GEOPHYSICS, VOL. 86, NO. 4 (JULY-AUGUST 2021); P. IM73–IM83, 19 FIGS., 2 TABLES.
10.1190/GEO2020-0572.1

ChannelSeg3D: Channel simulation and deep learning for channel interpretation in 3D seismic images

Hang Gao[1], Xinming Wu[1], and Guofeng Liu[2]

ABSTRACT

Seismic channel interpretation involves detecting channel structures, which often appear as meandering shapes in 3D seismic images. Many conventional methods are proposed for delineating channel structures using different seismic attributes. However, these methods are often sensitive to seismic discontinuities (e.g., noise and faults) that are not related to channels. We have adopted a convolutional neural network (CNN) method to improve automatic channel interpretation. The key problem in applying the CNN method into channel interpretation is the absence of labeled field seismic images for training the CNNs. To solve this problem, we adopt a workflow to automatically generate numerous synthetic training data sets with realistic channel structures. In this workflow, we first randomly simulate various meandering channel models based on geologic numerical simulation. We further simulate structural deformation in the form of stratigraphic folding referred to as "folding structures" and combine them with the previously generated channel models to create reflectivity models and the corresponding channel labels. Convolved with a wavelet, the reflectivity models can be transformed into learnable synthetic seismic volumes. By training the designed CNN with synthetic seismic data, we obtain a CNN that learns the characterization of channel structures. Although trained on only synthetic seismic volumes, this CNN shows outstanding performance on field seismic volumes. This indicates that the synthetic seismic images created in this workflow are realistic enough to train the CNN for channel interpretation in field seismic images.

INTRODUCTION

Channel interpretation in 3D seismic images is fundamental for developing fluvial reservoirs (Bridge and Tye, 2000), hydrocarbon reservoir characterization (Payenberg and Lang, 2003), well control (Suarez et al., 2008), and ancient channel geomorphology (Renhai et al., 2009). The main task in channel interpretation is to accurately detect channel bodies in complex and noisy seismic volumes. Because channels often appear as discontinuities in a seismic image (Hale, 2009), many seismic attributes including semblance (e.g., Marfurt et al., 1998; Hale, 2009), coherence (e.g., Marfurt et al., 1999; Li et al., 2017; Wu, 2017), eigenstructure-based coherence (Gersztenkorn and Marfurt, 1999), and sweetness (Hart, 2008) are used to detect channels by highlighting seismic discontinuities. Although these conventional methods can delineate channel edges, other discontinuous features resulting from noise and stratigraphic complexities are often highlighted as well, which degrades the performance of the methods. Some attributes such as k2 curvature (Wallet and Bradley, 2016) can extract the channel bodies but might also be sensitive to other discontinuity features unrelated to channels. In addition, these methods using seismic attributes are typically restricted to detect only the channel edges rather than highlight the channel bodies.

Recently, convolutional neural networks (CNNs) have shown superior performance over conventional approaches on computer vision tasks in natural images including object detection (e.g., Girshick et al., 2014; Girshick, 2015; Ren et al., 2015; Redmon et al., 2016), semantic segmentation (e.g., Ronneberger et al., 2015; Badrinarayanan et al., 2017; Chen et al., 2017a), and instance segmentation (e.g., Dai et al., 2016; He et al., 2017; Bolya et al., 2019; Xie et al., 2019). Inspired by the application of CNNs in computer vision, many researchers have introduced CNNs into

Manuscript received by the Editor 6 August 2020; revised manuscript received 5 March 2021; published ahead of production 9 April 2021; published online 10 June 2021.
[1]University of Science and Technology of China, Laboratory of Seismology and Physics of Earth's Interior, School of Earth and Space Sciences, Hefei 230026, China. E-mail: hanggao@mail.ustc.edu.cn; xinmwu@ustc.edu.cn (corresponding author).
[2]China University of Geosciences (Beijing), School of Geophysics and Information Technology, Beijing 10083, China. E-mail: liugf@cugb.edu.cn.

DOI:10.1190/geo2020-0572.1

geophysical problems including enhancement of seismic imaging (Zhang et al., 2019), salt-body delineation (Di et al., 2018; Shi et al., 2019; Di and AlRegib, 2020), and fault detection (Zhao and Mukhopadhyay, 2018; Di et al., 2019; Wu et al., 2019).

Channel detection in 3D seismic images also can be considered as a problem of image segmentation (Pham et al., 2019) by using a CNN. However, the main limitation of applying a CNN in channel interpretation is that training data sets are inaccessible. Wu et al. (2019, 2020) suggest that synthetic training data sets with simulated realistic structures can be used to effectively train CNNs for seismic structural interpretation in field images. Therefore, we can use a similar idea to create diverse structure models with structural deformation in the form of stratigraphic folding and channel structures to automatically yield rich training data.

