

Automatic fault detection on seismic images using a multiscale attention convolutional neural network

Kai Gao¹, Lianjie Huang¹, Yingcai Zheng², Rongrong Lin², Hao Hu², and Trenton Cladouhos³

ABSTRACT

High-fidelity fault detection on seismic images is one of the most important and challenging topics in the field of automatic seismic interpretation. Conventional hand-picking-based and semi-human-intervened fault-detection approaches are being replaced by fully automatic methods thanks to the development of machine learning. We have developed a novel multiscale attention convolutional neural network (MACNN) to improve machine-learning-based automatic end-to-end fault detection on seismic images. The most important characteristics of our MACNN fault-detection method are that it uses a multiscale spatial-channel attention mechanism to merge and refine encoder feature maps of different spatial resolutions. The new architecture enables our MACNN to more effectively learn and exploit contextual information embedded in the encoder feature maps. We determine through several synthetic data and field data examples that our MACNN tends to produce higher resolution, higher fidelity fault maps from complex seismic images compared to those of the conventional fault-detection convolutional neural network, thus leading to improved geologic fidelity and interpretability of detected faults.

INTRODUCTION

Faults are usually observed as planar discontinuities in rocks. Geologically, faults result from local to regional stress-induced brittle failure that eventually leads to visible displacement of rock formations along a fault plane (Fossen, 2016). Faults are geologically important because they are usually the key to infer evolution of the stress or

strain environment. Faults also play a critical role in geophysical exploration. For example, faults are vital to understanding oil and gas migration and reservoir sealing and provide important geologic guidance for drilling and extraction activities (Hardman and Booth, 1991). In geothermal exploration, particularly in hydrogeothermal systems, connected fault systems act as pathways for geothermal fluid injection, flow, and extraction and are therefore the key factor to characterize before and during geothermal production (Gan and Elsworth, 2014; Gao et al., 2020). In geologic carbon storage, detecting and monitoring possible faults in caprocks are crucial for determining the feasibility, sealing integrity, and long-term safety of a target reservoir for CO₂ storage (Villarrasa and Carrera, 2015).

Local- or regional-scale fault detection and interpretation are usually performed based on time or depth migration seismic images. Faults are generally picked by hand based on expert inference and geologic prior knowledge associated with a certain region. In addition to inherently subjective or even biased pickings because of interpreter dependence, picking by hand also becomes cumbersome or even infeasible for tracking complex fault systems, particularly for large-scale 3D images.

Faults are usually associated with relative displacements of reflectors on seismic images or image volumes. Semi-human-intervened fault-detection approaches can therefore compute image attributes associated with such a reflector displacement to assist in fault identification, including coherence (Marfurt et al., 1999; Wu, 2017) and semblance (Marfurt et al., 1998). The improved versions of such approaches rely on strategies such as constraining, smoothing, or enhancing computed coherence or semblance attributes to provide better fault detection (Bakker, 2002; Pedersen et al., 2005; Cohen et al., 2006; Wu and Fomel, 2018). In addition to questionable accuracy and reliability in delineating faults from complex field data images, these image attribute-based approaches can be fairly computationally expensive, particularly for large-scale 3D

Manuscript received by the Editor 25 December 2020; revised manuscript received 10 September 2021; published ahead of production 8 October 2021; published online 23 November 2021.

¹Los Alamos National Laboratory, Geophysics Group, Los Alamos, New Mexico 87545, USA. E-mail: kaigao@lanl.gov (corresponding author); ljh@lanl.gov.

²University of Houston, Department of Earth and Atmospheric Sciences, Houston, Texas 77204, USA. E-mail: yzheng24@central.uh.edu; rong2lin@gmail.com; hhu4@uh.edu.

³Cyrq Energy Inc., Seattle, Washington 98103, USA. E-mail: trenton.cladouhos@cyrqenergy.com.

© 2022 Society of Exploration Geophysicists. All rights reserved.

seismic images. Noises with a wavenumber spectrum similar to that of reflectors in seismic images can also obviously deteriorate the produced fault map, a factor that is almost inevitable in practice. Usually, these image attribute-based approaches heavily rely on the expert's choice of parameters for different images, and they are usually computationally expensive because of intensive pixel-based image attribute computation, analysis, and selection.

Seismic interpretation is marching fast into a new era thanks to the development of machine learning. For seismic interpretation, automatic fault detection based on machine learning achieves notably superior outcomes compared with conventional methods. Wu et al. (2019) develop an automatic end-to-end fault-detection method based on convolutional neural networks (CNNs) (LeCun et al., 1998), or more specifically U-Net, following and improving on prior works by Di et al. (2018), Guo et al. (2018), Xiong et al. (2018), and others based on pixel-level classification neural network architecture. The method of Wu et al. (2019) is a simplified version of the U-Net developed initially for image segmentation (Ronneberger et al., 2015). In terms of neural network architecture, the U-Net is a special type of CNN that contains an encoding branch and a decoding branch. Differing from standard CNNs in which the input sequentially flows through convolutional layers, U-Net uses a series of skip connections to convey contextual information between the encoder branch and the decoder branch in addition to simple downsampling or upsampling within each branch. These pairs of convolutional layers connected by skip connections have different "distances," i.e., a highest level encoder layer has a skip connection with a highest level decoder layer, and a lower level encoder layer has a skip connection with a lower level decoder layer, making the CNN visually a U-shaped neural network concatenated by long/short skip connections between the encoder/decoder branches. Notably, a skip connection connects a pair of encoder/decoder layers at the same resolution level. Wu et al. (2019) demonstrate that the U-Net-based approach provides superior detection accuracy along with significantly improved computational efficiency compared with conventional attribute-based approaches. Di et al. (2019) incorporate geologic information into a CNN to produce fault maps with improved geologic interpretability of detected faults. Gao et al. (2021) develop a nested residual U-Net for fault detection to improve the geologic interpretability of detected faults. Methods that focus on reducing the necessary amount of training data/labels are also reported in recent works (Cunha et al., 2020).

In our study, we find that the U-Net-based fault-detection method can reliably identify fault maps from various kinds of seismic images in most cases. However, the U-Net method may generate inaccurate fault prediction for noisy seismic images, or when the fault system in an image is complex. In some cases, the faults predicted using the U-Net may suffer from insufficient spatial resolution. These limitations may result from the intrinsic architecture of the U-Net. In the U-Net, the feature maps of an encoder skip connect to the decoder at the same spatial level. The contextual information embedded in the encoder branch is generally insufficient; therefore, a raw skip connection from the encoder branch to the decoder branch may lead to insufficient representation of the fault features in seismic images. In addition, differing from medical or natural image segmentation applications, faults on seismic images can interlace with one another, forming complex fault systems, and the probability distributions associated with faults can be highly

imbalanced compared with the background image without faults. Hence, the rudimentary U-Net architecture may easily result in false fault prediction.

The limitations associated with the rudimentary U-Net architecture motivate us to develop a novel machine-learning architecture for fault detection. Specifically, rather than skip concatenating an encoder to its same-level decoder, we design a multiscale spatial-channel attention mechanism to refine an encoder prior to the skip concatenation. We feed the feature maps of all encoders in the encoder branch to the multiscale spatial-channel attention gated block, in which we enhance the neural network's capability to learn "where" to look for faults, and "what" are the most important features of faults, at all spatial resolutions. We then concatenate the refined encoder to the same-level decoder at all spatial levels. Such multiscale spatial-channel attention refinement of the encoder contextual information leads to an improved learning of the fault features in a seismic image at all spatial resolutions. This multiscale spatial-channel attention mechanism differs from a similar mechanism that only refines same-level encoder feature maps for concatenation (Khanh et al., 2020). With experience learned from extensive numerical tests, we also replace the usual double convolution in the encoders and decoders with a triple-convolution block that includes three convolutional layers, a skip concatenation, and a skip addition. Our multiscale attention CNN (MACNN) is inspired by recent developments of U-Net-based image segmentation and fault detection (e.g., Huang et al., 2020; Sultana et al., 2020; Zhou et al., 2020) particularly the U-Net 3+ architecture (Huang et al., 2020), the multiscale fusion mechanism (Qin et al., 2020; Gao et al., 2021), and the spatial-channel attention mechanism (Khanh et al., 2020). Yet, we incorporate important simplifications, modifications, and improvements based on these prior works, and we design this novel CNN architecture for fault detection on seismic structural images. We also adopt a smoothed dice loss function instead of the widely used balanced cross-entropy loss function in previous works to train our MACNN. In this sense, our MACNN can be viewed as an improved U-Net for automatic end-to-end fault detection.

We organize our paper as follows: In the "Methodology" section, we describe the architecture of our MACNN for fault detection and the method of preparing a proper training data set. We then demonstrate the superior fault-detection capability of our method over two conventional U-Net fault-detection methods, using several synthetic and real seismic image examples. We summarize our work in the "Conclusions" section.

METHODOLOGY

Architecture

We display the overall structure of our MACNN in Figure 1, which is slightly more complex than the conventional U-Net CNN for fault detection (Wu et al., 2019). U-Net and MACNN have an encoder branch and a decoder branch. Specifically, both architectures have three encoders and three decoders and a link at the lowest level of the neural network. In U-Net, the skip connection connects a pair of an encoder and a decoder at the same level to transfer contextual information through the entire neural network. In contrast, in our MACNN, we refine an encoder (encoder 1, 2, or 3 as displayed in Figure 1) using a multiscale spatial-channel attention block before concatenating each of them to the decoder at the same level. This multiscale spatial-channel attention block,

at each level, is built from all three encoders after proper up- or downsampling and convolutions. Another distinct feature of our MACNN is that each encoder or decoder in the neural network is not a simple double convolution as in the U-Net. Instead, it is a small triple-convolution block that consists of three convolutional layers, as well as a skip connection and a skip addition. The novel components in our MACNN compared with those of the U-Net are inspired by several previous works, such as the full-scale connection structure (Huang et al., 2020), the single-level spatial-channel attention mechanism (Khanh et al., 2020), and the nested U-Net architecture (Qin et al., 2020; Gao et al., 2021). Our MACNN exploits the advantages of these structures and constructs a simplified architecture.

We first take the procedure of refining encoder 1 as an example to explain how the multiscale spatial-channel attention encoding works. In the conventional U-Net, encoder 1 is skip concatenated to decoder 1 without any refinement; therefore, it only contains contextual information at this spatial level. In our MACNN, to refine encoder 1 using multiscale contextual information before skip concatenating it to decoder 1, we build a multiscale spatial-channel attention gated block with the architecture displayed in Figure 2, which consists of a spatial attention map and a channel attention map. In the following, we use a tuple (C, D, H, W) to represent a feature map, in which C , D , H , and W are the number of channels, the depth, the height, and the width of the feature map, respectively. In the 2D case, the only relevant dimensions are C , H , and W per conventions of the machine-learning community.

The purpose of the spatial attention displayed in Figure 2a is to enhance the capability of our neural network on where to look for faults. To build the spatial attention map, we make no changes to the input encoder 1 because the spatial size of the final spatial/channel attention map is $(D, H, W) = (D_1, H_1, W_1)$; we upsample encoder 2 from (C_2, D_2, H_2, W_2) to (C_2, D, H, W) using a trilinear (or bilinear in the 2D case) interpolation of ratio 2; we upsample encoder 3 from (C_3, D_3, H_3, W_3) to (C_3, D, H, W) using a trilinear (or bilinear in the 2D case) interpolation of ratio 4. For each of the upsampled (or unchanged) encoders, we perform three operations: a channel average pooling, a channel max pooling, and a convolution with a kernel size of $(1, 1, 1)$, all compressing the number of channels from C to 1, and we concatenate the three feature maps into a $(3, D, H, W)$ feature map, followed by a convolution with a kernel size of $(7, 7, 7)$ that compresses the number of channels from 3 to 1. We then add the three outputs into a $(1, D, H, W)$ feature map, followed by a sigmoid activation function, to create the final spatial attention map of size $(1, D, H, W)$. The reason for this design is that we use multitype spatial operations (average pooling, max pooling, and size 1 convolution) and large convolution kernels ((7, 7, 7) in this case) to gain a sufficiently large receptive field, so that the spatial attention map gives the neural network an improved spatial contextual representation capability. The process of building the spatial attention map can be written as

$$F_{(3,D,H,W)}^i = (\mathcal{A} \circ \mathcal{R}[E_{(C_i, D_i, H_i, W_i)}^i]) \oplus (\mathcal{M} \circ \mathcal{R}[E_{(C_i, D_i, H_i, W_i)}^i]) \\ \oplus (\mathcal{K} \circ \mathcal{R}[E_{(C_i, D_i, H_i, W_i)}^i]), \quad (1)$$

$$S_{(1,D,H,W)} = \sigma \circ \mathcal{H}[F_{(3,D,H,W)}^i], \quad (2)$$

where $E_{(C_i, D_i, H_i, W_i)}^i$ ($i = 1, 2, 3$) is the encoder i ; \mathcal{R} is a resampling operation that transforms an input encoder of size (C_i, D_i, H_i, W_i) to the target spatial size (C_i, D, H, W) ; \mathcal{A} is an average pooling operated along the channel dimension that generates a feature map of size $(1, D, H, W)$; \mathcal{M} is a max pooling operated along the channel dimension that generates a feature map of size $(1, D, H, W)$; and \mathcal{K} is a convolution operation with a kernel of size $(1, 1, 1)$ that generates a feature map of size $(1, D, H, W)$. In addition, \mathcal{H} is a convolution operation with a kernel of size $(1, 1, 1)$ that generates a feature map of size $(1, D, H, W)$ from the concatenated feature map of size $(3, D, H, W)$, σ is a sigmoid activation function applied to a feature map, and the symbol \oplus denotes the concatenation along the channel dimension. The term $S_{(1,D,H,W)}$ is the final spatial attention map. The symbol “◦” represents the composition of functions. For instance, $\mathcal{A} \circ \mathcal{B}[x]$ means first applying function \mathcal{B} to x and then applying function \mathcal{A} to $\mathcal{B}[x]$. This representation also applies to the following equations.

