Geophysical Journal International

Geophys. J. Int. (2019) **219,** 2097–2109
Advance Access publication 2019 September 17
GJI Marine geosciences and applied geophysics

doi: 10.1093/gji/ggz418

Multitask learning for local seismic image processing: fault detection, structure-oriented smoothing with edge-preserving, and seismic normal estimation by using a single convolutional neural network

Xinming Wu [1], Luming Liang,[2] Yunzhi Shi,[3] Zhicheng Geng[3] and Sergey Fomel[3]

[1] *School of Earth and Space Sciences, University of Science and Technology of China, Hefei, China. E-mail: xinmwu@ustc.edu.cn*
[2] *Microsoft Applied Science Group, Redmond, WA, USA*
[3] *Bureau of Economic Geology, The University of Texas at Austin, TX, USA*

Accepted 2019 September 14. Received 2019 September 5; in original form 2019 April 11

SUMMARY

Fault detection in a seismic image is a key step of structural interpretation. Structure-oriented smoothing with edge-preserving removes noise while enhancing seismic structures and sharpening structural edges in a seismic image, which, therefore, facilitates and accelerates the seismic structural interpretation. Estimating seismic normal vectors or reflection slopes is a basic step for many other seismic data processing tasks. All the three seismic image processing tasks are related to each other as they all involve the analysis of seismic structural features. In conventional seismic image processing schemes, however, these three tasks are often independently performed by different algorithms and challenges remain in each of them. We propose to simultaneously perform all the three tasks by using a single convolutional neural network (CNN). To train the network, we automatically create thousands of 3-D noisy synthetic seismic images and corresponding ground truth of fault images, clean seismic images and seismic normal vectors. Although trained with only the synthetic data sets, the network automatically learns to accurately perform all the three image processing tasks in a general seismic image. Multiple field examples show that the network is significantly superior to the conventional methods in all the three tasks of computing a more accurate and sharper fault detection, a smoothed seismic volume with better enhanced structures and structural edges, and more accurate seismic normal vectors or reflection slopes. Using a Titan Xp GPU, the training processing takes about 8 hr and the trained model takes only half a second to process a seismic volume with 128 $\times$ 128 $\times$ 128 image samples.

Key words: Image processing; Neural networks, fuzzy logic; Seismic noise.

1 INTRODUCTION

A 3-D seismic image (like the one in Fig. 1a) contains rich structural features of the subsurface. Seismic image processes often involve extracting and enhancing the seismic structure features for interpreting the subsurface. The most common seismic image processing tasks may include (1) calculating seismic attributes to detect faults and fractures (Fig. 1b), (2) smoothing the seismic image to remove noise while enhancing structures and sharpening structural edges (Fig. 1c) and (3) estimating a seismic normal vector field (illustrated as spatially oriented ellipsoids in Fig. 1d) to measure the local orientations of the folding structures (reflections) apparent in the seismic image. Although numerous methods have been proposed for all the individual seismic image processing tasks, challenges still remain in each of them.

Faults are often recognized as lateral reflection discontinuities in a seismic image. Based on this observation, numerous attributes including semblance (Marfurt *et al.* 1998), coherency (Marfurt *et al.* 1999; Li & Lu 2014; Karimi *et al.* 2015), variance (Van Bemmel & Pepper 2000; Randen *et al.* 2001), curvature (Roberts 2001; Al-Dossary & Marfurt 2006; Di & Gao 2016) and fault likelihood (Hale 2013; Wu & Hale 2016) are proposed to detect faults by highlighting the reflection discontinuities in a seismic image. In calculating most of these attributes (e.g. Gersztenkorn & Marfurt 1999; Marfurt *et al.* 1999; Hale 2009b, 2013; Karimi *et al.* 2015), reflection slopes are first estimated from the seismic image and then used to measure reflection discontinuity along the reflection orientations, which is helpful to remove artefacts due to folding structures. These attributes, based on measuring reflection discontinuities, can be sensitive to noise and stratigraphic features that are

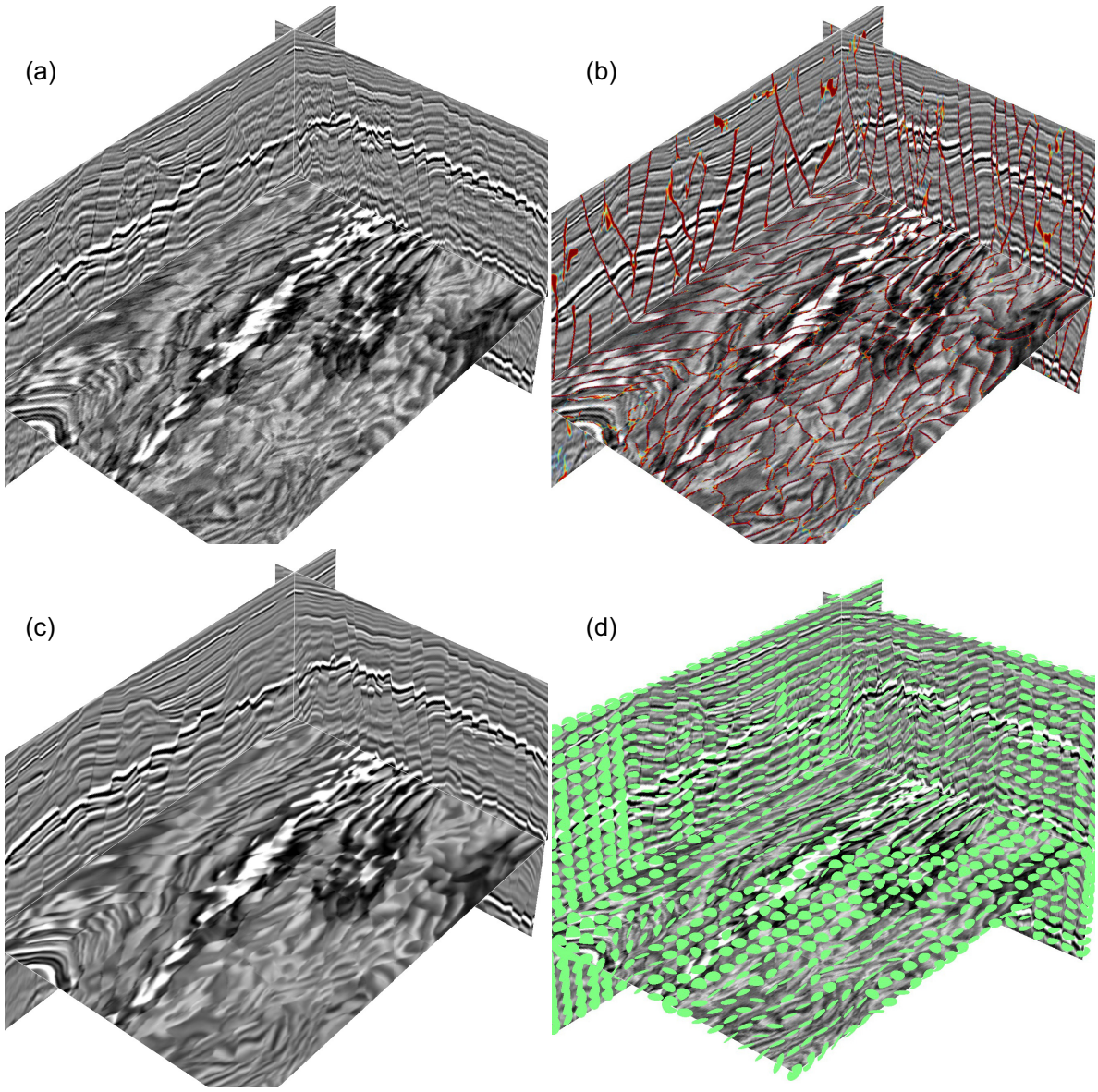

Figure 1. From an input 3-D seismic amplitude volume (a), we simultaneously detect faults (b), enhance the seismic reflections with edge-preserving (c) and estimate the local reflection orientations [illustrated as spatially oriented ellipsoids in (d)] by using a single CNN.

unrelated to faults but are also apparent as discontinuities within a seismic image. Therefore, some extra processing such as ant tracking (Pedersen *et al.* 2002, 2003) and optimal surface voting (Wu & Fomel 2018a) are required to further enhance the fault features while suppressing the noisy features in a fault attribute image.

Structure-oriented smoothing with edge-preserving is a widely used processing in either common images (Weickert 1997, 1998, 2001) or seismic images (Fehmers & Höcker 2003; Hale 2009b; Liu *et al.* 2010; Wu & Guo 2019) to enhance structural features while sharpening structurally meaningful edges in an input image. By removing noise and enhancing structures in a seismic image, this image processing is helpful to facilitate seismic structural (Fehmers & Höcker 2003) and stratigraphic (Wu & Guo 2019) interpretation. The structure-oriented smoothing with edge-preserving, however, requires first estimating structural orientations and detecting edges within the input image. The accuracy of smoothing and edge-preserving highly depends on the accurate estimation of structural orientations and edges, each of which can be a challenging task in practice.

