

Applications of supervised deep learning for seismic interpretation and inversion

York Zheng¹, Qie Zhang², Anar Yusifov², and Yunzhi Shi³

<https://doi.org/10.1190/tle38070526.1>.

Abstract

Recent advances in machine learning and its applications in various sectors are generating a new wave of experiments and solutions to solve geophysical problems in the oil and gas industry. We present two separate case studies in which supervised deep learning is used as an alternative to conventional techniques. The first case is an example of image classification applied to seismic interpretation. A convolutional neural network (CNN) is trained to pick faults automatically in 3D seismic volumes. Every sample in the input seismic image is classified as either a nonfault or fault with a certain dip and azimuth that are predicted simultaneously. The second case is an example of elastic model building—casting prestack seismic inversion as a machine learning regression problem. A CNN is trained to make predictions of 1D velocity and density profiles from input seismic records. In both case studies, we demonstrate that CNN models trained from synthetic data can be used to make efficient and effective predictions on field data. While results from the first example show that high-quality fault picks can be predicted from migrated seismic images, we find that it is more challenging in the prestack seismic inversion case where constraining the subsurface geologic variations and careful preconditioning of input seismic data are important for obtaining reasonably reliable results. This observation matches our experience using conventional workflows and methods, which also respond to improved signal to noise after migration and stack, and the inherent subsurface ambiguity makes unique parameter inversion difficult.

Introduction

Interpretation of geologic features and inference of reservoir properties are key to the success of hydrocarbon exploration and production. Often the processes involved in extracting useful subsurface information from seismic data are labor intensive, time consuming, subjective, and computationally demanding. In seismic interpretation, accurate delineation of subsurface structures is a necessary and routine process. Automation of this task ensures timely delivery of interpretation products that can be used readily for prospect identification, well planning, reservoir modeling, and geohazard analysis. While many automated or semiautomated tools exist for picking seismic horizons, fault picking remains a laborious exercise. Attribute-based techniques such as semblance (Marfurt et al., 1998), coherency (Marfurt et al., 1999), and variance (Van Bemmel and Pepper, 2000) alleviate some of the pain points associated with manual and mundane aspects of fault picking. They can be considered as early machine learning approaches where faults are extracted from hand-crafted feature maps that are based on

attribute analyses. However, the image of a fault, as well as its surrounding sediments, can be very complicated in seismic data and may not be described adequately by a single or a few attributes alone. Past experiences also suggest that fault attributes can be sensitive to stratigraphic variations and noise in the data.

Before seismic interpretation, a rigorous data processing sequence is required to convert the raw seismic data into an image that represents the subsurface structures. The clarity and geometry of these structures are heavily influenced by the velocity model used during prestack depth migration. The migration velocity model building process is not trivial, due to the fact that estimating earth parameters from seismic data directly is a highly nonlinear and complex problem. Currently, most prestack seismic inversion techniques rely on local optimization schemes, during which an initial earth model estimate is updated iteratively to minimize the difference between observed and modeled seismic data. Techniques such as traveltime tomography and full-waveform inversion (FWI) have been proven to be effective for acoustic model building for many geologic scenarios, but each still has its own limitations and often requires human intervention (Kosloff et al., 1996; Jones, 2010; Virieux et al., 2014). Furthermore, large-scale applications of multiparameter inversion such as elastic model building remain challenging practically and computationally with conventional workflows.

An alternative approach can be considered. Deep learning can directly map the relationship between a seismic image and its corresponding label (e.g., a fault or a velocity model) where the feature maps are derived by machines automatically. The training process allows a deep learning algorithm to abstract a large amount of data in high dimensions and learn a complex mapping function. In particular, the convolutional neural network (CNN) is a widely used deep learning method proven to be state of the art for computer vision problems (Russakovsky et al., 2015) including image classification, localization, and segmentation. In recent years, CNN has been applied to a variety of seismic problems including interpretation and inversion.

In our case of fault interpretation, we generally follow the Wu et al. (2019b) approach, referred to as FaultNet, to generate 3D synthetic training data, train a CNN model, and then apply the model to our real seismic images. Training using synthetically generated data provides us with several advantages, which are discussed in more detail in the next section. Our results show that the CNN model can yield high-quality fault picks from different seismic images across various regions. From the FaultNet probability volume, fault “sticks” can then be extracted automatically, which greatly improves the efficiency of fault interpretation.

¹BP, Sunbury-on-Thames, UK. E-mail: york.zheng@bp.com.

²BP, Houston, Texas, USA. E-mail: qie.zhang@bp.com; yusia4@bp.com.

³University of Texas at Austin, Austin, Texas, USA. E-mail: yzshi08@utexas.edu.

In the case of velocity model building, we adopt the approaches of Zheng and Zhang (2018). To appreciate the difficulty in constructing a neural network that generalizes well to all possible geologic scenarios, we take a pragmatic approach by restricting the earth models to a specific layered geology with large vertical contrasts but small lateral variations. Synthetic shot records are generated from random 1D elastic model profiles with layer properties and thickness sampled from Gaussian and uniform distributions, respectively. The results show that all three elastic parameters (P-wave velocity, S-wave velocity, and density) can be estimated simultaneously in a highly efficient manner using a CNN model trained using these synthetic shot data and corresponding elastic

models. The predictions on real seismic data show a noticeable uplift in the inversion result from that of traveltome tomography.

Case 1: Fault interpretation

Method. There are two basic ways of using CNN to predict faults in a seismic image. One way is to treat it as an image segmentation problem (e.g., Wu et al., 2019a) where the output of the neural networks is a “mask” image with the same size as the input image (Figure 1a). Every pixel or voxel in the input image is classified, often by a binary label such as 0 for nonfault and 1 for fault. A neural net architecture like SegNet (Badrinarayanan et al., 2017) or U-net (Ronneberger et al., 2015) can be used for this application.