To construct the structure model with geologic significance for channel interpretation, we build meandering channel models based on geologic numerical simulation. The key parameter in simulating the meandering channels is the migration rate. Ikeda et al. (1981) expound the basic theory of meandering and present the assumption of the nominal migration rate. Howard and Knutson (1984) put forward the adjusted migration rate and introduce the influence of upstream channel geometry to the simulation. Sylvester et al. (2019) further simplify the calculation of the adjusted migration rate and therefore improve the efficiency of simulating meandering channel models. Based on the channel simulation method by Sylvester et al. (2019), we propose a workflow to randomly build numerous meandering channel models and embed them into initial reflectivity models with flat layers (see Figure 1). We further simulate lateral compression to generate deformed "folded" anticlinal structures in these reflectivity models with channels. Then, we convolve the reflectivity models with Ricker wavelets and add random noise to obtain synthetic seismic images. By randomly selecting the parameters in the methods of simulating channels and folding structures, we are able to create numerous and diverse synthetic seismic volumes with realistic structural and channel features. Because the channels are well-defined in the simulation, we can automatically obtain the corresponding labels (or ground truth) of the 3D channels simulated in the synthetic seismic volumes.

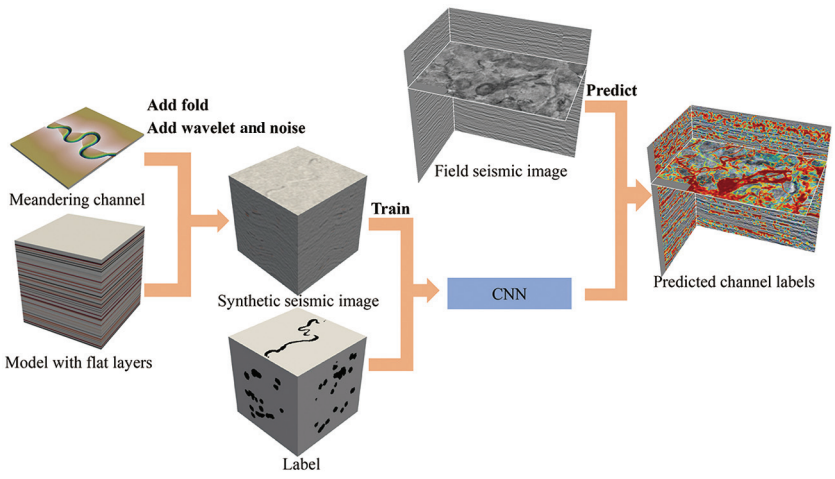


Figure 1. Flowchart of the main tasks in our workflow. We embed meandering channels into initial reflectivity models and simulate the structural deformation to generate geologic models containing channels. Then, we convolve them with a Ricker wavelet to build synthetic seismic images and the corresponding labels. Trained on these synthetic images, our CNN can extract the meandering channels in field seismic images.

With the automatically generated synthetic seismic images and the corresponding labels, we are able to train a 3D end-to-end CNN modified from the 2D DeepLabv3+ (Chen et al., 2018). This CNN, trained on only synthetic seismic images, works well to accurately highlight fluvial channel bodies in field seismic images, whereas traditional methods can detect only channel boundaries.

METHODOLOGY

We propose a supervised 3D CNN method to automatically detect channels in 3D seismic images. Training such a CNN requires numerous training data set pairs including seismic images and the corresponding labels (known control labels) of channels in the images. We therefore first introduce a numerical workflow to randomly generate numerous synthetic data sets with realistic meandering channels and folded structures.

Channel simulation

To numerically simulate the morphology of channels, we start with an initially straight channel (with some random noise), which is then iteratively migrated to obtain a meandering channel. Howard and Knutson (1984) introduce a simulation approach in which an adjusted migration rate is first calculated and then the channel migration is iteratively simulated according to the migration rate. After repeated iterations, the modeled channel evolves from the straight form to a meandering one. By randomly choosing the parameters for simulation, channels with a range of meandering rates can be created automatically in batches to build structure models for the next step of training a CNN.

Initialization

To build a channel model morphologically, we first simulate the meandering pattern of the channel without considering the channel width. We construct a rectangular coordinate system with x- and y-axes, which, respectively, are parallel and perpendicular to the direction of the initial channel. The z-axis is perpendicular to the X-Y plane downward in the depth direction of the channel. The channel can be simplified as its centerline, which is represented by a series of evenly sampled nodes or samples. The centerline is divided into three parts: upstream, midstream, and downstream. In this simulation, the upstream and downstream stay the same whereas some random perturbations are added to the midstream to initiate the migration simulation. Because of these perturbations, the channel model starts migrating from a straight channel to a meandering one.

