

FaultNet3D: Predicting Fault Probabilities, Strikes, and Dips With a Single Convolutional Neural Network

Xinming Wu¹, Yunzhi Shi¹, Sergey Fomel, Luming Liang¹, Qie Zhang, and Anar Z. Yusifov

Abstract—We simultaneously estimate fault probabilities, strikes, and dips directly from a seismic image by using a single convolutional neural network (CNN). In this method, we assume a local 3-D fault is a plane defined by a single combination of strike and dip angles. We assume the fault strikes and dips, respectively, are in the ranges of $[0^\circ, 360^\circ)$ and $[64^\circ, 85^\circ)$, which are divided into 577 classes corresponding to the situation of no fault and 576 different combinations of strikes and dips. We construct a 7-layer CNN to classify the fault strike and dip in a local seismic cube and obtain the classification probability at the same time. With the fault probability, strike and dip estimated at some seismic pixel, we further compute a fault cube (centered at the pixel) with fault features elongated along the fault plane. By sliding the classification window within a full seismic image, we are able to obtain a lot of overlapping fault cubes which are stacked to compute three full images of enhanced and continuous fault probabilities, strikes, and dips. To train the CNN model, we propose an effective and efficient workflow to automatically create 900 000 synthetic seismic cubes and the corresponding fault class labels. Although trained with only synthetic data sets, our CNN model can be applied to accurately estimate fault probabilities, strikes, and dips within field seismic images that are acquired at totally different surveys. With the estimated three fault images, we further construct fault cells that are represented as small 3-D squares, each square is colored by fault probability and oriented by fault strike and dip. We recursively link the fault cells by following the fault strikes and dips to finally construct fault skins, which are simple linked data structures to represent fault surfaces.

Index Terms—Convolutional neural network (CNN), deep learning, fault, fault dip, fault probability, fault strike, seismic interpretation.

I. INTRODUCTION

INTERPRETING faults from seismic reflection images is a key step for building structural models [1], [2] and reservoir characterization [3], [4]. In a seismic image, the most dominant features are reflection events corresponding to geologic horizons [5]–[7] while faults are often recognized as

reflection discontinuities. Therefore, fault interpretation typically requires first computing a fault probability image from the seismic image so that the faults are highlighted while the reflections are removed. Such a fault probability image can be estimated from numerous types of seismic attributes that measure reflection discontinuity or continuity. Such attributes include semblance [8], [9], coherency [10]–[14], variance [15], [16], and gradient magnitude [17]. A fault probability image based on discontinuity measurement, however, often highlights other discontinuous seismic features including stratigraphic features (e.g., channels and unconformities) and noise as shown in Fig. 1(a)–(e). To further enhance fault features while suppressing the features unrelated to faults, several researchers [18]–[22] propose to apply fault-oriented averaging or smoothing in computing a fault probability image like the fault likelihood image [20], [21] shown in Fig. 1(f). Other researchers propose to directly enhance the fault features in a fault probability image by using ant tracking [23] or optimal surface voting [24] as shown in Fig. 1(g).

Calculating a fault probability image of highlighting fault positions is only a necessary step for fault interpretation. A further step of extracting individual fault surfaces from a fault probability image is still not trivial because the fault geometry in 3-D can be highly complicated and the 3-D fault surfaces are typically not on the seismic sampling grid. In addition to a fault probability image, fault surface extraction also requires estimating fault strikes and dips, which indicate the directions to grow the off-grid fault surfaces as discussed in [21], [22] and [24]. 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 [9], [25], [26] and plane-wave destruction [27], fail to estimate fault orientations. Therefore, some researchers [18]–[20], [22] 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 [Fig. 1(f)], especially in areas where the faults are close to each other. To improve the whole fault interpretation process of computing a better fault detection image [Fig. 1(h)] and extracting individual fault surfaces, we propose to use a convolutional neural network (CNN) [28], [29] to simultaneously estimate accurate fault probabilities, fault strikes, and dips, which are further used to automatically construct fault surfaces.

Manuscript received January 1, 2019; revised May 4, 2019; accepted June 20, 2019. Date of publication July 25, 2019; date of current version October 31, 2019. (Corresponding author: Xinming Wu.)

X. Wu is with the School of Earth and Space Sciences, University of Science and Technology of China, Hefei 230026, China (e-mail: xinwucwp@gmail.com)

Y. Shi and S. Fomel are with the Bureau of Economic Geology, The University of Texas at Austin, Austin, TX 78712 USA (e-mail: yzshi08@utexas.edu; sergey.fomel@beg.utexas.edu).

L. Liang is with the Applied Science Group, Microsoft, Redmond, WA 98052 USA (e-mail: llmpass@gmail.com).

Q. Zhang and A. Z. Yusifov are with BP America, Inc., Houston, TX 77079-2604 USA.

Color versions of one or more of the figures in this article are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TGRS.2019.2925003

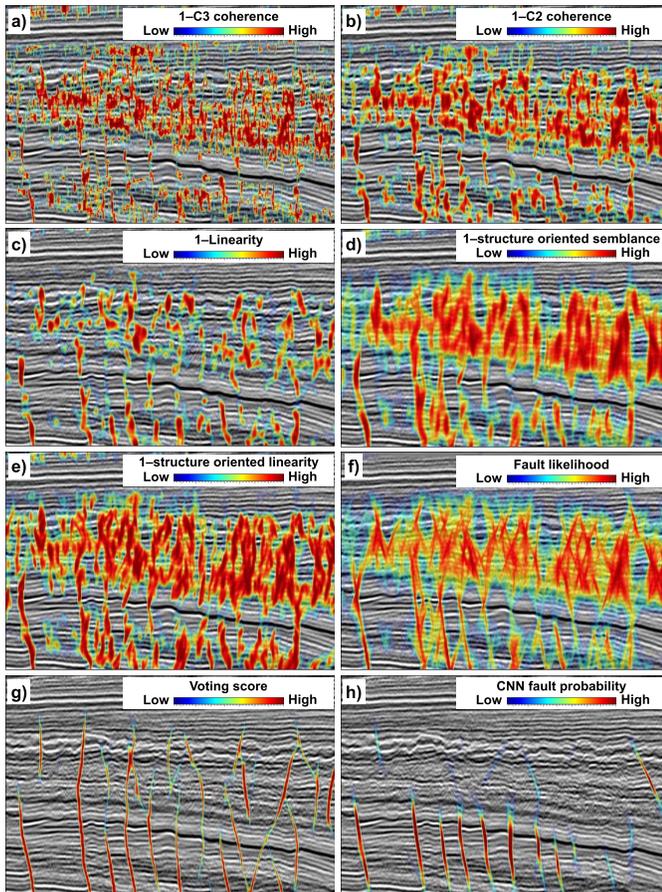


Fig. 1. Fault detection by using (a)–(g) seven commonly used methods and (h) our CNN-based method. Compared to the methods of (a) C3 [11], (b) C2 [10], (c) planarity [9], (d) structure-oriented planarity [14], and (e) structure-oriented semblance [9], (f) fault likelihood [20], [21], and (g) optimal surface voting [24] methods perform better fault detections. (h) Our CNN method achieves the best performance in obtaining an accurate and clean fault detection.

The CNN has been proven to be the most powerful method in solving computer vision problems including image classification [30]–[32], object detection [33]–[35], and segmentation [36], [37]. The image classification was previously a challenging problem for conventional classification methods based on image descriptors [38], [39]. By using the CNN, the error rate of classifying 1000 types of images on ImageNet 2015 benchmark was significantly reduced to only 3.57% [32], which is even lower than the error rate 5.1% of human classification [40] on the same benchmark data set. Recently, some CNN methods have been introduced to detect faults by pixelwise classification of fault or nonfault [41]–[46]. These methods, however, provide only a fault probability image as most of the conventional fault detection methods.

In this paper, we propose to use a single CNN to not only predict fault probabilities but also estimate fault strikes and dips at the same time. We consider the fault orientation estimation as an image classification problem by assuming a local 3-D fault within a small seismic image is a plane that is uniquely defined by a single combination of fault strike and dip as shown in Fig. 2. Inspired by the success of CNN in variant

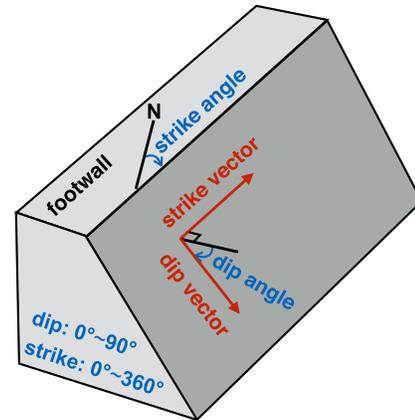


Fig. 2. 3-D fault geometry: a local 3-D fault plane is defined by a combination of strike and dip angles.

classification tasks, we propose to use CNN to classify fault orientations within the small seismic images. We construct a CNN model with six convolutional layers followed by a fully connected layer and a final output vector of a fault orientation prediction. To train the model, we propose an effective and efficient way (Fig. 3) to automatically create many synthetic seismic image cubes and the corresponding labels. Although trained with only synthetic seismic images, our CNN model can be applied to accurately estimate fault orientations in field seismic images that are acquired at variant surveys. With the estimated fault orientations, we then compute fault probability cubes with fault-oriented planar features. All the fault probability cubes share the same size and same sampling grid as the extracted seismic cubes. These probability cubes are further stacked to compute full images of fault probabilities, strikes, and dips. In stacking, the fault probability cubes (3-D) or patches (2-D) with fault-oriented features, the reliable predictions (often with consistent fault orientations) are enhanced while the unreliable predictions with inconsistent or outlier orientations will be suppressed. This stacking, therefore, helps improving the signal-to-noise ratio and spatial continuity of fault features in the final stacked fault probability image. This stacking also provides continuous strike and dip angles values in the finally stacked fault strike and dip images, where the angles are computed as weighted averages (weighted by the fault probability cubes) of the predicted fault angles. Using the three fault images, we finally automatically construct fault surfaces by using a linked data structure [21].

II. 2-D FAULT CLASSIFICATION

As shown in Fig. 2, the fault geometry is defined by a combination of fault strike and dip angles in 3-D space, which means that we should work on 3-D seismic images, instead of 2-D inline or crossline seismic images, to well interpret faults. For examples, faults can be unobvious in some 2-D seismic images that are nearly parallel to the faults or intersect the faults with small angles. However, to simplify the explanation of our CNN-based fault classification method, we start with the problem of 2-D fault classification in this

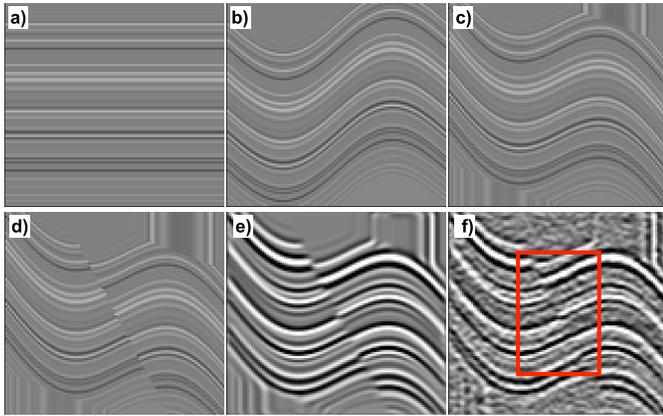


Fig. 3. Illustration of the main steps in creating a synthetic seismic image patch.

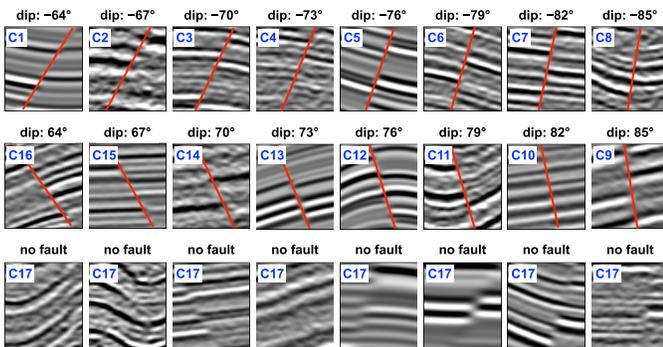


Fig. 4. Synthetic training images are labeled as 17 different classes (from C1 to C17) that correspond to 16 different fault dip angles and (Bottom row) class of no fault.

section and then discuss a 3-D CNN for 3-D fault classification in Section III.