The segmentation approach demands more convolutional layers and hence more computation, but this can be cost effective because a large stride (step of window advance) can be used for a sliding window during inference.

An alternative way is to treat fault prediction as an image classification problem (e.g., X. Wu et al., 2018; Xiong et al., 2018; Wu et al., 2019b) where CNN classifies only a particular location (typically the center point) of an image as a fault or not (Figure 1b). An interesting fact is that although the CNN algorithm is not explicitly instructed which location of the image to classify, it finds out the location by learning from a large pool of training data. Other than predicting only a binary class (fault or nonfault), the Wu et al. (2019b) method shows that a CNN model can predict the orientation of a fault at the same time. Knowing that the dip and azimuth of a fault is important, it greatly helps to connect fault points and stack fault planes (X. Wu et al., 2018; Wu et al., 2019b) in the postprocessing step after inference. Our approach is an image classifier that predicts the fault dip and azimuth together with the fault probability only at the image center (Figure 1c).

Training. To train our CNN model, 3D synthetic data are generated by following the workflow of the Wu et al. (2019b) approach where six steps are taken: (1) generating a horizontal reflectivity model, (2) folding, (3) shearing, (4) faulting, (5) convolving with a wavelet, and (6) adding noise. To match the low-angle faults that occasionally occur in our field data, we set the range of fault dips in our synthetic training data to be from 13.5° to 88.5° instead of from 63° to 86° as used by Wu et al. (2019b).

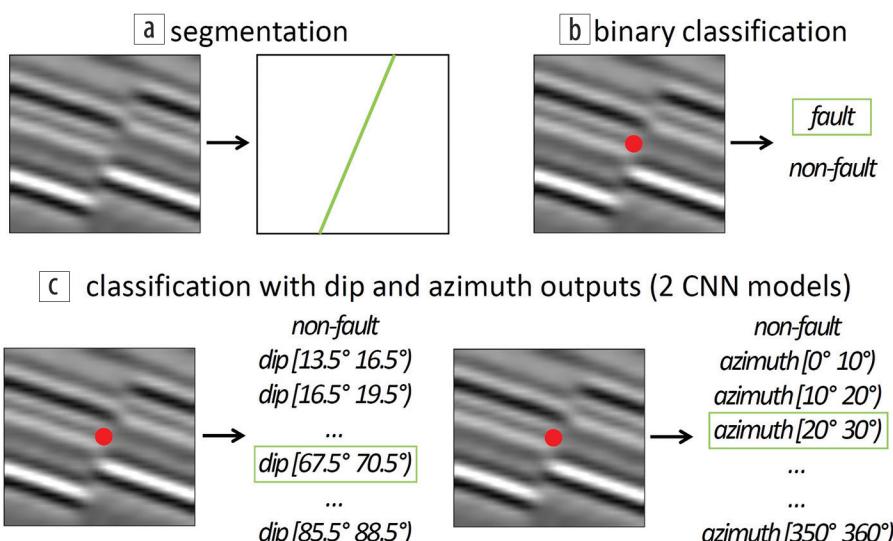


Figure 1. Various approaches for 3D fault labeling where (c) is our approach. A 3D image cube is used in actual processing, although a 2D slice is shown for illustration.

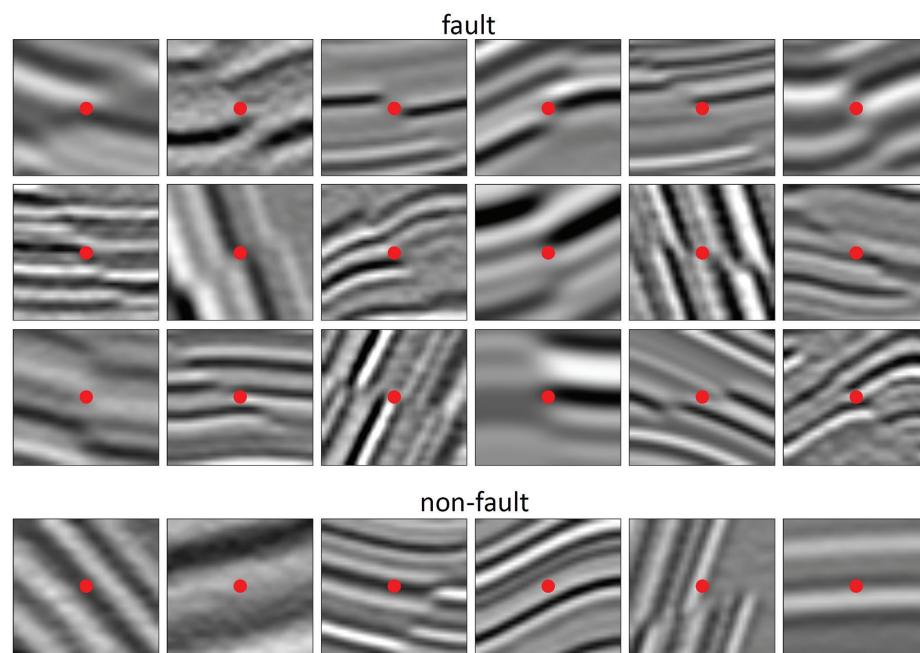


Figure 2. 2D slice examples extracted from 3D synthetic training data. Top three rows show examples of a fault at the image center. The bottom row shows that no faults pass through the center.