Migration

The overall simulation process can be best understood by depicting a single step of the migration. In the channel migration, it is necessary to update the migration rate at each node of the channel centerline. According to Nanson and Hickin (1983), the local migration rate at each node can be related to the bend radius and

channel width. With this consideration, the local migration rate (also regarded as nominal migration rate) can be calculated as

$$R_0 = k_1 \frac{W}{R}, \tag{1}$$

where R_0 is the nominal migration rate, W is the channel width, R is the bend radius of each node, and k_1 is a constant migration rate. Equation 1 indicates that the nominal migration rate is proportional to the ratio of the channel width to the bend radius. However, the channel simulation does not work well when the migration rate is updated with equation 1 because not only the local curvature but also the upstream geometry should be taken into consideration while the meandering channel migrates. Howard and Knutson (1984) propose a method to improve the calculation of the migration rate by taking the upstream channel geometry into account. In this method, the adjusted migration rate can be computed as

$$R_1(s) = \Omega R_0(s) + \Gamma\left[\int_0^\infty G(\xi)d\xi\right]^{-1}\left[\int_0^\infty R_0(s-\xi)G(\xi)d\xi\right], \tag{2}$$

where ξ is the distance upstream from s, Ω and Γ are the weighting factors, and $G(\xi)$ is a weighting function defined as

$$G(\xi) = e^{-\alpha\xi}, \tag{3}$$

where α is an adjustable parameter (Sylvester et al., 2019). The term α can be calculated as $\alpha = 2k(C_f/D)$, where k is a constant scaling parameter, C_f is the Chezy friction factor, and D is the water depth.

Based on equations 1–3, the adjusted migration rate at each node along the channel centerline is iteratively updated during the migration. The new location of a migrating channel node is computed according to the updated migration rate:

$$\begin{cases} x_{i+1} = x_i + R_1\Delta t\dfrac{dy}{\sqrt{dx^2+dy^2+dz^2}}, \\ y_{i+1} = y_i - R_1\Delta t\dfrac{dx}{\sqrt{dx^2+dy^2+dz^2}}, \end{cases} \tag{4}$$

where x_i, y_i, and z_i represent the coordinates of the migrating channel node at the ith iteration; Δt is a time interval; and dx, dy, and dz are the first derivation of x, y, and z, respectively, which can be calculated with finite-difference methods in discrete data. Using equation 4, the new channel centerline at each iteration can be calculated (Figure 2). Based on this workflow, the migration of a meandering channel is iteratively updated as shown in Figure 3.

Cutoff

Although the simulated channel continuously migrates during the repeated iterations, the migration cannot continue endlessly because of neck cutoffs. As rivers naturally meander, neck cutoffs occur where the local bend intersects itself forming a new primary channel centerline. To simulate the neck cutoffs, the bend of a meandering channel should be limited by a selected cutoff distance. As shown in Figure 4, the meandering channel is straightened at the extremely bended positions that are identified by the selected cutoff distance.

Using this workflow, the geometry of the channel centerline can be simulated by repeated iterations with the model modified from the Python meanderpy (Sylvester et al., 2019). By randomly choosing the parameters in the simulation, various channel models can be generated for the next step of building structure models with rich channel features.

Building labeled models

After generating various meandering channels, we then propose a simple way to incorporate the channels into 3D layered models with various folded structures. In this method, we begin with an initial reflectivity model whose layers are all flat as shown in Figure 5. The vertical reflectivity values in this model are randomly generated in the range of [−1,1].

Based on our meandering channel simulation, various channel styles can be created by choosing the parameters such as channel width and depth. Then, we embed narrow and wide channels in the

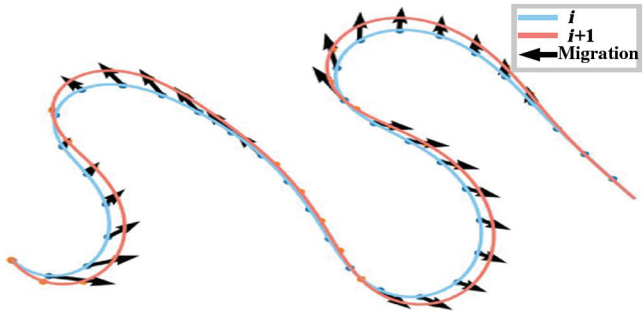


Figure 2. Based on equation 4, we simulate the channel migration from the ith (the blue curve) to the (i+1)th iteration (the red curve).

