

Convolutional neural networks for fault interpretation in seismic images

Xinming Wu*, Yunzhi Shi, Sergey Fomel, The University of Texas at Austin; Luming Liang, Uber

SUMMARY

We propose an automatic fault interpretation method by using convolutional neural networks (CNN). In this method, we construct a 7-layer CNN to first estimate fault orientations (dips and strikes) within small image patches that are extracted from a full seismic image. With the estimated fault orientations, we then construct anisotropic Gaussian functions that mainly extend along the estimated fault dips and strikes. We finally stack all the locally fault-oriented Gaussian functions to generate a fault probability image. Although trained by using only synthetic seismic images, the CNN model can accurately estimate fault orientations within real seismic images. The fault probability image, computed from the estimated fault orientations, displays cleaner, more accurate, and more continuous fault features than those in the conventional fault attribute images.

INTRODUCTION

Faults in a seismic image (Figure 1a) are often recognized as laterally high discontinuity or low continuity of reflections. Based on this observation, numerous methods have been proposed to highlight faults by computing attributes that measure reflection continuity such as semblance (Marfurt et al., 1998; Hale, 2009) and coherency (Marfurt et al., 1999; Li and Lu, 2014; Karimi et al., 2015; Wu, 2017), or reflection discontinuity such as variance (Van Bemmelen and Pepper, 2000; Randen et al., 2001) and gradient magnitude (Aqrawi and Boe, 2011). However, these attributes can be sensitive to noise and stratigraphic features, which also produce reflection discontinuities.

To enhance fault features while suppressing the features unrelated to faults, some authors (Neff et al., 2000; Cohen et al., 2006; Hale, 2013; Wu and Hale, 2016; Wu and Zhu, 2017) suggest to apply some extent of fault-oriented averaging or smoothing in computing a fault attribute image. Computing such a fault-oriented attribute image (Figure 1c), however, requires first estimating fault orientations such as fault strikes and dips. On the other hand, estimating fault orientations from a seismic image is highly challenging. In a seismic image, reflections are the dominant features while faults are embedded by the reflections, which makes the conventional structure orientation estimation methods, such as structure tensors (Van Vliet and Verbeek, 1995; Hale, 2009; Wu and Janson, 2017) and plane wave destruction (Fomel, 2002), fail to estimate fault orientations. Therefore, some researchers (Neff et al., 2000; Cohen et al., 2006; Hale, 2013; Wu and Zhu, 2017) estimate fault strikes and dips by scanning over all possible fault orientations to find the strikes and dips that maximize fault attribute or any measurement of reflection discontinuity. The fault-oriented scanning method, however, can generate false fault orientations and related false fault features, especially in areas where the faults are close to each other as in Figure 1c.

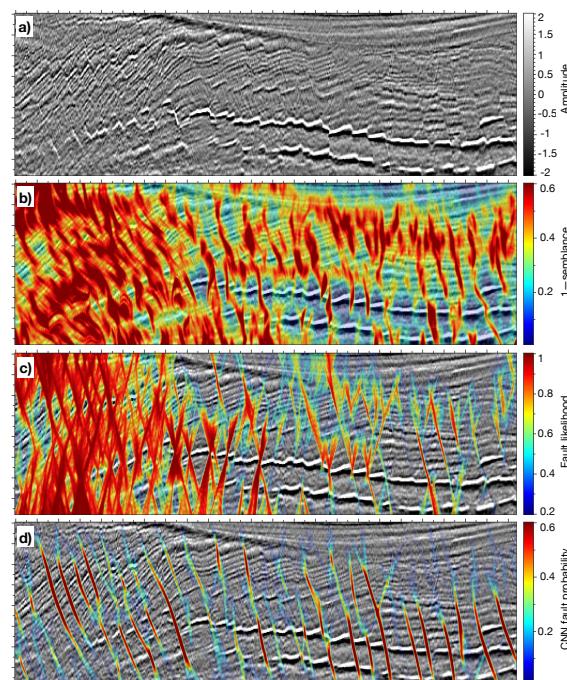


Figure 1: A 2D seismic image (a) displayed with 1-semblance (b), fault likelihood (c), and CNN fault probability (d).