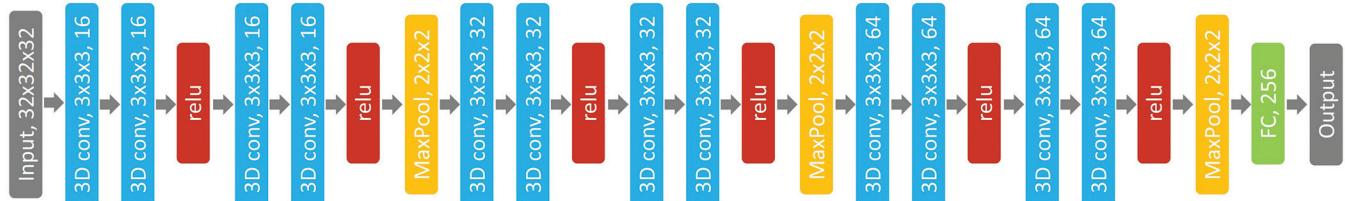


Figure 3. Our deep CNN architecture for both 3D fault dip and azimuth predictions. They only differ in the output layer and are trained separately.

Another parameter is the fault azimuth, which is left to span the full range of 0° to 360° . Figure 2 shows some examples of our synthetic training data for both fault and nonfault cases. Note in the bottom row: a fault not passing through the image center is still classified as a nonfault.

We use synthetic data for training in this study. Some benefits are: (1) no human labeling is required, (2) the difficulty of manually labeling fault dips and azimuths in field 3D data is avoided, (3) an unlimited number of training data and labels is possible, (4) a complete population of all possible fault dips and azimuths is guaranteed, (5) the ground truth labels are known, and (6) inaccuracy in manual picks following fault truncations is eliminated. The assumptions associated with our current synthetic training data include: no more than one fault in a 3D image cube, a linear local fault plane, and no fault plane reflectivity. Even with these assumptions, the training data set still covers a wide spectrum of fault geometries. Further generalizing our synthetic data and including field data for training is the subject of future study.

When predicting faults using a center point classifier (Figures 1b and 1c), we need to move a sliding window across the whole seismic image, typically voxel by voxel, which requires significant computation time. Therefore, we limit our 3D training data cube to $32 \times 32 \times 32$ samples. The center point of an image cube is labeled as a fault only if a fault plane passes through the center within a distance boundary of one sample and the fault slip is greater than one sample. We generate 100,000 3D image cubes for training and 10,000 for validation. Further increasing the training data size has negligible benefits on the performance. We also carefully design our synthetic training data so the number of training data in each classification category is balanced for the dip CNN (26 categories) and azimuth CNN (37 categories), respectively.

We employ two CNN models (Figure 3) to estimate fault dips and azimuths, respectively, instead of fitting both estimations in one CNN model as in Wu et al. (2019b). The two models are derived separately. They only differ in the output layer where the dip CNN has 26 output categories (a nonfault class plus 25 dip bins centered at 15° , 18° , 21° , ..., and 87° with a bin size 3°) and the azimuth CNN has 37 output categories (a nonfault class plus 36 azimuth bins centered at 5° , 15° , 25° , ..., and 355° with a bin size 10°). Both CNN models also predict whether the center of an image cube is a fault or not. If it is a fault, the dip or azimuth category is predicted at the same time (Figure 1c). We can claim an image center as a fault when both CNN models vote yes. Then we can assign the probability of the fault to be the average of the predicted dip and azimuth probabilities from those two models.

We double the number of convolutional layers in the X. Wu et al. (2018) and Wu et al. (2019b) CNN model to make our deep

CNN architecture (Figure 3). Our CNN consists of 12 3D convolutional layers using a uniform kernel size $3 \times 3 \times 3$. The channel number of convolutional layers starts at 16 and then doubles after every max pooling. A ReLU activation is applied after every two convolutional layers and a max pooling after every four convolutional layers. A fully connected (FC) layer with 256 neurons connects the convolutional layers and the output layer where a 50% dropout is applied after FC for regularization. In the output layer, a softmax classifier is used to output the probability associated with each category where the max probability indicates the predicted category. Our deep CNN model has approximately 1.6 million parameters and converges in less than 30 minutes using four GPUs. We also experimented with ResNet (He et al., 2016) using both basic block and bottleneck structures and various numbers of layers. However, we did not achieve a noticeably better performance than our CNN model.

Results. We used synthetic training data to derive a 3D deep CNN model for automated fault detection. It has been tested in 3D seismic images from the Gulf of Mexico (GoM), North Sea, offshore Mexico, Trinidad shelf, and Trinidad deepwater. The results show that this CNN model seems capable of producing consistently high-quality fault picks.

We share two field data results from our 3D FaultNet inference. Although 2D views are shown, our fault inference is processed in 3D. The first one is from a subsalt asset, Thunder Horse, in the deepwater GoM. Figures 4a and 4b show thinned and “fat” fault results, respectively, for two different lines. Thinned faults are derived solely from fat faults, and fat faults are generated by stacking fault planes that are described by 3D anisotropic Gaussian functions and oriented by the predicted fault dips and azimuths (Wu et al., 2019b). Our fault predictions seem robust and able to catch the majority of the human picks. At the same time, they also provide details of complex fault geometries such as relay faults. A depth slice comparison in Figure 4c shows that our predicted faults appear less noisy than the traditional coherency attribute.