Estimating normal vectors or slopes of seismic reflections is a basic seismic image processing step for many other geophysical tasks including calculating slope-guided attributes (Gersztenkorn & Marfurt 1999; Marfurt *et al.* 1999; Hale 2009b, 2013), structure-oriented smoothing (Fehmers & Höcker 2003; Hale 2009b; Wu & Guo 2019), horizon picking (Lomask *et al.* 2006; Fomel 2010; Wu & Fomel 2018b; Di *et al.* 2018), structure-guided interpolation (Hale 2009a) and regularization (Clapp *et al.* 2004; Wu 2017). Multiple methods including structure tensors (Bakker 2002; Hale 2009b), plane wave destruction (Fomel 2002), coherence scanning (Marfurt 2006), and directional structure tensors (Wu & Janson 2017) have been proposed to estimate reflection slopes or orientations from a seismic image. As discussed by Wu & Hale (2015), all these methods perform well in estimating orientations for structures with only one locally dominant orientation but yield inaccurately smooth

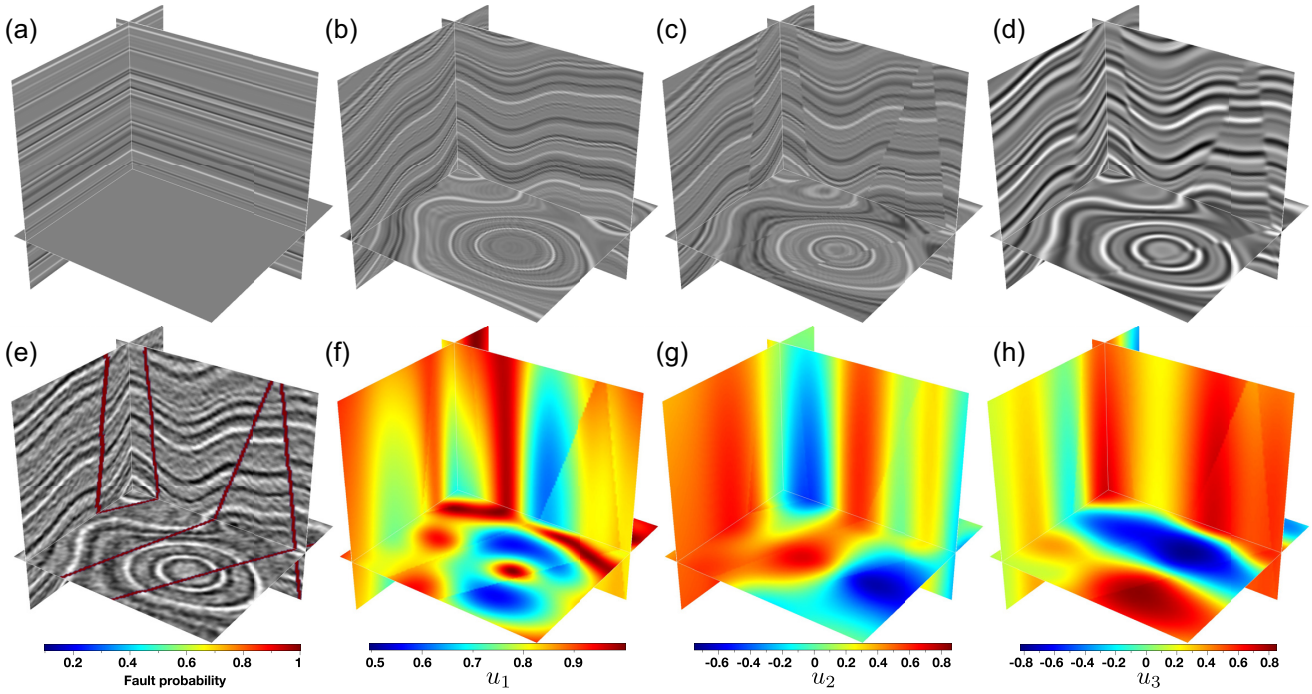

Figure 2. The workflow of creating 3-D synthetic training data sets. We first generate a horizontal reflectivity model (a) with a sequence of random values. We then sequentially add folding (b) and faulting (c) to the reflectivity model. We finally convolve the reflectivity model with a Ricker wavelet to obtain a clean seismic image (d) and add some random noise to obtain a noisy image (e). At the same time of generating the seismic image, we record the ground truth of the fault positions (e) and the three components (f–h) of the seismic normal vectors.

orientation estimation near structural edges where multiple structures meet. In order to solve this problem, some authors (Brox *et al.* 2006; Wu & Hale 2015) propose to construct non-linear structure tensors to more accurately estimate discontinuous orientations. Constructing the nonlinear structure tensors, however, involves the other two processes of edge detection and edge-preserving smoothing.

With successful applications in natural image processing tasks of classification (Krizhevsky *et al.* 2012; Zeiler & Fergus 2014; He *et al.* 2016), segmentation (Ronneberger *et al.* 2015; Badrinarayanan *et al.* 2017), object detection (Girshick *et al.* 2014; Ren *et al.* 2015; He *et al.* 2017) and so on, the machine learning methods have been recently introduced into seismic data processing tasks for seismic interpretation (Wu *et al.* 2019, 2019; Shi *et al.* 2019; Pham *et al.* 2019), inversion (Yang & Ma 2019; Li *et al.* 2019) and denoising (Yu *et al.* 2018; Zhu *et al.* 2018; Wang & Nealon 2019). As a subfield of machine learning, multitask learning (MTL, Caruana 1997; Argyriou *et al.* 2007; Ruder 2017) is proposed to simultaneously solve multiple tasks that are related to each other. By exploiting commonalities and differences across the multiple tasks, learning multiple related tasks from data simultaneously improves efficiency and prediction accuracy compared to learning these tasks individually (Evgeniou & Pontil 2004). MTL improves the generalization of the network by sharing representation and knowledge when learning multiple tasks in parallel. The representation and knowledge learned for each task can help other tasks be better learned (Caruana 1997). In addition, by sharing the training samples of multiple related tasks, MTL is helpful to solve the problem of lacking samples in each task (Caruana 1997; Zhang & Yang 2017). In this paper, we design a MTL network to simultaneously solve the above three seismic image processing tasks of detecting faults, structure-oriented smoothing with edge-preserving and estimating reflection orientations.

As discussed above, the three image processing tasks are coupled with each other and they all involve the analysis of seismic structural features. In conventional methods, however, the three tasks are independently performed by different algorithms, each of which needs to be carefully designed. Performing one processing typically requires accurate results precomputed from the other processes, which actually limits the performance of each processing in the conventional methods. Based on these observations, we solve the three seismic image processing tasks at the same time by taking the advantage of the similarities between the tasks. Specifically, we design a single convolutional neural network (CNN) to simultaneously perform all the three image processing tasks. This CNN consists of a feature extraction network followed by a set of three prediction networks. The feature extraction is implemented by using a simplified U-net architecture (Ronneberger *et al.* 2015; Wu *et al.* 2019), which efficiently computes multiscale structural features that are useful for all the three image processing tasks. The prediction part is implemented by three sets of residual blocks, which take the same output features of the U-net as inputs and make three predictions of a fault image, a smoothed image with enhanced structures and sharpen edges, and a seismic normal vector field. To train the network, we automatically generate noisy synthetic seismic images and the corresponding ground truth of fault positions, clean seismic images and seismic normal vectors. Although trained with only synthetic data sets, the network works well to perform the three image processing tasks in field data sets that are acquired at totally different surveys.

2 TRAINING DATA SETS

Training and validating a CNN model often requires a large amount of images and corresponding labels. In training a CNN to detect faults, enhance a seismic image and estimate seismic reflection orientations, we need a lot of 3-D noisy seismic amplitude volumes as inputs and the corresponding target volumes of fault labels, clean seismic amplitudes and seismic normal vectors. It might not be difficult to obtain a lot of input seismic amplitude volumes by extract subvolumes from large field seismic data sets and apply various types of data augmentation to the extracted volumes. However, it is impossible to fully label all the faults within the field seismic volumes or to obtain the corresponding clean seismic volumes and accurate reflection slopes. To solve this problem, we automatically create a lot of synthetic data sets to train and validate our CNN model.

Fig. 2 shows the workflow (Wu *et al.* 2019) of creating synthetic seismic amplitude volumes and the corresponding label volumes. In this workflow, we start with an initially flat reflectivity model (Fig. 2a), which is created by horizontally extending a 1-D trace of random reflectivity values. We then add a combination of folding and dipping structures (Fig. 2b) in this reflectivity model. We further add faulting to obtain a folded and faulted reflectivity model (Fig. 2c). We finally convolve the reflectivity model with a Ricker wavelet to simulate a synthetic seismic volume with clean amplitude values shown in Fig. 2(d). A noisy seismic amplitude volume in Fig. 2(e) is further obtained by add some random noise. Note that we apply the convolution after the faulting because the convolution is helpful to blur the sharp faults (Fig. 2c) and therefore make them more realistic (Fig. 2d). In addition, the convolution is applied in directions perpendicular to the reflectivity structures as discussed by Wu *et al.* (2019). At the same time of creating the folding and faulting structures within the reflectivity model, we also record the fault positions (overlaid with the noisy seismic volume in Fig. 2e) and the three components (Figs 2f–h) of the normal vector field that defines the folding structures in the 3-D volume. The normal vectors are unit vectors, which are pointing downward (with the positive vertical component $u_1(x, y, z) > 0$ as shown in Fig. 2f) and perpendicular to the folding structures everywhere in the 3-D volume. In training our deep CNN (Fig. 3), the input is a noisy seismic amplitude volume (Fig. 2e), the expected outputs include a fault image with ones at fault positions and zeros elsewhere (Fig. 2e), a clean seismic amplitude volume (Fig. 2d), and three components(Figs 2f–h) of normal vector field. By using the workflow, we automatically created 100 sets of 3-D volumes, each volume contains $128 \times 128 \times 128$ samples. We further apply data augmentation to these 100 sets of volumes and obtain a lot more training data sets.