Convolutional neural networks (CNN) (LeCun et al., 1989, 1998) have proven to be the most powerful method in solving computer vision problems including image classification (Krizhevsky et al., 2012; Zeiler and Fergus, 2014; He et al., 2016), object detection (Girshick et al., 2014; Ren et al., 2015; He et al., 2017) and segmentation (Ronneberger et al., 2015; Badrinarayanan et al., 2017). The image classification was previously a challenging problem for conventional classification methods based on image descriptors (e.g., Cortes and Vapnik, 1995; Lowe, 1999). By using the CNN, the error rate of classifying 1000 types of images on ImageNet 2015 benchmark was significantly reduced to only 3.57% (He et al., 2016), which is even lower than the error rate 5.1% of human classification (Russakovsky et al., 2015) on the same benchmark dataset.

In this paper, we consider fault orientation estimation as an image classification problem by assuming a local fault within a small seismic image is a line (2D) or plane (3D) that is uniquely defined by a single fault orientation. Inspired by the success of CNN in variant classification tasks, we propose to use CNN to classify fault orientations within the small seismic images. We construct a CNN model with 6 convolutional layers followed by a fully connected layers and a final output vector of a fault orientation prediction. We train the CNN model by using 200,000 synthetic seismic image patches that are automatically computed as illustrated in Figure 2. We further apply the trained CNN model to classify the orientations of the faults in small seismic patches that are extracted from

CNN for fault interpretation

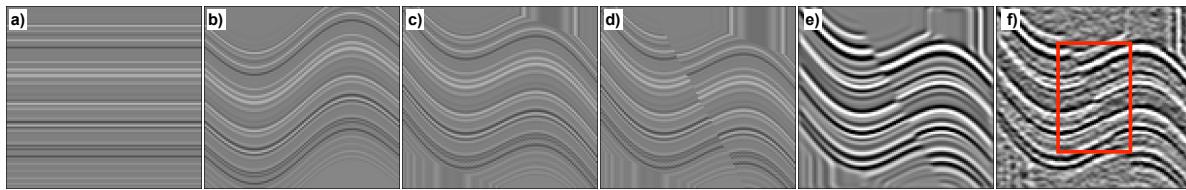


Figure 2: Illustration of the main steps in creating a synthetic seismic image patch.

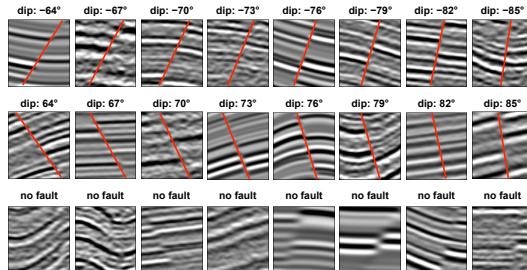


Figure 3: Synthetic training images with and without faults.

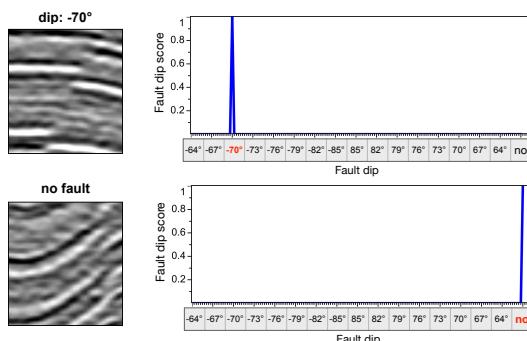


Figure 4: Synthetic training images (left column) and the corresponding label vectors (right column).

an input full seismic image. With the estimated fault orientations, we compute fault probability patches with fault-oriented linear features. All the fault probability patches have the same size of the extracted seismic patches and are further stacked to compute a full fault probability image as shown in Figure 1d.

METHOD

A fault is often locally linear (2D) or planar (3D) and therefore the fault geometry within a local window can be uniquely defined by a single apparent dip (2D) or a combination of dip and strike (3D). As shown in Figure 3, knowing the fault dips, we can simply draw lines (red lines) along the dips to approximate the faults in the small seismic patches. Thus, a key step of fault interpretation is to first estimate fault orientations.