A. 2-D Fault Classes

A 2-D fault is locally linear and therefore the fault geometry within a local 2-D window can be uniquely defined by a single apparent dip. As shown in Fig. 4, if the fault dips are known, then we can simply draw lines (red lines) along the dips to approximate the faults in the small seismic patches. Therefore, a key step for 2-D fault interpretation is to first estimate fault dips.

We assume the dip angles of 2-D 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 three dip angles as one fault class. For example, we assume the faults with dip angles -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 Fig. 4. We assume the faults, which do not pass through the center image pixel, all belong to the class of “no fault” [see Fig. 4 (bottom row)]. With these assumptions, we consider fault interpretation within the small seismic patches (Fig. 4) as a problem of fault dip classification by using CNN. To achieve this, the 17 different representative dip classes need to be first learned by a CNN

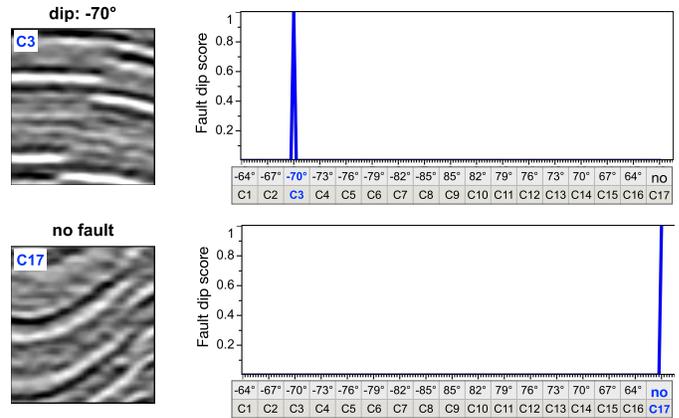


Fig. 5. (Left column) Synthetic training images and (Right column) corresponding label vectors.

model. We create a large number of synthetic seismic patches (Fig. 4) for each fault dip class to train the CNN model.

B. 2-D Training Data Sets

Training and validating a CNN model often requires a huge amount of images and corresponding labels. Manually labeling faults and estimating fault orientations would be extremely time-consuming and subjective. Therefore, we propose an effective and efficient way to create numerous synthetic seismic images and corresponding fault labels.

To create a seismic image, we first generate a horizontal reflectivity model $r(x, y)$ [Fig. 3(a)] with a sequence of random reflectivity values that are in the range of $[-1, 1]$. We then create a sinusoidal folding structure [Fig. 3(b)] by vertically shifting the model, where the vertical shifts are defined as a sine function

$$s_1(x, y) = a \sin(b + cx). \quad (1)$$

By randomly choosing combinations of the three parameters (a , b and c) from the predefined ranges, we are able to create a lot of models with unique folding structures. To further increase the complexity of the folding structure, we also apply some linear shearing [Fig. 3(c)]

$$s_2(x, y) = dx + e \quad (2)$$

where the two parameters d and e , again, are randomly chosen from some predefined ranges. Note that both the shift maps $s_1(x, y)$ and $s_2(x, y)$ are laterally varying but vertically invariant. With the two shift maps, we use a sinc interpolation to sequentially shift the reflectivity model $r(x, y)$ to obtain the folded models $r(x, y + s_1(x, y))$ [Fig. 3(b)] and $r(x, y + s_1 + s_2)$ [Fig. 3(c)].

After obtaining a folded model [Fig. 3(c)], we further add some linear faulting [Fig. 3(d)] in the model and the faulting can be defined by different dips and slips. The fault dip angle is randomly chosen from the range of $[-63^\circ, -86^\circ] \cup [86^\circ, 63^\circ]$. The fault throws (vertical components of slips) are randomly chosen from the range of between -30 and 30 samples.

After creating the folding and faulting in the reflectivity model, we then compute a synthetic seismic image [Fig. 3(e)] by convolving a Ricker wavelet with the model in directions perpendicular to the folding structures. The peak frequency of the Ricker wavelet is randomly chosen from a predefined range, which should include all possible frequencies in field seismic data sets. Note that the convolution is applied after (not before) the folding and faulting because the convolution can blur the sharp discontinuities near faults and, therefore, make the faults look more realistic. We further add some random noise [Fig. 3(f)] to increase the realism of the synthetic seismic image. The noise is simply additive white Gaussian noise and the rms signal-to-noise ratio is randomly chosen from the range between 0 and 0.8. To eliminate the boundary artifacts, we finally crop the center patch [red box in Fig. 3(f)] 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 200000 training images and 20000 validating images used in this paper. All the training and validating images are normalized so that the amplitude values in each image are in the range of $[-1, 1]$. Fig. 4 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 the class of no fault. We created about 10000 training images for each fault dip class and about 40000 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 interpretation. On the other hand, classifying a fault image as a wrong but neighboring fault dip class would not be a significant error.

Labeling the synthetic seismic patches is straightforward because we know exactly how the faults are oriented within the patches. Fig. 5 shows two representative seismic patches with [Fig. 5(a)] and without [Fig. 5(b)] faults, where we label each seismic patch by using a vectors of 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 0 elsewhere. For the image patches without faults, we assign the value of 1 to the last element in the label vector.

C. 2-D CNN Architecture

With an input seismic patch (48×32 pixels), we classify the corresponding fault dip class by using a CNN model (Fig. 6) with six 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 2 layers (COV3, COV4) and the last 2 layers (COV5, COV6), respectively, contain 32 filters and 64 filters with the same size of 3×3 and stride of 1. Following [47], we apply the batch normalization to the output feature maps

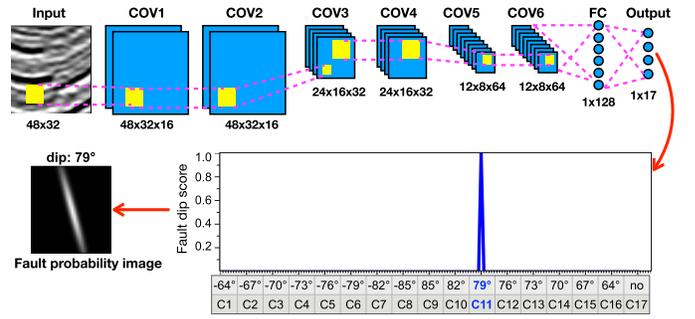


Fig. 6. Outline of the 2-D CNN architecture used to predict the fault class (defined by fault dip) from an input seismic amplitude patch.

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 [30]. 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 six convolutional layers, we flatten out the 2-D feature maps into a 1-D flat, fully connected (FC) layer 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 is a vector of 17 components and each component represents a specific class of fault dip. The component with the highest probability indicates the predicted dip angle of the fault passing through the center pixel of the input image patch. With the predicted dip angle and the highest prediction probability, we can further construct a fault probability image (with the same size and sampling grid as the input seismic patch) with linear fault features oriented by the dip angle. Such a fault probability image patch $f(\mathbf{x})$ is computed as a Gaussian function oriented by the predicted fault dip angle and scaled by the prediction probability p as follows:

$$f(\mathbf{x}) = p \exp \left[-\frac{1}{2} (\mathbf{x} - \mathbf{x}_c)^T \mathbf{R}^T \mathbf{S} \mathbf{R} (\mathbf{x} - \mathbf{x}_c) \right] \quad (3)$$

where \mathbf{x}_c represents the center pixel. \mathbf{R} and \mathbf{S} are the following 2×2 matrices:

$$\mathbf{R} = \begin{bmatrix} \mathbf{u}^T \\ \mathbf{w}^T \end{bmatrix}, \text{ and } \mathbf{S} = \begin{bmatrix} \frac{1}{\sigma_u^2} & 0 \\ 0 & \frac{1}{\sigma_w^2} \end{bmatrix}, \quad (4)$$

where the unit column vectors \mathbf{u} and \mathbf{w} , respectively, represent the fault dip and normal vectors, which are all calculated from the predicted fault dip angle. We set $\sigma_w = 1$ and $\sigma_u = 16$ to obtain the fault-oriented linear features in the fault probability image patch [see the image shown in Fig. 6 (bottom left) or the images in Fig. 7 (right column)].

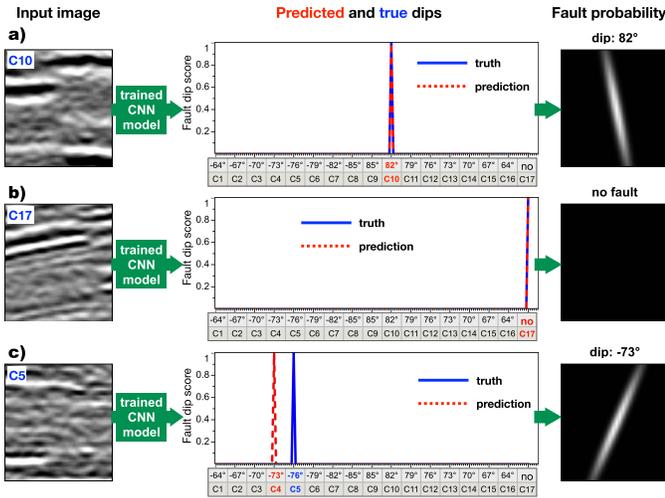


Fig. 7. (Left column) Input seismic patches, (Middle column) predicted fault dips, and (Right column) fault probability image patches that are computed as anisotropic Gaussian functions oriented by the predicted dips.

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 [48] took only around 8.8 min for 12 epochs. The final mean categorical accuracy on the validation images is about 0.89.

Fig. 7 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 [Fig. 7(a)] and without [Fig. 7(b)] fault. With the predicted fault dips, we construct the corresponding fault probability patches (with the same size of the input seismic patches) with dip-oriented anisotropic Gaussians [see Fig. 7(right column)]. The predicted dip for the third seismic patch [the left image in Fig. 7(c)] is inaccurate but is still close to the true fault dip [the middle image in Fig. 7(c)]. This inaccurate dip prediction is not a significant error for the fault interpretation because the fault probability patch [the right image in Fig. 7(c)], 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 but are not much different from the true dips. Fig. 8 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 optimized to automatically compute multiple appropriate feature maps (especially those at the third and fifth layers in Fig. 8) to detect the fault position. More importantly, the linear image features (white color) in these feature maps are consistently aligned in the true fault dip direction. Based on these consistently oriented linear features, the final classification will unlikely predict a fault dip that is much different from the true dip.

D. 2-D Applications

In applying the trained CNN model to a full seismic image [Fig. 9(a)], we suggest to deconvolve the seismic image

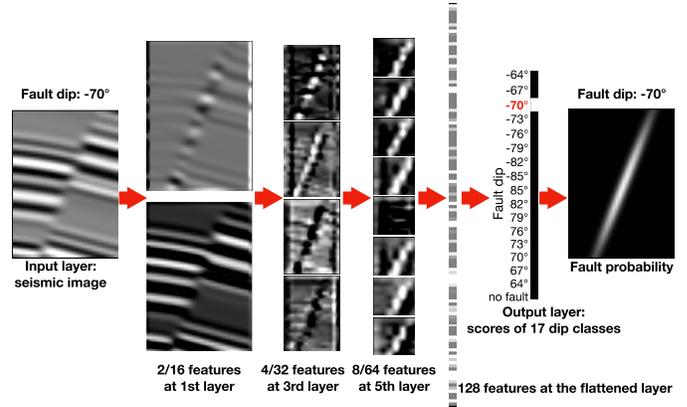


Fig. 8. Illustration of feature maps in some CNN layers.

and then compute a new seismic image by convolving the deconvolved image with a wavelet that is used in generating the synthetic seismic images. By doing this, the structures of the real seismic image are preserved in the new seismic image which, however, contains more consistent frequencies with the training images than the original seismic image. We then use a sliding window with 48×32 pixels to extract overlapping patches from the new seismic image. Each of the extracted seismic patches is normalized so that the amplitude values are in the range of $[-1, 1]$ as in the training images. We further 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 patches to compute full images of the fault probabilities [Fig. 9(b)] and fault dips [Fig. 9(c)]. The fault probability image [Fig. 9(b)] shows clean and continuous fault features to detect the fault positions while the fault dip image [Fig. 9(c)] indicate the fault orientation. Note that this stacking is helpful to obtain enhanced fault images because only the consistently oriented fault predictions are preserved while the unreliable predictions with inconsistent fault orientations are suppressed in this stacking processing. In addition, by stacking the patches with fault features elongated along the fault orientations, we are able to fill potential gaps (mislabeled fault samples) and obtain a fault probability image with continuous fault features as shown in Fig. 9(b). In this stacking processing, the final fault dip image [Fig. 9(c)] is computed as a weighted average (weighted by the fault probability patches) of the predicted fault dips. Therefore, the final stacked fault dip image actually contains continuous fault dip angles although the network only classifies specified fault dip angles.

