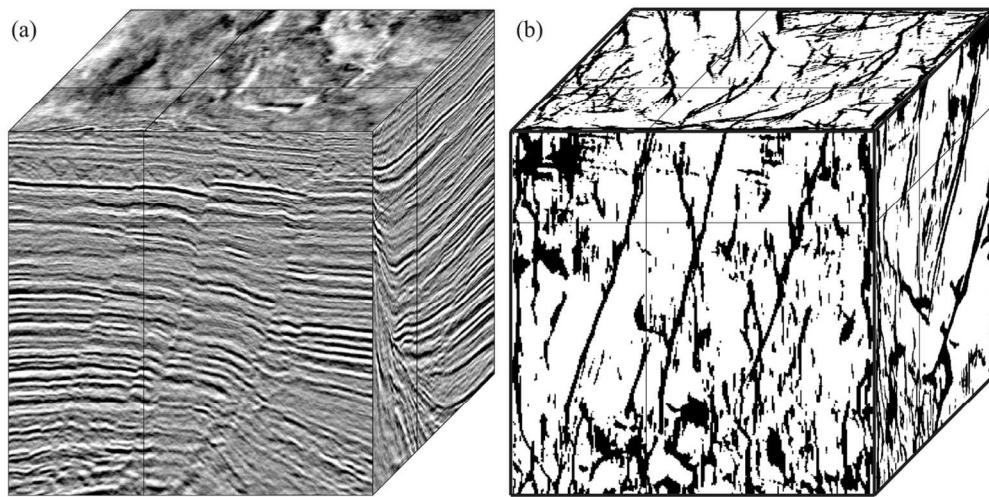


Figure 2. One of the **training cubes** and its annotation, (a) the seismic image cube with (b) every point annotated as fault (1, black) or non-fault (0, white).

120x60mm (300 x 300 DPI)

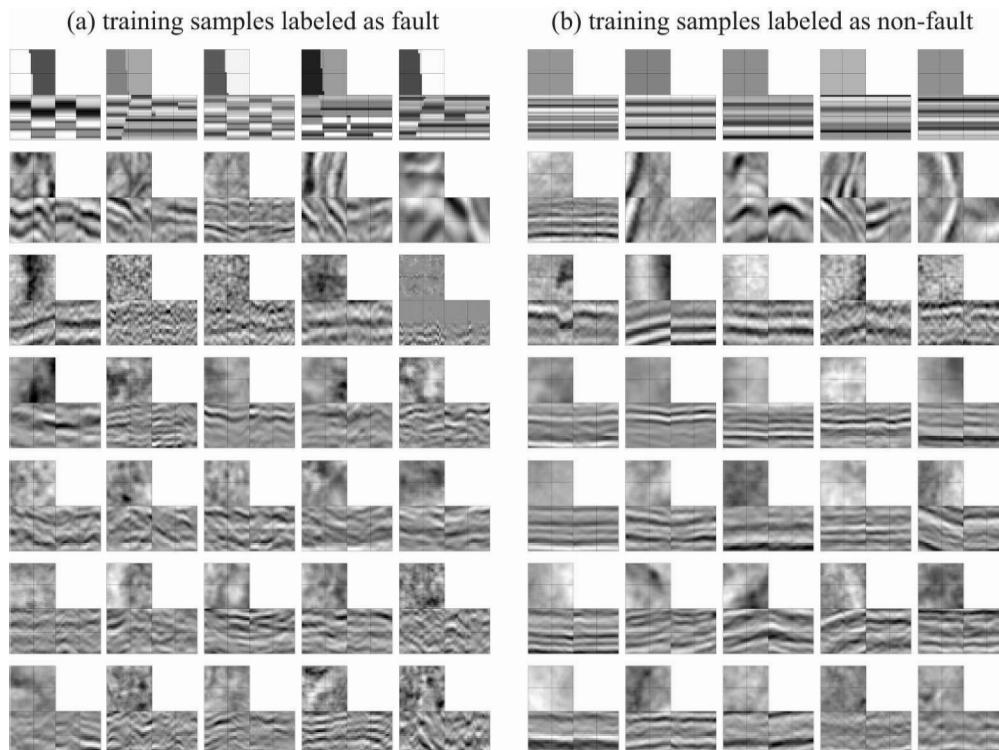


Figure 3. Samples labeled as (a) fault and (b) non-fault. All samples are randomly selected from the training dataset, except the first row showing synthetic samples. Each sample is composed of front, side and top panels which are inline, cross-line and time slices, respectively. The size of each slice is 24×24. The spatial grid size is 25 meters while the vertical time sampling interval is either 0.002 or 0.004 seconds for different cubes.