We assume the dip angles of 2D faults are in the range of $[-63^\circ, -86^\circ] \cup [86^\circ, 63^\circ]$, where the faults extend more vertically than horizontally. We further define every 3 dip angles as one fault class. For example, we assume the faults with dip angels -63° , -64° and -65° belong to the same fault class. We therefore totally have 17 different fault classes including 8 classes for negative fault dips, 8 classes for positive dips and 1

class for no fault, as shown in Figure 3. We assume the faults, that do not pass through the center image pixel, all belong to the class of "no fault" (the bottom row in Figure 3). With these assumptions, we consider fault interpretation within the small seismic patches (Figure 3) as a problem of fault dip classification by using CNN. To achieve this, the 17 different representative dip classes need to be first learnt by a CNN model. We create a large number of synthetic seismic patches (Figure 3) for each fault dip class to train the CNN model.

Training datasets

Training and validating a CNN model often requires a huge amount of images and corresponding labellings. Manually labelling the images would be extremely time-consuming. Therefore, we propose an effective and efficient way to create numerous synthetic seismic images and corresponding labellings.

To create a seismic image, we first generate a horizontal reflectivity model $r(x, y)$ (Figure 2a) with a sequence of random reflectivity values that are in the range of $[-1, 1]$. We then create a sinusoidal folding structure (Figure 2b) by vertically shifting the model, where the vertical shifts are defined as a sine function with three parameters $s_1(x) = a \sin(b + cx)$. We also apply some linear shearing (Figure 2c) $s_2(x) = dx + e$ to further increase the variation of the folding structure. We further add some linear faulting in the model and the faulting can be defined by different dips and slips. After creating the folding and faulting in the reflectivity model, we then compute a synthetic seismic image (Figure 2e) by convolving a Ricker wavelet with the model in directions perpendicular to the reflectivity folding structures. We further add some random noise (Figure 2f) to increase the realism of the synthetic seismic image. To eliminate the boundary artifacts, we finally crop the center patch (red box in Figure 2f) of the noisy seismic image to obtain a final training seismic patch with 48×32 pixels. By randomly choosing the reflectivity sequence, sinusoidal folding parameters, linear shearing parameters, faulting parameters, wavelet peak frequencies, and noise intensities, we are able to create numerous unique synthetic seismic patches, which can be much more than the 200,000 training images and 20,000 validating images used in this paper.

Figure 3 shows 36 representative training image patches that are automatically computed as discussed above. The top and middle images correspond to 16 different fault dip classes while the bottom images correspond to class of no fault. We created about 10,000 training images for each fault dip class and about 40,000 images for the class of no fault. Using more images for the class of no fault in training a CNN model is helpful to reduce the error of misclassifying a no fault image as a fault image, which will be a significant error in the fault interpreta-

CNN for fault interpretation

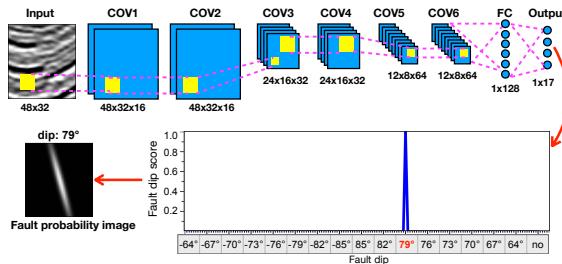


Figure 5: Outline of the CNN architecture used to estimate fault orientation from an input seismic patch.