We may not expect the faults to be as thick as the features apparent in the CNN fault probability image [Fig. 9(b)]. To improve the resolution of fault features, we retain only the probability and dip values on the ridges of the fault probability [Fig. 9(b)] and set zeros elsewhere to obtain the images of thinned fault probabilities [Fig. 9(c)] and thinned dips [Fig. 9(d)]. Note that the estimated fault dips can be either positive or negative in the ranges of $[-63^\circ, -86^\circ] \cup [86^\circ, 63^\circ]$. We have taken the absolute of the dips for the visualization purpose in Fig. 9(c) and (d).

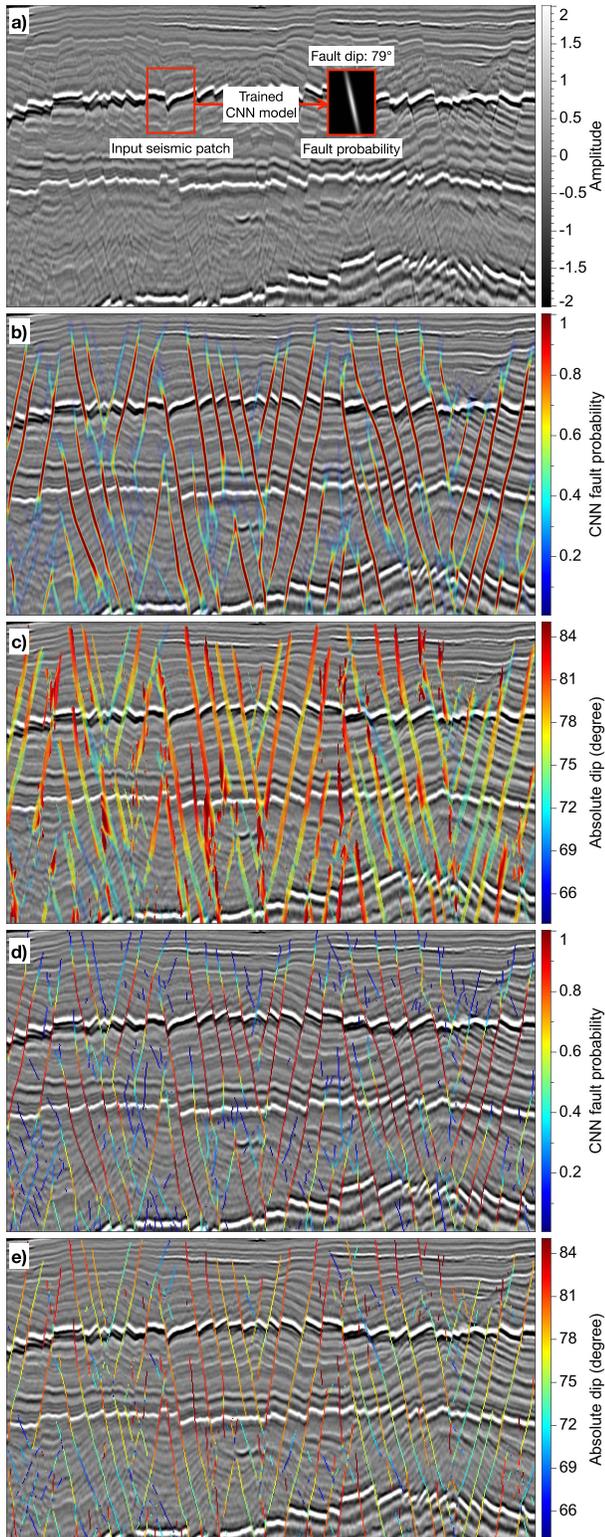


Fig. 9. Given a seismic image (a), we use the trained CNN model to predict the fault dip for every image patch and construct a corresponding fault probability patch. We further stack all the overlapping patches to compute full images of fault probabilities (b) and dips (c), which are thinned in (d) and (e), respectively.

To further verify our CNN-based method, we apply the trained CNN model together with another seven commonly used methods to another 2-D field example in Fig. 1.

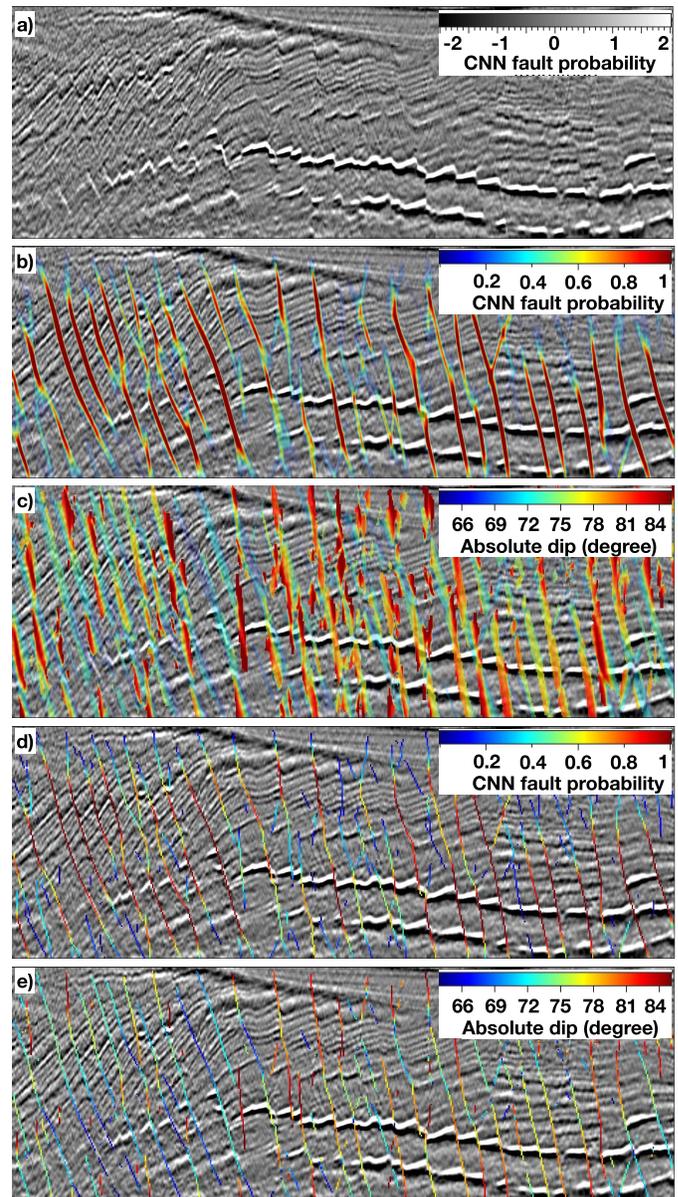


Fig. 10. 2-D seismic amplitude image (2) is displayed with fault probabilities (b) and dips (c), which are thinned in (d) and (e), respectively.

Fig. 1(a)–(h) shows all the eight fault detection results that are, respectively, computed by using the methods of C3 [11], C2 [10], planarity [9], structure-oriented semblance [9], structure-oriented planarity [14], likelihood [20], [21], optimal surface voting [24], and our CNN method. The input for the optimal surface voting method is the linearity image [Fig. 1(c)] and the input for all the other methods is the background seismic amplitude image. Compared to the first five methods, the fault likelihood [Fig. 1(f)] and optimal surface voting [Fig. 1(g)] methods provide better fault detections where the fault features are less noisy and can be more continuously tracked. Our CNN-based method achieves the best performance in computing an accurate and clean fault detection [Fig. 1(h)].

Fig. 10(a) shows a more complicated example, where the seismic image is pretty noisy and the seismic reflections

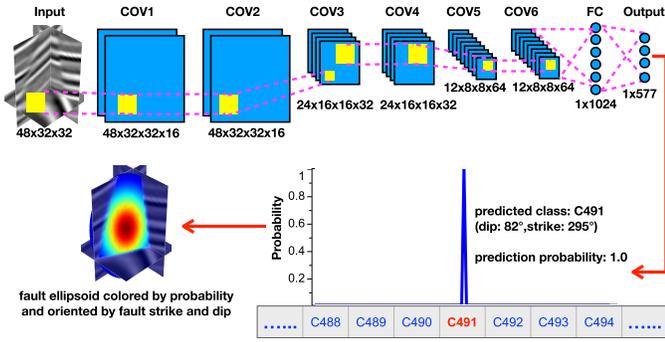


Fig. 11. Outline of the 3-D CNN architecture used to predict the fault class (defined by fault strike and dip) from an input seismic amplitude cube. (Bottom right) In the output vector with 577 elements, the element with the highest probability indicates the predicted fault class C497, which corresponds to a local fault plane with the strike of 295° and the dip of 82° . With the predicted fault strike and dip angles, we further construct (Bottom left) 3-D fault-oriented anisotropic Gaussian to approximate the fault in the input seismic cube.

are not well imaged. However, our trained CNN model still worked well to accurately detect most of the faults and estimate the corresponding fault dips as shown in the fault images of probabilities [Fig. 10(b)] and dips [Fig. 10(c)]. The thinned fault probability [Fig. 10(d)] and dip [Fig. 10(e)] images provide a better view of the fault detection with clearer and sharper fault features.

Remember that our CNN model was trained with only synthetic seismic image patches as those shown in Fig. 4. This trained model, however, works well to detect faults and predict fault orientations in variant field examples (Figs. 1, 9, and 10) that are acquired at totally different surveys.

III. 3-D FAULT CLASSIFICATION

As shown in Fig. 2, the fault geometry is 3-D and a local fault plane is defined by a combination of strike and dip angles. The fault dips estimated from 2-D inline or crossline seismic images are apparent dips, which are not the true fault dips if the faults do not perpendicularly intersect the 2-D seismic images. Moreover, some faults may disappear in a 2-D seismic image if the faults are nearly parallel to the direction of extracting the seismic image. This means that we should work on 3-D seismic images, instead of 2-D images, to reasonably and fully interpret faults. We therefore construct a 3-D CNN (Fig. 11) to simultaneously estimate the fault strike and dip angles from a 3-D seismic cube, which is, again, considered as an image classification problem as estimating fault dips in 2-D.

A. 3-D Fault Classes

Compared to the 2-D fault classes, we have much more classes in 3-D, where we define a fault class in a local 3-D window by a combination of strike and dip angles. Theoretically, the fault strike and dip angles can be in the full ranges of $[0^\circ, 360^\circ]$ and $[0^\circ, 90^\circ]$, respectively. To reduce the number of fault classes, we consider the fault dips only in a smaller range $[64^\circ, 85^\circ]$ by assuming most faults are more vertical than horizontal but cannot be perfectly vertical.

We further define a 3-D fault class for every 5° in strike angles and 3° in dip angles. We, therefore, totally have 577 fault classes in 3-D including the class of no fault as follows:

$$\frac{360^\circ}{5^\circ} \times \frac{85^\circ - 64^\circ}{3^\circ} + 1 \text{ (no fault)} = 577. \quad (5)$$

B. 3-D CNN Architecture

Although there are much more 3-D fault classes than the 2-D fault classes, we do not need a significantly more complicated CNN for the 3-D fault classification. Similar to the 2-D fault classification, we construct a 3-D CNN architecture (Fig. 11) with six convolutional layers followed by a fully connected layer and an output layer. As shown in Fig. 11, the input is a 3-D seismic cube with $48(\text{vertical}) \times 32(\text{inline}) \times 32(\text{crossline})$ pixels. At every convolutional layer, we define exactly the same number of the filter as in 2-D, but the filter in 3-D is now with the size of $3 \times 3 \times 3$ and the stride of 1. We apply the batch normalization and ReLU activation after each convolutional layer and an average pooling after every two convolutional layers. In the fully connected layer, we increased the number of elements from 128 in 2-D to 1024 in 3-D. The final output layer is a larger vector containing 577 elements, which correspond to the 577 different fault classes defined in 3-D.