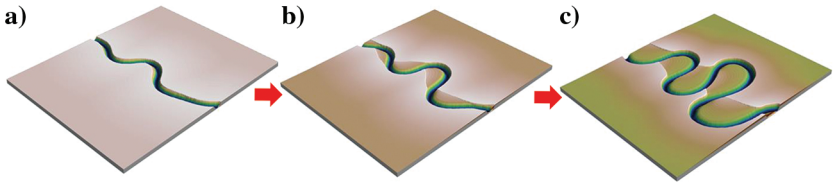


Figure 3. The migration of meandering channels. From (a) to (c), the channel gradually meanders.

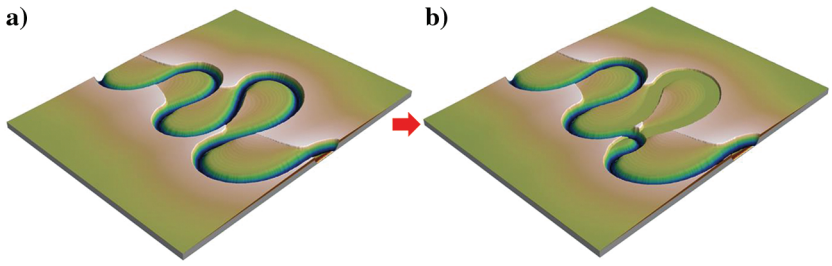


Figure 4. The cutoff of the meandering channel. The bend of the channel reaches its limit in (a). Neck cutoffs occur where the local bend intersects itself as in (b).

flat reflectivity model, where the depth and orientation of placing each channel are randomly chosen. In our reflectivity models, the reflectivity values inside the channel bodies are set to be relatively higher than the background reflectivities. We then define relatively high reflectivities within the channel bodies to obtain a reflectivity model with various channel features as shown in Figure 5. The reflectivity value within each channel body is also randomly chosen from a predefined range to increase the diversity of the simulated models. We further follow the workflow proposed by Wu et al. (2020) and simulate folding structures by shearing the layers and channels vertically. The shearing can be decomposed into shift fields $S_1(X,Y,Z)$ and $S_2(X,Y,Z)$ which, respectively, tilt and bend the layers:

$$\begin{cases} S_1(X,Y,Z) = aX + bY + c_0, \\ S_2(X,Y,Z) = \frac{1.5}{Z_{\max}} Z \sum_{k=1}^{k=N} b_k e^{\frac{(X-c_k)^2+(Y-d_k)^2}{2\sigma_k^2}}, \end{cases} \quad (5)$$

where a, b, and c_0 control the dipping structure, N is the number of 2D Gaussian functions, (c_k, d_k) means the center of each Gaussian function, and σ_k and b_k are the half-width and amplitude of the kth Gaussian function, respectively. By using these shearing shift fields, the initially flat reflectivity model (Figure 6a) in the original space (X,Y,Z) is mapped to the folding space $(X,Y,Z+S_1+S_2)$ as shown in Figure 6c. By following this method, we create a reflectivity model with diverse folding structures and deformed meandering channel features.

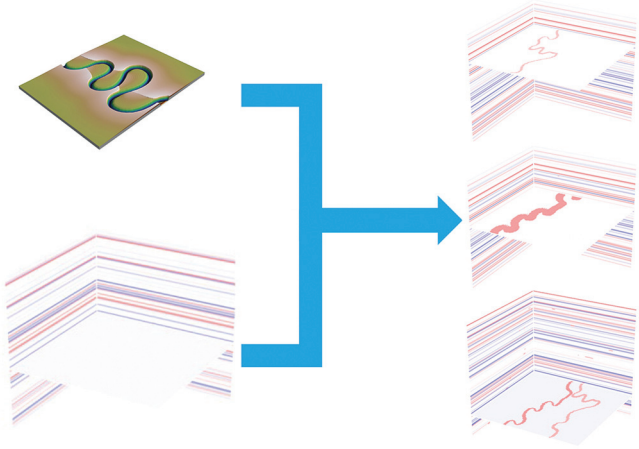


Figure 5. We generate the reflectivity models with diverse channels by combining the meandering channels with the initial models.

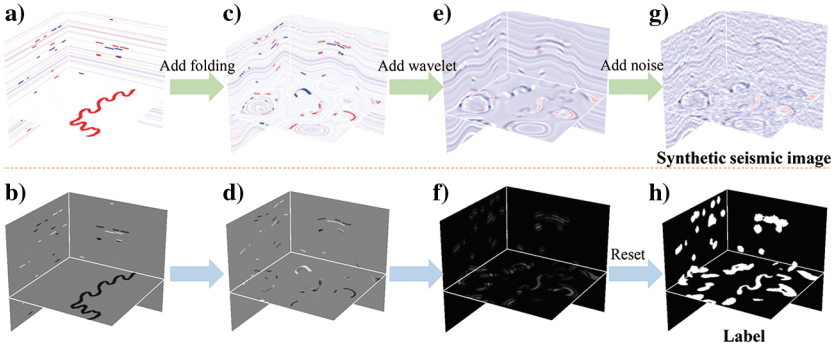


Figure 6. By adding folding structures to the initial reflectivity models of (a) flat layers and (b) channels, we obtain (c and d) the corresponding folded models, which are further convolved with a Ricker wavelet to obtain (e) synthetic seismic and (f) label images. (g) Noise is added to the synthetic seismic image to make it more realistic, and (h) the synthetic channel image is reset to a binary label image.