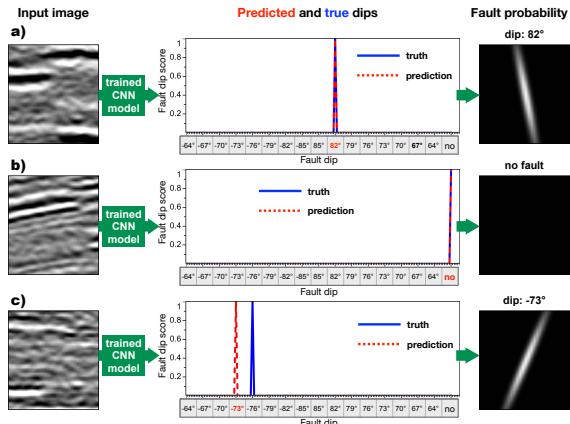


Figure 6: Input seismic patches (left column), predicted fault dips (middle column), and fault probability image patches (right column) that are computed as anisotropic Gaussian functions oriented by the predicted dips.

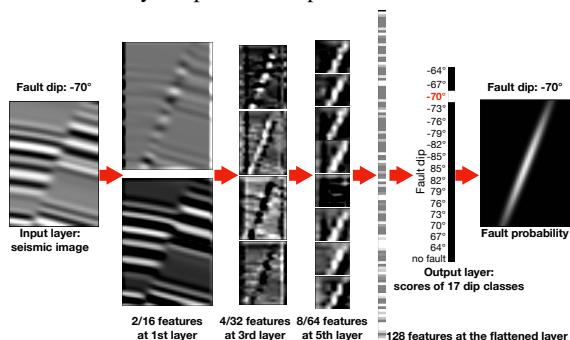


Figure 7: Illustration of feature maps in some CNN layers.

tion. On the other hand, classifying a fault image as a wrong but neighboring fault dip class would not be a significant error. Labelling the synthetic seismic patches is straightforward because we know exactly how the faults are oriented within the patches. Figure 4 shows two representative seismic patches with (Figure 4a) and without (Figure 4b) faults, where we label the seismic patches by using two vectors with 17 elements corresponding to 17 different fault dip classes. We define such a label vector by simply setting its value to 1 at the position of the true dip but zeros elsewhere. For the image patches without faults, we assign the value of 1 to the last element of the label vector.

CNN architecture

With an input seismic patch (48×32 pixels), we classify the corresponding fault dip class by using a CNN model (Figure 5) with 6 convolutional layers followed by a fully connected layer. The first two convolutional layers (COV1, COV2) contain 16 filters of size 3×3 and stride of 1. Similarly, the second two layers (COV3, COV4) and the last two layers (COV5, COV6), respectively, contain 32 filters and 64 filters with the same size of 3×3 and stride of 1. Following Ioffe and Szegedy (2015), we apply the batch normalization to the output feature maps right after each convolutional layer and before activation. The activation is implemented by a nonlinear ReLU function ($f(x) = \max(0, x)$), which has been shown to provide better fitting abilities than the sigmoid function (Krizhevsky et al., 2012). After every two convolutional layers, we apply an average pooling to every resulting feature map to take the average over every 2×2 spatially neighboring pixels with a stride of 2. This average pooling contributes to increased performance despite the reduction of resolution.

After the 6 convolutional layers, we flatten out the 2D feature maps into a 1D flat fully connected layer (FC) of 128 elements. Before feeding the FC elements to the output layer, we adopt a dropout regularization method to randomly set 50% of the elements to zeros, which is helpful to avoid overfitting and encourage the sparsity of the neurons. The output softmax layer has 17 output units and each unit represents a specific class of fault dip. The unit with the highest score indicates the predicted dip angle of the fault passing through the center pixel of the input image patch. We train the CNN model by using 200,000 synthetic seismic patches and validate it on another 20,000 synthetic seismic patches. By using a Titan Xp GPU, the whole training process with the Adadelta optimization method (Zeiler, 2012) took only around 8.8 minutes for 12 epochs. The final mean categorical accuracy on the validation images is about 0.89.