Our second example is from a 3D ocean-bottom-node image (phase 2) in the Trinidad shelf. Figures 5a and 5c show our FaultNet probability results for two different lines derived from our deep CNN model. Figures 5b and 5d show the corresponding fault sticks (digitized fault lines) automatically extracted from our FaultNet probability volume using the automatic fault extraction tool in PaleoScan. The fault sticks look sharp and continuous, following fault truncations precisely. These fully automatically generated fault sticks can be viewed and edited in interpretation software. Although our fault predictions are not flawless, they can easily create a basis for any fault framework. Weeks to months of manual picking time may be saved if a set of dense fault sticks

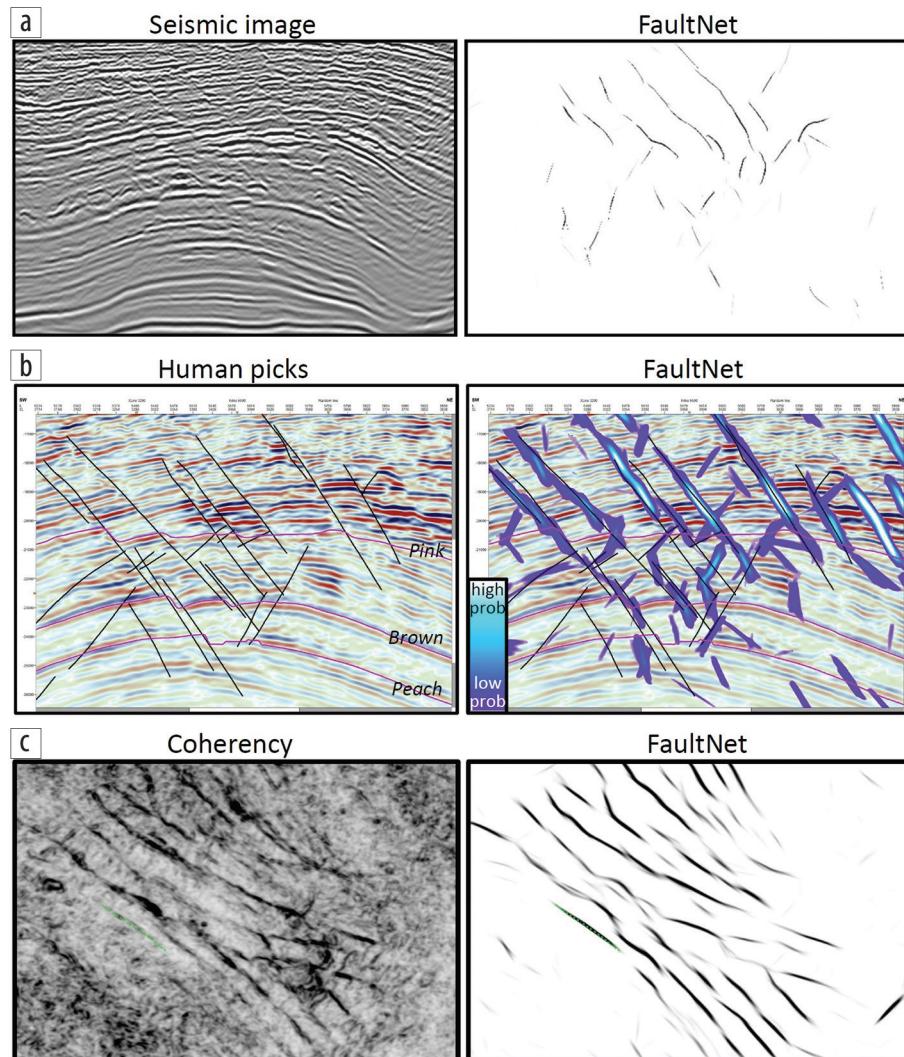


Figure 4. Thunder Horse (GoM) FaultNet results: (a) thinned faults, (b) hot display of fat faults compared to human picks, and (c) a comparison of coherency and FaultNet at a depth slice.

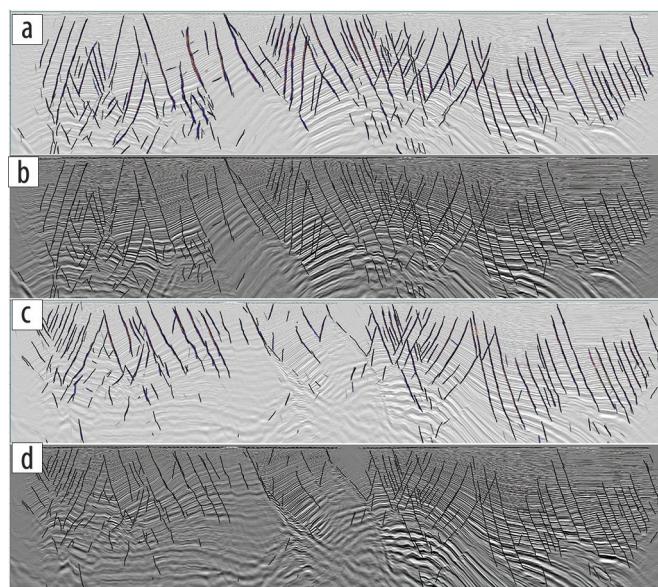


Figure 5. Trinidad (shelf) seismic images overlaid with fault predictions. (a) and (c) Our FaultNet probability results for two different lines derived from our deep CNN model. (b) and (d) The corresponding fault sticks (digitized fault lines) automatically extracted from our FaultNet probability results (a) and (c), respectively.

can be populated automatically and accurately in a large 3D image volume.

Case 2: Prestack seismic inversion

Method. Different approaches of using deep learning for seismic velocity inversion have been proposed in the past, and they largely fall into two categories. One approach is to embed the concepts of deep learning into a conventional inversion framework where components of the workflow such as forward modeling and gradient computation are replaced by deep learning algorithms (Richardson, 2018). A more straightforward approach, as shown in this study (Figure 6), is to replace the entire inversion by inference using a supervised deep learning model. Similar to the approaches of Araya-Polo et al. (2018) and Y. Wu et al. (2018), we choose a CNN model, but training is done using seismic gathers as the training data and 1D earth models as the labels. The trained CNN represents a nonlinear function that maps an input seismic record to an earth model. It is interesting to note that this approach is different from

conventional inversion methods in terms of optimization objectives. In our supervised learning, the cost function is based on the misfit of the model parameters, whereas it is expressed in terms of data misfit in conventional methods such as FWI or traveltime tomography.