In this output vector, the element with the highest probability indicates the predicted fault class that corresponds to a specific combination of fault strike and dip angles. As an example shown in Fig. 11 (bottom right), the element with the highest probability ($p = 1.0$ in this example) indicates the fault class C497, which is defined by the strike of 295° and the dip of 82° . With the predicted fault orientation and the prediction probability, we further construct a fault probability cube (with the same size of the input seismic cube) with fault-oriented planar features as shown in the image of Fig. 11 (bottom left). Such a fault probability cube $f(\mathbf{x})$ is computed as a 3-D anisotropic Gaussian function oriented by the predicted fault orientations and scaled by the prediction probability p as follows:

$$f(\mathbf{x}) = p \exp \left[-\frac{1}{2} (\mathbf{x} - \mathbf{x}_c)^\top \mathbf{R}^\top \mathbf{S} \mathbf{R} (\mathbf{x} - \mathbf{x}_c) \right]. \quad (6)$$

Here, \mathbf{x}_c represent the center pixel. \mathbf{R} and \mathbf{S} are the following $3 \times 3 \times 3$ matrices:

$$\mathbf{R} = \begin{bmatrix} \mathbf{u}^\top \\ \mathbf{v}^\top \\ \mathbf{w}^\top \end{bmatrix}, \quad \text{and} \quad \mathbf{S} = \begin{bmatrix} \frac{1}{\sigma_u^2} & 0 & 0 \\ 0 & \frac{1}{\sigma_v^2} & 0 \\ 0 & 0 & \frac{1}{\sigma_w^2} \end{bmatrix}, \quad (7)$$

where the unit column vectors \mathbf{u} , \mathbf{v} , and \mathbf{w} , respectively, represent the fault dip, strike, and normal vectors that are all calculated from the predicted fault strike and dip angles. We set $\sigma_w = 1$, $\sigma_v = 16$ and $\sigma_u = 16$ to obtain the fault-oriented planar features in the fault probability image cube (see image in Fig. 11 (bottom left)).

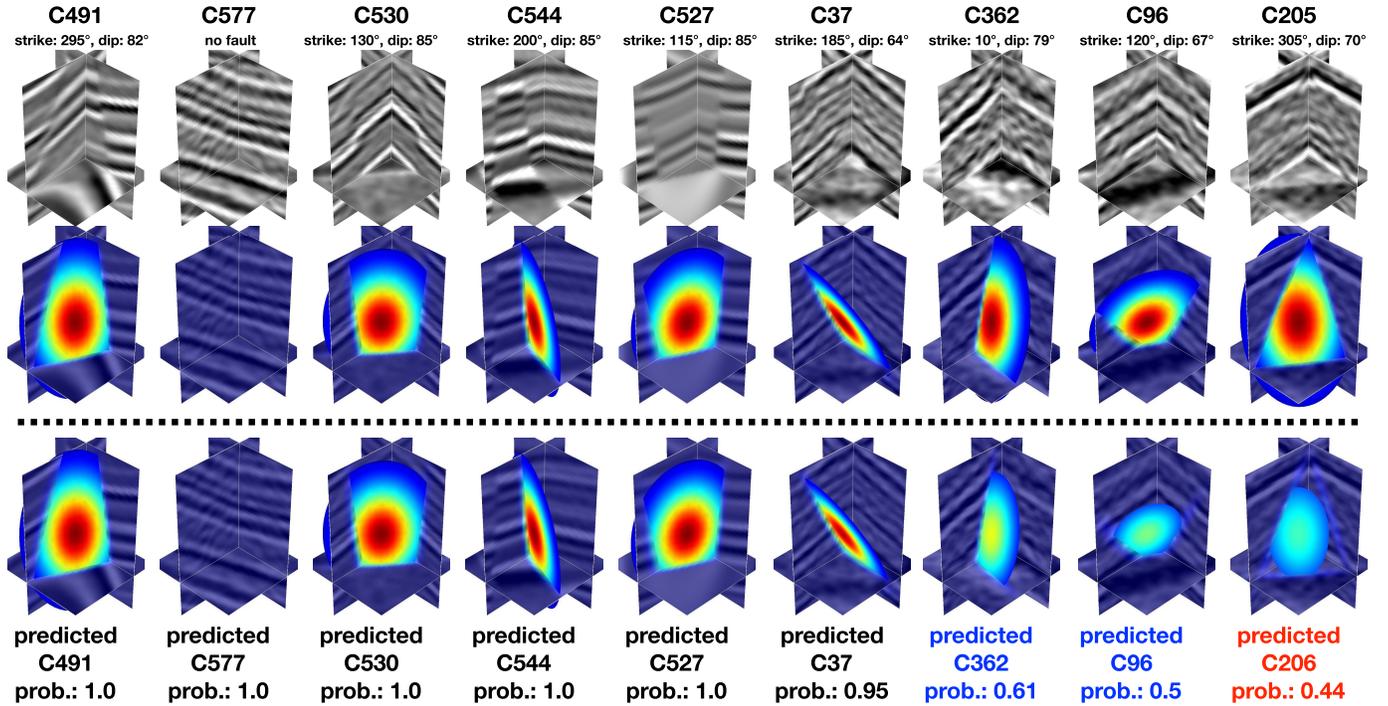


Fig. 12. First row shows nine synthetic seismic cubes, where the true fault classes (defined by fault strikes and dips) are marked on the top of the first row. The ellipsoids in the second row represent 3-D anisotropic Gaussians that are oriented by the true fault strikes and dips. The ellipsoids in the third row represent 3-D anisotropic Gaussians that are oriented by the predicted fault strikes and dips (predicted from the top seismic cubes) and scaled by the prediction probabilities.

C. 3-D Training Data Sets

Training a CNN model for 3-D fault classification requires much more training data sets because we have much more fault classes in 3-D. It is impossible to manually label all the fault classes with different fault strike and dip angles. By using a similar workflow as in 2-D, we automatically created 900 000 synthetic seismic cubes with 1500 image cubes for each fault class and 36 000 image cubes for the class of no fault. We also created 57 700 synthetic seismic cubes (100 cubes for each fault class) to validate the training processing. By using a Titan Xp GPU, the whole training process on the 900 000 synthetic cubes took only around 4 h for 12 epochs. The final mean categorical accuracy on the validation cubes is about 0.92, which is even higher than the 2-D fault classification.

The first row of images in Fig. 12 shows nine of the synthetic seismic cubes that were created to validate the 3-D CNN model. The true fault classes for all the image cubes are marked on top of the image cubes. To visualize the local fault planes, in these synthetic seismic cubes, we use 3-D planar ellipsoids that are oriented by the true fault strikes and dips as shown in the second row of Fig. 12. The color on each ellipsoid represents fault probabilities defined by an anisotropic Gaussian as in (6) ($p = 1$). The three axes (in the directions of fault strike, dip, and normal vectors) of each ellipsoid are proportional to the σ_u , σ_v , and σ_w [see (7)].

The third row of Fig. 12 shows the fault classification results of the input seismic cubes (the first row) by using a trained CNN model. The predicted fault classes (fault strikes and dips) and prediction probabilities are marked at the bottom

of the third row. The predicted classes for the first six seismic cubes exactly match the true fault classes (marked on the top of the first row) and the prediction probabilities are pretty high ($p \geq 0.95$). The prediction probabilities for the seventh and eighth image cubes are relatively lower but the predicted classes still match the true fault classes. The prediction probability for the last image cube is low ($p = 0.44$) and the predicted class C206 is inaccurate but is still close to the true fault class C205. Note that the prediction probabilities are relatively lower for the last three seismic cubes because they are noisier and the faults are less obvious in these cubes. The anisotropic ellipsoids in the third row are oriented by the predicted fault strike and dip angles and their sizes are scaled by the prediction probabilities. These ellipsoids are all consistently aligned with the true fault planes shown in the second row. As the prediction probabilities for the last three seismic cubes are relatively smaller, the spatial extensions of the last three ellipsoids in the third row are relatively smaller. These three ellipsoids are also mostly colored by blue, which means that the fault probabilities on the ellipsoids are relatively lower.

The way of constructing such anisotropic fault ellipsoids is beneficial for the next section of detecting faults in 3-D full seismic images, where we stack the overlapping ellipsoids to calculate full images of fault probabilities, strikes and dips. The ellipsoids corresponding to higher prediction probabilities will contribute more in the stacking because they are constructed to have larger spatial extensions and higher fault probabilities.

D. 3-D Applications

In applying the 3-D CNN model (trained with 3-D synthetic data sets) to a full 3-D seismic amplitude image [Fig. 13(a)], we again deconvolve the seismic image and then compute a new seismic image by convolving with a consistent wavelet used in generating the synthetic training images. From this new seismic image, we then use a 3-D sliding window ($48 \times 32 \times 32$ pixels) to extract seismic cubes (one per every two samples in all directions) that are overlapped with each other. Each seismic cube is normalized so that the amplitude values are in the range of $[-1, 1]$ as in the training images. We further feed these seismic cubes to the CNN model and obtain a predicted fault class and the corresponding prediction probability for each cube. The predicted fault class indicates the fault strike and dip or the class of no fault. If there is a fault predicted in a seismic cube, we then use the predicted fault strike and dip to construct a 3-D fault-oriented Gaussian [see (6)] centered at that cube to represent the predicted 3-D fault. The colorful anisotropic ellipsoids shown in Fig. 13(b) represent 3-D visualizations of a subset of such fault-oriented Gaussians. These 3-D ellipsoids visually look like planes because they mainly extend in the fault strike and dip directions, and are very thin in the fault normal directions. The color on each ellipsoid represents fault probability which is highest at the center and gradually decreases in directions along the fault plane. The ellipsoids at some positions have smaller extensions or sizes because the prediction probabilities p at the positions are relatively lower ($p \leq 1$) and the fault-oriented Gaussians are scaled by the lower probabilities p as in (6). The fault probabilities (represented by colors on each ellipsoid) on these smaller ellipsoids are also relatively lower.

Fig. 13(c) shows the full set of fault ellipsoids constructed at all the positions where faults are predicted at the inline, crossline, and horizontal seismic slices. We observe that these ellipsoids are consistently aligned and overlapped with each other. We stack the fault probabilities, strikes and dips of all these overlapped ellipsoids to obtain three full images of fault probabilities, strikes and dips as shown in Fig. 14(a)–(c), respectively. In this stacking, the ellipsoids with higher prediction probabilities will contribute more to computing the fault images because they have larger spatial extensions and higher fault probabilities. The stacked fault probability image [Fig. 14(a)] accurately highlights most of the faults in the seismic image and the fault features in this probability image can be clearly and continuously tracked.

Fig. 14(d)–(f), respectively, shows the thinned fault probabilities, strikes, and dips, where we have nonzero values only at the ridges of the fault probabilities [Fig. 14(a)] and zeros elsewhere. In finding the ridges of fault probabilities shown in Fig. 14(a), we look for the peaks in the fault normal directions, which are computed from the fault strikes [Fig. 14(b)] and dips [Fig. 14(b)]. As shown in Fig. 15(a), we further construct fault cells [21] at the locations with nonzero values in the thinned fault images. These fault cells are displayed as small squares and each square is colored by fault probability and oriented by fault strike and dip, which means that these cells can actually represent all the three fault

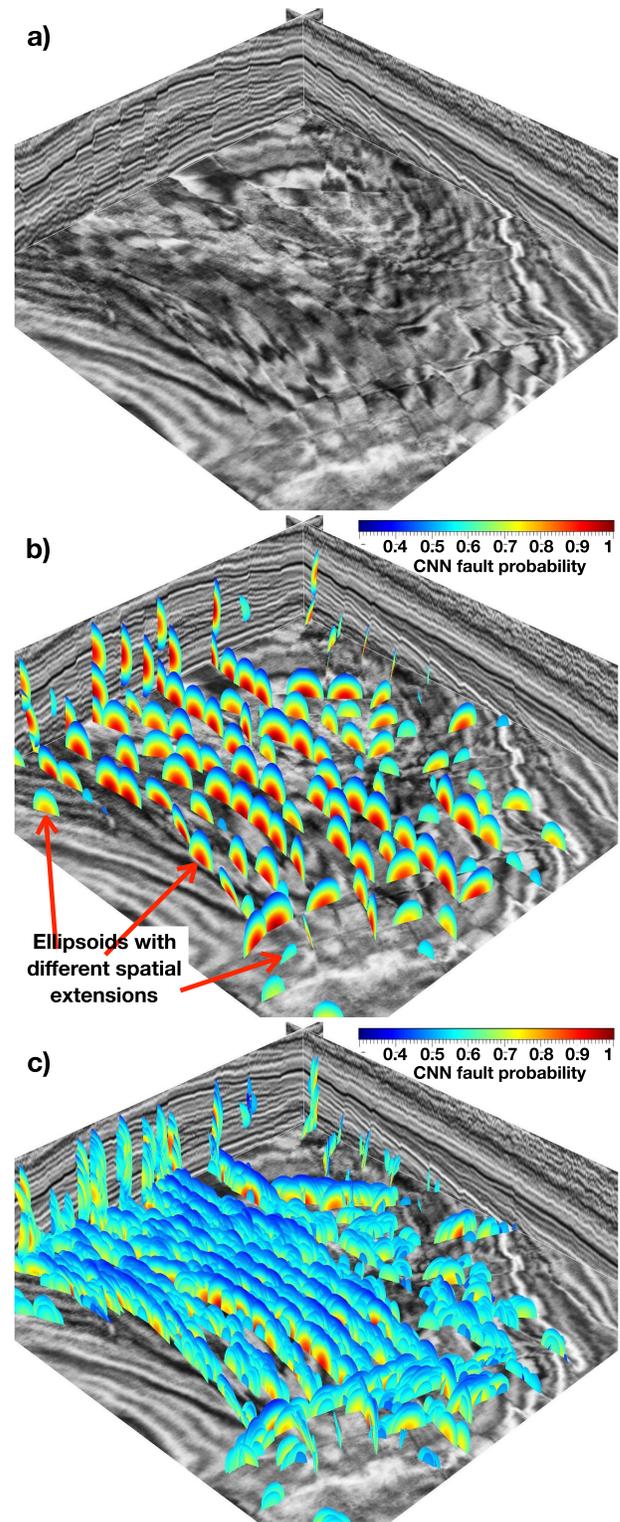


Fig. 13. (a) 3-D seismic image is displayed with (b) subset and (c) full set of anisotropic ellipsoids that are colored by fault probabilities and oriented by fault strikes and dips. The fault probabilities, strikes, and dips are simultaneously estimated using our CNN model that is trained with only synthetic data sets.

images of thinned probabilities, strikes and dips. We finally link the consistently oriented fault cells to obtain the individual fault skins [21] or surfaces [Fig. 15(b)], which are constructed by using a simple linked data structure discussed in [21].