Figure 6 shows three of the final validation tests, where the trained CNN model provided exactly accurate fault dip estimation for the first two seismic patches with (Figure 6a) and without (Figure 6b) fault. With the predicted fault dips, we construct the corresponding fault probability patches (with the same size of the input seismic patches) as dip-oriented anisotropic Gaussians (right column in Figure 6). The predicted dip for the third seismic patch (left image in Figure 6c) is inaccurate but is still close to the true fault dip (middle image in Figure 6c). This inaccurate dip prediction is not a significant error for the fault interpretation because the fault probability patch (right image in Figure 6c), constructed using the predicted dip, can still provide a good approximation of the fault. We checked through the 20,000 validation tests and found that the inaccurately predicted dips appear mostly close to the true dips. Figure 7 shows how the CNN model works to estimate fault dip from an input seismic patch. After the training process, the filters in the convolutional layers have been updated to automatically compute multiple appropriate feature maps (especially those at the 3rd and 5th layers in Figure 7) to detect the fault. Based on these appropriate fault feature maps, the fault dip can be accurately estimated in the output layer.

CNN for fault interpretation

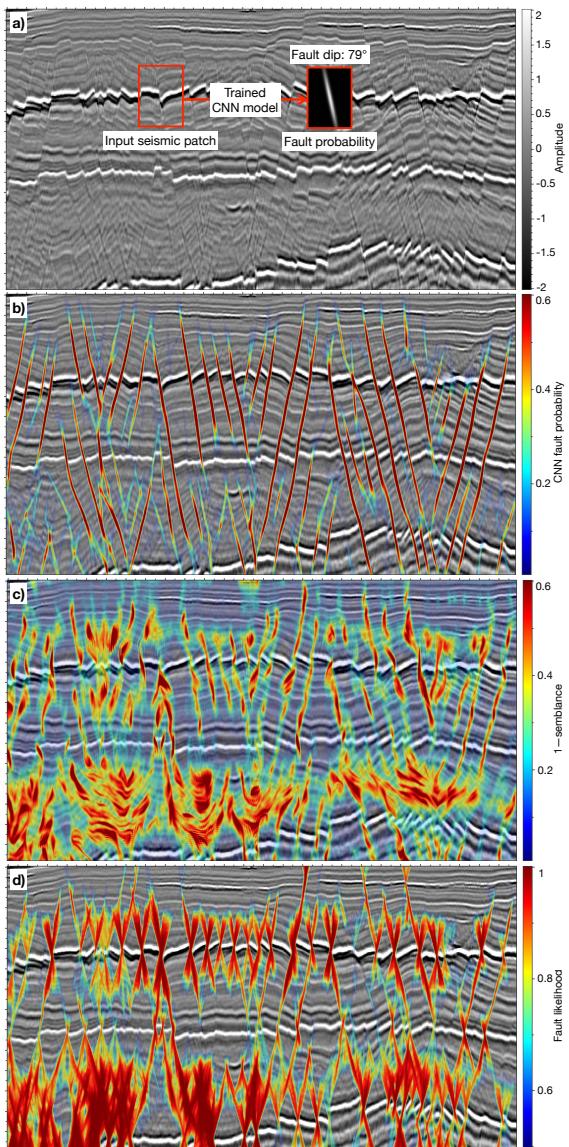


Figure 8: Given a seismic image (a), we use the trained CNN model to predict the fault dip for every overlapping image patch and construct a corresponding fault probability patch. We further stack all the overlapping fault probability patches to compute a CNN fault probability image in (b), which shows much cleaner and more continuous fault features than those in the 1-semblance (Hale, 2009) image (Figure 8c) and fault likelihood (Hale, 2013; Wu and Hale, 2016) image (Figure 8d).

APPLICATIONS

In applying the trained CNN model to a full seismic image (Figure 8a), we first extract overlapping seismic patches with 48×32 pixels. We then feed the patches to the CNN model to estimate a fault dip for each patch and further compute fault probability patches according to the estimated dips. We finally stack all the overlapping fault probability patches to compute a full fault probability image (Figure 8b), which displays much cleaner and more continuous fault features than those in the 1-semblance (Hale, 2009) image (Figure 8c) and fault like-