We also build a channel attention map using the architecture displayed in Figure 2b to enhance the capability of our MACNN on what to look for on encoder feature maps. We use global average pooling and max pooling to build two feature maps, each with a size of $(C, 1, 1, 1)$. Then, we create a feature map of size $(N, 1, 1, 1)$ and add the feature maps from all three encoders to create a final channel attention map of size $(C, 1, 1, 1)$ after a series of convolutions and a sigmoid activation. The process can be written as

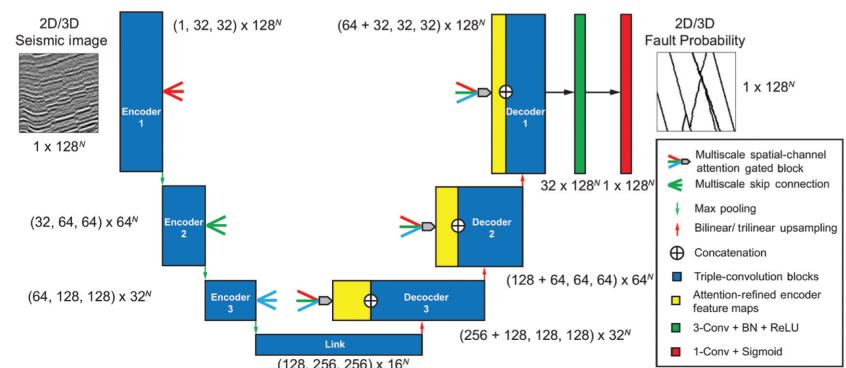


Figure 1. The overall architecture of our MACNN for fault detection. The input to the neural network is a 2D or 3D seismic image, and the output is a same-size fault probability map with a value range from 0 to 1. “ReLU” represents the rectified linear unit (e.g., Watt et al., 2020), “BN” represents the batch normalization, “Conv” represents a convolutional layer, and “Sigmoid” represents the sigmoid activation function. The numbers $C \times X^N$ associated an encoder or decoder represent a set of C feature maps with the height and width (or depth, height, and width in the 3D case) being equal to X , where $N = 2$ or 3 is the number of dimensions. All of the upsampling operations in our MACNN use bilinear (or trilinear in the 3D case) interpolation, in contrast to the U-Net that uses simple nearest-neighbor interpolations. The tuple (L_1, L_2, L_3) in the sketch represents the number of feature maps in the triple-convolution block. In the prediction phase after training, the size X of an input seismic image can be different along different spatial dimensions, with a value other than 128.

$$F_{(N,1,1,1)}^i = \mathcal{K} \circ \mathcal{A} \circ \mathcal{R} \left[E_{(C,D_i,H_i,W_i)}^i \right], \quad (3)$$

$$G_{(N,1,1,1)}^i = \mathcal{K} \circ \mathcal{M} \circ \mathcal{R} \left[E_{(C,D_i,H_i,W_i)}^i \right], \quad (4)$$

$$W_{(C,1,1,1)} = \sigma \circ \mathcal{H} \left[\sum_{i=1}^3 \left(F_{(N,1,1,1)}^i + G_{(N,1,1,1)}^i \right) \right], \quad (5)$$

where \mathcal{R} is the up- or downsampling operation that resizes the input encoder E_i from size (C_i, D_i, H_i, W_i) to size (C_i, D, H, W) ; \mathcal{A} is a global average-pooling operation that transforms the input feature map to size $(C_i, 1, 1, 1)$; \mathcal{M} is a global max-pooling operation that transforms the input feature map to size $(C_i, 1, 1, 1)$; and \mathcal{K} is a convolution with a kernel size of $(1, 1, 1)$ that generates a feature map of size $(N, 1, 1, 1)$. In addition, \mathcal{H} is a convolution with a kernel size of $(1, 1, 1)$ that generates a feature map of size $(C, 1, 1, 1)$ and $W_{(C,1,1,1)}$ is the final channel attention map.

We then obtain the multiscale spatial-channel attention-refined E' for an input encoder map E using a process displayed in Figure 2c:

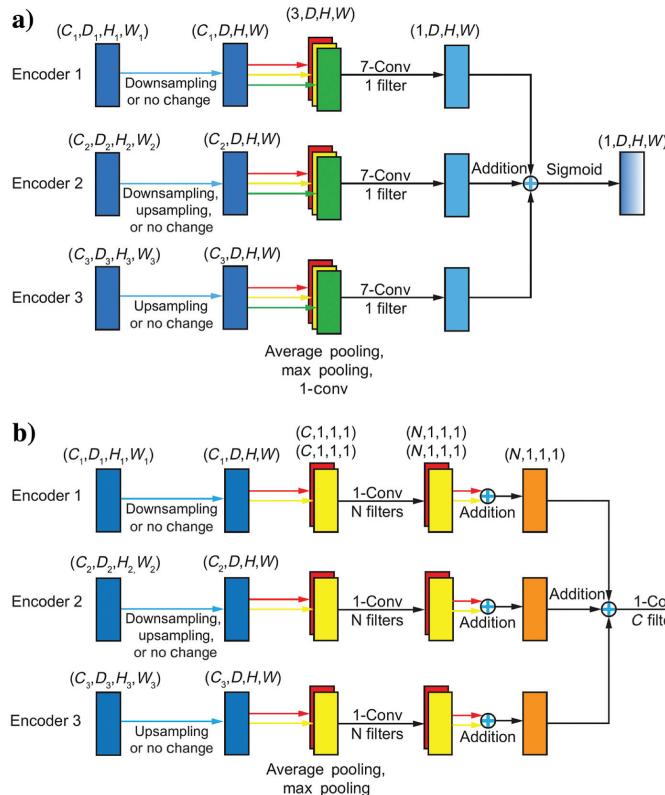
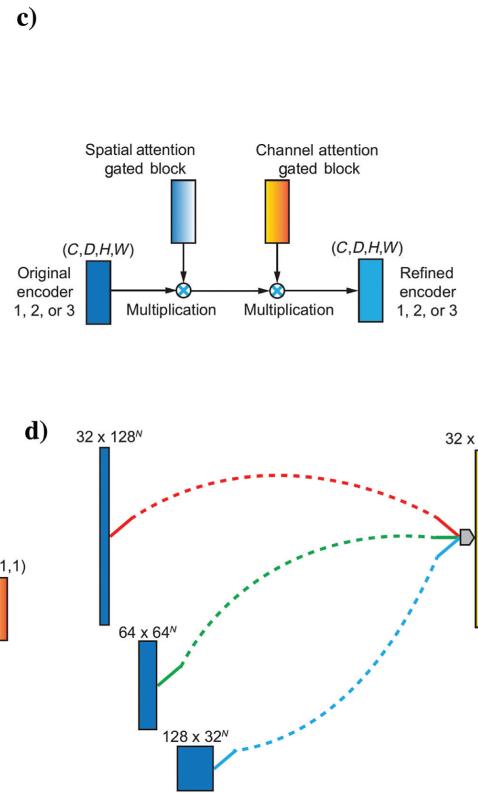


Figure 2. The architecture of the multiscale spatial-channel attention gated block. (a-c) The spatial attention gated block, channel attention gated block, and the final composite spatial-channel attention gated block for creating attention-refined encoder feature maps from the feature maps of three encoders at different spatial scales, respectively. The actual values of C , H , and W (or C , D , H , and W in the 3D case) at three different spatial scales are consistent with those given in Figure 1. (d) The procedure of refining encoder 1 using the multiscale spatial-channel attention block.

$$\begin{aligned} E'_{(C,D,H,W)} &= E_{(C,D,H,W)} \times \mathcal{T}_C[S_{(1,D,H,W)}] \\ &\times \mathcal{T}_{DHW}[W_{(C,1,1,1)}], \end{aligned} \quad (6)$$

where \mathcal{T}_C is a tile process that repeats the spatial attention map $S_{(1,D,H,W)}$ along the channel dimension by a factor of C and \mathcal{T}_{DHW} is a tile process that repeats the channel attention map $W_{(C,1,1,1)}$ along the three spatial dimensions by a factor of D , H , and W , respectively. The product “ \times ” in equation 6 represents the element-wise product between two feature maps. The entire encoder refinement procedure can be represented by a simple sketch in Figure 2d. Afterward, we skip concatenate the spatial-channel attention-refined encoder 1 to the same-level decoder 1 as displayed in Figure 1. This refinement procedure enables encoder 1 to learn fault features of different spatial resolutions. We use a similar procedure but with different target sizes to refine encoders 2 and 3, and skip concatenate them to the corresponding decoders 2 and 3.

Another essential component of MACNN is the triple-convolution architecture for the encoders and decoders. We display the architecture of this triple convolution in Figure 3. For a specific encoder or decoder, the spatial size of the three convolutional layers is the same, with optionally different numbers of channels. The distinct feature of this triple-convolution structure compared with the double-convolution structure in the conventional U-Net is that in addition to one additional convolutional layer, we set a skip connection as well as a skip addition between the first and third convolutional layers. In



addition, we use a dilational convolution with a dilation rate of two in the middle convolutional layer to improve the receptive field of feature learning for this encoder or decoder.

It is worth noting that Khanh et al. (2020) design a similar spatial-channel attention mechanism to improve the U-Net. However, their spatial-channel attention only refines the encoder feature maps at the same spatial level, whereas our multiscale attention mechanism builds the spatial and channel attention maps using all encoders. In addition, their spatial-channel attention mechanism combines the encoder and decoder feature maps to create spatial attention and channel attention maps, a process that can complicate the training process based on our numerical experiments, whereas our multi-scale attention block only uses encoder features maps to build these attention maps. Note that the triple-convolution block is different from the local U-Net block in the nested U-Net architecture (Qin et al., 2020; Gao et al., 2021). In the triple-convolution block, all of the encoders/decoders use only three convolutional layers with the middle one being a two-dilational convolutional layer. This simplification makes the training faster compared with nested U-Net structures.

Implementation

A proper loss function is crucial for a machine-learning training process and robust generalization. To train our MACNN for 2D and 3D fault detection, we use a smoothed dice loss function (Sudre et al., 2017; Jadon, 2020):

$$\mathcal{L}(p, g) = 1 - \frac{2 \sum_{i=1}^N p_i g_i + 1}{\sum_{i=1}^N p_i + \sum_{i=1}^N g_i + 1}, \quad (7)$$

where $0 \leq p_i \leq 1$ is the predicted fault probability value of the i th image pixel and g_i is the ground truth with values of 0 (nonfault) or 1 (fault) of the i th image pixel. The dice loss measures the global overlap between the neural network prediction and the ground truth; thus, it is suitable for training a neural network with an imbalanced data set. There are alternative definitions for a dice loss, and the one we choose is a form suitable for our fault prediction task. Through extensive numerical tests, we find that a neural network trained by the smoothed dice loss function usually generates higher resolution fault predictions with fewer uncertainties compared to the counterpart trained with the balanced cross-entropy loss function.

Supervised learning generally requires a representative and sufficiently large training data set to provide unbiased, reliable generalization (Chollet, 2017). For fault detection, this requirement means that the training data set should properly include a sufficiently large amount of seismic structures with geologically realistic faults, accompanied by fault labels, for the neural network to learn how to recognize faults on seismic images.