Building synthetic seismic images

The workflow proposed above creates the reflectivity model, which contains various meandering channels as shown in Figure 6a. Then, the structural deformation is simulated to generate folded structures hosting the channel features. (Figure 6c). To generate synthetic seismic images for training the CNN, we further convolve the reflectivity model with a Ricker wavelet (with a randomly chosen peak frequency) as shown in Figure 6e. To make the synthetic seismic images more realistic, we add some random noise into them to create random noise to add complexity to the seismic volumes (Figure 6g).

To obtain the corresponding 3D model of the channel labels, we apply exactly the same workflow to the corresponding stratigraphically flat channel model containing nonzero reflectivity values only within the channel bodies (Figure 6b). We first apply exactly the same vertical shearing as in Figure 6c to deform the channels (Figure 6d). We then apply the same wavelet as in Figure 6e to obtain a channel image (Figure 6f), which displays channel features similar to those shown in Figure 6e. However, this channel image (Figure 6f) cannot be directly used as the label image of channels because the steps of structural folding and convolving the wavelet often introduce discontinuities and gaps into the channel features. We therefore smooth the channel image (Figure 6f) to remove the potential discontinuities and fill in the gaps. We then set a threshold value to reset the smoothed channel image to obtain a binary image (Figure 6h) with ones within the channel bodies and zeros elsewhere. In this way, we are able to obtain a training data pair including a 3D seismic image (Figure 6g) and the corresponding binary label image (Figure 6h) of the channels.

By randomly altering parameters throughout the complete workflow, folded structures, and synthetic seismic images, we can generate numerous learnable synthetic seismic data sets for training the CNN. In total, we generate 200 pairs of data sets for training our CNN with Figure 7 showing four individual pairs. We observe that each data set shows unique structural and channel features.

CNN for channel interpretation

Channel interpretation in 3D seismic images can be considered as a 3D image segmentation problem. Differing from conventional segmentation in 2D natural images, channel interpretation requires us to achieve the volumetric segmentation of meandering channel bodies. Inspired by DeepLabv3+ (Chen et al., 2018), we design a 3D CNN to highlight the channel bodies in 3D seismic volumes. Trained on only synthetic seismic images and the corresponding labels, this CNN can successfully identify meandering channels in synthetic and field seismic images.

CNN architecture

Figure 8 shows our 3D CNN architecture modified from DeepLabv3+ (Chen et al., 2018). This CNN is the latest release of the DeepLab series, which has achieved excellent performance in the field of natural image semantic segmentation. For example, DeepLabv3+ attains state-of-the-art performance on Cityscapes, which consists of urban street scenes for semantic segmentation tests.

Recently, the encoder-decoder network has been widely used in the semantic segmentation field (e.g., Ronneberger et al., 2015; Badrinarayanan et al., 2017; Chen et al., 2018). Therefore, we use the encoder-decoder module as the main architecture of our CNN. The encoder module extracts the low-level and high-level features from the input images. The decoder module concatenates low-level features (local features with a small receptive field) into high-level features (global features with a large receptive field) to recover the sharp channel boundaries.

The encoder contains two modules: the deep convolutional neural network (DCNN) and the atrous spatial pyramid pooling (ASPP). In this work, we use ResNet-101 (He et al., 2016) as the DCNN module to achieve the feature extraction from the input image as shown in Figure 9. ResNet-101 mainly consists of a convolution layer, a max-pooling layer, and four ResBlocks with different parameters. The convolution layer and the max-pooling layer before the ResBlocks can reduce the size of the input images to decrease the memory requirement and computational costs. He et al. (2016) propose a two-layer building block and a three-layer building block called the "bottleneck." These four ResBlocks, respectively, contain 3, 4, 23, and 3 bottleneck blocks with different strides of 1, 2, 2, and 1. The three-layer structure contains a 3×3 and two 1×1 convolution layers as shown in Figure 10. This block can efficiently reduce the amount of calculations. Therefore, we apply the bottleneck as the main body of ResBlock to obtain a deeper neural network.