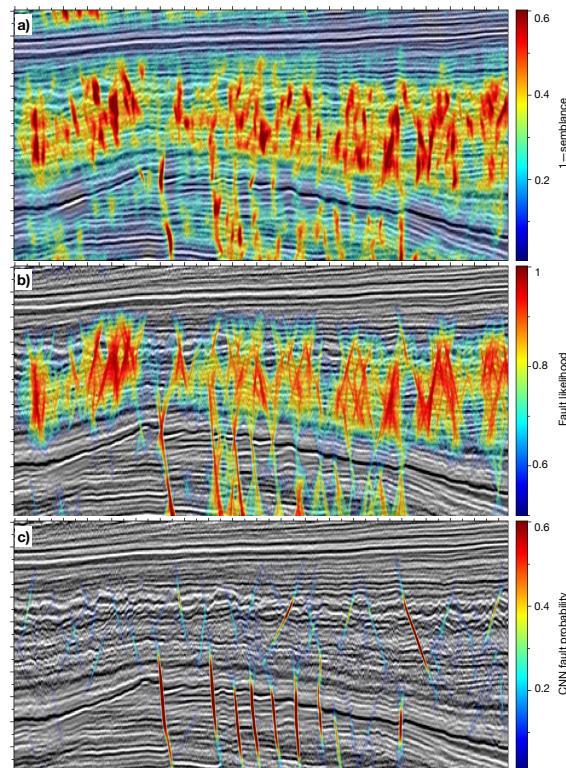


Figure 9: A 2D seismic image (a) displayed with 1-semblance (b), fault likelihood (c), and CNN fault probability (d).

lihood (Hale, 2013; Wu and Hale, 2016) image (Figure 8d). Figures 1 and 9 show two more complicated examples, where both the semblance and fault likelihood images cannot provide clean and accurate fault features. However, the trained CNN model still works well to accurately predict fault dips and further yield clean and accurate fault probability images.

CONCLUSIONS

We have proposed a CNN method to accurately estimate fault dips and further compute an accurate and clean fault probability image. We train the CNN model completely on synthetic seismic patches without any manual labeling. The trained model works well in multiple real seismic images to compute better fault images than most of the conventional fault detection methods. This method is currently tested on only 2D examples but can be easily extended to 3D cases. We actually expect this method to work better in 3D, where the fault geometry can be better described and more information in 3D cubes can be used by a CNN model to predict fault orientations.

ACKNOWLEDGMENTS

This research is financially supported by the sponsors of the Texas Consortium for Computation Seismology (TCCS). We also acknowledge the support of NVIDIA Corporation with the donation of the Titan Xp GPU used for this research.