Training a CNN model that generalizes across a large variety of geology is challenging and requires a massive set of training data (Zheng and Zhang, 2018). Therefore, we focus on a field-specific case study where the geology is more or less laterally consistent. Prior knowledge from a land survey is used where imaging challenges exist for traditional inversion methods due to the presence of large vertical contrasts and low-velocity zones. By constraining our earth models to represent the subsurface structure of a particular field, we can reduce the parameter space significantly, decrease the ambiguity of inferred velocity parameters, and limit the amount of training data used. Furthermore, by assuming a local 1D geology, we can efficiently build synthetic elastic profiles (V_p , V_s , and density) and simulate the corresponding elastic seismic data.

Training. Similar to FaultNet, our workflow here begins with the construction of a set of elastic synthetic training data. These include a collection of 1D earth models and their corresponding prestack seismic gathers. To create the synthetic earth models, we divide the subsurface section into 14 macrolayers and obtain the statistics from a set of true well data in a layer-by-layer manner. The statistics of each layer are characterized by a set of mean elastic property trends, standard deviations, and a layer thickness distribution. We choose normal distributions for each layer's elastic parameters and a uniform distribution for the layer thickness; and sample from them to create a suite of elastic models with different yet statistically consistent vertical profiles. Shot gathers are modeled using the reflectivity method (Kennett, 1983)

to simulate the full isotropic elastic response (particle velocity) recorded on the vertical geophone component. In total, the training data set consists of 10,000 synthetic examples. A further 2000 examples are used for validation and testing.

A CNN is constructed consisting of multiple 2D convolutional layers with 3×3 filter kernels. Unlike the classification in FaultNet, here inversion is treated as a regression problem. Therefore, we do not use a classifier for the output and instead introduce two FC layers at the end. Between sets of convolutional layers, a max-pooling layer and a dropout layer are used for downsampling and to control overfitting. A range of neural network parameters was tested, and it was found that network architecture and optimization hyperparameters have the most impact on the performance of the CNN model. As expected, the biggest computation cost is in the training, but it is still a

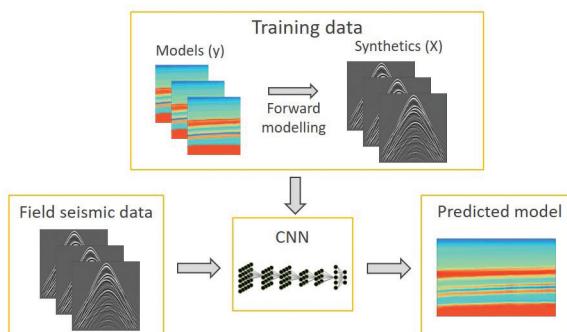


Figure 6. The proposed deep learning (CNN) workflow for prestack seismic inversion.

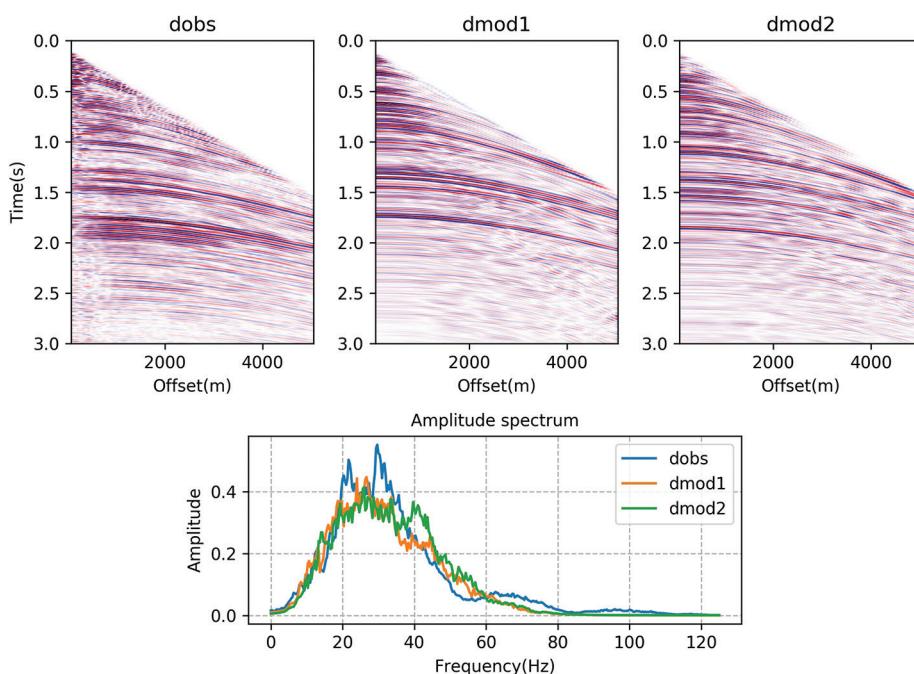


Figure 7. Comparison of a real CMP gather and two random synthetic gathers (top row) and their amplitude spectra (bottom row) after preprocessing.

fraction of the cost compared with conventional inversion means such as tomography or FWI. In total, training across four GPUs takes only about 30 minutes; prediction along a line of 1000 gathers on a single CPU takes just under 3 minutes.