Following the DCNN module, the ASPP module (Chen et al., 2017b) is constructed to further process the output of DCNN. The ASPP module contains a pooling layer, a 1×1 convolutional layer, and three 3×3 atrous convolutional layers with a sampling rate of 6, 12, and 18, respectively (see Figure 11). The atrous convolution can increase the receptive field without losing the resolution. Based on the combination of convolutional layers with multiple sampling rates, the ASPP module can capture the context of input images at different scales and concatenate the feature maps with diverse dilation rates. After the concatenation of different feature maps, we apply a convolution layer to reduce the number of feature maps from 1280 to 256.

The decoder (below the horizontal dashed line in Figure 8) receives two inputs. The first is the low-level features extracted from the first ResBlock in DCNN, which provides the detailed information. The second contains the high-level features generated by ASPP, which provides the semantic information. The decoder reduces the number of the low-level features and upsam-

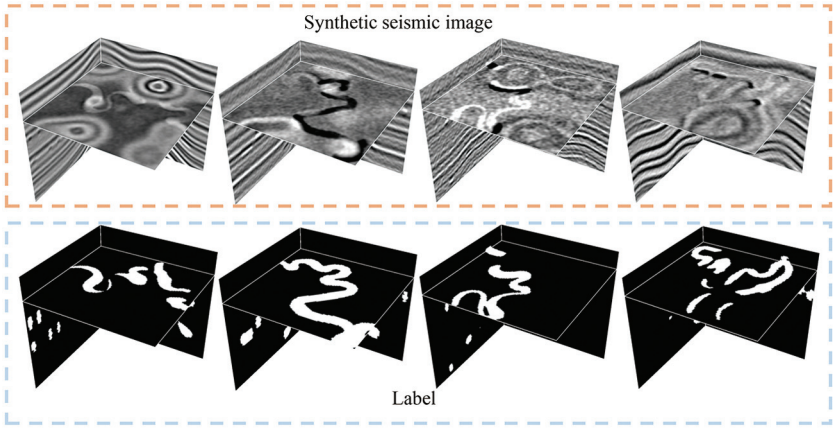


Figure 7. Automatically generated seismic images and the corresponding channel labels.

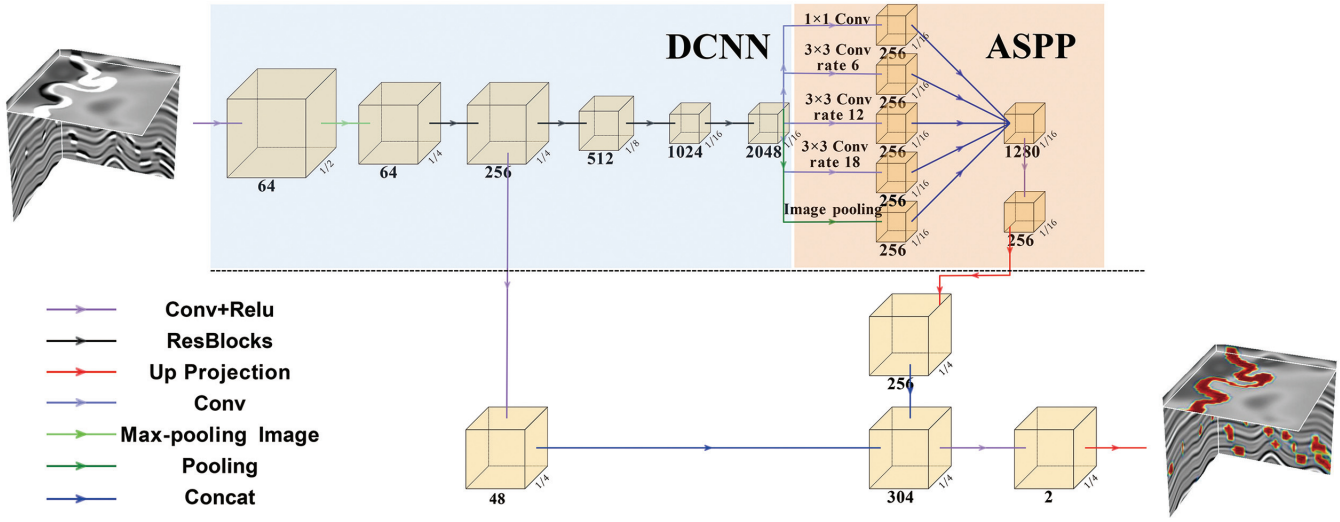


Figure 8. The architecture of our 3D CNN for volumetric channel segmentation in a 3D seismic image.

ples the high-level features by four times. After the upsampling, the detailed low-level features and the semantic high-level features are concatenated and then fed to a regular convolutional layer, which reduces the number of feature maps from 304 to 2. Then, an upsampling layer is applied to increase the size of the features by four times to keep the size of the output consistent with the input. Finally, the decoder outputs a one-hot encoding which represents the probability distribution of the channel structures and background.