242x181mm (300 x 300 DPI)

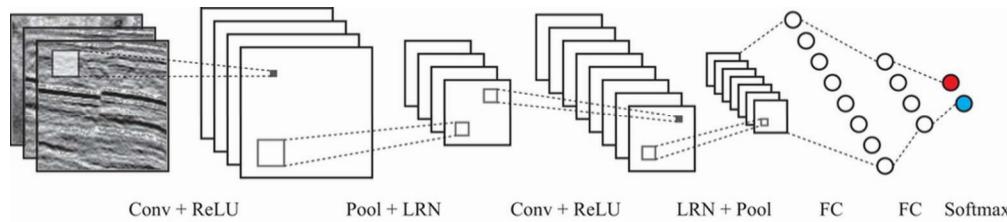


Figure 4. A schematic of the network architecture. Overall, the network contains two convolution layers (Conv) and two fully-connected (FC) layers followed by a softmax classifier, which gives the label prediction with probabilities. The squares and dashed lines on the images indicate the convolution and pool operations in the network, the circles are neurons. Red circle indicates the output unit for fault probability while blue indicates non-fault probability. Rectified linear activation unit (ReLU), max-pooling (Pool) and local response normalization (LRN) are also used in the network.

82x17mm (300 x 300 DPI)

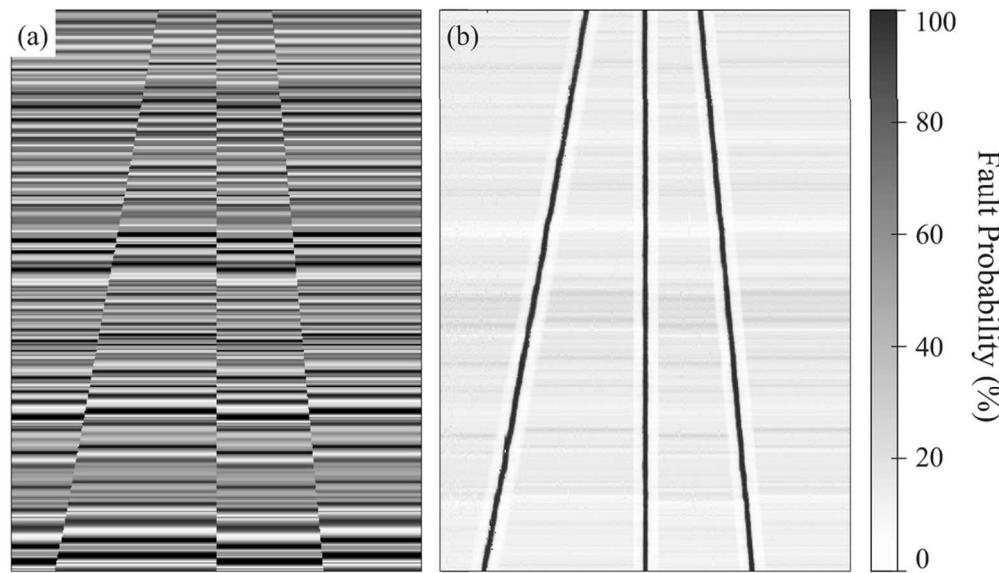


Figure 5. CNN testing results on a simple synthetic seismic data, (a) an inline section showing artificial seismic faults and (b) seismic faults probability obtained from CNN model. The section size is 376×376. The spatial grid size is 25 meters while the vertical time sampling interval is 0.002 seconds.