We use synthetic 2D or 3D seismic images with faults and their corresponding faults to train our neural network. We use the following procedure to generate a 2D seismic image with faults: (1) Generate a 1D random reflectivity model, (2) smooth a random-valued array using a Gaussian filter with a random value σ to mimic the fluctuation of a seismic reflector in the horizontal direction, and assign the generated reflectivity to these deformed seismic reflectors, (3) map these reflectors to a regularly sampled mesh, (4) convolve the reflectivity model with a Ricker or time-integrated Ricker wavelet with a randomly chosen center frequency, (5) add a random number of faults with random dip angles and spatial locations in the 2D

image, and shift one of the two blocks with respect to the fault with positive or negative displacement, and (6) add a random level of noises to the image, and slightly smooth the image to smear the sharp fault discontinuity. To generate a 3D seismic image with faults, we adopt a similar procedure as that of the 2D case, except that we assume the layers on the two sides of a 3D fault surface can shift with respect to each other in a direction parameterized by three parameters: dip, strike, and rake. As in the 2D case, we add a random number of faults when generating a 3D image with faults. Using this method, we generate a total of 10,000 2D images of size 128×128 as the training data set and a total of 100 2D images as the test data set to train and test the 2D version of our neural network, respectively. We generate a total of 1000 3D images of size $128 \times 128 \times 128$ as the training data set and a total of 100 3D images as the test data set to train and test the 3D version of our neural network, respectively. The training and test data sets have no overlapping to ensure an unbiased assessment of the trained neural network.

Because we train our MACNN using these synthetic images/faults and conduct prediction on either synthetic or field seismic images using the trained neural network, there is no additional computational cost when predicting faults on field-data images except for the computational cost in the prediction phase. Similar to any other supervised machine-learning methods, the performance of the U-Net and our MACNN in predicting faults on seismic images that are not included in the training data set is strongly influenced by the statistical characteristics of the training data/labels. In other words, given a training data set with evidently different features, MACNN is likely to give different fault prediction even for the same image. It is an open question on how to produce the smallest possible, representative, high-quality training data for a given supervised learning algorithm to provide reliable generalization. For MACNN, the numerical examples in the “Numerical examples” section demonstrate that the training data created in this manner lead to generally good predictions for different types of images.

Although our MACNN architecture seems complex, it only introduces a moderately greater amount of parameters on top of the U-Net. For example, the 2D and 3D versions of the U-Net fault detection (Wu et al., 2019) contain approximately 3.1 and 9.4 M learnable weights, respectively, whereas our 2D and 3D MACNN contain approximately 3.9 and 11.7 M learnable weights, respectively.

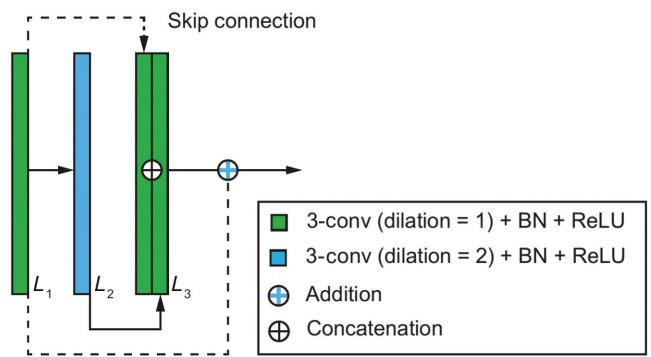


Figure 3. A sketch of the triple-convolution block for the encoders and decoders. The terms L_1 , L_2 , and L_3 are the numbers of channels of the three convolutional layers.

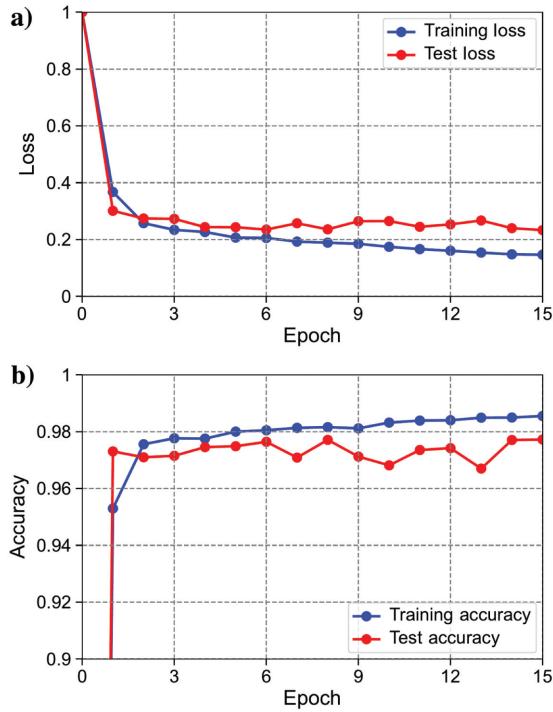


Figure 4. (a) Comparison between the training loss (the blue curve) and the test loss (the red curve) in training the MACNN. (b) Comparison between the training accuracy (the blue curve) and the test accuracy (the red curve) in training the MACNN.

Table 1. Comparison of three different metrics, including the precision, accuracy, and recall score, among the three different neural networks for the 2D test data set. Bold font denotes the best scores.

Method/metrics	Accuracy	Precision	Recall
U-Net-1	0.975	0.737	0.766
U-Net-2	0.975	0.756	0.701
MACNN	0.979	0.798	0.796

Table 2. Comparison of three different metrics, including the precision, accuracy, and recall score, among the three different neural networks for the 3D test data set. Bold font denotes the best scores.

Method/metrics	Accuracy	Precision	Recall
U-Net-1	0.960	0.529	0.942
U-Net-2	0.980	0.791	0.740
MACNN	0.983	0.818	0.796

We use the open-source machine-learning framework PyTorch (Paszke et al., 2019) to implement our MACNN. We enclose our implementation of MACNN in a distributed training framework, PyTorch Lightning (Falcon, 2019), to accelerate the training process. We use the Adam optimization (Kingma and Ba, 2017; Watt et al., 2020) with an initial learning rate of 0.0001 to train our neural network. We use a cosine annealing learning rate scheduler (Loshchilov and Hutter, 2017) to reduce the learning rate over epochs. We train 2D MACNN on a total of 4 NVIDIA P100 graphics processing unit (GPU) cards, and we train 3D MACNN on a total of 16 NVIDIA P100 GPU cards. Therefore, the effective batch size for training our MACNN is four in the 2D case and 16 in the 3D case. Figure 4 displays the loss and accuracy curves of training our MACNN. The accuracy curve in Figure 4b indicates that our MACNN reaches accuracy of approximately 98% for the training and test data sets.

In the training phase, each epoch of training the 2D MACNN takes approximately 30 s, whereas each epoch of training the 3D MACNN takes approximately 10 min. The prediction phase of our MACNN is nearly in real time. For instance, predicting faults on a 2D image sampled by 512×1024 grid points takes only several seconds, whereas predicting faults on a 3D image sampled by $256 \times 512 \times 256$ grid points takes approximately half a minute, both on a 36-core central processing unit.

To demonstrate the advantage of our MACNN for fault detection, we also implement two other architectures based on U-Net. The first one is the conventional U-Net provided by Wu et al. (2019) with a balanced cross-entropy loss. We denote this architecture as U-Net-1. The second one is the same U-Net as the first one except that we add a batch-normalization layer before each of the rectified linear unit (ReLU) activation layers in the neural network. Batch normalization is a technique that reshifts and rescales the input feature map; thus, it can mitigate the internal covariate shift issue and stabilize the learning (Watt et al., 2020). We also use the smoothed dice loss as the loss function to train this batch-normalized U-Net. We refer to this architecture as U-Net-2 for convenience. We train and test these two neural networks (their 2D and 3D versions) using the same training and test data sets as those for our MACNN, respectively. All three architectures are fully CNNs. Therefore, they can directly detect faults on 2D or 3D seismic images of any size without any changes to their respective architecture.

Tables 1 and 2 compare three performance metrics among these three architectures for 2D and 3D fault detection, respectively, including the accuracy, precision, and recall score. The three metrics are defined as

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{FN} + \text{TN}}, \quad (8)$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \quad (9)$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \quad (10)$$

where

$$TP = \sum_{i=1}^N (\text{truth} = 1) \text{ and } (\text{prediction} = 1), \quad (11)$$

$$TN = \sum_{i=1}^N (\text{truth} = 0) \text{ and } (\text{prediction} = 0), \quad (12)$$

$$FP = \sum_{i=1}^N (\text{truth} = 0) \text{ and } (\text{prediction} = 1), \quad (13)$$

$$FN = \sum_{i=1}^N (\text{truth} = 1) \text{ and } (\text{prediction} = 0), \quad (14)$$

where N is the total number of pixels in a 2D or 3D seismic image and we use “and” in these expressions to denote the logical operation. Based on the definitions, the accuracy quantifies how many of the predictions (the fault and nonfault pixels in this case) are correct, the precision quantifies how many of the positive predictions (the fault pixels in this case) are correct, and the recall score quantifies how many true positive pixels (the ground-truth fault pixels in this case) are predicted correctly by a neural network. For these metrics, a higher value indicates a better prediction. In some cases, the precision and the recall score may change in opposite directions with respect to each other.

We observe that, for two dimensions and three dimensions, our MACNN gains some advantages over the conventional U-Net structure in accuracy and precision, and in some cases, the recall score as well. Using batch normalization and the smoothed dice loss in U-Net-2 may or may not improve the performance measured by these three metrics over U-Net-1 structure. However, we must remark that these metrics can possibly be misleading for the synthetic and field seismic images in evaluating the performance of U-Net-1 and U-Net-2. We find that, in most cases, adding batch normalization and using the dice loss function generally improves the resolution and the quality of fault prediction. This phenomenon also indicates the complexity in reasonably and accurately evaluating the performance of a neural network for predicting faults, particularly for field seismic images.

Synthetic image benchmark

The comparison among the U-Nets and our MACNN for five of the synthetic seismic images in the validation data set is shown in Figure 5a. The five rows of Figure 5 represent the selected five images and ground-truth/detected faults. The first column displays the seismic images, the second column shows the ground-truth fault maps, and the third to the fifth columns show the faults detected using U-Net-1, U-Net-2, and our MACNN, respectively. U-Net-1 and U-Net-2 produce generally accurate fault predictions. However, there are cases in which these two architectures misidentify faults, miss subtle faults, or generate low-probability fault pixels, particularly when the two or several faults interlace or are close to one another. For example, for the third image, U-Net-1 and U-Net-2

produce incorrect predictions for the three intersecting faults in the left part of the image. In contrast, our MACNN correctly detects the intersection pattern of the three faults.

We observe a similar prediction accuracy improvement of our MACNN compared with the conventional architectures in the 3D case. Figure 6 displays an example with three randomly oriented faults. All three architectures produce generally good fault predictions, yet U-Net-1 and U-Net-2 show some inaccuracies in the intersection part — U-Net-1 has a low-resolution prediction as displayed in Figure 6b on this intersection, whereas U-Net-2 gives a broken prediction as depicted in Figure 6c. In contrast, our MACNN produces a prediction in Figure 6d that is almost identical to the ground truth in Figure 6a. In Figure 7a, we display a 3D seismic image and the faults on the image. The difficulty of predicting faults on this image is that it contains a strike-slip fault that merely generates a small displacement across the horizontal direction, and this fault is almost unidentifiable in the vertical direction. Figure 7b–7d displays the fault prediction results produced using U-Net-1, U-Net-2, and our MACNN, respectively. Clearly, this strike-slip fault is not accurately identified using either U-Net-1 or U-Net-2 — U-Net-1 almost completely misses this fault, whereas U-Net-2 only produces a broken prediction. In contrast, the MACNN prediction in Figure 7d is close to the ground truth, where the horizontal strike-slip pattern is well identified.