2.1 Data augmentation

All the tasks of detecting faults, enhancing the seismic amplitude volume and estimate local structural orientations involve only local image processing within local windows. This means that we do not need the whole large volumes but only small cubes of training data sets to train the network for these three local image processing tasks. Therefore, the first type of data augmentation is simply to extract overlapping subvolumes [with 64 (vertical) $\times$ 56 (inline) $\times$ 56 (crossline) samples] from the automatically created larger training data sets. We choose vertically elongated subvolumes because faults often appear more vertical than horizontal within a seismic amplitude volume.

The second type of data augmentation is to rotate the extracted subvolumes around the vertical time or depth axis. To avoid interpolation or artefacts near boundaries, we rotate a volume by only three options of 90°, 180° and 270°. If we rotate an input seismic volume by θ, we simply apply the same rotation to the corresponding fault volume and clean seismic volume to make them consistent with the input seismic volume. For the normal vector field, however, we need to first transform the vector field by applying a rotation matrix (defined by θ) as follows:

$$\begin{bmatrix} \tilde{u}_3(x,y,z) \\ \tilde{u}_2(x,y,z) \\ \tilde{u}_1(x,y,z) \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u_3(x,y,z) \\ u_2(x,y,z) \\ u_1(x,y,z) \end{bmatrix}, \quad (1)$$

where $u_1(x, y, z)$, $u_2(x, y, z)$ and $u_3(x, y, z)$, respectively, are the vertical, inline and crossline components of the original normal vector field, as shown in Figs 2(f)–(h). After the transformation, we then rotate each volume of $\tilde{u}_1(x, y, z)$, $\tilde{u}_2(x, y, z)$ and $\tilde{u}_3(x, y, z)$ to make them consistent with the rotated input seismic volume. By applying the two types of data augmentation, we significantly increase the diversity of the data sets and totally obtain 9000 sets of data sets. We use 90 per cent of the data sets for training and the rest for validation.

3 SEISMIC IMAGE PROCESSING BY CNN

All the three seismic image processing tasks of detecting faults, enhancing the seismic image, and estimating seismic structural orientations involve the analysis of geometric features apparent in a 3-D seismic image. In calculating an attribute for fault detection, we often take into account the reflection orientations or slopes to compute structure-oriented or slope-guided coherence (Gersztenkorn & Marfurt 1999; Karimi *et al.* 2015) or semblance (Marfurt *et al.* 1999; Hale 2009b, 2013). In enhancing a seismic image, we expect to smooth the image along seismic structural orientations and preserve the discontinuities near faults (Fehmers & Höcker 2003; Hale 2009b; Wu & Guo 2019), which requires first estimating structure orientations and edges. Estimating seismic structural orientations (Hale 2009b; Wu & Janson 2017) is especially a direct analysis of the seismic geometric features. Therefore, the three image processing tasks are coupled with each other and all involve the analysis of seismic structures. Based on these observations, we consider to perform all the three tasks by using a MTL CNN.

3.1 CNN architecture

Fig. 3 shows the CNN that we use for simultaneously perform the three image processing tasks. With an input noisy seismic amplitude volume on the left, the CNN simultaneously computes three outputs (on the right) of a fault volume, a smoothed seismic amplitude volume with enhanced reflections and sharpen faults, and a normal vector field with three components. The first part of shared layers (on the left of the dashed black line in Fig. 3) forms an encoder-decoder network which is simplified from the original U-net (Ronneberger *et al.* 2015) by reducing the number of convolutional layers and the number of features at each layer. This simplified U-net has been successfully applied to the 3-D fault segmentation (Wu *et al.* 2019), where details of the network are discussed. With downsampling and upsampling structures, this U-net provides an effective way to not

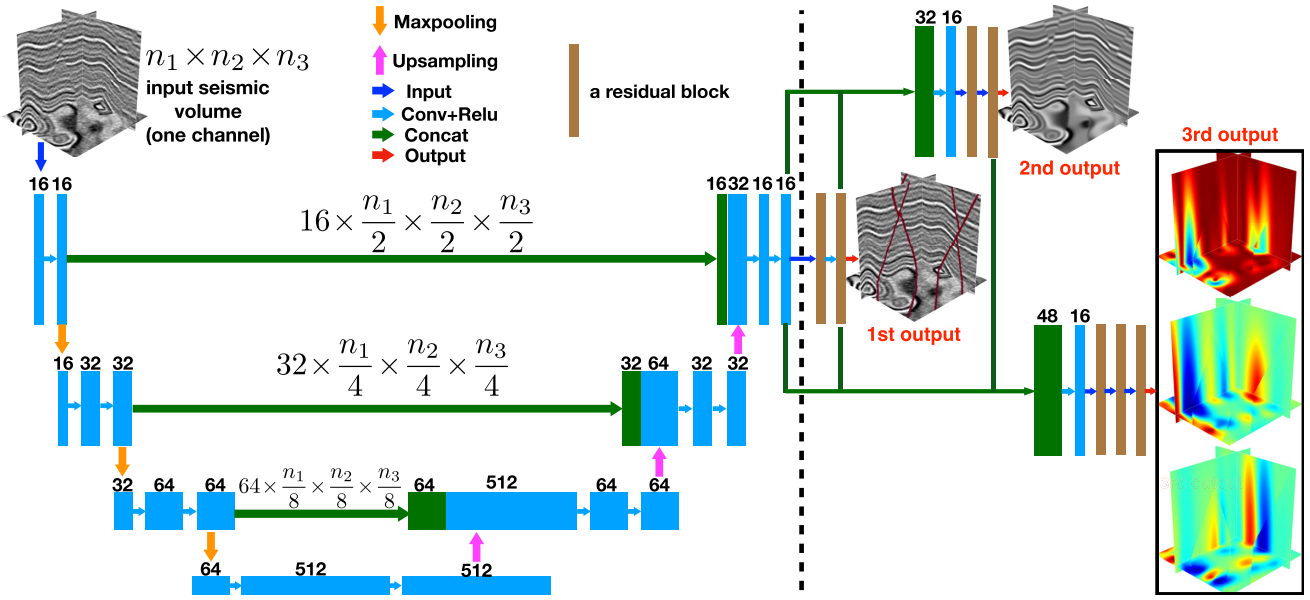

Figure 3. A deep convolutional neural network to simultaneously detect faults, compute a smoothed image with edge-preserving, and estimate seismic normal vectors from an input seismic image.

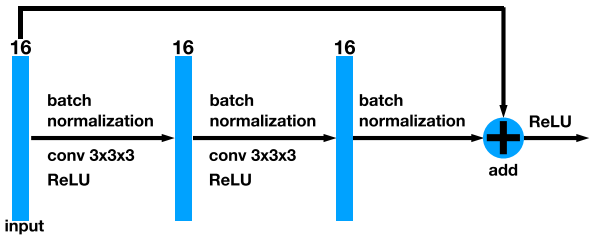

Figure 4. The architecture of a residual block.

only save GPU memory and computational costs but also compute seismic structural features in multiple scales.

The output 16 features of the U-net are further used as the shared inputs for the second part of prediction network (on the right of the dashed black line in Fig. 3), which contains three separate sets of residual blocks (brown blocks in Fig. 3) to compute three final outputs as shown on the right-hand side of the network (Fig. 3). As shown in Fig. 4, each residual block is designed as a skip connection over two convolutional layers (each layer generates 16 features), followed by batch normalization layers and non-linear activations (ReLU). With the 16 input features, computed from the U-net, the first set of two residual blocks computes the 1st final output of a fault image.

The 16 U-net features and the 16 output features of the first set of residual blocks are concatenated together as inputs for the second set of two residual blocks to compute the 2nd final output, which is a smoothed seismic amplitude volume with enhanced reflections and sharpen fault discontinuities. Similar to conventional structure-oriented smoothing with edge preserving (Fehmers & Höcker 2003; Hale 2009b; Wu & Guo 2019), we include the 16 fault features (outputs of the first set of residual blocks) in the input features and expect them to help preserve the fault edges in calculating the smoothed seismic volume. Before feeding these concatenated 32 feature maps into the residual blocks, we add an extra convolutional layer (with 16 features) to make sure the number of input feature maps is consistent with the number of features (16 features) defined at each layer of the residual blocks, so that we can consistently add the input features and output features at the end of the residual block as shown in Fig. 4.