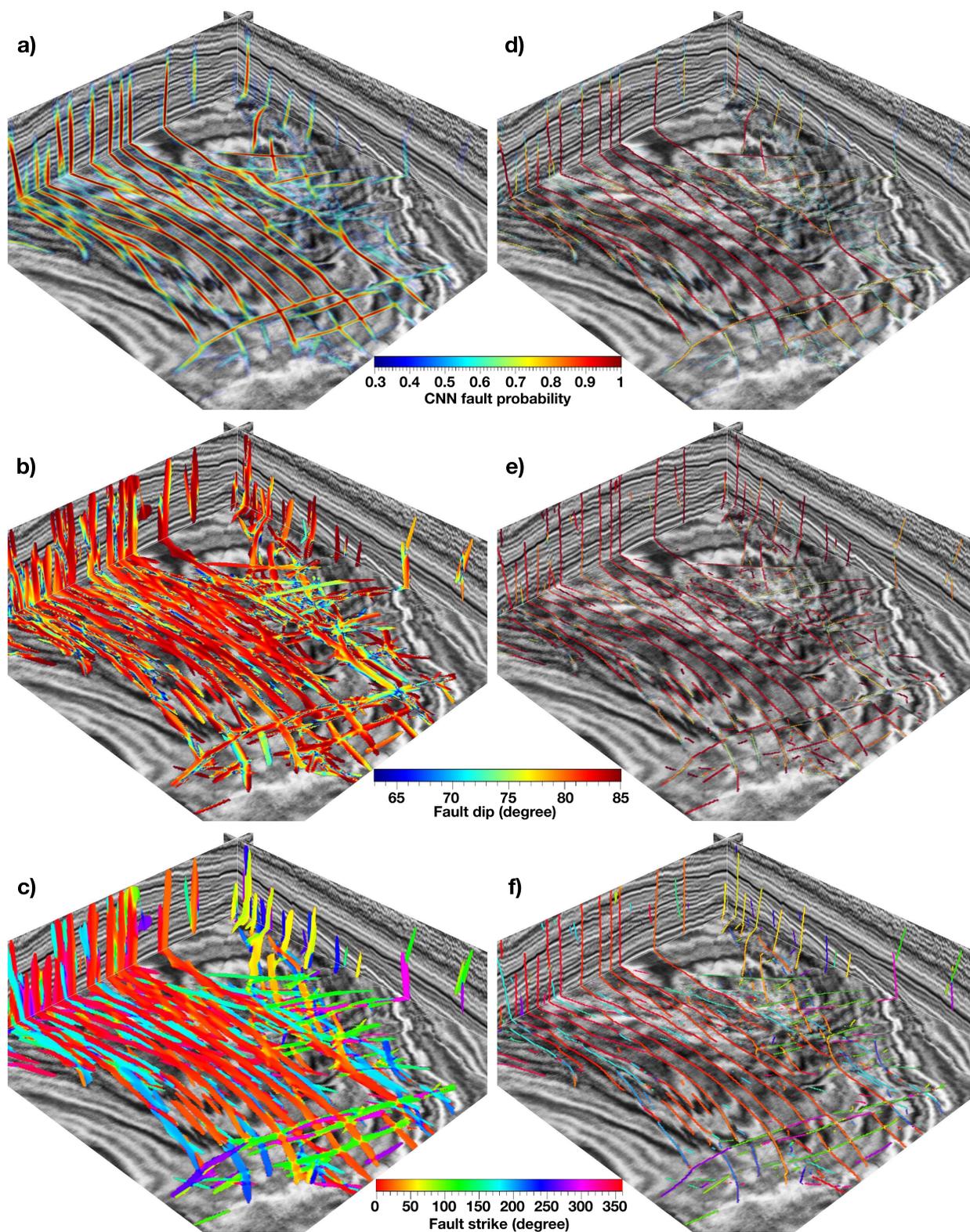


Fig. 14. We stack the overlapping ellipsoids in Fig. 13(b) to compute three images of (a) fault probabilities, (b) dips, and (c) strikes, which are further thinned in (d)–(f), respectively.

Fig. 17 shows another 3-D seismic amplitude image that was acquired at the Campos Basin, offshore Brazil. The reflections in this seismic image are intensively faulted due to the

salt bodies at the bottom of the image. From this 3-D seismic image (Fig. 16), we simultaneously compute three fault images of thinned probabilities [Fig. 17(a)], dips [Fig. 17(b)], and

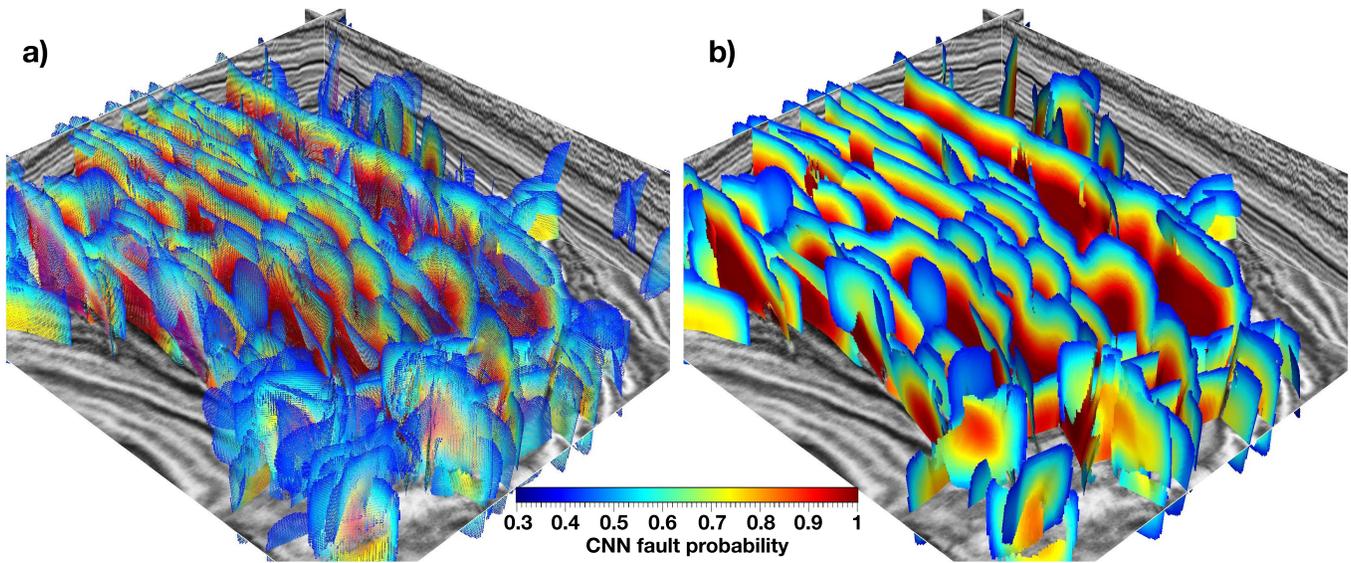


Fig. 15. From the three thinned fault images [Figs. 14(d)–14(f)], we further construct fault cells (a) at the locations with nonzero values in the thinned probability image [Fig. 14(d)]. These fault cells are displayed as small squares and each square is colored by fault probability and oriented by fault strike and dip. We finally link the consistently oriented fault cells to construct fault skins/surfaces (b), which are also colored by fault probabilities.

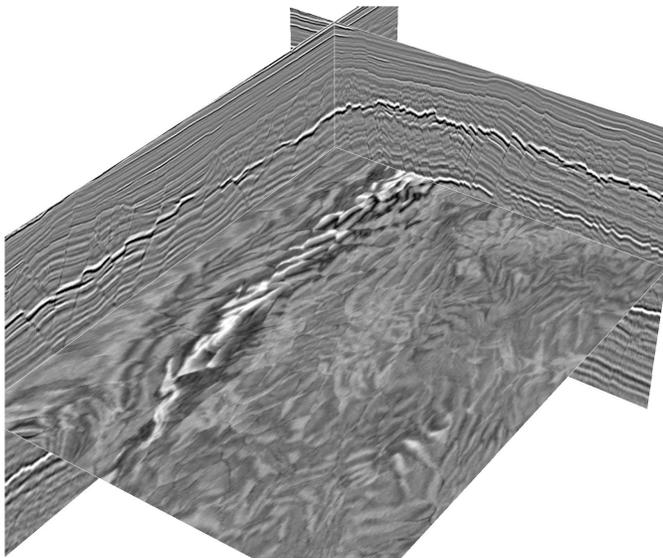


Fig. 16. 3-D seismic image with intensive faults.

strikes [Fig. 17(c)] by using our CNN model (trained with only synthetic data sets) and the poststacking processing. In these thinned fault images, numerous closely spaced faults are accurately, clearly, and continuously labeled. The horizontal slice shown in Fig. 17(a) displays a clear pattern of polygonal faults that may be associated with salt diapirs [49], [50]. From these three thinned fault images, we define the consistently oriented fault cells shown in Fig. 17(d), which are further linked to construct fault surfaces that are colored by probabilities and strikes shown in Fig. 17(e) and (f), respectively.

Fig. 18 shows the final 3-D example, where the 3-D seismic image was acquired at Opunake 3-D survey, east of offshore Maui field, New Zealand [51]. From this 3-D seismic image,

we again compute three thinned fault images of probabilities [Fig. 18(a)], strikes (not shown), and dips (not shown) by using the same CNN model (trained with only synthetic data sets) and the poststacking processing. Fig. 18(b) shows the fault surfaces (colored by fault probabilities) that are automatically constructed from the three thinned fault images. Fig. 18(c) and (d) shows a different 3-D view of the thinned fault probability image and the extracted fault surfaces, where we can observe that the complicated branch fault surfaces are successfully constructed.

E. Comparison and Evaluation

As a comparison, we also apply the FaultSeg3D [46] (a CNN-based fault segmentation method) to the three 3-D examples. As discussed in [46], the FaultSeg3D is significantly superior to the conventional methods in detecting faults from a seismic image. To compare the proposed CNN-based classification method with the FaultSeg3D method, we display the fault probability images computed by using the two methods side by side in Fig. 19. We observe that the two methods provide similar fault detection results which are all good in general.

However, if we look more carefully, some gaps (denoted by yellow arrows) and noisy features are observed in the results [Fig. 19(d)–(f)] of the FaultSeg3D method. The fault features in the results [Fig. 19(a)–(c)] of the proposed method are relatively more continuous and clearer to track. Moreover, as shown in Figs. 14 and 17, the proposed method can simultaneously estimate fault strikes and dips as well as the fault probabilities [Fig. 19(a)–(c)] while the FaultSeg3D method can estimate only the fault probabilities [Fig. 19(d)–(f)]. The estimated fault strikes and dips are important for fault population analysis [52] and are required for automatically constructing fault surfaces as shown in Figs. 15, 17, and 18.