100x57mm (300 × 300 DPI)

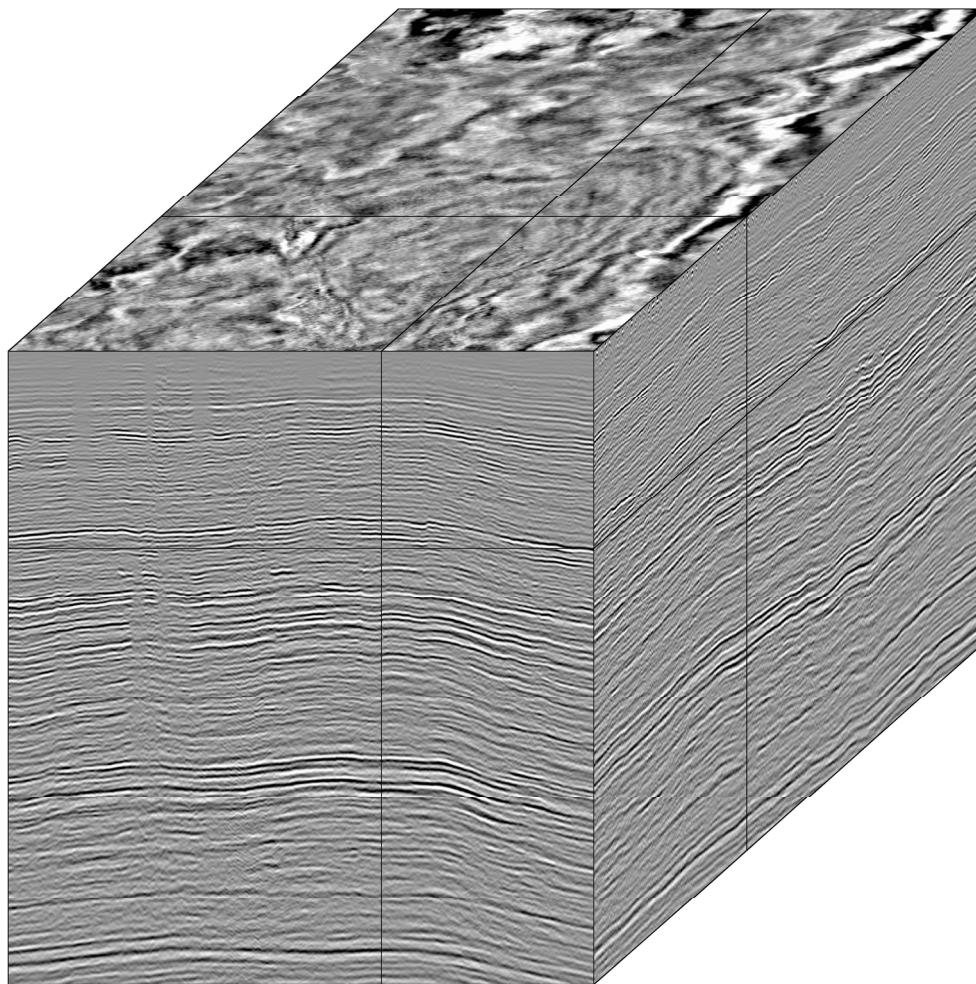


Figure 6. The seismic image cube of real data for CNN **testing**. The cube size is $1000 \times 655 \times 1083$. The spatial grid size is 25 meters while the vertical time sampling interval is 0.002 seconds.

200x200mm (300 x 300 DPI)