For the application to real data, we find that some preconditioning of the input seismic gathers is required. Premigration common-midpoint gathers are used as inputs following a standard sequence of preprocessing steps that mainly consists of noise attenuation, statics corrections, gain corrections, trace regularization, and muting of some early arrivals. A time-varying gain is applied to the synthetic data to roughly mimic the amount of attenuation observed in the field data. To match the frequency bandwidth between the synthetic and field data, a bandpass filter with corner frequencies of 10 and 50 Hz is applied to the synthetic data. These preprocessing steps help mitigate some of the inherent differences between the field data and synthetic data, thus allowing a CNN model derived from synthetic data to be applicable for real data predictions. For visual inspection, a real CMP record and two randomly selected synthetic gathers are shown in Figure 7.

The comparison shows a general agreement between the field and synthetic data in both time and frequency domain after preprocessing.

Results. The performance of the CNN is tested on synthetic gathers first and then on real data. Results of three randomly selected synthetic test cases are compared with the true elastic logs (synthetic models) as shown in Figure 8. The predicted elastic profiles show a good match with the truth where both the background trends and the strong impedance contrasts are recovered well. It is worth noting that the predicted profiles appear to be blocky but smooth and not overfitted to the short wavelength Gaussian noise that was added to the synthetic models.

The predicted elastic model using real data along a 2D line is shown in Figure 9. For comparison, the predicted P-wave velocity model is shown next to that obtained from reflection traveltime tomography in Figure 10. The predicted P-wave velocity section contains more details compared with the conventional tomography result using the same data. It is also encouraging to see that the CNN result exhibits subtle structural variations laterally, which is not quite evident in the tomographic model. It should also be

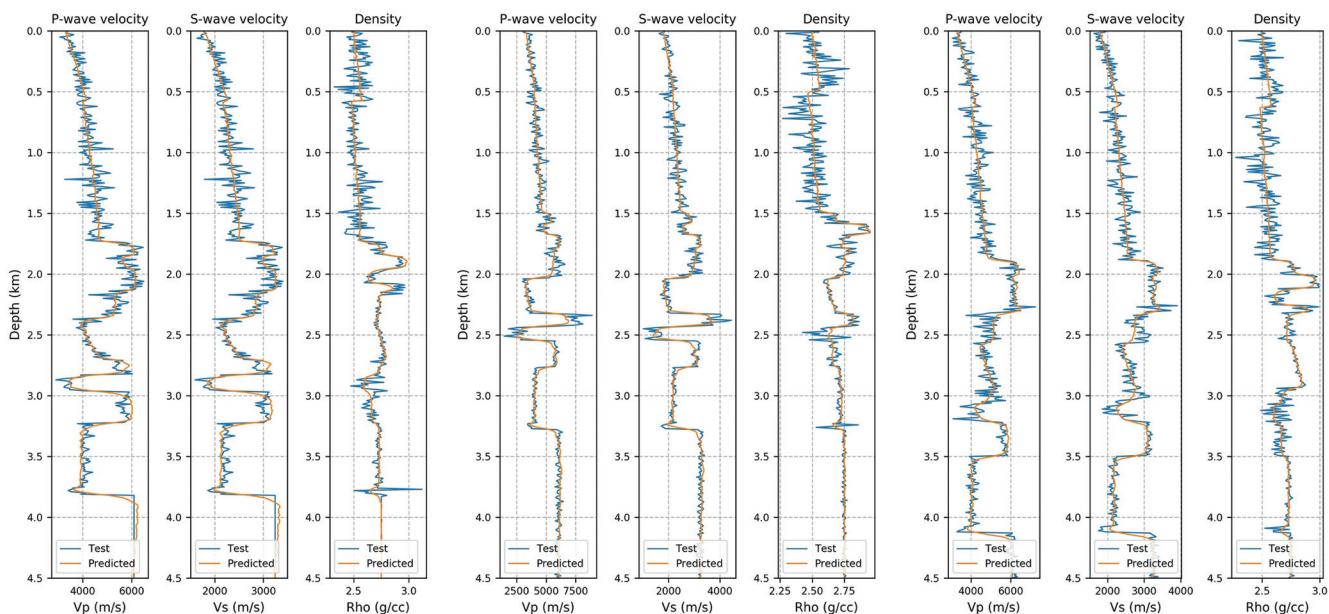


Figure 8. Comparison of true (blue) and predicted (orange) 1D elastic models for three random tests.

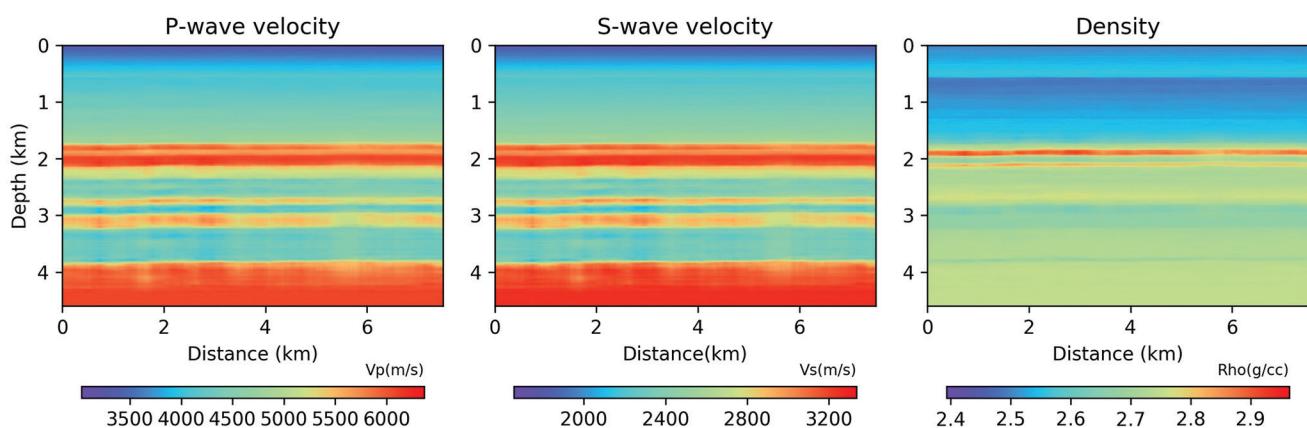


Figure 9. Predicted 2D elastic models (V_p , V_s , and density) by CNN using real data from a land survey.

noted that we invert only for V_p with tomography, not V_s and density.