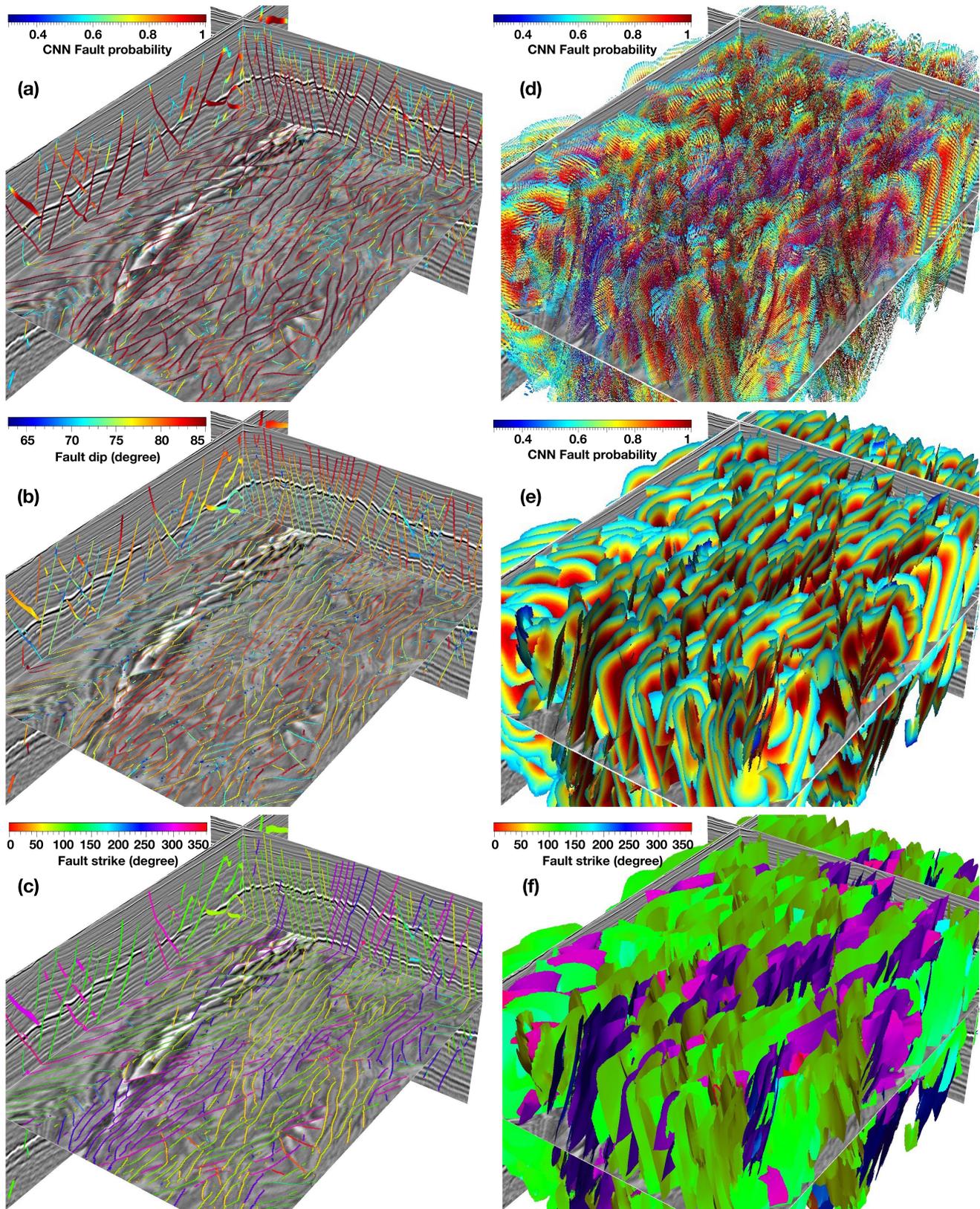


Fig. 17. With an input seismic image (Fig. 16), we are able to simultaneously estimate three images of (a) fault probabilities, (b) dips, and (c) strikes by using our CNN model trained with only synthetic data sets. From these three fault images, we further construct (d) fault cells that are further linked to form fault skins colored by (e) fault probabilities and (f) strikes.

The disadvantage of the proposed method is that it is computational expensive because the fault classification typically requires applying a sliding window through the whole seismic

volume. Fortunately, by using the postprocessing of stacking fault-oriented Gaussians, we do not need to apply the sliding window at every image pixel. To save computational cost,

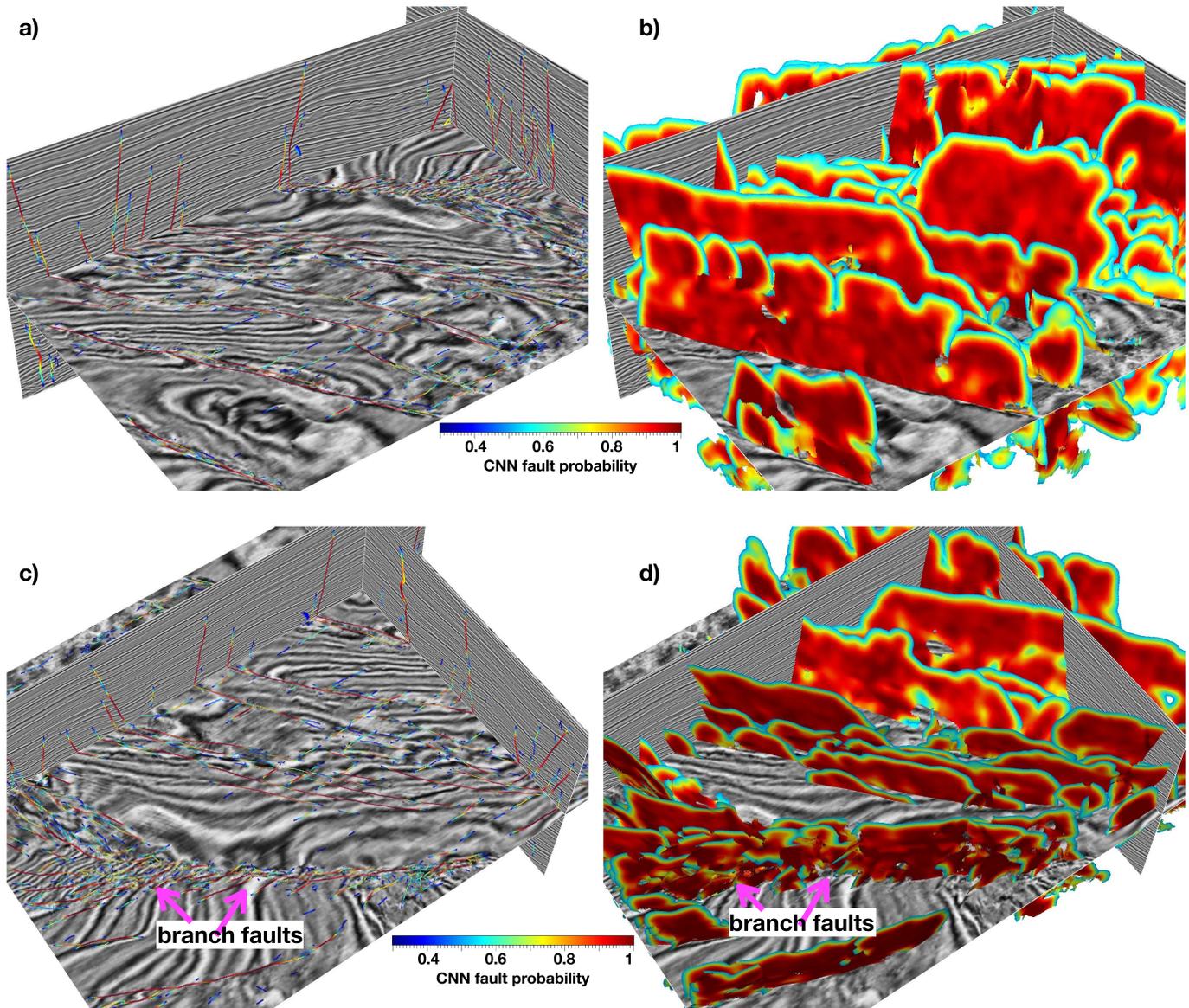


Fig. 18. 3-D seismic image is displayed with (a) CNN fault probabilities and (b) automatically extracted fault surfaces. (c) and (d) Different 3-D view of the CNN fault probability image and the fault surfaces, where we can observe branch faults with a complicated geometry.

we apply the sliding window at every two image pixels (in all directions) and the poststacking processing will interpolate back a full fault image. In this way, the proposed method requires around 21 min to estimate the three fault images of probabilities, strikes, and dips in the first 3-D example [with $100 \times 400 \times 420$ samples shown in Fig. 13(a)] by using a Titan Xp GPU. By using the same GPU, the FaultSeg3D method requires less than 1 min to compute the fault probability image shown in Fig. 19(d). To further improve the computational efficiency of the proposed method, a potential way is to first use a more efficient method to roughly detect fault positions in the seismic image and then apply the proposed CNN-based classification method at only these detected fault positions to estimate fault strikes and dips as well as better fault probabilities.

To further quantitatively evaluate the proposed CNN-based classification method, we created two synthetic seismic images

with $128 \times 128 \times 128$ samples as shown in Fig. 20(a). The advantage of using synthetic seismic images is that we know the true faults as shown in Fig. 20(b), which can be used to calibrate the results of fault detection methods. Fig. 20(c)–(j) shows eight fault detection results that are, respectively, computed by using the methods of C2 [10], planarity [9], structure-oriented linearity [14], structure-oriented semblance [9], fault likelihood [20], [21], optimal surface voting [24], CNN-based segmentation [46], and our CNN-based classification. The inputs for the optimal surface voting method are the planarity volumes [Fig. 20(e)] and the inputs for all the other methods are the seismic amplitude volumes [Fig. 20(a)]. The first four methods [Fig. 20(c)–(f)], based on measuring seismic reflection discontinuities, can detect some parts of the faults and also highlight many noisy features that are unrelated to faults. In addition, the faults cannot be continuously tracked in the fault detections of these four methods. Compared to the

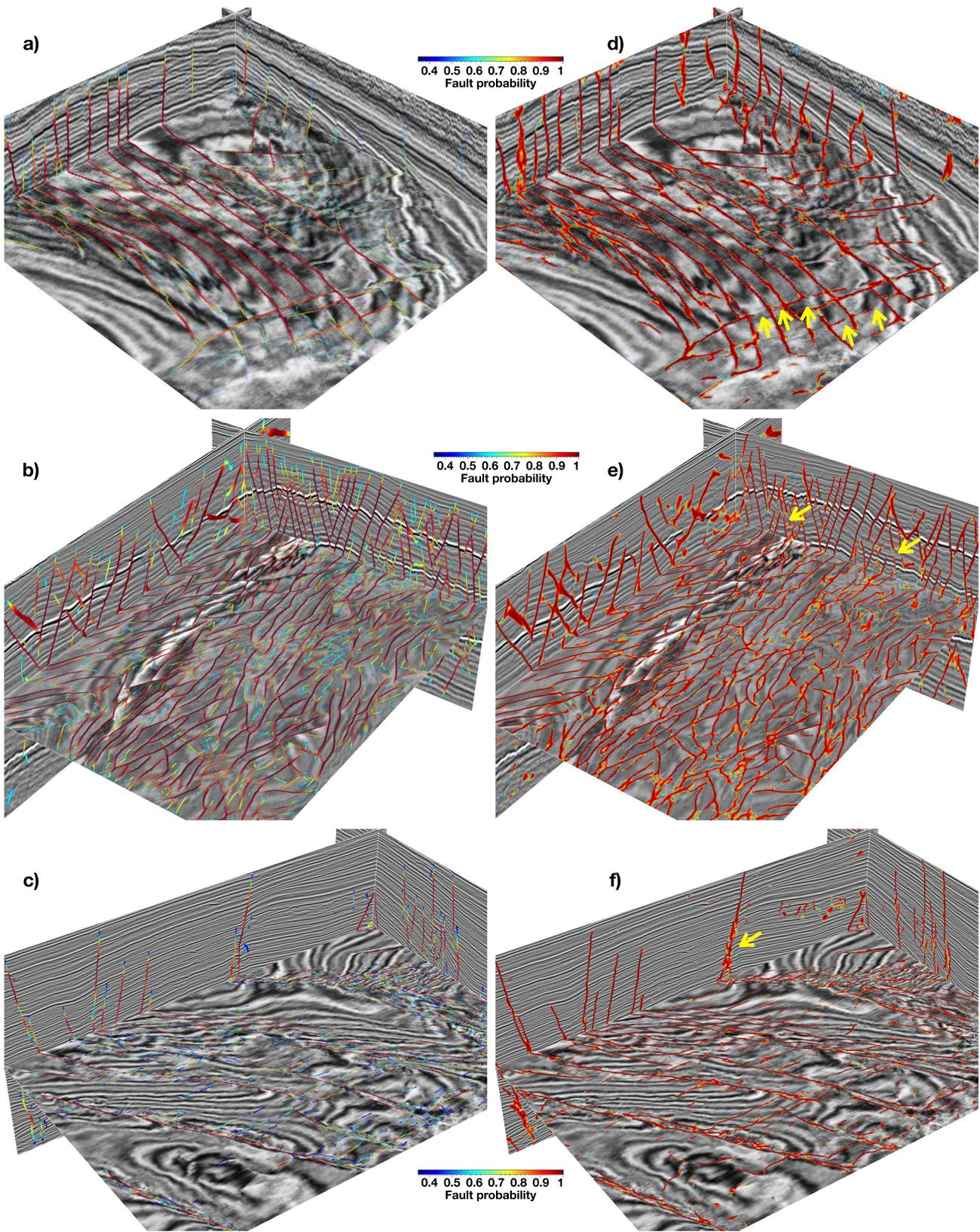


Fig. 19. As a comparison, the fault probability images computed by (a)–(c) proposed method and (d)–(f) CNN-based fault segmentation method [46] are displayed side by side.