REFERENCES

- Aqrawi, A. A., and T. H. Boe, 2011, Improved fault segmentation using a dip guided and modified 3D Sobel filter: 81st Annual International Meeting, SEG, Expanded Abstracts, 999–1003, <https://doi.org/10.1190/1.3628241>.
- 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**, 2481–2495, <https://doi.org/10.1109/TPAMI.2016.2644615>.
- Cohen, I., N. Coult, and A. A. Vassiliou, 2006, Detection and extraction of fault surfaces in 3D seismic data: Geophysics, **71**, no. 4, P21–P27, <https://doi.org/10.1190/1.2215357>.
- Cortes, C., and V. Vapnik, 1995, Support-vector networks: Machine learning, **20**, 273–297.
- Fomel, S., 2002, Applications of plane-wave destruction filters: Geophysics, **67**, 1946–1960, <https://doi.org/10.1190/1.1527095>.
- Girshick, R., J. Donahue, T. Darrell, and J. Malik, 2014, Rich feature hierarchies for accurate object detection and semantic segmentation: Proceedings of the IEEE conference on computer vision and pattern recognition, 580–587.
- Hale, D., 2009, Structure-oriented smoothing and semblance: CWP Report 635.
- Hale, D., 2013, Methods to compute fault images, extract fault surfaces, and estimate fault throws from 3D seismic images: Geophysics, **78**, no. 2, O33–O43, <https://doi.org/10.1190/geo2012-0331.1>.
- He, K., G. Gkioxari, P. Dollar, and R. Girshick, 2017, Mask r-cnn: The IEEE International Conference on Computer Vision (ICCV), 2980–2988.
- He, K., X. Zhang, S. Ren, and J. Sun, 2016, Deep residual learning for image recognition: Proceedings of the IEEE conference on computer vision and pattern recognition, 770–778.
- Ioffe, S., and C. Szegedy, 2015, Batch normalization: Accelerating deep network training by reducing internal covariate shift: arXiv preprint arXiv:1502.03167.
- Karimi, P., S. Fomel, L. Wood, and D. Dunlap, 2015, Predictive coherence: Interpretation, **3**, no. 4, SAE1–SAE7, <https://doi.org/10.1190/INT-2015-0030.1>.
- Krizhevsky, A., I. Sutskever, and G. E. Hinton, 2012, Imagenet classification with deep convolutional neural networks: Advances in neural information processing systems, 1097–1105.
- LeCun, Y., B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, 1989, Backpropagation applied to handwritten zip code recognition: Neural computation, **1**, 541–551, <https://doi.org/10.1162/neco.1989.1.4.541>.
- LeCun, Y., L. Bottou, Y. Bengio, and P. Haffner, 1998, Gradient-based learning applied to document recognition: Proceedings of the IEEE, **86**, 2278–2324, <https://doi.org/10.1109/5.726791>.
- Li, F., and W. Lu, 2014, Coherence attribute at different spectral scales: Interpretation, **2**, no. 1, SA99–SA106, <https://doi.org/10.1190/INT-2013-0089.1>.
- Lowe, D. G., 1999, Object recognition from local scale- invariant features: Proceedings of the Seventh IEEE International Conference on Computer Vision, 1150–1157.
- 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**, 1150–1165, <https://doi.org/10.1190/1.1444415>.
- Marfurt, K. J., V. Sudhaker, A. Gerszenkorn, K. D. Crawford, and S. E. Nissen, 1999, Coherency calculations in the presence of structural dip: Geophysics, **64**, 104–111, <https://doi.org/10.1190/1.1444508>.
- Neff, D. B., J. R. Grismore, and W. A. Lucas, 2000, Automated seismic fault detection and picking: US Patent 6, **018**, 498.
- Randen, T., S. I. Pedersen, and L. Sønneland, 2001, Automatic extraction of fault surfaces from three-dimensional seismic data: 81st Annual International Meeting, SEG, Expanded Abstracts, 551–554, <https://doi.org/10.1190/1.1816675>.
- Ren, S., K. He, R. Girshick, and J. Sun, 2015, Faster r-cnn: Towards real-time object detection with region proposal networks: Advances in neural information processing systems, 91–99.
- Ronneberger, O., P. Fischer, and T. Brox, 2015, U-net: Convolutional networks for biomedical image segmentation: International Conference on Medical image computing and computer-assisted intervention, Springer, 234–241.
- Russakovsky, O., J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al., 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: US Patent 6, 151, 555.
- Van Vliet, L. J., and P. W. Verbeek, 1995, Estimators for orientation and anisotropy in digitized images: Proceedings of the first annual conference of the Advanced School for Computing and Imaging ASC'95, Heijen (The Netherlands), 442–450.
- Wu, X., 2017, Directional structure-tensor based coherence to detect seismic faults and channels: Geophysics, **82**, no. 2, A13–A17, <https://doi.org/10.1190/geo2016-0473.1>.
- Wu, X., and D. Hale, 2016, 3D seismic image processing for faults: Geophysics, **81**, no. 2, IM1–IM11, <https://doi.org/10.1190/geo2015-0380.1>.
- Wu, X., and X. Janson, 2017, Directional structure tensors in estimating seismic structural and stratigraphic orientations: Geophysical Journal International, **210**, 534548.
- Wu, X., and Z. Zhu, 2017, Methods to enhance seismic faults and construct fault surfaces: Computers and Geosciences, **107**, 37–48.
- Zeiler, M. D., 2012, Adadelta: An adaptive learning rate method: arXiv preprint arXiv:1212.5701.
- Zeiler, M. D., and R. Fergus, 2014, Visualizing and understanding convolutional networks, in D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, eds, Computer Vision — ECCV 2014: Springer **869**, 818–833.