However, there are some variations in dynamic range of the elastic parameters across the section. These are due to residual differences between the field data and synthetic training data that are not balanced adequately during preprocessing. When the difference is large, predictions tend to become less stable.

The comparison of 1D profiles extracted at the well location clearly shows that P-wave velocity predicted by the CNN matches the well log better than tomography. Although we find that inversion quality is sensitive to the wavelet, it appears that the result is not hampered by the lack of low-frequency content in the input seismic data. This is due to the fact that the low-wavenumber component is implicitly set during the training model specification, and the CNN essentially recovers the medium-(blockiness) and high-wavenumber (contrasts) structures of the elastic parameters.

Conclusions

Separate CNN models are trained for 3D automated fault predictions (classification) and 1D elastic prestack seismic inversion (regression). In the fault interpretation case, the model rapidly outputs the fault probability, dip, and azimuth simultaneously. Applications on real data show that the CNN model can produce reliable fault picks in various seismic images from different regions. The automated workflow and the output fault sticks may significantly reduce the cycle time of a fault interpretation project — potentially from weeks or months to hours or days.

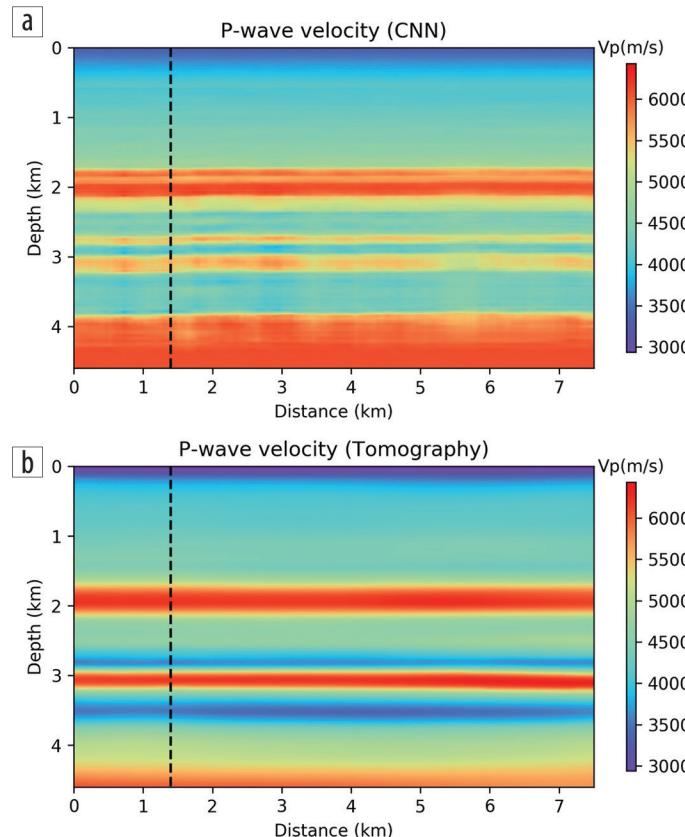


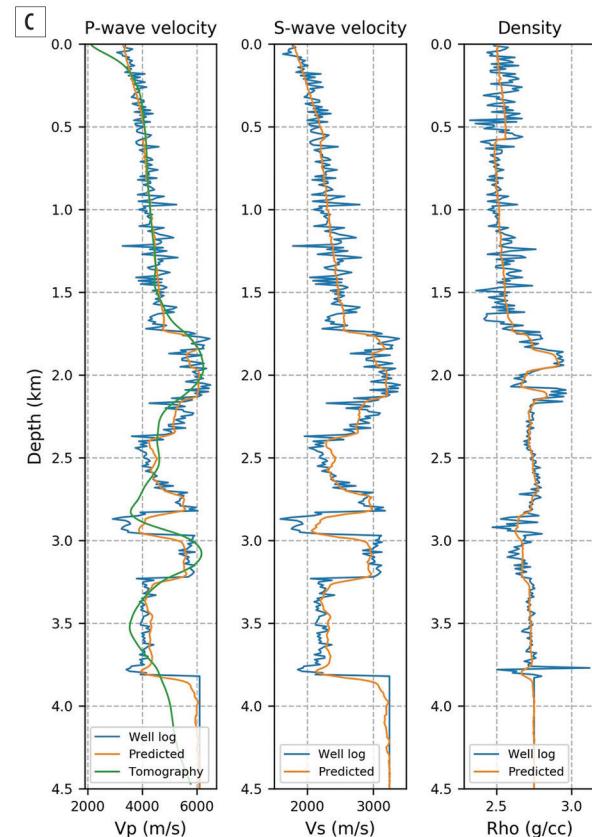
Figure 10. (a) Comparison of the P-wave velocity models obtained from CNN and (b) traveltome tomography. (c) Comparison of 1D profiles at the well location. Well logs are in blue, CNN predictions are in orange, and V_p model from tomography is in green.

For seismic inversion, we show that CNN can be trained on synthetic data to predict simultaneously elastic parameters from real seismic gathers. We take a local field-oriented approach to this problem. By limiting training to a specific geologic scenario on land and using available well data, we can construct training data to represent locally varying subsurface parameters. Inversion of real prestack data by CNN shows results that are comparable with conventional model building methods, but robustness can be limited by the inherent difference between field data and synthetic training data.