To better evaluate the model performance, we compare our model with a 3D U-Net model in Table 1. We also test the performance of different normalizations using the same model. We prepare three methods trained on the same training data sets to predict on the same test sets. The results in Table 1 show that, although using the same normalization method, the performance of DeepLabv3+ is better than a 3D U-Net. The "Mean-Std normalization" is more appropriate for this problem than the min-max normalization.

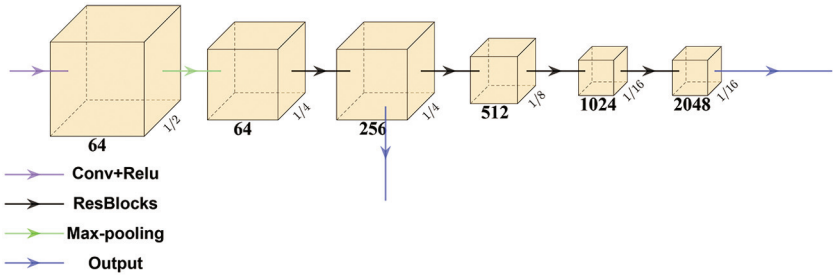


Figure 9. The DCNN module implemented with ResNet-101.

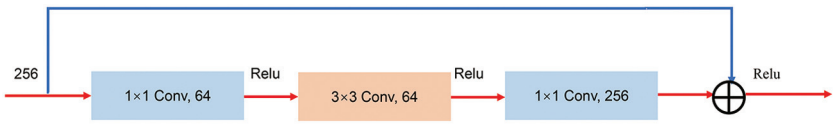


Figure 10. The three-layer bottleneck layer that contains one 3 × 3 and two 1 × 1 convolution layers.

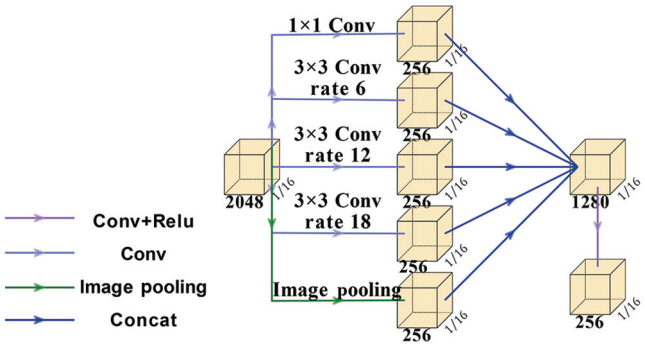


Figure 11. The ASPP module, which can capture the multiscale features of the channels.

Table 1. Comparison of different methods on test data sets.

Method	Normalization	Precision (%)	Recall (%)	mIoU (%)	F1 score (%)
DeepLabv3+	Mean-Std	**95.68**	**96.41**	**92.41**	**96.04**
DeepLabv3+	Min-Max	95.00	95.38	91.19	95.36
3D-U-Net	Mean-Std	95.53	96.20	92.08	95.86

Note: DeepLabv3+ with Mean-Std achieve better performance on Precision, Recall, mIOU, and F1 score compared with the other methods.

The training process of CNN

Based on synthetic seismic images and the corresponding labels, we train the 3D CNN shown in Figure 8. The input sizes of training data sets and labels for the 3D CNN are $128 \times 128 \times 128$. Before feeding the training data into the CNN, we normalize each seismic image separately using the Mean-Std normalization as follows:

$$\mathbf{x}' = \frac{\mathbf{x} - \mu}{\sigma}, \tag{6}$$

where μ and σ, respectively, are the mean and standard deviation of input data $\mathbf{x}$. This normalization is more robust than the min-max normalization, which is sensitive to the maximum and minimum of the sample.

Considering that the channel detection in a 3D seismic image is a binary segmentation problem, we use the following widely used binary cross-entropy loss function L to train our CNN:

$$L = -\frac{1}{N}\sum_{i=1}^{N}(y_i \cdot \log(p(y_i)) + (1 - y_i) \cdot \log(1 - p(y_i))), \tag{7}$$

where y_i is the label of the ith voxel, $p(y_i)$ is the predicted probability of the ith voxel, and N is the number of samples.

Due to the extreme imbalance between the categories (the pixels of the background are far more than those of the channel bodies in a seismic image), we use a category-balanced cross-entropy loss in our methods to improve the training processing. Before the calculation of loss, we estimate the number of pixels of channel and background to update the weight parameters. Then, we apply the weight parameters to balance the loss of categories:

$$\begin{cases} L = -\frac{1}{N}\sum_{i=1}^{N}(w_i \cdot y_i \cdot \log(p(y_i)) + (1 - y_i) \cdot \log(1 - p(y_i))), \\ w_i = \frac{N_{\text{background}}}{N_{\text{channel}}}, \end{cases} \tag{8}$$

where N_{channel} is the number of pixels corresponding to the channel label and $N_{\text{background}}$ is the number of pixels of the background. The weight parameters w_i can adjust the loss of the channel category to balance the weights between categories.