To further demonstrate the advantage of our MACNN architecture, we show in Figure 8a a comparison among the U-Nets and our MACNN on a synthetic image with different levels of random noise. The seven rows in Figure 8 represent a synthetic image with low to high levels of added random noise, and their corresponding ground-truth and detected faults. With an increasing level of random noise, U-Net-1 and U-Net-2 produce fault maps with more and more

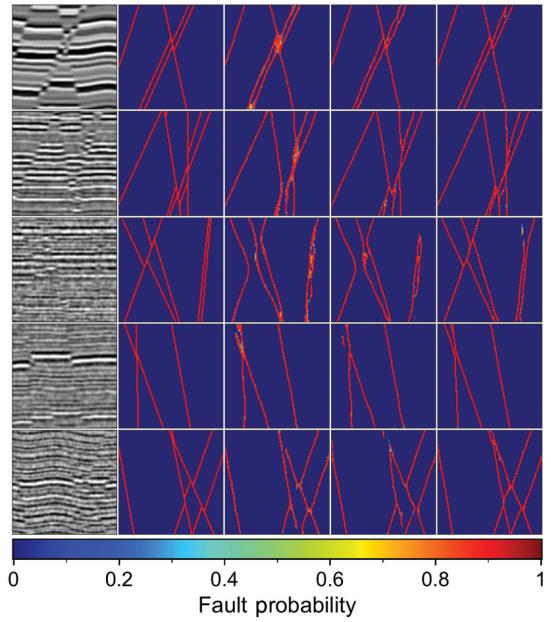


Figure 5. Comparison among faults detected using U-Net-1, U-Net-2, and our MACNN for five 2D synthetic images. Rows 1–5 correspond to five different seismic images. Columns 1–5 represent (1) the seismic images, (2) the ground-truth faults, (3) the faults detected using U-Net-1, (4) the faults detected using U-Net-2, and (5) the faults detected using our MACNN, respectively.

erroneous fault pixels that are increasingly difficult to interpret. Some fault pixels predicted by U-Net-1 also have low probability. In contrast, our MACNN produces fault maps that are closer to the ground truth compared with U-Net-1 and U-Net-2. Remarkably, even at the seventh row in which the image is extremely noisy, our MACNN can still predict one of the long faults with some accuracy.

These evaluations on two dimensions and three dimensions based on synthetic images demonstrate the superior fault prediction capability of our MACNN compared with the conventional architectures. The evaluations also demonstrate that, even though our MACNN does not outperform conventional architectures significantly based on the three metrics, in practice, it is able to produce much more interpretable and reliable fault maps. This phenomenon implies that it can be difficult to completely and accurately quantify the quality of a fault-detection result using simple numeric metrics and that a more suitable metric is worth further investigation.

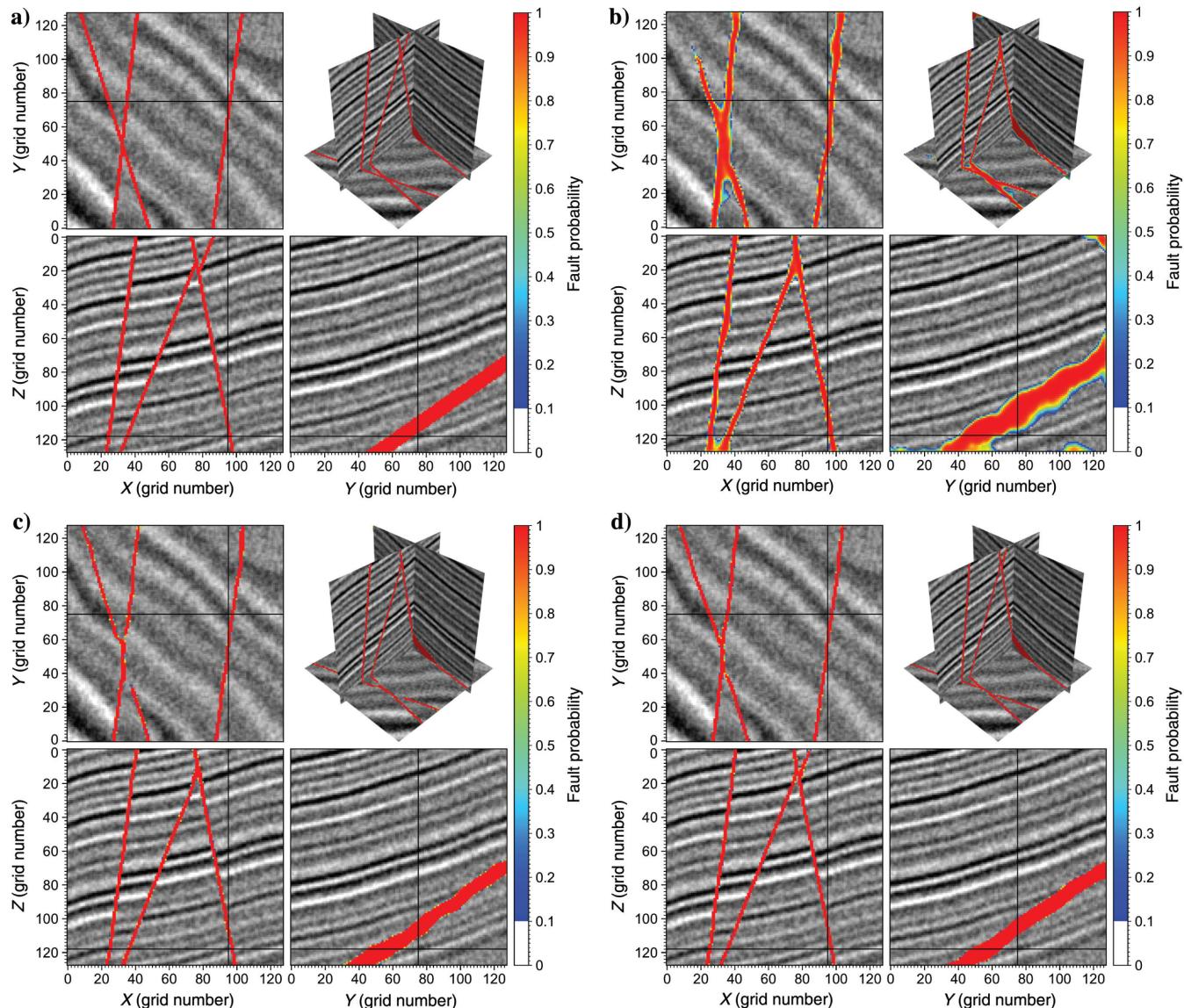


Figure 6. Comparison among faults detected on a 3D seismic image using (b) U-Net-1, (c) U-Net-2, and (d) our MACNN. (a) The ground truth fault overlying on the image.

FIELD DATA EXAMPLES

We demonstrate the superior fault-detection capability of MACNN using several field data examples.

Soda Lake geothermal field images

In the first example, we compare the two U-Net architectures and our MACNN in detecting faults on seismic images from the Soda Lake geothermal field.

Figure 9a shows a 2D P-wave velocity model built using geologic features of the Soda Lake geothermal field (Gao et al., 2020). We use the dashed red lines to indicate the positions of three major faults in the horizontal center. These three faults are implicitly imposed on the velocity model when we built the velocity model; i.e., they appear as weak lateral velocity changes in the model instead of explicit faults. Several other faults are similarly built into

the model. We build an S-wave velocity model using the same structural pattern, and we generate synthetic elastic multicomponent data based on this elastic model using a high-order standard staggered-grid finite-difference method (Virieux, 1986).

Figure 10a shows a 2D seismic image produced using a vector elastic reverse time migration method (Gao and Huang, 2019). The dimension of the seismic image is 133 grid points in the vertical direction and 745 grid points in the horizontal direction, with actual physical dimensions of 0.6×3.4 km. We process the image using a spatial windowed root-mean-square normalization and a nonlinear anisotropic diffusion filtering (Wu and Guo, 2018) to improve the visibility of faults for our fault-detection test purpose.

With the neural networks pretrained using synthetic images as described in the “Methodology” section, we perform fault detection using U-Net-1, U-Net-2, and our MACNN, with the prediction results shown in Figure 10b–10d, respectively. This seismic

image of synthetic data contains little noise, and the faults on the image can easily be identified even by human eyes. Therefore, all three methods produce a fault map with almost every possible fault delineated. However, in the region denoted by the dashed yellow circle, U-Net-1 and U-Net-2 produce a prediction with some inaccuracies. The fault on the right predicted by U-Net-1 is broken, whereas the fault on the right predicted by U-Net-2 is shorter than the ground truth. Only the MACNN prediction is very close to the ground truth.

We perform another three fault detections on a 2D seismic image in Figure 11a generated using field seismic data from the Soda Lake geothermal field in Nevada, USA. Figure 11b–11d displays the fault-detection results produced using U-Net-1, U-Net-2, and our MACNN, respectively. Although all three methods produce a high-resolution fault prediction, the pattern and accuracy differ. The fault prediction produced by the U-Net-1 has a few places with

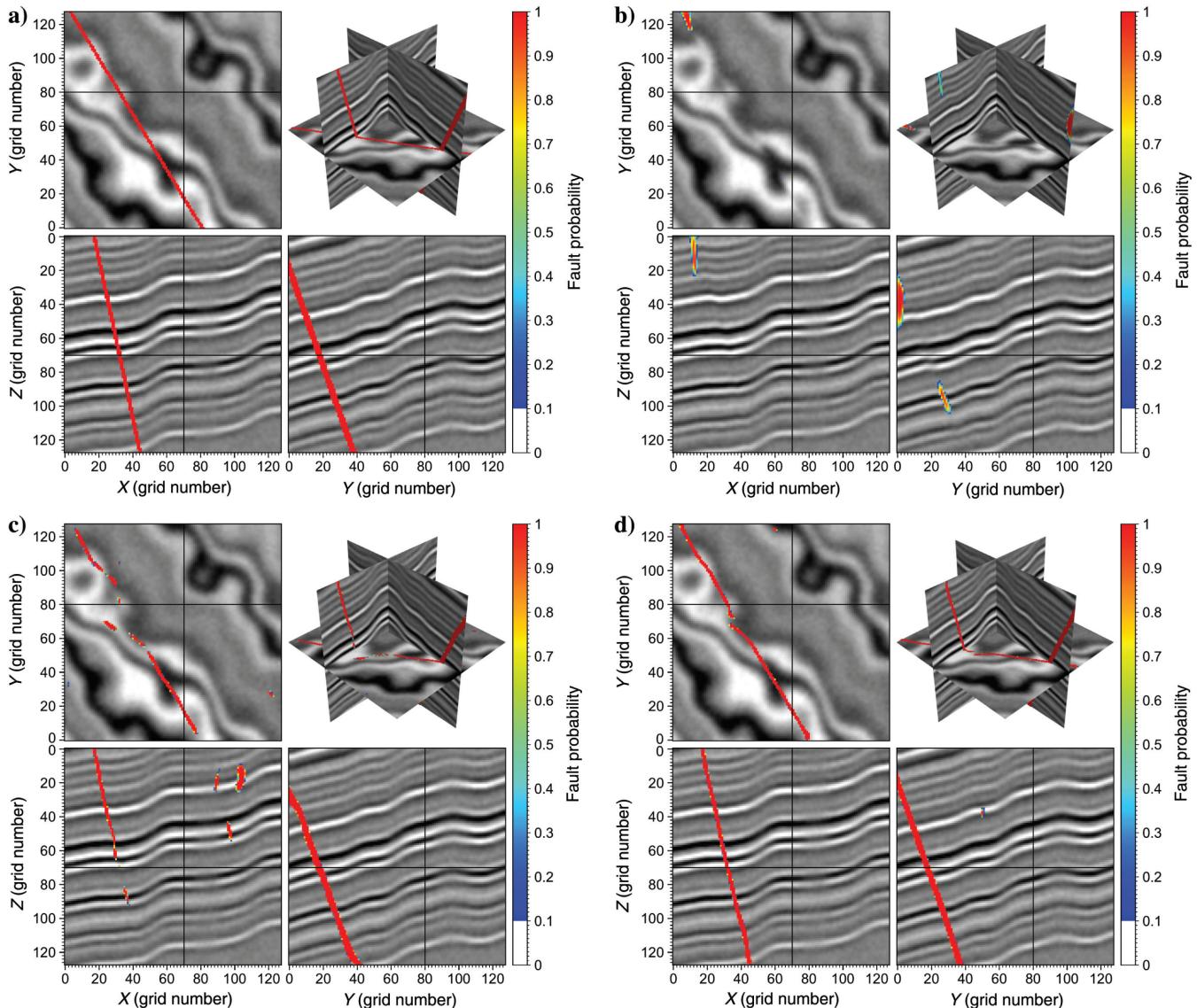


Figure 7. Comparison among faults detected on another 3D seismic image using (b) U-Net-1, (c) U-Net-2, and (d) our MACNN. (a) The ground truth fault overlying on the image.

low-probability fault pixels, where two or several faults smear into each other, including at the positions $(x, z) = (1.8, 0.5)$, $(2.25, 0.7)$, and $(3.4, 0.8)$ km. In addition, we use three ellipses to highlight the regions with faults. Visually, the predicted faults produced by MACNN in these three regions are more continuous, and thus are more geologically interpretable, than those predicted by the other methods.