The 16 U-net features, together with the previously computed outputs of the two sets of residual blocks, are all concatenated together as inputs for a convolutional layer (with 16 features) followed by the third set of three residual blocks to compute the 3rd final output, which is a seismic normal vector field with three components $u_1(x, y, z)$ (vertical), $u_2(x, y, z)$ (inline) and $u_3(x, y, z)$ (crossline) corresponding three channels in the output. This is similar to estimate discontinuous or multiple orientations by using non-linear structure tensors (Brox *et al.* 2006; Wu & Hale 2015), where structural edges and non-linear smoothing with edge-preserving are used to construct the non-linear tensors. Estimating seismic normal vector field is a more challenging regression problem for a convolutional network, we therefore add one more residual block than in calculating the other two outputs. The normal vector field and reflection slopes $p_2(x, y, z) = -\frac{u_2(x,y,z)}{u_1(x,y,z)}$ (inline) and $p_3(x, y, z) = -\frac{u_3(x,y,z)}{u_1(x,y,z)}$ (crossline) can be used to equivalently describe the 3-D local orientations of reflections. However, we suggest to estimate the normal vector field, instead of reflection slopes, because the output normal vectors are all normalized as unit vectors for both training and prediction. On the other hand, the reflection slopes cannot be normalized (normalized slopes will no longer be consistent with the structures in the input seismic volume) and the range of the slopes are various for different input seismic volumes, which makes the slope estimation less stable than the normal vector field estimation by using a CNN.

In designing this CNN architecture, we find that the fault segmentation is an easier task for the CNN compared to the other two tasks. We therefore first compute a fault segmentation with less convolutional layers and then compute a smoothed seismic image and a seismic normal vector field with deeper convolutional layers. It has been shown by Wu *et al.* (2019) that only the U-net (first part of this CNN architecture) is good enough to perform fault segmentation well. In this CNN architecture (Fig. 3), we still add two more residual blocks followed by the U-net to compute a fault segmentation because we expect the output features of the U-net contain more general features (not just fault features) that can be shared for the

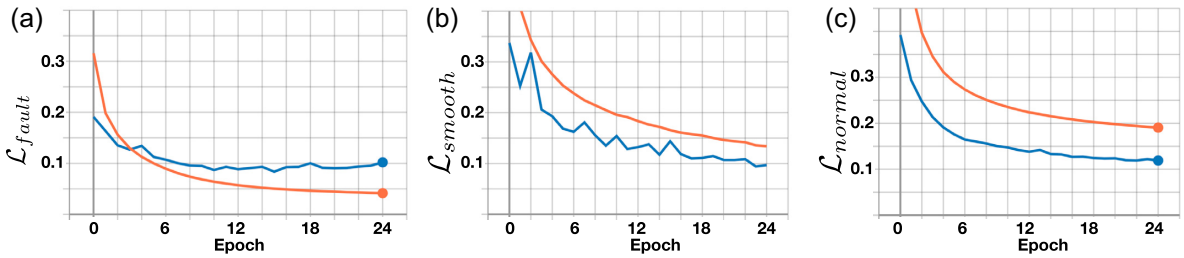

Figure 5. Three losses of the model for training (orange curves) and validation (blue curves) data sets.

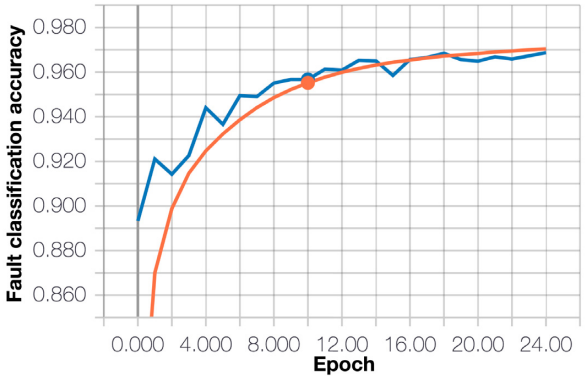

Figure 6. Mean accuracy rate across all predictions for fault classifications in the training (orange curve) and validation (blue curve) data sets.

other two tasks. In our experiments, estimating the seismic normal vector field is the most challenging task (among the three tasks) for the CNN. We therefore consider the seismic normal estimation as the final output with the deepest layers. In addition, it is consistent with using the non-linear structure tensors (Brox *et al.* 2006) to estimate discontinuous orientations, where the structural edges are first detected and an edge-preserving smoothing is applied for the orientation estimation.

3.2 Loss functions

In training the network (Fig. 3) for predicting the three results of a fault image, an enhanced seismic volume and a normal vector field, we use three different loss functions. Fault detection is a classic binary segmentation problem, where the binary cross-entropy loss function is widely used. However, the regular cross-entropy loss might not be suitable to measure the errors of fault segmentation, where the number of fault samples (labelled by ones) and non-fault samples (labelled by zeros) in a fault image are highly imbalanced (more than 90 per cent of samples are zeros) as discussed by Wu *et al.* (2019). We therefore use a balanced cross-entropy loss function as discussed by Xie & Tu (2015):

$$\mathcal{L}_{fault} = -\beta \sum_{i=0}^{i=N} y_i \log(p_i) - (1-\beta) \sum_{i=0}^{i=N} (1-y_i) \log(1-p_i), \quad (2)$$

where $\beta = \frac{\sum_{i=0}^{i=N}(1-y_i)}{N}$ represents the ratio between non-fault pixels and the total image pixels while $1-\beta$ denotes the ratio of fault pixels.

Estimating a smoothed seismic volume is a classic regression problem. We therefore use the following most common regression loss function:

$$\mathcal{L}_{smooth} = \frac{\sum_{i=1}^{N}(y_i - y_i^p)^2}{N}, \quad (3)$$

which measures the mean square error (MSE) between the target (y_i) and predicted (y_i^p) seismic amplitude volumes.

In estimating a seismic normal vector field, the output has three channels corresponding to the vertical, inline and crossline components of the normal vectors. To measure the error between the predicted and true normal vectors at each sample, we use the following normalized cosine similarity loss function:

$$\mathcal{L}_{normal} = \frac{1}{N} \sum_{i=1}^{N} \left(1 - \frac{\langle \mathbf{u}_i^p, \mathbf{u}_i^y \rangle}{\sqrt{\langle \mathbf{u}_i^p, \mathbf{u}_i^p \rangle}} \right), \quad (4)$$

where $\mathbf{u}_i^p$ and $\mathbf{u}_i^y$ represent the predicted and true normal vector fields. The true normal vectors are unit vectors while the predicted normals are not necessary unit vectors. We therefore normalize the predicted normals to unit vectors at each sample and backpropagate this normalization. To be consistent, we also normalize the predicted normals when applying the trained CNN model to any test data set.

We train the network by using a combination of these three loss functions. We notice that the gradients or errors measured by the last cosine similarity loss function (eq. 4) are relatively smaller than those of the other two loss functions. In addition, the gradients need to be backpropagated more layers to update the network as the seismic normal estimation is the final output with the deepest layers as shown in Fig. 3. To be able to effectively backpropagate the gradients measured by the cosine similarity loss function, we amplify the gradients by the scale of 10 before the backpropagation.

3.3 Training and validation

We train the CNN by using 8100 sets of input seismic volumes and the corresponding target volumes that are extracted from the automatically generated synthetic data sets by using the workflow shown in Fig. 2. The validation data set contains another 900 sets of data sets, which are not included in the training data set. We normalize both the input noisy seismic amplitude volumes and the target clean amplitude volumes before the training and validation. In this normalization, the input and target seismic amplitude volumes are not independently normalized. Instead, each pair of the input and target seismic volumes are subtracted by the same mean of the input seismic volume and then divided by the same standard deviation of the input seismic volume. In this way, we are able to recover the seismic amplitude values by using only the mean and standard deviation of the input seismic volume but without the need of the mean or standard deviation of the true seismic volume which are unknown in the prediction.

As the three seismic image processing tasks of detecting faults, enhancing the seismic amplitude volume and estimating seismic