first four methods, the fault likelihood [Fig. 20(g)], optimal surface voting [Fig. 20(h)], and CNN-based fault segmentation [Fig. 20(i)] methods provide better fault detections where

the fault features are less noisy and can be more continuously tracked. However, we can still observe some noisy features as denoted by white arrows in the fault detection

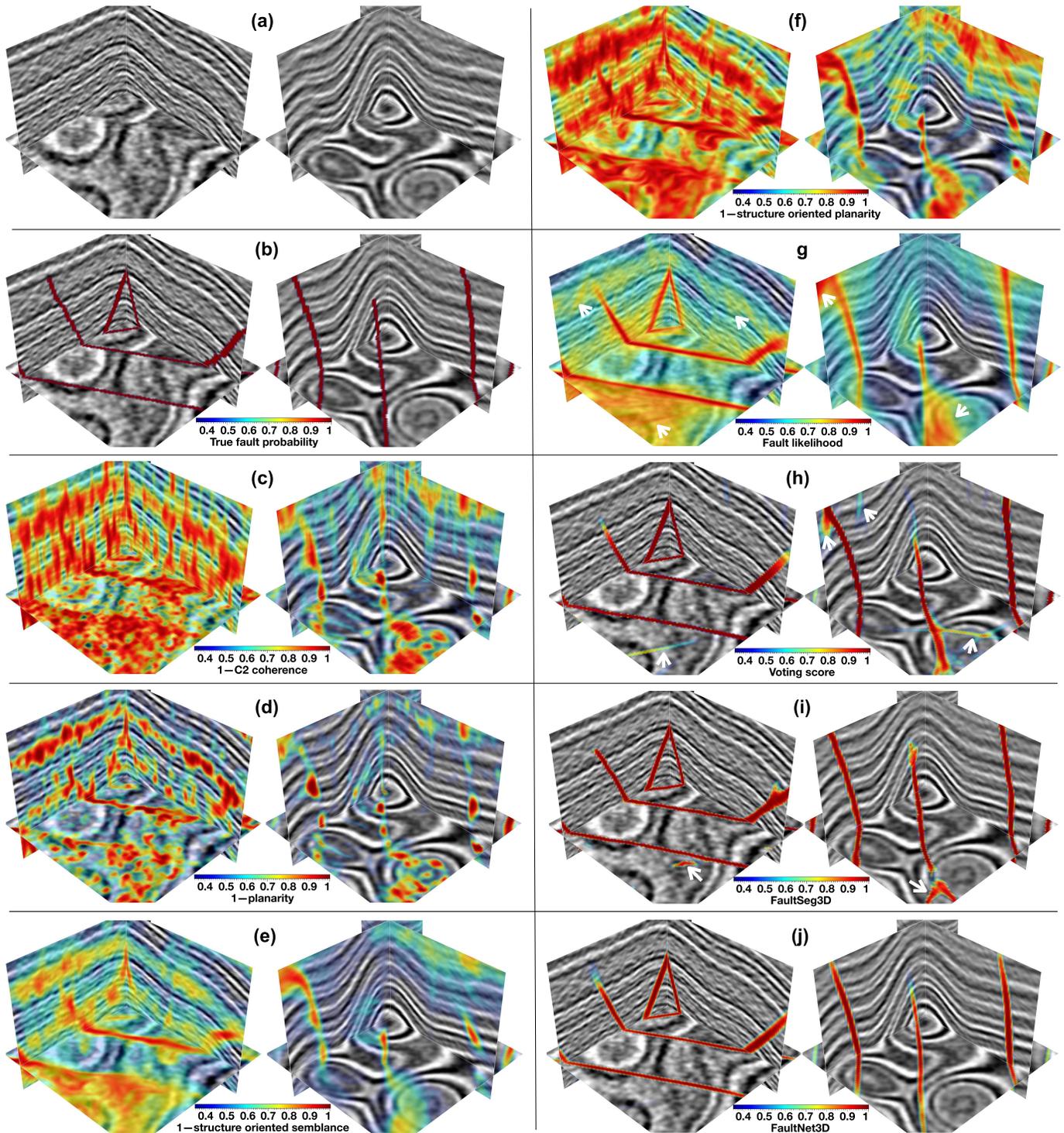


Fig. 20. (a) Two 3-D synthetic seismic images are overlaid with (b) true fault probabilities, and the fault detection results by eight methods. The first four methods of (c) C2 [10], (d) planarity [9], (e) structure-oriented planarity [14], and (f) structure-oriented semblance [9] can detect some parts of the faults but the fault features are noisy and cannot be continuously tracked. The methods of (g) fault likelihood [20], [21], (h) optimal surface voting [24], and CNN-based fault segmentation [46] perform better fault detections with enhanced fault features. (j) Our method achieves the best performance in obtaining an accurate and clean fault detection.

results [Fig. 20(g)–(i)]. Our CNN-based classification method achieves the best performance in computing clean and accurate fault detections, which are most consistent with the true faults shown in Fig. 20(b). More importantly, the network of our CNN-based classification method is much simpler than

the fault segmentation method [46] and can simultaneously estimate the fault orientations (strikes and dips) in addition to detecting the fault positions. The fault orientations are important for the followed step of automatically constructing fault surfaces.

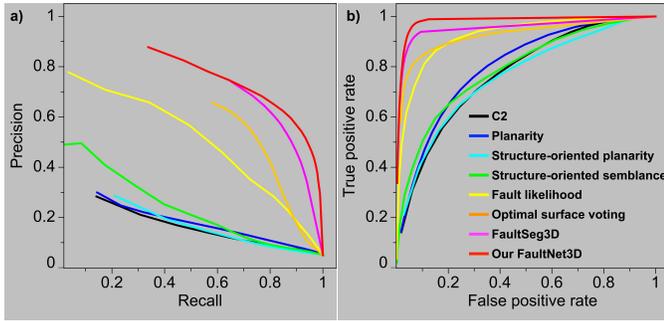


Fig. 21. (a) Precision–recall and (b) ROC curves are used to evaluate the eight fault detection methods on the synthetic seismic images in Fig. 20. The two CNN-based methods of FaultSeg3D [46] (magenta curves) and FaultNet3D (red curves) perform significantly better than the other methods. In addition, our FaultNet3D method still provides higher fault detection accuracy than the FaultSeg3D method although the network (Fig. 11) of the former is much simpler than the later. More importantly, our FaultNet3D can simultaneously estimate the fault strikes and dips at the same time as detecting the faults.

Based on all the above fault detections (Fig. 20) on the two synthetic seismic images, we calculate the precision–recall [53] and receiver operator characteristic (ROC) [54] curves in Fig. 21 to quantitatively evaluate the performances of the eight fault detection methods. The precision–recall curves [Fig. 21(a)] show that our CNN-based classification method (red curve) provides the highest precisions for all the choices of recall. The CNN-based segmentation method (magenta curve), with a more complicated U-net architecture [46], can provide high fault-detection precisions which, however, are still relatively lower than our proposed classification network with a much simpler architecture (Fig. 11). The precisions of the optimal surface voting [orange curve in Fig. 21(a)] and fault likelihood [yellow curve in Fig. 21(a)] methods are relatively lower than the two CNN-based methods but are higher than the other four methods. The ROC curves in Fig. 21(b) provide similar evaluations of the methods. Note that this quantitative comparison on such synthetic benchmark data sets might be biased as the two CNN-based methods are trained on synthetic data sets generated in a similar fashion as the benchmark data sets, whereas other methods do not have any prior knowledge about the benchmark data. However, this quantitative comparison is consistent with the qualitative comparison on a field data shown in Figs. 1 and 19.

IV. CONCLUSION

Instead of predicting only fault probabilities as in the previous CNN-based fault classification methods, we have proposed to use a single CNN to simultaneously estimate fault probabilities, strikes, and dips from an input seismic image, which is formulated as a classification problem. We may formulate the strike and dip estimation as a regression problem, which, however, often requires a more complicated or deeper network than our network used for classification. In addition, most samples of a seismic volume are nonfault samples, and faults are very sparse features appear in only very limited and narrow areas in the seismic volume, which makes it a highly challenging problem to directly estimate fault strikes and

dips from a seismic volume using a regression network. Our formulation of fault classification enables us to simultaneously estimate fault probabilities, strikes, and dips directly from a seismic volume by using a very simple network with only six convolutional layers.

The output probabilities, strikes, and dips of our network are not the final fault images for fault interpretation. In fact, we further use the estimated strikes and dips to compute fault-oriented Gaussians (weighted by the estimated fault probabilities) to stack for the final fault images of probabilities, strikes, and dips, where the signal-to-noise ratio and continuity of the fault features are improved compared to the original outputs of the network. Although the network can predict only discrete fault strike and dip angles with a specified resolution, the stacking will provide continuous angle values, each of which is computed as a weighted average of nearby estimated strikes and dips. In this stacking, only the reliable fault predictions (often with consistent fault strikes and dips) are enhanced and preserved while the noisy or unreliable fault predictions (often with outlier or inconsistent strikes and dips) are suppressed. In addition, the stacking of fault-oriented Gaussians (elongated along fault planes) is helpful to interpolate or fill potential gaps (where faults are mislabeled) to obtain a final fault image with continuous fault features. This also enables us to apply the classification sliding window in a sparse sampling grid as the stacking will interpolate back a full fault image, which significantly increases the computational efficiency compared to applying the classification in every image sample. With the finally stacked fault images of probabilities, strikes, and dips, we further construct oriented fault cells, which are consistently connected to construct fault skins (surfaces) by using a simple linked data structure.

We train the CNN model completely on synthetic seismic data sets without any manual labeling. The trained model, however, works robustly in multiple field seismic images (acquired at totally different surveys) to compute better fault images than most of the conventional fault detection methods. We have discussed both 2-D and 3-D CNN for fault classification, however, we recommend using the 3-D CNN to perform fault classifications in 3-D seismic images because the fault geometry can be better described in 3-D and more information in 3-D cubes can be used by a CNN model to more accurately predict fault orientations.

Some limitations remain in our method. One arises from the way in which we generate synthetic training data sets where the faults are assumed to extend more vertically than horizontally with relatively high dip angles. The trained network may fail to detect the thrust faults or listric faults which are more horizontally extended with lower dip angles. To be able to detect these faults, we probably need to train another network with training data sets containing such faults and use a laterally elongated window instead of a vertically elongated window in this paper. Another limitation is that our network is designed and trained to estimate only fault probabilities, strikes, and dips. Further research would be worthwhile to construct a network to estimate fault slips as well.

ACKNOWLEDGMENT

The authors would like to thank NVIDIA Corporation for the support and donation of the Titan Xp GPU used for this work. They would also like to thank two anonymous reviewers and Z. Geng for their constructive suggestions that help improve this paper.