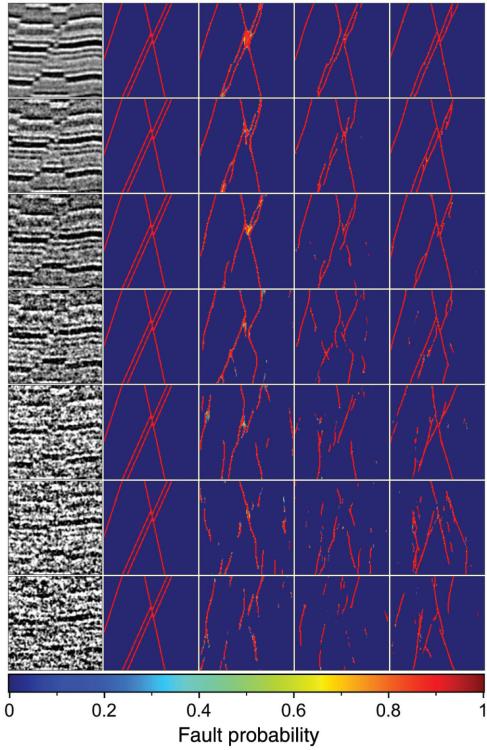


Figure 8. Comparison among faults detected using the U-Nets and our MACNN for a synthetic image with low to high levels of random noise. Rows 1–7 correspond to the same seismic image with seven different levels (low to high) of random noise. Columns 1–5 show (1) the seismic images, (2) the ground-truth faults, (3) the faults detected using U-Net-1, (4) the faults detected using U-Net-2, and (5) the faults detected using our MACNN, respectively.

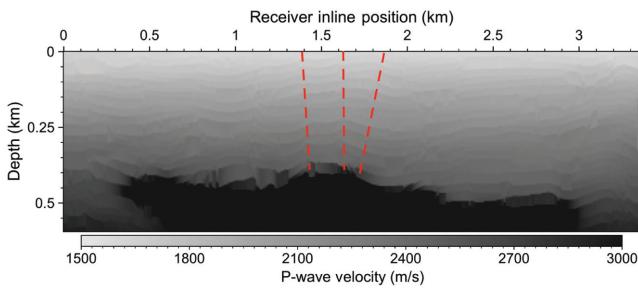


Figure 9. A 2D velocity model built using geologic features of the Soda Lake geothermal field, with the dashed red lines indicating three of the many implicit faults that we impose to the model during the model building. The maximum value of the P-wave velocity is 4800 m/s. We display the model with an upper clip value of 3000 m/s for better visualization.

Opunake image

In the second example, we compare the faults detected using the three aforementioned methods for the field data seismic image shown in Figure 12a. This 2D image is extracted from a New Zealand Opunake 3D seismic image, which is an open-source field seismic image hosted at the SEG Wiki website (SEG, 2020b). The image is sampled at 251 grid points in the vertical direction and 1001 grid points in the horizontal direction, and it contains a complex fault system that consists of several major faults and numerous small-scale faults. Except the six or seven major faults that can be easily picked and interpreted by an interpreter, almost all of the

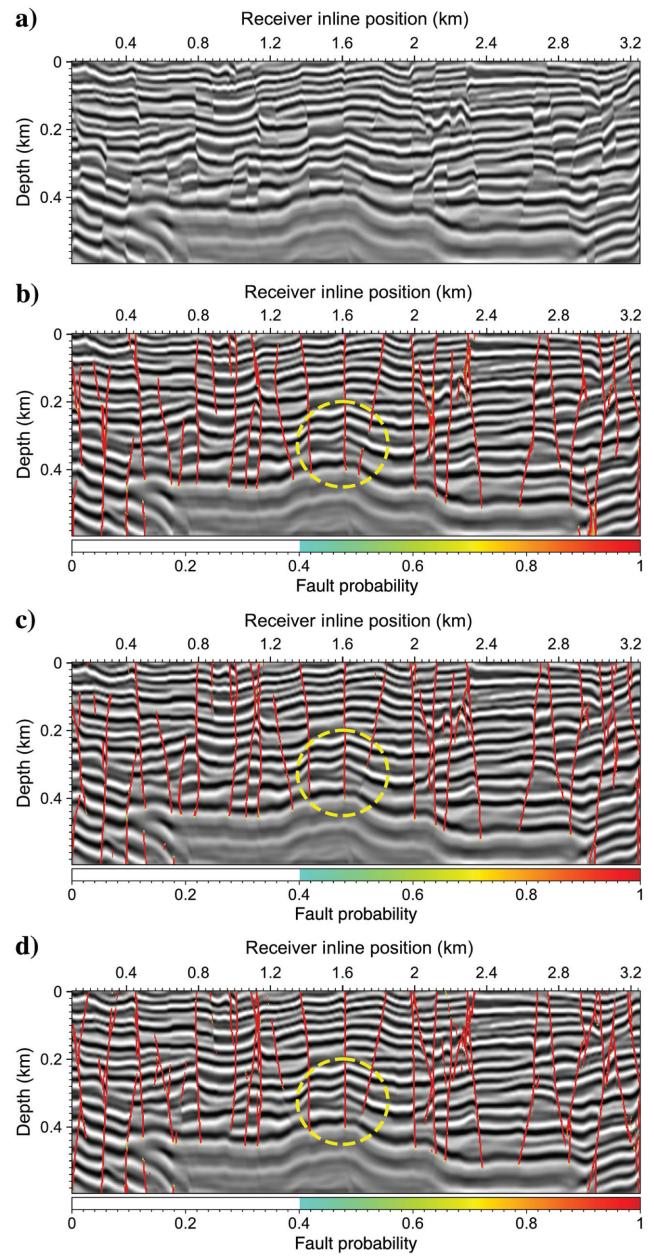


Figure 10. (a-d) The seismic image associated with the P-wave velocity model in Figure 9, the image overlaid by the detected faults from U-Net-1, U-Net-2, and our MACNN, respectively.

small-scale faults are very difficult, if not impossible, to pick on the image even with prior geologic knowledge.

Figure 12b–12d displays the faults detected using the U-Net-1, U-Net-2, and our MACNN, respectively, overlying the seismic image. In the region highlighted by the cyan ellipse, the faults predicted by U-Net-1 smear into each other and they have relatively lower probabilities compared with the other fault pixels of the prediction result. U-Net-2, trained with batch normalization and dice loss, produces a cleaner image in this region, but a short fault to the right of the major fault seems to be a false prediction. In contrast, our MACNN prediction for this region is of high resolution and clean.

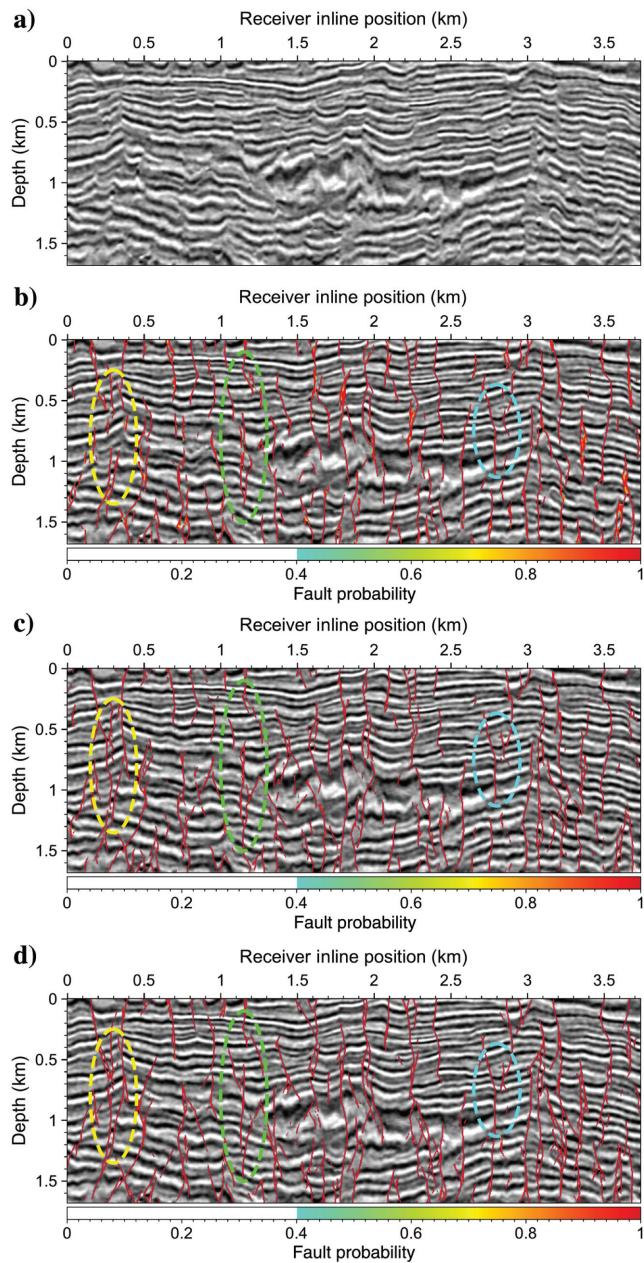


Figure 11. (a) A 2D slice of the 3D seismic image produced using field seismic data and 3D reverse time migration (Gao et al., 2020). (b-d) The same image overlaid by the faults detected by U-Net-1, U-Net-2, and our MACNN, respectively.

The comparison among the prediction results in the green rectangle region resembles that in the cyan ellipse. Our MACNN produces the visually best and the most geologically plausible fault map among the three results. For the yellow rectangle region, the three results seem fairly close, yet U-Net-1 predicts a long fault beneath the left major

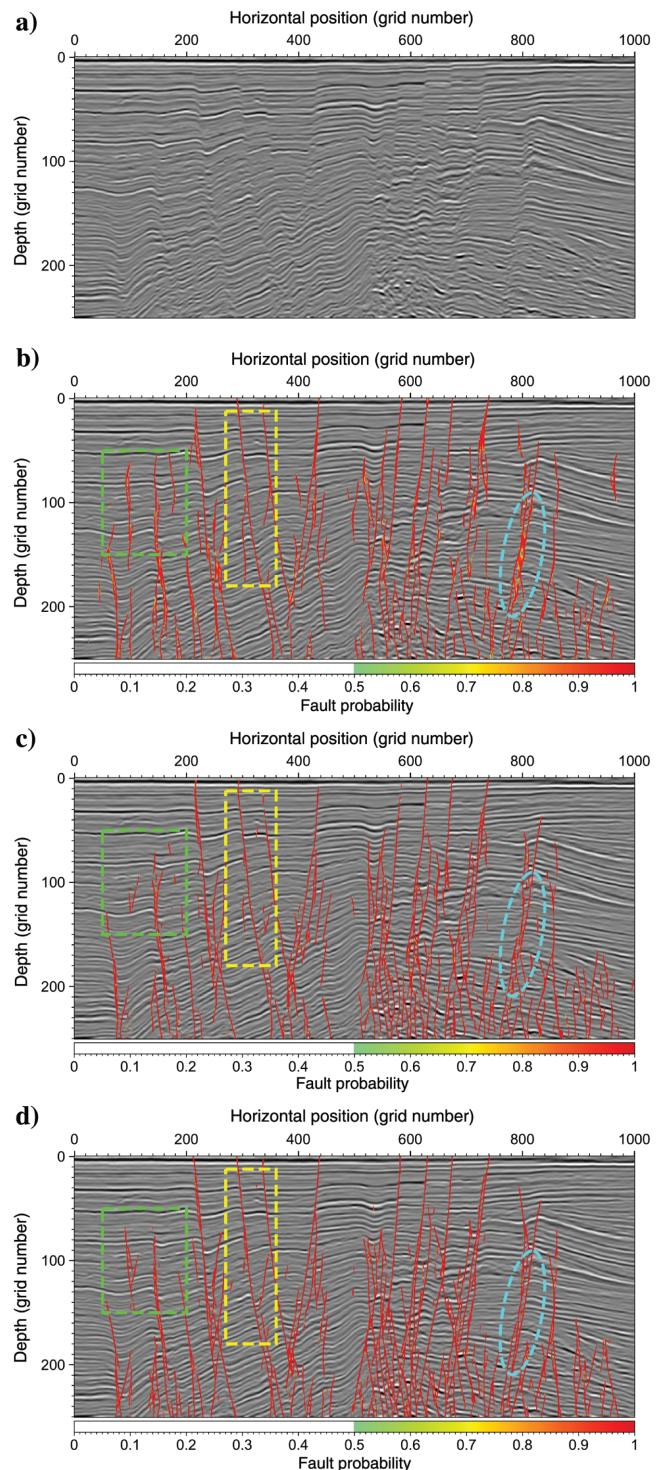


Figure 12. (a) A 2D slice of the 3D Opunake field seismic image. (b-d) The image in (a) overlaid by the faults predicted using U-Net-1, U-Net-2, and our MACNN, respectively.