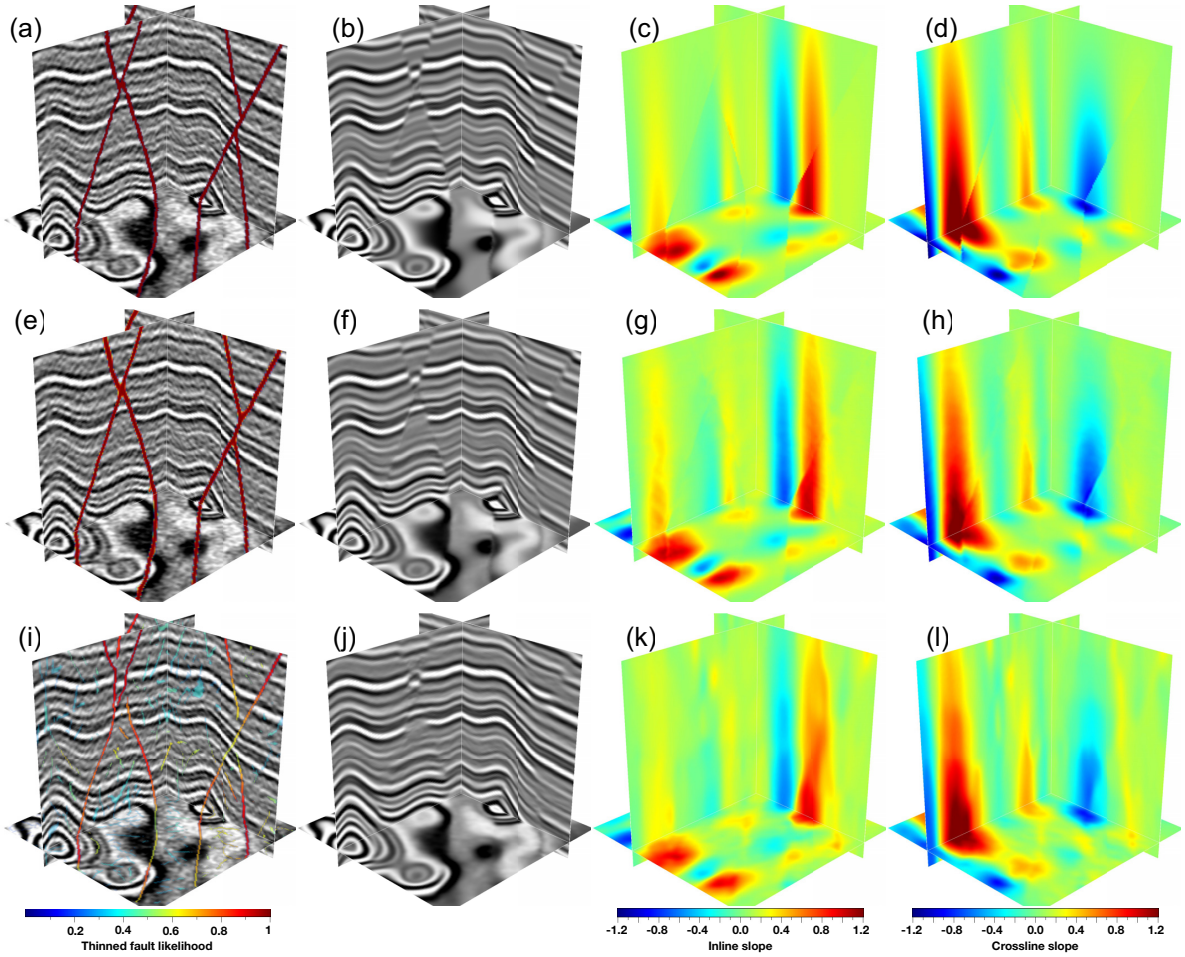

Figure 7. Top row: ground truth of (a) faults (overlaid with the input noisy seismic image), (b) the clean seismic image, (c) inline and (d) crossline slopes. Middle row: (e) fault detection, (f) smoothed seismic image with edge-preserving, (g) inline and (h) crossline slopes are simultaneously computed from the input noisy seismic image by using the proposed CNN (Fig. 3). Bottom row: conventional methods of (i) fault detection by calculating a TFL (thinned fault likelihood) image, (j) smoothed image by structure-oriented smoothing, (k) inline and (l) crossline slopes estimated by structure tensors.

normal vectors are local image processing, we train the CNN by using small sub-volumes with 64 × 56 × 56 samples, which significantly increases the number and diversity of the training data sets and greatly improves the training efficiency. Although trained with small subvolumes, this CNN model can be applied to real seismic volumes with different dimensions. The only limitation is that each dimension needs to be divisible by 8 as we have three times of downsampling (reduce the dimension of an input by half) and upsampling (double the dimension of an input) included in the CNN architecture.

We feed the 3-D noisy seismic volumes to the neural network in batches and each batch contains 8 volumes, which consist of 2 original seismic volumes and the same volumes rotated around the vertical axis by 90°, 180° and 270°. We use the Adam method (Kingma & Ba 2014) to optimize the network parameters and set the learning rate to be 0.0001. We train the network with 25 epochs and all the 8100 training data sets are processed at each epoch. As shown in Figs 5 and 6, all the three loss functions for both training (orange curves) and validation (blue curves) converges to small values while the fault classification accuracy gradually increases to 97 per cent after 25 epochs.

To verify the CNN model trained with 25 epochs, we apply this trained model to the synthetic seismic volume with 128 × 128 × 128 samples (Fig. 7a), which was not included in the training data sets. Figs 7(a)–(d) are the ground truth of the fault image (overlaid with the noisy seismic volume), clean seismic volume, inline and crossline slope volumes, respectively. By using the trained CNN model, we simultaneously predict a fault probability image (Fig. 7e), a clean seismic volume (Fig. 7f), inline (Fig. 7g) and crossline (Fig. 7g) slope volumes. Using one GPU (Titan Xp), the CNN took only half second to predict all the results shown in the middle row of Fig. 7. All of the predictions match well with the ground truth shown in the top row. Note that the CNN model actually predicts a seismic normal vector field with three components $u_1(x, y, z)$, $u_2(x, y, z)$ and $u_3(x, y, z)$. The inline (Fig. 7g) and crossline (Fig. 7h) slopes are converted from the estimated normal vectors by $p_2(x, y, z) = -\frac{u_2(x,y,z)}{u_1(x,y,z)}$ and $p_3(x, y, z) = -\frac{u_3(x,y,z)}{u_1(x,y,z)}$. The motivation of displaying the two slope volumes, instead of the three components of the normal vector field, is simply to reduce the number of figures in this paper.

In comparison, we also use conventional methods to perform the three seismic image processing tasks as shown in the bottom row of Fig. 7. Fig. 7(i) shows a TFL (thinned fault likelihood) image (Hale 2013; Wu & Hale 2016). Fig. 7(j) shows a smoothed seismic amplitude volume computed by using the conventional anisotropic and spatially varying diffusion (Hale 2009b; Wu & Guo 2019) guided by

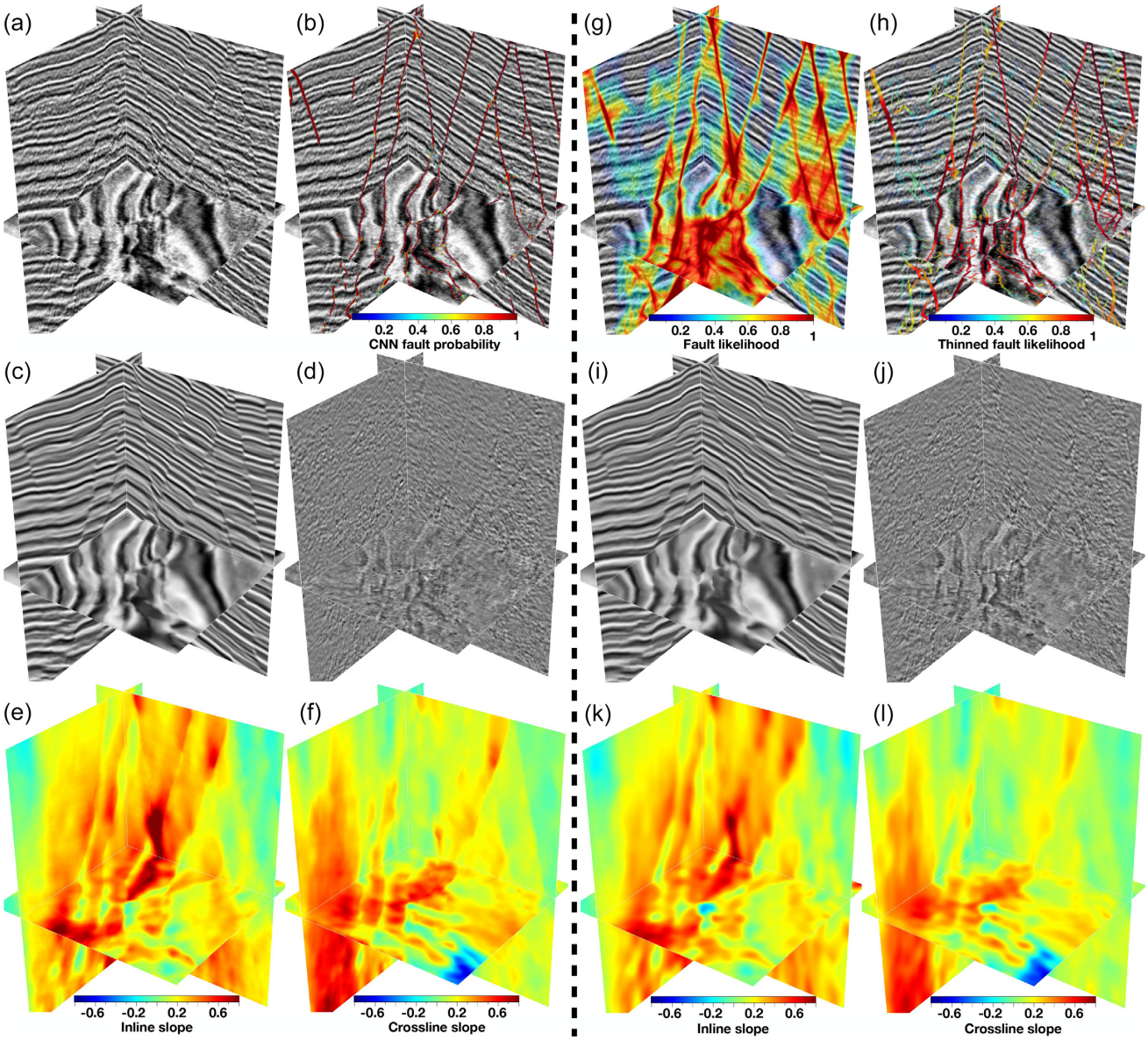

Figure 8. With an input seismic volume (a), the proposed CNN simultaneously computes a clean and sharp fault image (b), a smoothed seismic image (c) with structures enhanced while noise removed (d), inline (e) and crossline (f) slopes. As a comparison, we use three conventional methods of fault likelihood scanning (Hale 2013; Wu & Hale 2016), structure-oriented smoothing (Hale 2009b), and structure tensors (Hale 2009b), respectively, to compute a fault likelihood image before (g) and after (h) thinning, a smoothed image (i) with noise removed (j), and inline (k) and crossline (l) slopes from the same seismic image (a).

structural orientations and weighted by the fault attribute (Fig. 7i). Figs 7(k) and (l) show the inline and crossline slopes estimated by using the conventional structure tensors (Hale 2009b). From the comparison, the CNN is significantly superior to the conventional methods in all the three image processing tasks because it can provide a more accurate and sharper fault detection, a smoothed seismic volume with better enhanced reflections and better preserved faults, and more accurate inline and crossline slopes with discontinuities better preserved near faults. In addition, we are able to perform all the three image processing tasks by using a single CNN, instead of carefully designing a different algorithm for each task in the conventional methods.