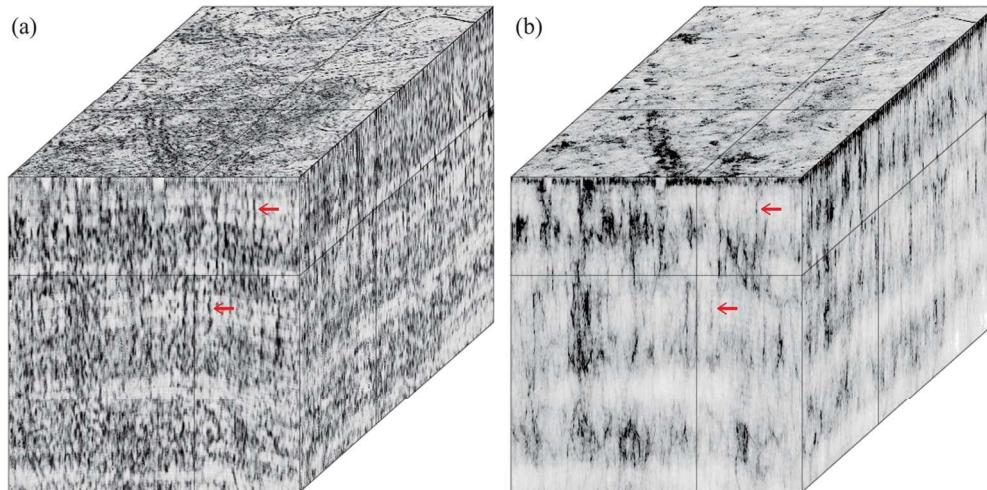


Figure 7. Real data example showing (a) the fault probability cube from the CNN prediction, compared with (b) the corresponding coherence cube. The cube size is $1000 \times 655 \times 1083$. The spatial grid size is 25 meters while the vertical time sampling interval is 0.002 seconds.

120x60mm (300 x 300 DPI)

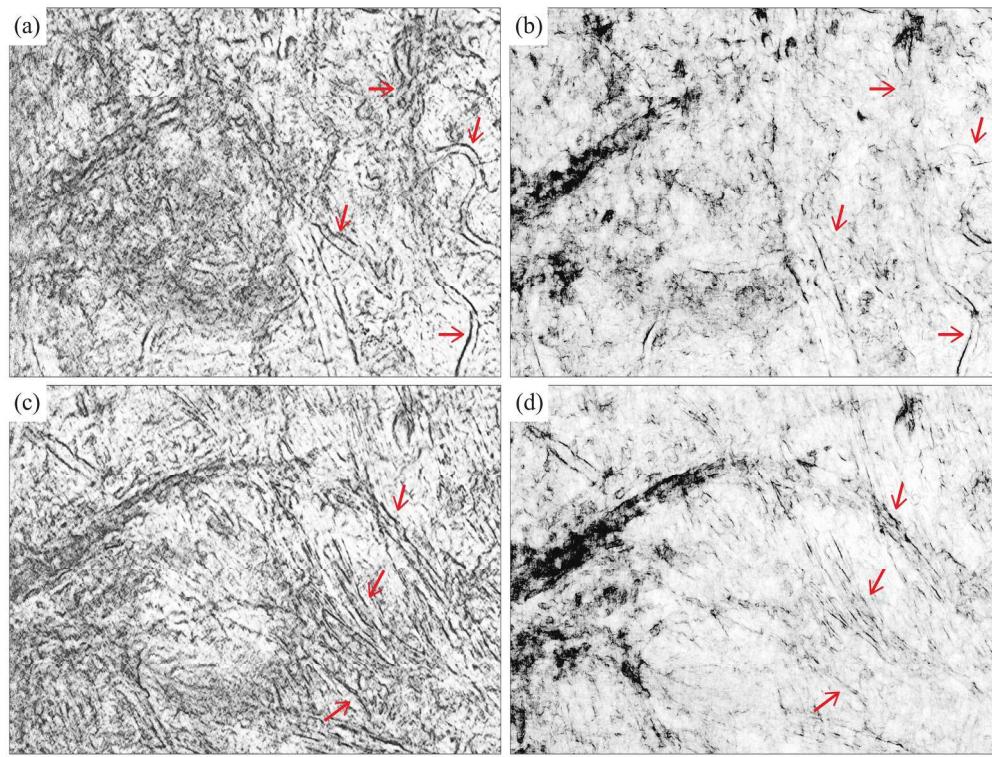


Figure 8. Time slices of fault probability [(a) and (c)] from CNN prediction, compared with seismic coherence [(b) and (d)], at two different time slices, T=312 (top row) and T=387 (bottom row), respectively. The section size is 655×1083. The spatial grid size is 25 meters.

200x151mm (300 x 300 DPI)

DATA AND MATERIALS AVAILABILITY

Data associated with this research are confidential and cannot be released.