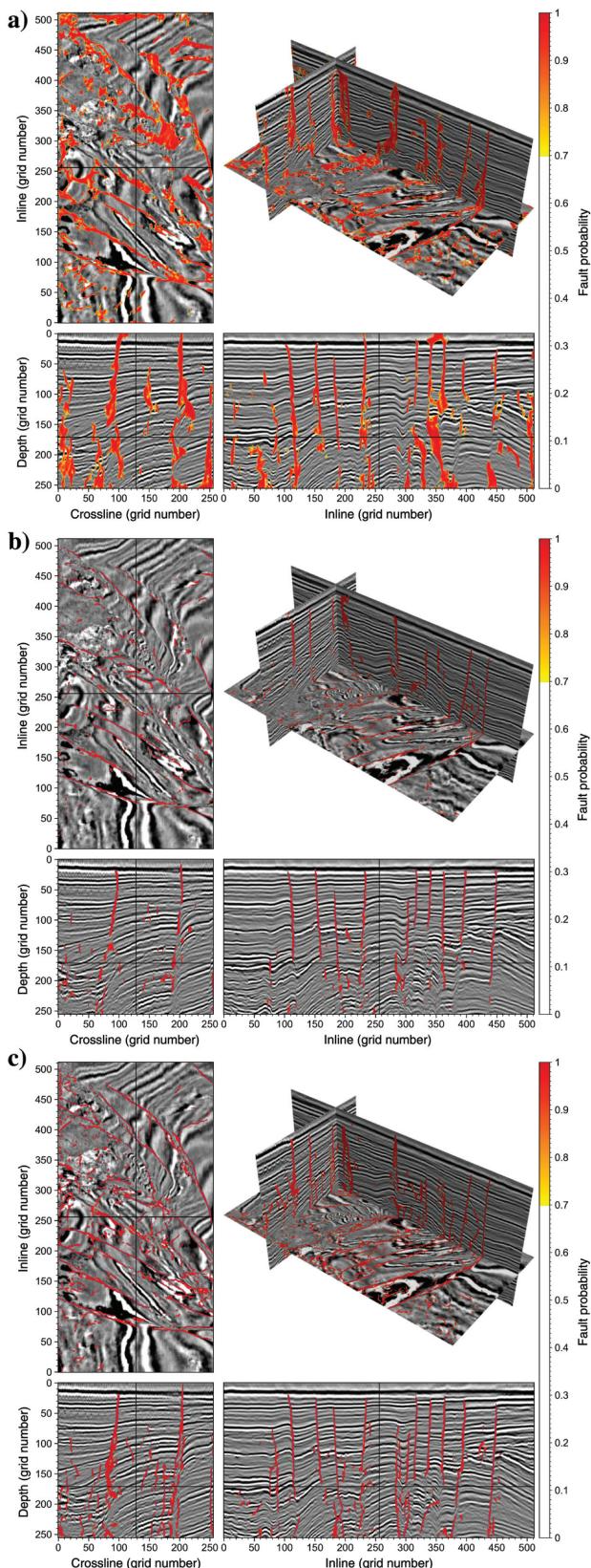


Figure 13. The upper portion of the 3D Opunake field seismic image overlaid by the faults predicted using (a) U-Net-1, (b) U-Net-2, and (c) our MACNN.

fault, which may not be particularly geologically plausible because we do not see a typical lateral discontinuity associated with one or several faults. U-Net-2 predicts a fault on the right with a partially broken path, which is not particularly geologically plausible. In contrast, our MACNN generates a fault map that is the most geologically interpretable among the three results, revealing all possible major and minor faults, while avoiding excessive detection.

Furthermore, we validate the efficacy and accuracy of our MACNN for detecting faults on the 3D Opunake image. We select the upper part of the entire 3D image, and the selected region is sampled at a total of $256 \times 512 \times 256$ grid points. Figure 13a–13c displays the 3D image overlaid by the faults predicted using U-Net-1, U-Net-2, and our MACNN, respectively. The fault map generated by U-Net-1 is visually of lower spatial resolution compared with those produced by U-Net-2 and our MACNN. This indicates that using a batch normalization strategy and the dice loss function can improve the resolution and reduce the uncertainty of fault prediction. The fault map produced by U-Net-2 misses some fairly obvious faults at the lower part of the image. In contrast, our MACNN produces a relatively higher resolution and higher probability fault map compared with the two U-Net architectures, particularly in the deeper part of the 3D image where the signal-to-noise ratios are relatively lower than those in the upper and middle regions.

Kerry image

In the third example, we compare the predicted faults by the three architectures on a New Zealand Kerry image publicly available at the SEG Wiki ([SEG, 2020a](#)).

We first perform fault detection on a 2D image shown in Figure 14, which is extracted from the 3D seismic image volume. The image is sampled at 310 grid points in the vertical direction and 690 grid points in the horizontal direction. The image is locally weighted so that it is spatially more balanced compared with the original image. The upper half of the image develops several major normal faults with opposite tilt angles with respect to the horizontal center. In the deeper region of the model, the image quality is relatively low, and we observe a few small-scale faults that cut through three or four major deep reflectors.

We display the faults detected using the two U-Net structures and our MACNN in Figure 14b–14d, respectively. Overall, there are more faults on the U-Net-1 fault map than on the U-Net-2 and MACNN-generated fault maps. U-Net-1 seems to be oversensitive to this image. For example, in the upper half of the image, the fault map produced using U-Net-1 appears to have several short faults. These faults are generally isolated in the image and do not have a specific orientation that is related to the dominant dip of the major faults, making them look like artifacts. In contrast, the fault maps produced using U-Net-2 and our MACNN in Figure 14c and 14d are evidently cleaner than the U-Net-1 result. Further comparing U-Net-2 and our MACNN fault maps, we find that our MACNN not only detects all major faults but also produces more continuous and less crooked faults compared to U-Net-2; thus, the MACNN fault map potentially facilitates more reliable seismic interpretation.

Furthermore, we perform 3D fault detection on the upper part of 3D Kerry image volume and we compare the fault-detection results of the three methods in Figure 15b–15d. The comparison on the 3D fault-detection results resembles that for the Opunake

3D image in Figure 13. We find that the fault map produced using U-Net-1 is of relatively lower resolution compared with the other two results. Meanwhile, the fault map produced using U-Net-2 misses some visually obvious faults compared to the result

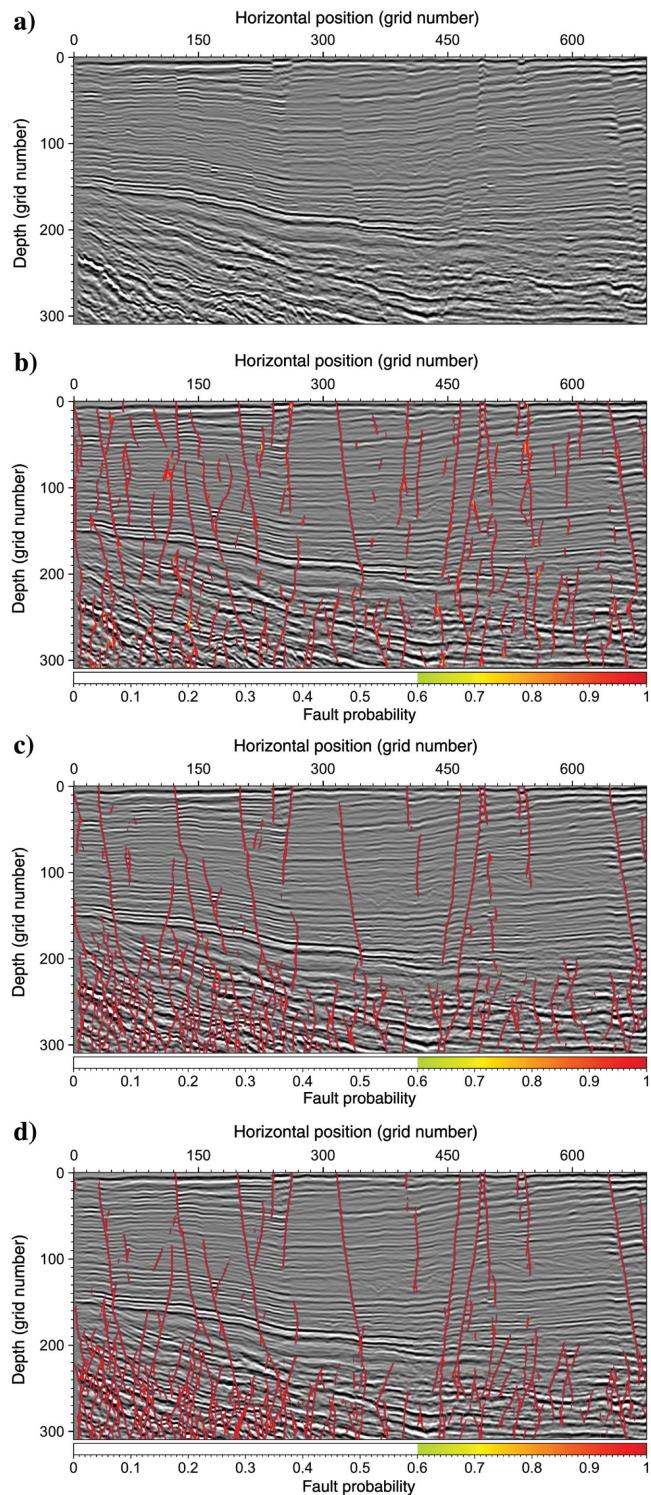


Figure 14. (a) A 2D slice of the 3D Kerry field seismic image. (b-d) The image in (a) overlaid by the faults predicted using U-Net-1, U-Net-2, and our MACNN, respectively.

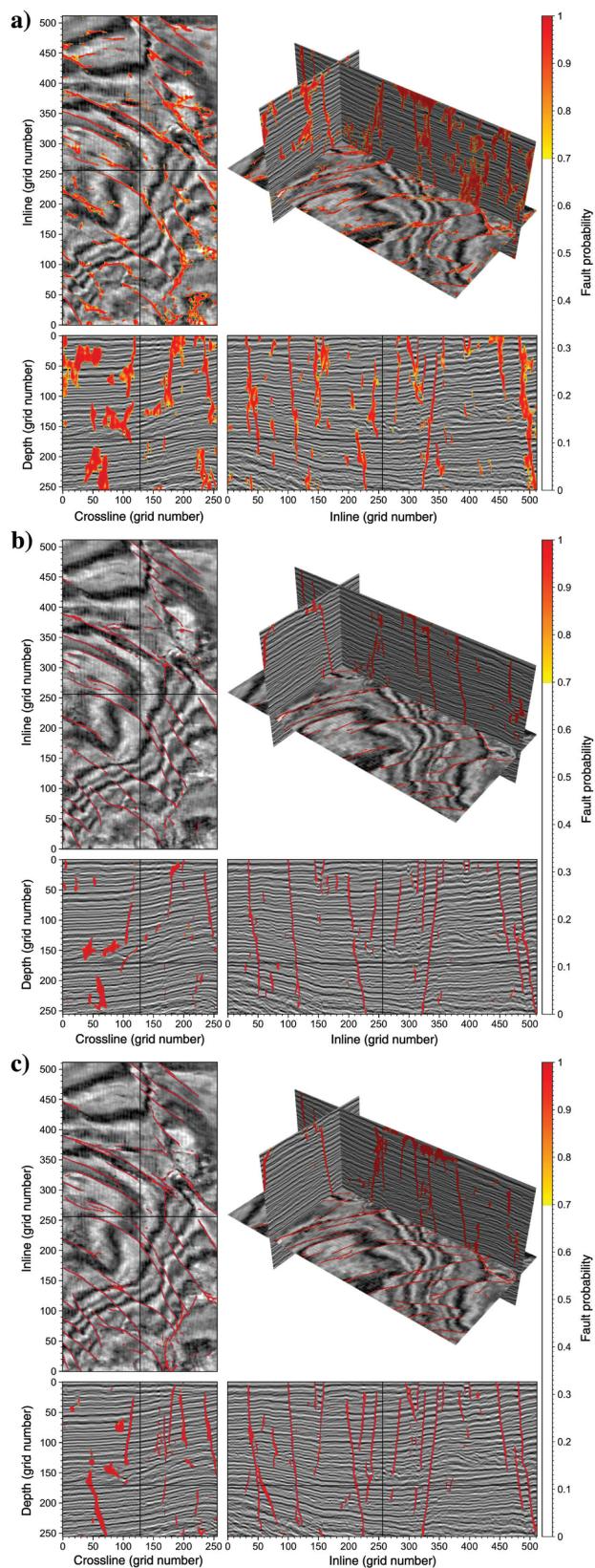


Figure 15. The upper to middle portion of the 3D Kerry field seismic image overlaid by the faults predicted using (a) U-Net-1, (b) U-Net-2, and (c) our MACNN.

produced using our MACNN, particularly on the inline-depth slice plane. The comparison indicates that our MACNN architecture reduces the uncertainties for 3D fault detection compared to the conventional U-Net architectures.