3.4 Applications

In addition to the validation in the synthetic examples (Fig. 7), we further verify the same CNN model, trained with only synthetic data set, on four field seismic volumes that are acquired at different surveys. To be consistent with the synthetic training seismic volumes, each of the field seismic volumes is subtracted by its mean and then divided by its standard deviation to obtain a normalized volume. This normalization modifies the seismic amplitude values but does not change the seismic structure patterns (which depend on amplitude variations). This means that this normalization does not affect our image processing tasks that are based on analyzing seismic structure patterns.

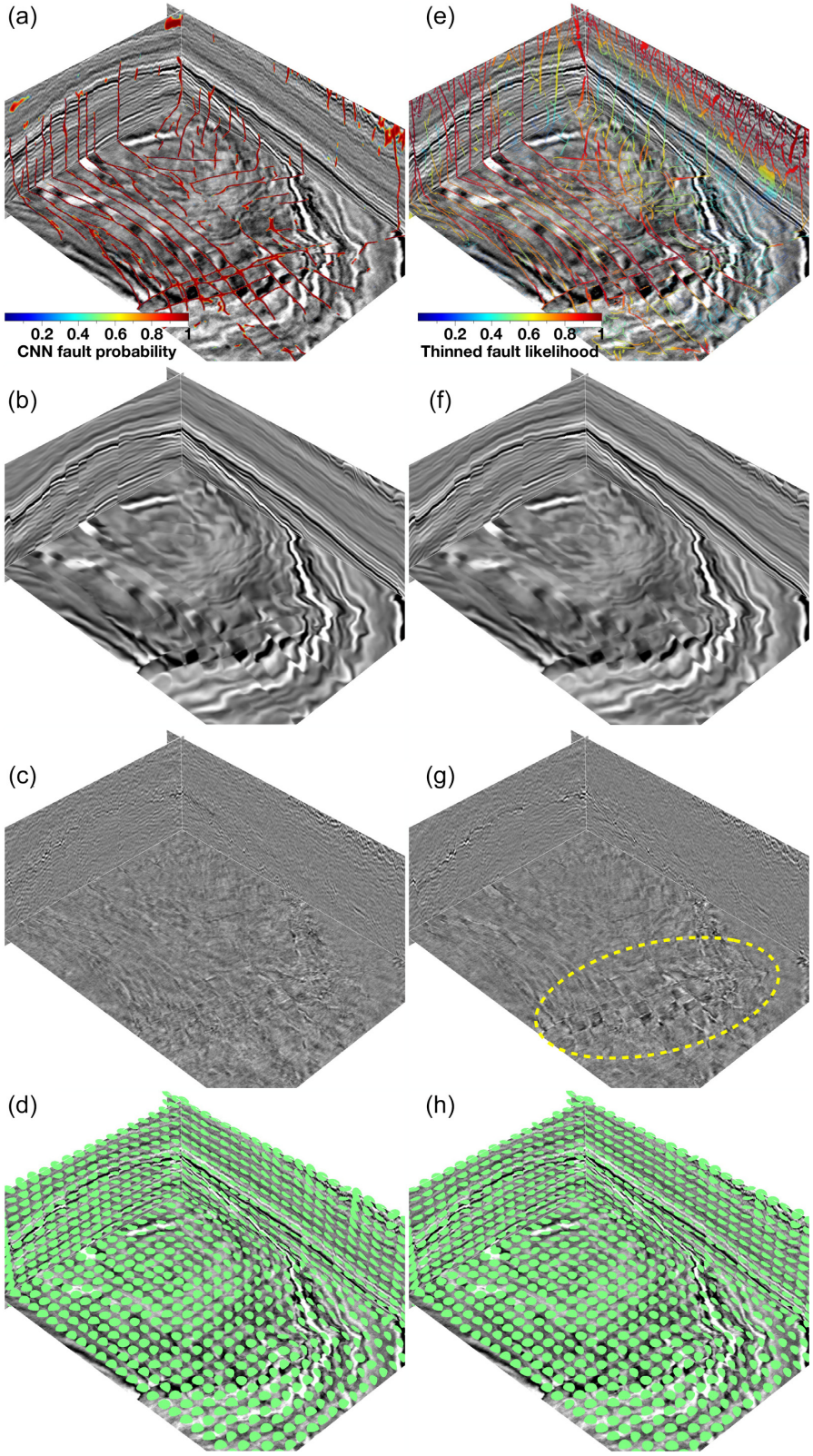

Figure 9. With an input seismic image [background grey image in (a)], the proposed CNN simultaneously performs three image processing tasks of computing a clean and sharp fault probability image (a), a smoothed seismic image (b) with enhanced reflections and faults while noise removed (c), and a seismic normal vector field (d). As a comparison, we use three conventional methods of fault likelihood scanning (e), structure-oriented smoothing (f–g), and structure tensors (h) to perform the corresponding three image processing tasks on the same input seismic image. The dashed yellow ellipse in (g) shows that the conventional structural oriented smoothing method removes more meaningful structure features than our CNN-based method.