In both case studies shown here, we demonstrate that a CNN, trained on synthetic data, can be applied to real seismic data. We also highlight the advantages of using deep learning for seismic problems with respect to cost effectiveness and cycle-time reduction. It is important to appreciate that currently a significant effort is still required to design and optimize a supervised deep learning workflow to produce results that are comparable to those from conventional workflows. **TE**

Acknowledgments

We would like to thank BP and our partners ExxonMobil and EOG Resources Trinidad Ltd. for permission to publish this work. We would also like to thank many of our BP colleagues on the technology and asset teams for helpful discussions. Finally, we would like to thank Eliis for developing their seismic interpretation software PaleoScan, which was used to extract fault sticks from our FaultNet attribute volume presented in this paper.



Data and materials availability

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

Corresponding author: york.zheng@bp.com

References

- Araya-Polo, M., J. Jennings, A. Adler, and T. Dahlke, 2018, Deep learning tomography: The Leading Edge, **37**, no. 1, 58–66, <https://doi.org/10.1190/tle37010058.1>.
- Badrinarayanan, V., A. Kendall, and R. Cipolla, 2017, SegNet: A deep convolutional encoder-decoder architecture for image segmentation: IEEE Transactions on Pattern Analysis and Machine Intelligence, **39**, no. 12, <https://doi.org/10.1109/TPAMI.2016.2644615>.
- He, K., X. Zhang, S. Ren, and J. Sun, 2016, Deep residual learning for image recognition: Proceedings of the Conference on Computer Vision and Pattern Recognition, IEEE, 770–778, <https://doi.org/10.1109/CVPR.2016.90>.
- Jones, I., 2010, Tutorial: Velocity estimation via ray-based tomography: First Break, **28**, no. 2, 45–52, <https://doi.org/10.3997/1365-2397.2010006>.
- Kennett, B. L. N., 1983, Seismic wave propagation in stratified media: Cambridge University Press.
- Kosloff, D., J. W. C. Sherwood, Z. Koren, E. Machet, and Y. Falkovitz, 1996, Velocity and interface depth determination by tomography of depth migrated gathers: Geophysics, **61**, no. 5, 1511–1523, <https://doi.org/10.1190/1.1444076>.
- Marfurt, K. J., R. L. Kirlin, S. L. Farmer, and M. S. Bahorich, 1998, 3-D seismic attributes using a semblance-based coherency algorithm: Geophysics, **63**, no. 4, 1150–1165, <https://doi.org/10.1190/1.1444415>.
- Marfurt, K. J., V. Sudhaker, A. Gersztenkorn, K. D. Crawford, and S. E. Nissen, 1999, Coherency calculations in the presence of structural dip: Geophysics, **64**, no. 1, 104–111, <https://doi.org/10.1190/1.1444508>.
- Richardson, A., 2018, Seismic full-waveform inversion using deep learning tools and techniques: arXiv:1801.07232.
- Ronneberger, O., P. Fischer, and T. Brox, 2015, U-net: Convolutional networks for biomedical image segmentation: Proceedings of the International Conference on Medical Image Computing and Computer-assisted Intervention, Springer, 234–241, https://doi.org/10.1007/978-3-319-24574-4_28.
- Russakovsky, O., 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, 2015, ImageNet large scale visual recognition challenge: International Journal of Computer Vision, **115**, 211–252, <https://doi.org/10.1007/s11263-015-0816-y>.
- Van Bemmelen, P. P., and R. E. Pepper, 2000, Seismic signal processing method and apparatus for generating a cube of variance values: U.S. Patent 6,151,555.
- Virieux, J., A. Asnaashari, R. Brossier, L. Métivier, A. Ribodetti, and W. Zhou, 2014, An introduction to full waveform inversion, in V. Grechka and K. Wapenaar, eds., Encyclopedia of Exploration Geophysics: SEG, R1-1–R1-40, <https://doi.org/10.1190/1.9781560803027.entry6>.
- Wu, Y., Y. Lin, and Z. Zhou, 2018, InversionNet: Accurate and efficient seismic waveform inversion with convolutional neural networks: 88th Annual International Meeting, SEG, Expanded Abstracts, 2096–2100, <https://doi.org/10.1190/segam2018-2998603.1>.
- Wu, X., Y. Shi, S. Fomel, and L. Liang, 2018, Convolutional neural networks for fault interpretation in seismic images: 88th Annual International Meeting, SEG, Expanded Abstracts, 1946–1950, <https://doi.org/10.1190/segam2018-2995341.1>.
- Wu, X., L. Liang, Y. Shi, and S. Fomel, 2019a, FaultSeg3D: Using synthetic data sets to train an end-to-end convolutional neural network for 3D seismic fault segmentation: Geophysics, **84**, no. 3, IM35–IM45, <https://doi.org/10.1190/geo2018-0646.1>.
- Wu, X., Y. Shi, S. Fomel, L. Liang, Q. Zhang, and A. Yusifov, 2019b, FaultNet3D: Predicting fault probabilities, strikes and dips with a common CNN: IEEE Transactions on Geoscience and Remote Sensing, <https://doi.org/10.1109/TGRS.2019.2925003>.
- Xiong, W., X. Ji, Y. Ma, Y. Wang, N. M. AlBenHassan, M. N. Ali, and Y. Luo, 2018, Seismic fault detection with convolutional neural network: Geophysics, **83**, no. 5, O97–O103, <https://doi.org/10.1190/geo2017-0666.1>.
- Zheng, Y., and Q. Zhang, 2018, Pre-stack seismic inversion using deep learning: Presented at 1st EAGE/PESGB Workshop on Machine Learning.