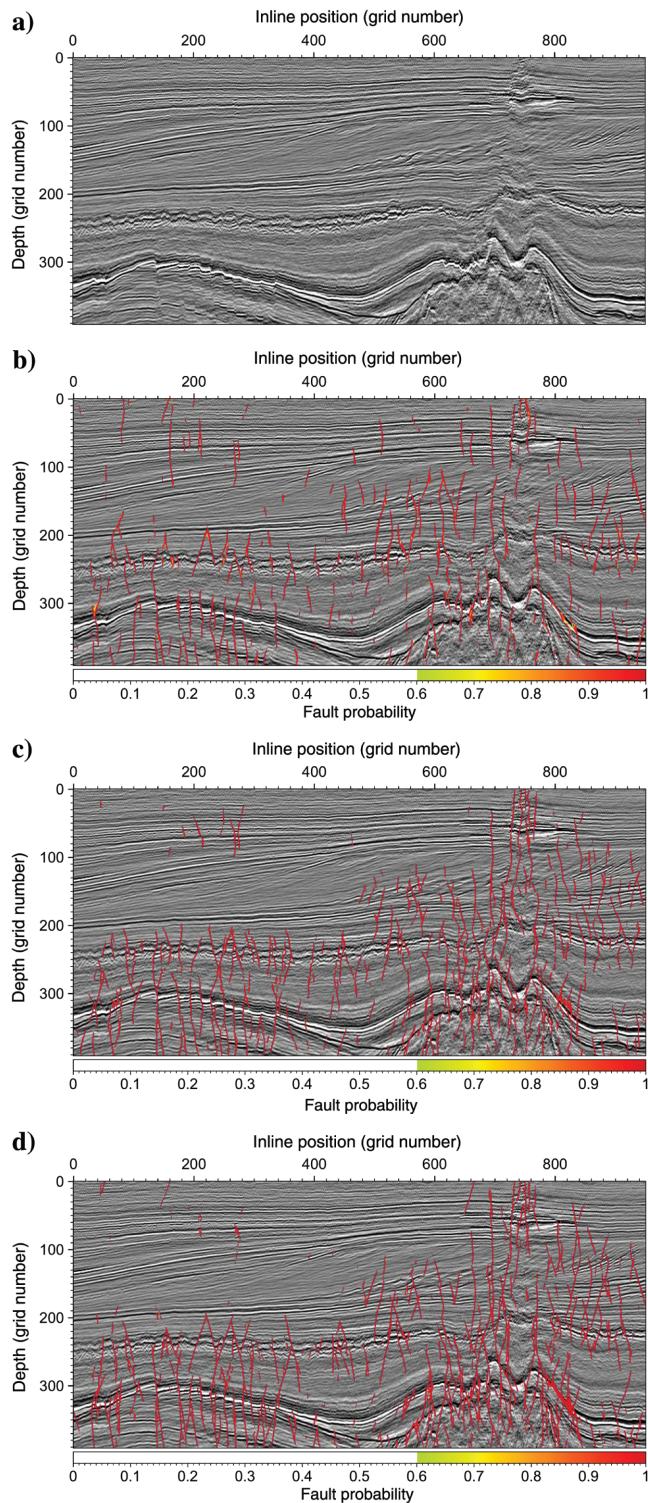


Figure 16. (a) An 2D inline slice of the North Sea 3D image. (b-d) The image in (a) overlaid by the faults predicted using U-Net-1, U-Net-2, and our MACNN, respectively.

North Sea F3 image

In the last example, we validate our MACNN on two field seismic images extracted from the 3D North Sea F3 seismic image volume.

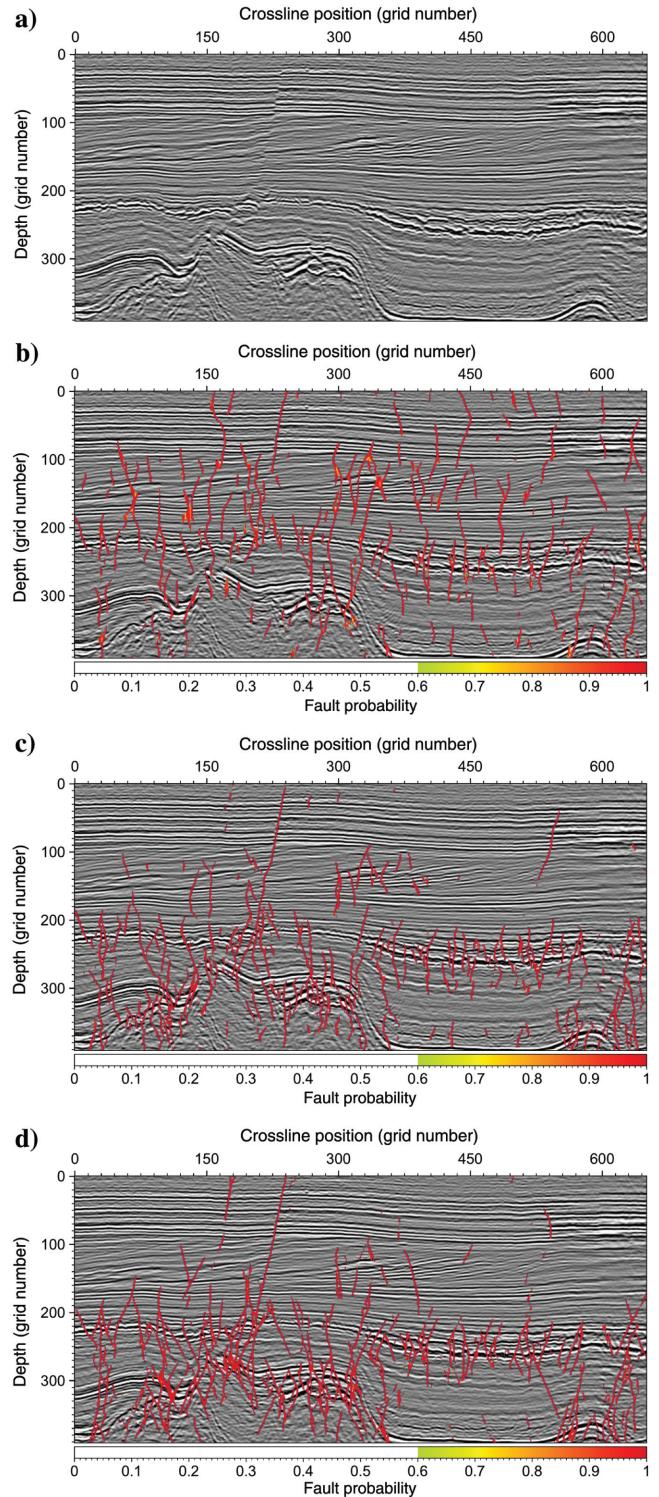


Figure 17. (a) An 2D crossline slice of the North Sea 3D image. (b-d) The image in (a) overlaid by the faults predicted using U-Net-1, U-Net-2, and our MACNN, respectively.

F3 is a geologic block in the Dutch sector of the North Sea, which was investigated using 3D seismics for oil and gas in the Upper Jurassic to Lower Cretaceous strata. A 3D seismic image volume associated with this area is made publicly available by TerraNubis ([TerraNubis, 2020](#)).

Figure 16a–16d displays a sliced 2D inline seismic image, the image overlaid by the faults predicted using U-Net-1, U-Net-2, and our MACNN, respectively. Similarly, Figure 17a–17d displays a sliced 2D crossline seismic image, the image overlaid by the faults predicted using U-Net-1, the U-Net-2, and our MACNN, respectively. We do not apply image processing to the sliced 2D images. Therefore, they contain some random noise. In Figures 16 and 17, we find that the U-Net-1 method can be oversensitive and produces numerous scattered faults that are unlikely to be true, particularly in the upper half of the images. The U-Net-2 method clearly reduces excessive detection compared to U-Net-1. Our MACNN produces more plausible fault maps compared to U-Net-2. For example, by comparing the U-Net-2 detection result in Figure 17c with our MACNN detection result in Figure 17d, we find that the faults in our MACNN detection result are generally straighter and thus more geologically plausible and interpretable. Geologically, crooked or curved faults can be more difficult to associate with a conventional geologic evolution interpretation than straight faults.

DISCUSSION

Fault detection using the U-Net and its derived or improved architectures (e.g., the MACNN developed in this paper) is essentially detecting the boundaries of “anomalies” with respect to the

“background.” Therefore, although all three architectures compared in this work are fully CNNs and can directly detect faults on 2D or 3D images of any size, the resolution of the input seismic image inevitably affects the detection result. The resolution mentioned here does not refer to the vertical resolution of reflectors in the sense of seismic migration, but to the grid sampling interval (or equivalently the ratio of the spatial dimensions) of a seismic image.

For instance, given a seismic image with an original dimension of $N_1 \times N_2 \times N_3$, the fault-detection result using the trained U-Net or our MACNN can be significantly different if we resample the image using the best available resampling algorithms to be such as $(2N_1) \times (4N_2) \times (8N_3)$, or $((1/2)N_1) \times ((1/4)N_2) \times ((1/8)N_3)$, etc. We take a field seismic image shown in Figure 16a as an example to analyze the effect of image sampling on fault detection. Here, we take our MACNN as an example to explain the influence of image resolution on the final detection result.

Figure 18 displays a total of nine fault-detection results generated using our MACNN on this image using three different vertical sampling intervals and three different horizontal sampling intervals. When the vertical and horizontal sampling intervals are large (Figure 18a), few faults are detected and the faults tend to be long and continuous, whereas at the smallest sampling intervals (Figure 18i), numerous faults are detected and the faults tend to be short and scattered. In general, using a smaller horizontal sampling interval to represent an image results in a higher horizontal resolution of the detected faults, whereas using a smaller vertical sampling interval to represent an image results in shorter faults. From the perspective of geologic interpretation, it is favorable to avoid fault-detection results such as those in Figure 18c, 18d, 18f, 18g, and 18i because these

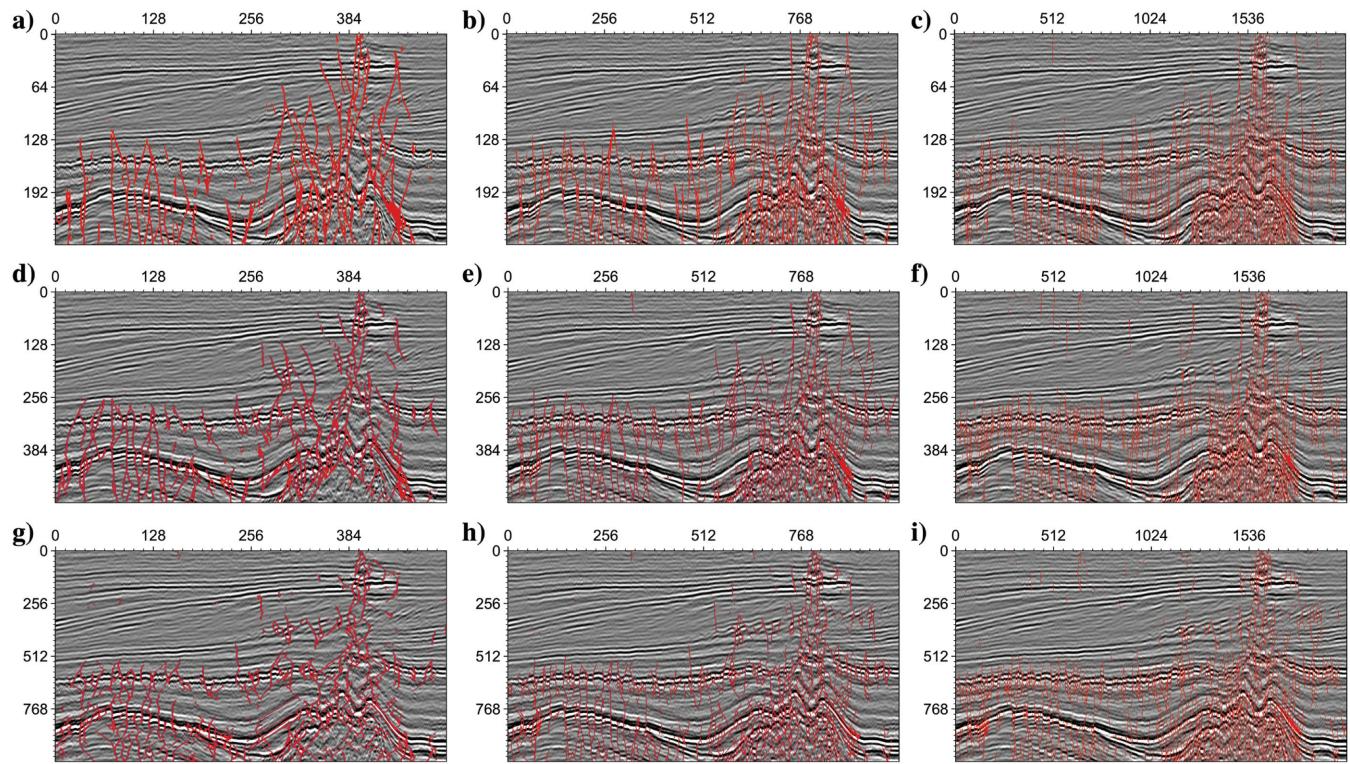


Figure 18. Fault-detection results by our MACNN overlying on a field image sampled by different sample intervals in the vertical and horizontal directions. Rows 1–3 have 256, 512, and 1024 grid samples along the vertical direction, respectively. Columns 1–3 have 512, 1024, and 2048 grid samples along the horizontal direction, respectively.

fault-detection results are likely to cause difficulties in deriving a plausible geologic evolution model.