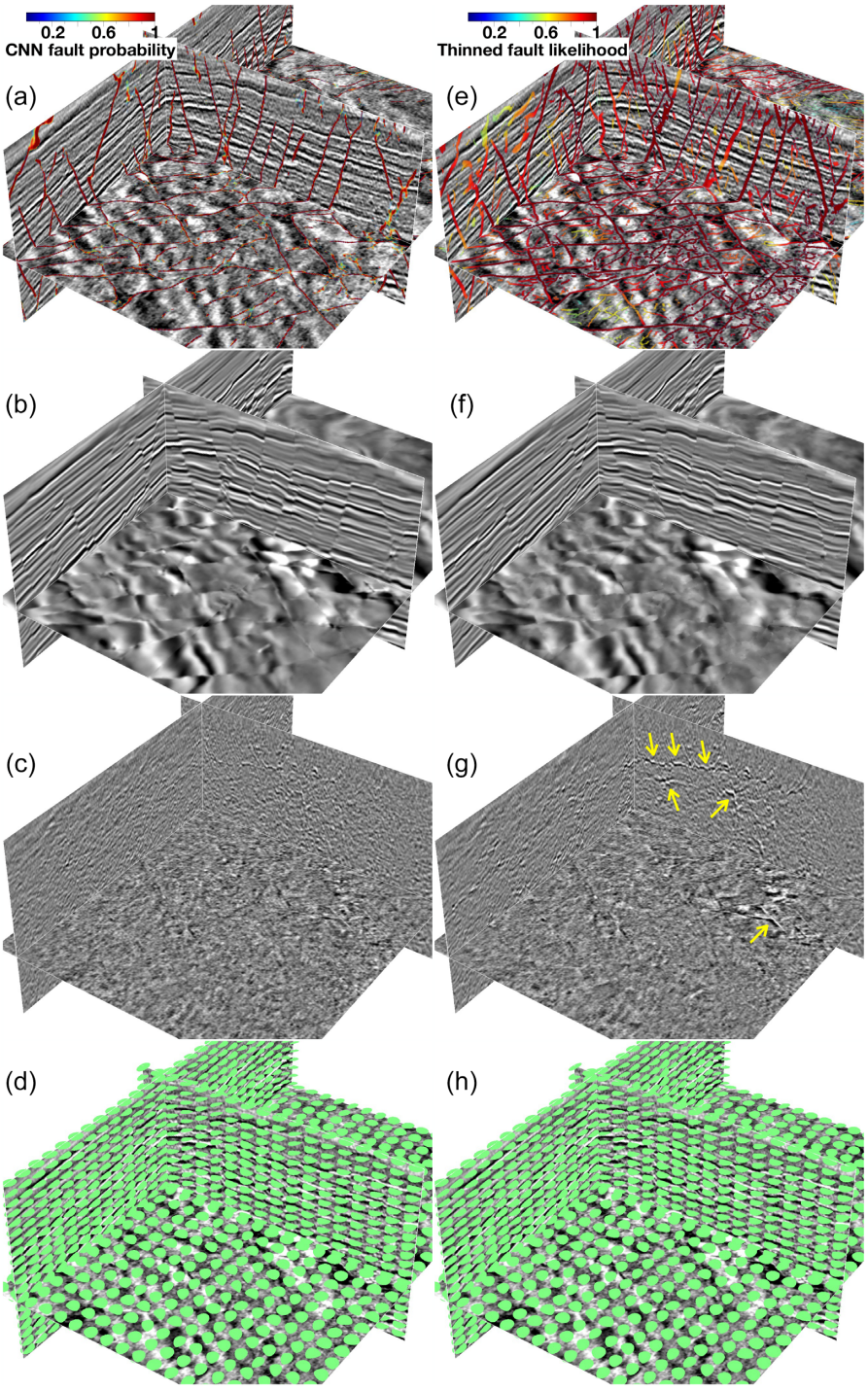

Figure 10. With an input seismic image [background grey image in (a)], the proposed CNN simultaneously performs three image processing tasks of computing a clean and sharp fault probability image (a), a smoothed seismic image (b) with enhanced reflections and faults while noise removed (c), and a seismic normal vector field (d). As a comparison, we use three conventional methods of fault likelihood scanning (e), structure-oriented smoothing (f–g), and structure tensors (h) to perform the corresponding three image processing tasks on the same input seismic image. The yellow arrows in (g) shows that the conventional smoothing method removes more meaningful structure features than the CNN-based method.

Fig. 1 a shows a seismic amplitude volume (240 (vertical) × 400 (inline) × 592 (crossline) samples) that is acquired at the Campos Basin, offshore Brazil. The reflections within the seismic volume are heavily faulted due to the salt bodies at the bottom of the volume. With this input seismic volume, the proposed CNN simultaneously computes a fault probability image overlaid with the seismic volume in Fig. 1(b), a smoothed seismic amplitude volume (Fig. 1c) with enhanced reflections and sharpen faults, and seismic normal vectors, which are illustrated by the spatially oriented ellipsoids in Fig. 1(d). In the output CNN fault probability image (Fig. 1b), most of the faults are clearly and accurately labeled except some subtle faults. The horizontal slice displays clear patterns of

polygonal faults that may be associated with salt diapirs (Rowan *et al.* 1999; Carruthers 2012). In the smoothed seismic amplitude volume (Fig. 1c), the seismic reflections are more clear and faults are sharper than those in the original seismic volume (Fig. 1a). The green ellipsoids (Fig. 1d), oriented by the estimated seismic normal vectors, consistently align with reflections within the seismic volume.

Fig. 8 shows another field example. With the input seismic image (Fig. 8a), the proposed CNN simultaneously computes three outputs of a fault probability image (Fig. 8b), a smoothed seismic amplitude image (Fig. 8d), and a normal vector field, which are further used to compute the inline (Fig. 8e) and crossline (Fig. 8f) slopes. In the smoothed seismic volume, noise are removed (Fig. 8d) where all the reflections and faults are significantly enhanced. In comparison, we first compute a fault likelihood image before (Fig. 8g) and after (Fig. 8h) thinning from the same input seismic image by using the fault likelihood scanning method (Hale 2013; Wu & Hale 2016). The fault likelihood images (Figs 8g and h) can detect most of the faults, however, are more noisy than the CNN fault probability image (Fig. 8b). We then compute a smoothed seismic amplitude image (Fig. 8i) by using the structure-oriented smoothing (Hale 2009b) guided by structural orientations and weighted by the thinned fault likelihood image. This conventional smoothing method can also enhance structures and preserve faults, however, requires first estimating structural orientations and detecting faults. We further estimate inline and (Fig. 8k) crossline (Fig. 8l) slopes from the same seismic image by using structure tensors (Hale 2009b). These slopes show similar feature patterns as in the slopes estimated by our CNN-based method. However, our CNN method better preserves slope discontinuities near the faults where the hanging wall and foot-wall structures are differently oriented.

Fig. 9 shows the 3rd filed example, where again, our CNN method simultaneously computes a fault probability image (Fig. 9a) with clean and sharp fault features, a smoothed seismic volume (Fig. 9b) with structural features enhanced while noise removed (Fig. 9c), and a seismic normal vector field (Fig. 9d). On the other hand side, by using the conventional methods, the computed fault likelihood image (Fig. 9e) is more noisy. The conventional structure-oriented smoothing removes noise but also removes more structural features (highlighted by the dashed yellow ellipse in Fig. 9g) compared to our CNN-based method. The conventional structure-tensor method and our CNN-based method provide consistent estimations of seismic normal vectors as illustrated by the green ellipsoids in Figs 9(h) and (d). Note that the removed noise in both the CNN-based method and the conventional structure-oriented smoothing is mostly random noise. Attenuating coherent noise apparent in seismic images is typically a more challenging task. However, the CNN-based method could be a potential way to effectively remove coherent noise by training the network with proper training data sets.

Fig. 10 shows the last example, where the input seismic amplitude image (background grey image in Fig. 10a) is relatively more noisy than the previous ones. Our CNN method still computes a clean fault probability image (Fig. 10a), a properly smoothed seismic image (Fig. 10b) with structural features enhanced while noise effectively removed (Fig. 10c), and accurate seismic normal vectors that are consistently aligned with the seismic reflections as shown in Fig. 10(d). As a comparison, again, we use the conventional methods to compute a thinned fault likelihood image (Fig. 10e), a smoothed seismic image (Fig. 10f) with noise removed (Fig. 10f), and a seismic normal vector field (Fig. 10g). We observe the fault features in the thinned fault likelihood image are much more noisy than the CNN fault probability image. The conventional structure-oriented smoothing removes some useful structural features (highlighted by yellow arrows in Fig. 10f) as well as noise.

In summary, although the CNN model is trained with only synthetic data sets, it works well to perform all the three image processing tasks of detecting faults, structure-oriented smoothing with edge-preserving and estimating seismic normal vectors in 3-D field seismic volumes that are acquired at totally different surveys. In addition, the seismic image processing using the trained CNN model is highly efficient. By using one Titan Xp GPU, processing the seismic volume (with 128 $\times$ 360 $\times$ 512 samples) in Fig. 9(a) takes around 6 s to compute the three outputs of a fault probability image (Fig. 9d), a smoothed seismic volume (Fig. 9b) and a seismic normal vector field with three components.

4 CONCLUSIONS

We have proposed a MTL CNN to simultaneously perform three seismic image processing tasks of detecting faults, structure-oriented smoothing with edge preserving, and estimating seismic local structure orientations (seismic normal vectors or reflection slopes). All the three tasks involve the analysis of seismic structural features and one depends on the others. Based on this observation, we design a CNN including a shared encoder-decoder network of extracting common features followed by three branches of prediction networks, which share the commonly extracted features and compute three outputs of a fault image, a smoothed seismic image and seismic normal vectors. All these three tasks involve only local image processing, we therefore use small image cubes (64 $\times$ 56 $\times$ 56 samples) to more efficiently train the CNN, which is further used to perform the image processing tasks on large image volumes. We train the neural network by using only synthetic data sets, which are all automatically generated by randomly adding folding, faulting and noise. Although trained with only synthetic data sets, the neural network can accurately perform all the three image processing tasks on field seismic volumes that are acquired at totally different areas. All the three tasks are based on the seismic structure patterns that are characterized by seismic amplitude variations (not the absolute amplitude values), which means that the normalization (subtracting and dividing a constant value) applied to the seismic images does not modify the structure patterns. This is an important reason why our network, trained with only synthetic data sets, could be successfully applied to different field data sets which share similar structure patterns with the training data sets and are consistently normalized as the training data sets.

ACKNOWLEDGEMENTS

This research was supported by the National Science Foundation of China under Grant 41974121. We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan Xp GPU used for this research.

REFERENCES

Al-Dossary, S. & Marfurt, K.J., 2006. 3D volumetric multispectral estimates of reflector curvature and rotation, *Geophysics,* **71**(5), P41–P51.

Argyriou, A., Evgeniou, T. & Pontil, M., 2007. Multi-task feature learning, in *NIPS'06 Proceedings of the 19th International Conference on Neural Information Processing Systems,* pp. 41–48, Canada, December 04–07, 2006, MIT Press.

Badrinarayanan, V., Kendall, A. & Cipolla, R., 2017. Segnet: A deep convolutional encoder-decoder architecture for image segmentation, *IEEE Trans. Pattern Anal. Mach. Intell.,* **39**(12), 2481–2495.

Bakker, P., 2002. Image structure analysis for seismic interpretation, *PhD thesis*, Delft University of Technology.

Brox, T., Weickert, J., Burgeth, B. & Mrázek, P., 2006. Nonlinear structure tensors, *Image Vis. Comput.,* **24**(1), 41–55.

Carruthers, T., 2012. Interaction of polygonal fault systems with salt diapirs, *PhD thesis*, Cardiff University.

Caruana, R., 1997. Multitask learning, *Mach. Learn.,* **28**(1), 41–75.

Clapp, R.G., Biondi, B.L. & Claerbout, J.F., 2004. Incorporating geologic information into reflection tomography, *Geophysics,* **69**(2), 533–546.