REFERENCES

- [1] G. Caumon, P. Collon-Drouaillet, C. Le Carlier de Veslud, S. Viseur, and J. Sausse, "Surface-based 3D modeling of geological structures," *Math. Geosci.*, vol. 41, no. 8, pp. 927–945, 2009.
- [2] X. Wu, "Building 3D subsurface models conforming to seismic structural and stratigraphic features," *Geophysics*, vol. 82, no. 3, pp. IM21–IM30, 2017.
- [3] J. Lomask, L. Hernandez, V. Liceras, A. Kumar, and A. Khadeeva, "A seismic to simulation unconventional workflow using automated fault-detection attributes," *Interpretation*, vol. 5, no. 3, pp. SJ41–SJ48, 2017.
- [4] M. Kufrasa, L. Słonka, P. Krzywiec, K. Dzwiniel, and J. Zacharski, "Fracture pattern of the lower paleozoic sedimentary cover in the Lublin basin of southeastern Poland derived from seismic attribute analysis and structural restoration," *Interpretation*, vol. 6, no. 3, pp. SH73–SH89, 2018.
- [5] P. R. Vail, R. G. Todd, and J. B. Sangree, "Seismic stratigraphy and global changes of sea level: Part 5. chronostratigraphic significance of seismic reflections: Section 2. application of seismic reflection configuration to stratigraphic interpretation," in *M 26: Seismic Stratigraphy—Applications to Hydrocarbon Exploration*, vol. 26. Tulsa, OK, USA: AAPG Bulletin, 1977, pp. 99–116.
- [6] X. Wu and D. Hale, "Horizon volumes with interpreted constraints," *Geophysics*, vol. 80, no. 2, pp. IM21–IM33, 2015.
- [7] X. Wu and S. Fomel, "Least-squares horizons with local slopes and multigrid correlations," *Geophysics*, vol. 83, no. 4, pp. IM29–IM40, 2018.
- [8] K. J. Marfurt, R. L. Kirlin, S. L. Farmer, and M. S. Bahorich, "3-D seismic attributes using a semblance-based coherency algorithm," *Geophysics*, vol. 63, no. 4, pp. 1150–1165, 1998.
- [9] D. Hale, "Structure-oriented smoothing and semblance," Center for Wave Phenomena, Colorado School of Mines, Tech. Rep. 635, 2009.
- [10] K. J. Marfurt, V. Sudhaker, A. Gersztenkorn, K. D. Crawford, and S. E. Nissen, "Coherency calculations in the presence of structural dip," *Geophysics*, vol. 64, pp. 104–111, Jan./Feb. 1999.
- [11] A. Gersztenkorn and K. J. Marfurt, "Eigenstructure-based coherence computations as an aid to 3-D structural and stratigraphic mapping," *Geophysics*, vol. 64, no. 5, pp. 1468–1479, 1999.
- [12] F. Li and W. Lu, "Coherence attribute at different spectral scales," *Interpretation*, vol. 2, no. 1, pp. SA99–SA106, 2014.
- [13] P. Karimi, S. Fomel, L. Wood, and D. Dunlap, "Predictive coherence," *Interpretation*, vol. 3, no. 4, pp. SAE1–SAE7, 2015.
- [14] X. Wu, "Directional structure-tensor based coherence to detect seismic faults and channels," *Geophysics*, vol. 82, no. 2, pp. A13–A17, 2017.
- [15] P. P. Van Bommel and R. E. Pepper, "Seismic signal processing method and apparatus for generating a cube of variance values," U.S. Patent 6151555, Nov. 21, 2000.
- [16] T. Randen *et al.*, "Automatic extraction of fault surfaces from three-dimensional seismic data," in *Proc. 81st Annu. Int. Meeting, SEG, Expanded Abstr.*, 2001, pp. 551–554.
- [17] A. A. Aqrabi and T. H. Boe, "Improved fault segmentation using a dip guided and modified 3D Sobel filter," in *Proc. 81st Annu. Int. Meeting, SEG, Expanded Abstr.*, 2011, pp. 999–1003.
- [18] D. B. Neff, J. R. Grismore, and W. A. Lucas, "Automated seismic fault detection and picking," U.S. Patent 6018498, Jan. 25, 2000.
- [19] I. Cohen, N. Coult, and A. A. Vassiliou, "Detection and extraction of fault surfaces in 3D seismic data," *Geophysics*, vol. 71, no. 4, pp. P21–P27, 2006.
- [20] D. Hale, "Methods to compute fault images, extract fault surfaces, and estimate fault throws from 3D seismic images," *Geophysics*, vol. 78, no. 2, pp. O33–O43, 2013.
- [21] X. Wu and D. Hale, "3D seismic image processing for faults," *Geophysics*, vol. 81, no. 2, pp. IM1–IM11, 2016.
- [22] X. Wu and Z. Zhu, "Methods to enhance seismic faults and construct fault surfaces," *Comput. Geosci.*, vol. 107, pp. 37–48, Oct. 2017.
- [23] S. I. Pedersen, T. Randen, L. Sonneland, and O. Steen, "Automatic fault extraction using artificial ants," in *Proc. 72nd Annu. Int. Meeting, SEG, Expanded Abstr.*, 2002, pp. 512–515.
- [24] X. Wu and S. Fomel, "Automatic fault interpretation with optimal surface voting," *Geophysics*, vol. 83, no. 5, pp. O67–O82, 2018.
- [25] L. J. Van Vliet and P. W. Verbeek, "Estimators for orientation and anisotropy in digitized images," in *Proc. 1st Annu. Conf. Adv. School Comput. Imag. (ASCI)*, Heijten, The Netherlands, 1995, pp. 442–450.
- [26] X. Wu and X. Janson, "Directional structure tensors in estimating seismic structural and stratigraphic orientations," *Geophys. J. Int.*, vol. 210, no. 1, pp. 534–548, 2017.
- [27] S. Fomel, "Applications of plane-wave destruction filters," *Geophysics*, vol. 67, no. 6, pp. 1946–1960, 2002.
- [28] Y. LeCun *et al.*, "Backpropagation applied to handwritten zip code recognition," *Neural Comput.*, vol. 1, no. 4, pp. 541–551, 1989.
- [29] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.
- [30] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.
- [31] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2014, pp. 818–833.
- [32] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 770–778.
- [33] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 580–587.
- [34] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 91–99.
- [35] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, 2017, pp. 2980–2988.
- [36] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. Med. Image Comput.-Assist. Intervent.* Cham, Switzerland: Springer, 2015, pp. 234–241.
- [37] V. Badrinarayanan, A. Kendall, and R. Cipolla, "SegNet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 12, pp. 2481–2495, Dec. 2017.
- [38] C. Cortes and V. Vapnik, "Support-vector networks," *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, 1995.
- [39] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proc. IEEE Int. Conf. Comput. Vis.*, vol. 2, Sep. 1999, pp. 1150–1157.
- [40] O. Russakovsky *et al.*, "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, Dec. 2015.
- [41] L. Huang, X. Dong, and T. E. Clec, "A scalable deep learning platform for identifying geologic features from seismic attributes," *Lead. Edge*, vol. 36, no. 3, pp. 249–256, 2017.
- [42] A. Guitton, "3D convolutional neural networks for fault interpretation," in *Proc. 80th EAGE Conf. Exhib.*, 2018, pp. 1–4.
- [43] T. Zhao and P. Mukhopadhyay, "A fault-detection workflow using deep learning and image processing," in *Proc. 88th Annu. Int. Meeting, SEG, Expanded Abstr.*, 2018, pp. 1966–1970.
- [44] H. Di, M. Shafiq, and G. AlRegib, "Patch-level MLP classification for improved fault detection," in *Proc. 88th Annu. Int. Meeting, SEG, Expanded Abstr.*, 2018, pp. 2211–2215.
- [45] B. Guo, L. Li, and Y. Luo, "A new method for automatic seismic fault detection using convolutional neural network," in *Proc. 88th Annu. Int. Meeting, SEG, Expanded Abstr.*, 2018, pp. 1951–1955.
- [46] X. Wu, L. Liang, Y. Shi, and S. Fomel, "FaultSeg3D: Using synthetic data sets to train an end-to-end convolutional neural network for 3D seismic fault segmentation," *Geophysics*, vol. 84, no. 3, pp. IM35–IM45, 2019.
- [47] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," 2015. *arXiv:1502.03167*. [Online]. Available: <https://arxiv.org/abs/1502.03167>
- [48] M. D. Zeiler, "ADADELTA: An adaptive learning rate method," 2012. *arXiv:1212.5701*. [Online]. Available: <https://arxiv.org/abs/1212.5701>
- [49] M. G. Rowan, M. P. Jackson, and B. D. Trudgill, "Salt-related fault families and fault welds in the northern gulf of Mexico," *AAPG Bull.*, vol. 83, no. 9, pp. 1454–1484, 1999.

- [50] T. Carruthers, "Interaction of polygonal fault systems with salt diapirs," Ph.D. dissertation, School Earth Ocean Sci., Cardiff Univ., Wales, U.K., 2012.
- [51] C. R. Cohen, L. J. Christianson, C. Bates, R. P. Laney, and G. A. Morton, "Pogo New Zealand's 3D seismic: New standards & structural/stratigraphic insights in the Taranaki basin," Pogo Producing Company, Houston, TX, USA, 2006.
- [52] G. F. Moore, B. B. Boston, A. F. Sacks, and D. M. Saffer, "Analysis of normal fault populations in the Kumano Forearc Basin, Nankai Trough, Japan: 1. Multiple orientations and generations of faults from 3-D coherency mapping," *Geochem., Geophys., Geosyst.*, vol. 14, no. 6, pp. 1989–2002, 2013.
- [53] D. R. Martin, C. C. Fowlkes, and J. Malik, "Learning to detect natural image boundaries using local brightness, color, and texture cues," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 5, pp. 530–549, May 2004.
- [54] F. J. Provost *et al.*, "The case against accuracy estimation for comparing induction algorithms," in *Proc. ICML*, vol. 98, 1998, pp. 445–453.



Xinming Wu received the Ph.D. degree in geophysics from the Colorado School of Mines, Golden, CO, USA, in 2016.

He was a member of the Center for Wave Phenomena, Colorado School of Mines. He was a Post-Doctoral Fellow with the Bureau of Economic Geology, University of Texas at Austin, Austin, TX, USA. He is currently a Professor with the School of Earth and Space Sciences, University of Science and Technology of China (USTC), Hefei, China. His research interests include image processing, machine

learning, 3-D seismic interpretation, subsurface modeling, and geophysical inversion.

Dr. Wu was a recipient of the Best Paper Award in Geophysics in 2016, the Best Student Poster Paper Award at the 2017 SEG Annual Convention, and the Honorable Mention Award for Best Paper at the 2018 SEG annual convention.



Yunzhi Shi received the B.S. degree in geophysics from the University of Science and Technology of China (USTC), Hefei, China, in 2015. He is currently pursuing the Ph.D. degree with the Texas Consortium for Computation Seismology (TCCS), The University of Texas at Austin, Austin, TX, USA.

He was an Intern at BP America, Inc., Houston, TX, USA. His research interests include deep-learning applications on interpretation tasks including fault detection, salt body classification, and stratigraphic correlation.



Sergey Fomel received the Ph.D. degree in geophysics from Stanford University, Stanford, CA, USA, in 2001.

He was with the Institute of Geophysics in Russia, Schlumberger Geco-Prakla, Gatwick, U.K., and the Lawrence Berkeley National Laboratory, Berkeley, CA, USA. He is currently a Professor with the Jackson School of Geosciences, The University of Texas at Austin, Austin, TX, USA, with a joint appointment between the Bureau of Economic Geology and the Department of Geological Sciences.

He is involved in the development of Madagascar—an open-source software package for geophysical data analysis.

Dr. Fomel was a recipient of a number of professional awards, including the J. Clarence Karcher Award from SEG in 2001 and the Conrad Schlumberger Award from EAGE in 2011.



Luming Liang received the B.S. degree from Central South University, Changsha, China, in 2005, and the M.S. and Ph.D. degrees from the Colorado School of Mines, Golden, CO, USA, in 2008 and 2014, respectively.

He was a Software Engineer with the Team of Information Extraction from Imagery, Advanced Technology Group, Uber, Boulder, CO, USA.

He was a Data & Applied Scientist with the Bing Geospatial Group, Microsoft, which had been acquired by Uber in 2015. He is currently a Senior

Researcher with the Applied Science Group, Microsoft, Redmond, WA, USA. He has authored or coauthored more than 30 papers in computer vision, graphics, and signal processing.



Qie Zhang received the Ph.D. degree in geophysics from the University of Missouri, Columbia, MO, USA, in 2009.

He is currently a Research Geophysicist with BP America, Inc., Houston, TX, USA. His research interests include deep learning and seismic imaging topics such as seismic acquisition, migration, and velocity inversion.



Anar Z. Yusifov received the M.Sc. degree in applied mathematics at Baku State University, Baku, Azerbaijan, in 2007.

He currently is a Senior Computational Scientist and a Research Performance and Productivity Advisor with BP America, Inc., Houston, TX, USA. His research interests include soft computing technologies and all aspects of software engineering in high performance computing.