Therefore, it is important in practice to prepare a suitably sampled seismic image for a machine-learning algorithm to predict faults. This requirement can be achieved using two approaches: First, one can resample a given seismic image to have a suitable horizontal and vertical resolution for fault detection. Second, one can filter a given image using, for example, simple wavenumber-domain filtering so that the image has a wavenumber spectrum similar to that of the training images in a statistical sense. In the examples shown in the previous section, we have prepared our images using one or both of these two approaches. We also foresee that any future machine-learning algorithm that can overcome such restrictions and be resolution-independent will likely become an important next-generation fault-detection method.

CONCLUSION

We have developed a novel fault-detection method using an MACNN. The neural network uses a multiscale spatial-channel attention mechanism, triple-convolution blocks, and a smoothed dice loss function to enhance the capability of machine-learning-based fully automatic fault detection. We train our neural network using 2D and 3D synthetic seismic images. The trained neural network can directly generalize to field seismic images without further training or adjustment. We demonstrate the improved fault-detection capability of our method compared with two conventional U-Net architectures using several synthetic and field data seismic images. We observe that our new method can produce higher resolution, higher fidelity fault-detection results even for noisy, complex field seismic images, leading to potentially improved geologic interpretability compared with conventional architectures.

ACKNOWLEDGMENTS

This work was supported by the U.S. Department of Energy (DOE) Office of Energy Efficiency and Renewable Energy through the Los Alamos National Laboratory (LANL). LANL is operated by Triad National Security, LLC, for the U.S. DOE National Nuclear Security Administration (NNSA) under contract no. 89233218CNA000001. This research used computing resources provided by the LANL Institutional Computing Program supported by the U.S. DOE NNSA under contract no. 89233218CNA000001. We thank Z. Jiang of Texas A&M University for the helpful discussions. We thank A. Malcolm, H. Di, and three anonymous reviewers for their careful and insightful comments that greatly improved the paper.

DATA AND MATERIALS AVAILABILITY

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

REFERENCES

- Bakker, P., 2002, Image structure analysis for seismic interpretation: Ph.D. thesis, Delft University of Technology.
- Chollet, F., 2017, Deep learning with Python: Manning Publications.
- Cohen, I., N. Coulter, and A. A. Vassiliou, 2006, Detection and extraction of fault surfaces in 3D seismic data: *Geophysics*, **71**, no. 4, P21–P27, doi: [10.1190/1.2215357](https://doi.org/10.1190/1.2215357).
- Cunha, A., A. Pochet, H. Lopes, and M. Gattass, 2020, Seismic fault detection in real data using transfer learning from a convolutional neural network pre-trained with synthetic seismic data: *Computers & Geosciences*, **135**, 104344, doi: [10.1016/j.cageo.2019.104344](https://doi.org/10.1016/j.cageo.2019.104344).
- Di, H., C. Li, S. Smith, and A. Abubakar, 2019, Machine learning-assisted seismic interpretation with geologic constraints: 89th Annual International Meeting, SEG, Expanded Abstracts, 5360–5364, doi: [10.1190/segam2019-w4-01.1](https://doi.org/10.1190/segam2019-w4-01.1).
- Di, H., M. Shafiq, and G. AlRegib, 2018, Patch-level MLP classification for improved fault detection: 88th Annual International Meeting, SEG, Expanded Abstracts, 2211–2215, doi: [10.1190/segam2018-2996921.1](https://doi.org/10.1190/segam2018-2996921.1).
- Falcon, W., 2019, Pytorch lightning: GitHub, github.com/PyTorchLightning/pytorch-lightning, accessed 12 January 2020.
- Fossen, H., 2016, Structural geology, 2nd ed.: Cambridge University Press.
- Gan, Q., and D. Elsworth, 2014, Analysis of fluid injection-induced fault reactivation and seismic slip in geothermal reservoirs: *Journal of Geophysical Research, Solid Earth*, **119**, 3340–3353, doi: [10.1002/2013JB010679](https://doi.org/10.1002/2013JB010679).
- Gao, K., and L. Huang, 2019, An efficient vector elastic reverse time migration method in the hybrid time and frequency domain for anisotropic media: *Geophysics*, **84**, no. 6, S511–S522, doi: [10.1190/geo2018-0644.1](https://doi.org/10.1190/geo2018-0644.1).
- Gao, K., L. Huang, and T. Cladouhos, 2020, Three-dimensional seismic characterization and imaging of the Soda Lake geothermal field: *Geothermics*, **90**, 101996, doi: [10.1016/j.geothermics.2020.101996](https://doi.org/10.1016/j.geothermics.2020.101996).
- Gao, K., L. Huang, and Y. Zheng, 2021, Fault detection on seismic structural images using a nested residual U-Net: *IEEE Transactions on Geoscience and Remote Sensing*, 1–15, doi: [10.1109/TGRS.2021.3073840](https://doi.org/10.1109/TGRS.2021.3073840).
- Guo, B., L. Li, and Y. Luo, 2018, A new method for automatic seismic fault detection using convolutional neural network: 88th Annual International Meeting, SEG, Expanded Abstracts, 1951–1955, doi: [10.1190/segam2018-2995894.1](https://doi.org/10.1190/segam2018-2995894.1).
- Hardman, R. F. P., and J. E. Booth, 1991, The significance of normal faults in the exploration and production of North Sea hydrocarbons: Geological Society, London, Special Publications 56, 1–13.
- Huang, H., L. Lin, R. Tong, H. Hu, Q. Zhang, Y. Iwamoto, X. Han, Y.-W. Chen, and J. Wu, 2020, UNet 3+: A full-scale connected UNet for medical image segmentation: arXiv:2004.08790.
- Jadon, S., 2020, A survey of loss functions for semantic segmentation: IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB), 1–7.
- Khanh, T. L. B., D. Dao, N.-H. Ho, H.-J. Yang, E.-T. Baek, G. Lee, S.-H. Kim, and S. Yoo, 2020, Enhancing U-Net with spatial-channel attention gate for abnormal tissue segmentation in medical imaging: *Applied Sciences*, **10**, 5729, doi: [10.3390/app10175729](https://doi.org/10.3390/app10175729).
- Kingma, D. P., and J. Ba, 2017, Adam: A method for stochastic optimization: arXiv:1412.6980.
- LeCun, Y., L. Bottou, Y. Bengio, and P. Haffner, 1998, Gradient-based learning applied to document recognition: *Proceedings of the IEEE*, **86**, 2278–2324, doi: [10.1109/5.726791](https://doi.org/10.1109/5.726791).
- Loshchilov, I., and F. Hutter, 2017, SGDR: Stochastic gradient descent with warm restarts: *Proceedings of the International Conference on Learning Representations*, 1–16.
- 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, doi: [10.1190/1.1444415](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, doi: [10.1190/1.1444508](https://doi.org/10.1190/1.1444508).
- Paszke, A., S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, 2019, PyTorch: An imperative style, high-performance deep learning library: *Advances in Neural Information Processing Systems* 32, Curran Associates, Inc., 8024–8035.
- Pedersen, S. I., T. Randen, L. Sonnenland, and Ø. Steen, 2005, Automatic fault extraction using artificial ants: 75th Annual International Meeting, SEG, Expanded Abstracts, 512–515, doi: [10.1190/1.1817297](https://doi.org/10.1190/1.1817297).
- Qin, X., Z. Zhang, C. Huang, M. Dehghan, O. R. Zaiane, and M. Jagersand, 2020, U2-Net: Going deeper with nested U-structure for salient object detection: *Pattern Recognition*, **106**, 107404, doi: [10.1016/j.patcog.2020.107404](https://doi.org/10.1016/j.patcog.2020.107404).
- Ronneberger, O., P. Fischer, and T. Brox, 2015, U-Net: Convolutional networks for biomedical image segmentation: *Proceedings of the Medical Image Computing and Computer-Assisted Intervention*, 234–241.
- SEG, 2020a, New Zealand Kerry 3D data: SEG Wiki, wiki.seg.org/wiki/Kerry-3D, accessed 12 January 2020.
- SEG, 2020b, New Zealand Opunake 3D data: SEG Wiki, wiki.seg.org/wiki/Opunake-3D, accessed 12 January 2020.
- Sudre, C. H., W. Li, T. Vercauteren, S. Ourselin, and M. Jorge Cardoso, 2017, Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations: *Deep Learning in Medical Image Analysis*

- and Multimodal Learning for Clinical Decision Support, Springer International Publishing, 240–248.
- Sultana, F., A. Sufian, and P. Dutta, 2020, Evolution of image segmentation using deep convolutional neural network: A survey: *Knowledge-Based Systems*, **201–202**, 106062, doi: [10.1016/j.knosys.2020.106062](https://doi.org/10.1016/j.knosys.2020.106062).
- TerraNubis, 2020, North Sea F3 3D seismic data and image: Public data, terranubis.com/datainfo/F3-Demo-2020, accessed 12 January 2020.
- Vilarrasa, V., and J. Carrera, 2015, Geologic carbon storage is unlikely to trigger large earthquakes and reactivate faults through which CO₂ could leak: *Proceedings of the National Academy of Sciences*, **112**, 5938–5943, doi: [10.1073/pnas.1413284112](https://doi.org/10.1073/pnas.1413284112).
- Virieux, J., 1986, P-SV wave propagation in heterogeneous media: Velocity-stress finite difference method: *Geophysics*, **51**, 889–901, doi: [10.1190/1.1442147](https://doi.org/10.1190/1.1442147).
- Watt, J., R. Borhani, and A. K. Katsaggelos, 2020, Machine learning refined: Foundations, algorithms, and applications, 2nd ed.: Cambridge University Press.
- 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](https://doi.org/10.1190/geo2016-0473.1).
- Wu, X., and S. Fomel, 2018, Automatic fault interpretation with optimal surface voting: *Geophysics*, **83**, no. 5, O67–O82, doi: [10.1190/geo2018-0115.1](https://doi.org/10.1190/geo2018-0115.1).
- Wu, X., and Z. Guo, 2018, Detecting faults and channels while enhancing seismic structural and stratigraphic features: *Interpretation*, **7**, no. 1, T155–T166, doi: [10.1190/INT-2017-0174.1](https://doi.org/10.1190/INT-2017-0174.1).
- Wu, X., L. Liang, Y. Shi, and S. Fomel, 2019, 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, doi: [10.1190/geo2018-0646.1](https://doi.org/10.1190/geo2018-0646.1).
- Xiong, W., X. Ji, Y. Ma, Y. Wang, N. M. AlBinHassan, M. N. Ali, and Y. Luo, 2018, Seismic fault detection with convolutional neural network: *Geophysics*, **83**, no. 5, O97–O103, doi: [10.1190/geo2017-0666.1](https://doi.org/10.1190/geo2017-0666.1).
- Zhou, Z., M. M. R. Siddiquee, N. Tajbakhsh, and J. Liang, 2020, UNet++: Redesigning skip connections to exploit multiscale features in image segmentation: arXiv:1912.05074.

Biographies and photographs of the authors are not available.