Di, H. & Gao, D., 2016. Efficient volumetric extraction of most positive/negative curvature and flexure for fracture characterization from 3D seismic data, *Geophys. Prospect.,* **64**(6), 1454–1468.

Di, H., Gao, D. & AlRegib, G., 2018. 3D structural-orientation vector guided autotracking for weak seismic reflections: a new tool for shale reservoir visualization and interpretation, *Interpretation,* **6**(4), SN47–SN56.

Evgeniou, T. & Pontil, M., 2004. Regularized multi–task learning, in *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining,* pp. 109–117, ACM.

Fehmers, G.C. & Höcker, C.F., 2003. Fast structural interpretation with structure-oriented filtering, *Geophysics,* **68**(4), 1286–1293.

Fomel, S., 2002. Applications of plane-wave destruction filters, *Geophysics,* **67**(6), 1946–1960.

Fomel, S., 2010. Predictive painting of 3D seismic volumes, *Geophysics,* **75**(4), A25–A30.

Gersztenkorn, A. & Marfurt, K.J., 1999. Eigenstructure-based coherence computations as an aid to 3-D structural and stratigraphic mapping, *Geophysics,* **64**(5), 1468–1479.

Girshick, R., Donahue, J., Darrell, T. & Malik, J., 2014. Rich feature hierarchies for accurate object detection and semantic segmentation, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition,* pp. 580–587.

Hale, D., 2009a. Image-guided blended neighbor interpolation of scattered data, in *79th Annual International Meeting, SEG, Expanded Abstracts,* pp. 1127–1131.

Hale, D., 2009b. Structure-oriented smoothing and semblance, *CWP Report 635*.

Hale, D., 2013. Methods to compute fault images, extract fault surfaces, and estimate fault throws from 3D seismic images, *Geophysics,* **78**(2), O33–O43.

He, K., Zhang, X., Ren, S. & Sun, J., 2016. Deep residual learning for image recognition, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition,* pp. 770–778.

He, K., Gkioxari, G., Dollár, P. & Girshick, R., 2017. Mask r-cnn, in *Computer Vision (ICCV), 2017 IEEE International Conference on,* pp. 2980–2988, IEEE.

Karimi, P., Fomel, S., Wood, L. & Dunlap, D., 2015. Predictive coherence, *Interpretation,* **3**(4), SAE1–SAE7.

Kingma, D.P. & Ba, J., 2014. Adam: a method for stochastic optimization, *CoRR,* **abs/1412.6980**.

Krizhevsky, A., Sutskever, I. & Hinton, G.E., 2012. Imagenet classification with deep convolutional neural networks, in *Advances in Neural Information Processing Systems,* pp. 1097–1105.

Li, F. & Lu, W., 2014. Coherence attribute at different spectral scales, *Interpretation,* **2**(1), SA99–SA106.

Li, S., Liu, B., Ren, Y., Chen, Y., Yang, S., Wang, Y. & Jiang, P., 2019. Deep learning inversion of seismic data, preprint (arXiv:1901.07733).

Liu, Y., Fomel, S. & Liu, G., 2010. Nonlinear structure-enhancing filtering using plane-wave prediction, *Geophys. Prospect.,* **58**(3), 415–427.

Lomask, J., Guitton, A., Fomel, S., Claerbout, J. & Valenciano, A.A., 2006. Flattening without picking, *Geophysics,* **71**(4), 13–20.

Marfurt, K.J., 2006. Robust estimates of 3D reflector dip and azimuth, *Geophysics,* **71**(4), P29–P40.

Marfurt, K.J., Kirlin, R.L., Farmer, S.L. & Bahorich, M.S., 1998. 3-D seismic attributes using a semblance-based coherency algorithm, *Geophysics,* **63**(4), 1150–1165.

Marfurt, K.J., Sudhaker, V., Gersztenkorn, A., Crawford, K.D. & Nissen, S.E., 1999. Coherency calculations in the presence of structural dip, *Geophysics,* **64**(1), 104–111.

Pedersen, S.I., Randen, T., Sonneland, L. & Steen, Ø., 2002. Automatic fault extraction using artificial ants, in *72nd Annual International Meeting, SEG, Expanded Abstracts,* pp. 512–515.

Pedersen, S.I., Skov, T., Hetlelid, A., Fayemendy, P., Randen, T. & Sønneland, L., 2003. New paradigm of fault interpretation, in *73rd Annual International Meeting, SEG, Expanded Abstracts,* pp. 350–353.

Pham, N., Fomel, S. & Dunlap, D., 2019. Automatic channel detection using deep learning, *Interpretation,* **7**(3), SE43–SE50.

Randen, T., Pedersen, S.I., Sønneland, L. *et al.*, 2001. Automatic extraction of fault surfaces from three-dimensional seismic data, in *81st Annual International Meeting, SEG, Expanded Abstracts,* pp. 551–554.

Ren, S., He, K., Girshick, R. & Sun, J., 2015. Faster r-cnn: towards real-time object detection with region proposal networks, in *Advances in Neural Information Processing Systems,* pp. 91–99.

Roberts, A., 2001. Curvature attributes and their application to 3D interpreted horizons, *First Break,* **19**(2), 85–100.

Ronneberger, O., Fischer, P. & Brox, T., 2015. U-net: Convolutional networks for biomedical image segmentation, in *International Conference on Medical Image Computing and Computer-Assisted Intervention,* pp. 234–241, Springer.

Rowan, M.G., Jackson, M.P. & Trudgill, B.D., 1999. Salt-related fault families and fault welds in the northern gulf of mexico, *AAPG Bull.,* **83**(9), 1454–1484.

Ruder, S., 2017. An overview of multi-task learning in deep neural networks, preprint (arXiv:1706.05098).

Shi, Y., Wu, X. & Fomel, S., 2019. Saltseg: automatic 3D salt segmentation using a deep convolutional neural network, *Interpretation,* **7,** SE113–SE122.

Van Bemmel, P.P. & Pepper, R.E., 2000. Seismic signal processing method and apparatus for generating a cube of variance values, US Patent 6,151,555.

Wang, E. & Nealon, J., 2019. Applying machine learning to 3D seismic image denoising and enhancement, *Interpretation,* **7**(3), SE131–SE139.

Weickert, J., 1997. A review of nonlinear diffusion filtering, in *Scale-Space Theory in Computer Vision, Vol. 1252 of* ***Lecture Notes in Computer Science****,* pp. 1–28, eds ter Haar Romeny, B., Florack, L., Koenderink, J. & Viergever, M., Springer Berlin Heidelberg.

Weickert, J., 1998. *Anisotropic Diffusion in Image Processing,* Teubner Stuttgart.

Weickert, J., 2001. Applications of nonlinear diffusion in image processing and computer vision, *Acta Math. Univ. Comenianae,* **70**(1), 33–50.

Wu, X., 2017. Structure-, stratigraphy-, and fault-guided regularization in geophysical inversion, *Geophys. J. Int.,* **210**(1), 184–195.

Wu, X. & Fomel, S., 2018a. Automatic fault interpretation with optimal surface voting, *Geophysics,* **83**(5), O67–O82.

Wu, X. & Fomel, S., 2018b. Least-squares horizons with local slopes and multigrid correlations, *Geophysics,* **83**(4), IM29–IM40.

Wu, X. & Guo, Z., 2019. Detecting faults and channels while enhancing seismic structural and stratigraphic features, *Interpretation,* **7,** T155–T166.

Wu, X. & Hale, D., 2015. 3D seismic image processing for unconformities, *Geophysics,* **80**(2), IM35–IM44.

Wu, X. & Hale, D., 2016. 3D seismic image processing for faults, *Geophysics,* **81**(2), IM1–IM11.

Wu, X. & Janson, X., 2017. Directional structure tensors in estimating seismic structural and stratigraphic orientations, *Geophys. J. Int.,* **210,** 534–548.

Wu, X., Liang, L., Shi, Y. & Fomel, S., 2019. FaultSeg3D: using synthetic datasets to train an end-to-end convolutional neural network for 3D seismic fault segmentation, *Geophysics,* **84,** IM35–IM45.

Wu, X., Shi, Y., Fomel, S., Liang, L., Zhang, Q. & Yusifov, A.Z., 2019. FaultNet3D: predicting fault probabilities, strikes, and dips with a single convolutional neural network, *IEEE Trans. Geosci. Rem. Sens.,* **99,** 1–18.

Xie, S. & Tu, Z., 2015. Holistically-nested edge detection, in *Proceedings of the IEEE International Conference on Computer Vision,* pp. 1395–1403.

Yang, F. & Ma, J., 2019. Deep-learning inversion: a next-generation seismic velocity model building method, *Geophysics,* **84**(4), R583–R599.

Yu, S., Ma, J. & Wang, W., 2018. Deep learning tutorial for denoising, preprint (arXiv:1810.11614).

Zeiler, M.D. & Fergus, R., 2014. Visualizing and understanding convolutional networks, in *European Conference on Computer Vision,* pp. 818–833, Springer.

Zhang, Y. & Yang, Q., 2017. An overview of multi-task learning, *Nat. Sci. Rev.,* **5**(1), 30–43.

Zhu, W., Mousavi, S.M. & Beroza, G.C., 2018. Seismic signal denoising and decomposition using deep neural networks, preprint (arXiv:1811.02695).