In this work, we apply the Adam optimizer (Kingma and Ba, 2014) in the training process to minimize the training loss and optimize the network parameters. The learning rate is initial-

ized to 1×10^{-4}. To better optimize the loss in the late training process, we set a learning rate scheme that reduces the learning rate to one tenth of the original when half of the epochs have been computed.

In total, we use 200 pairs of synthetic training data sets (such as those shown in Figure 7) to train our CNN. To further increase the diversity of the training data sets and improve the generalization of the CNN model, we augment the training data sets by rotating the training data 0°, 90°, 180°, and 270°. around the vertical z-axis (Figure 12). The loss curves on training (the blue curve) and validation (the orange curve) data sets are shown in Figure 13.

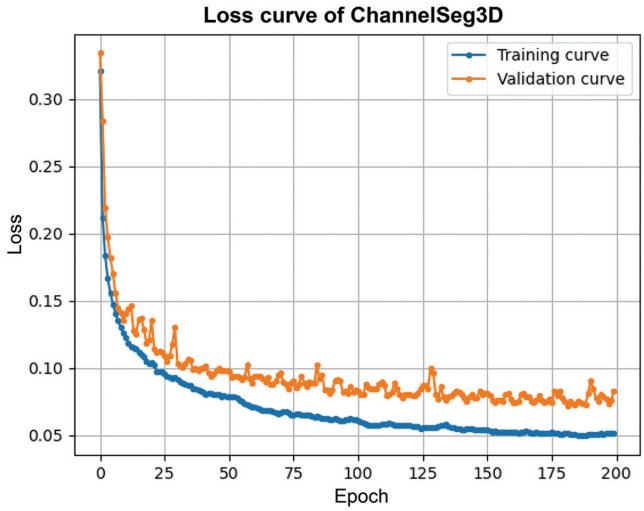


Figure 13. Loss curve of ChannelSeg3D using Mean-Std normalization on the training data sets.

APPLICATION

Evaluations against synthetic seismic data

The top row of Figure 14 shows four automatically generated synthetic seismic volumes that are not included in the training data sets. The middle row of Figure 14 shows the channel detections by using our trained CNN. We observe that the detected channels match well with the ground truth of the channels in the label images (the bottom row). This suggests that our CNN modified from 2D CNN is qualified to achieve the volumetric channel segmentation in 3D seismic volumes.

Evaluations against field seismic data

Considering the larger data size of field seismic images and the memory limitation, we divide the field data into smaller blocks and interpret them separately. To avoid the stitching effects (the effect on restoring small blocks to raw data) and the boundary effect (the effect on prediction on the boundary), we set an overlap (as the transition region) between neighboring blocks and average the predictions with the overlap.

To further test the performance of this workflow, this 3D CNN is applied to a 3D seismic image from the Atchafalaya Bay as shown in Figures 15 and 16 (Triezenberg et al., 2016). We choose two vertical slices with different depths to show the results in Figure 16. The meandering channel bodies in this field seismic data

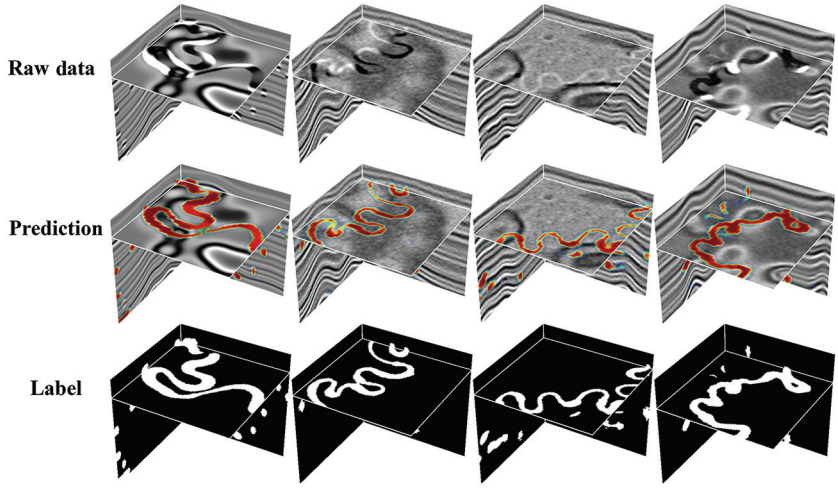


Figure 14. Our CNN works well in the synthetic seismic images.

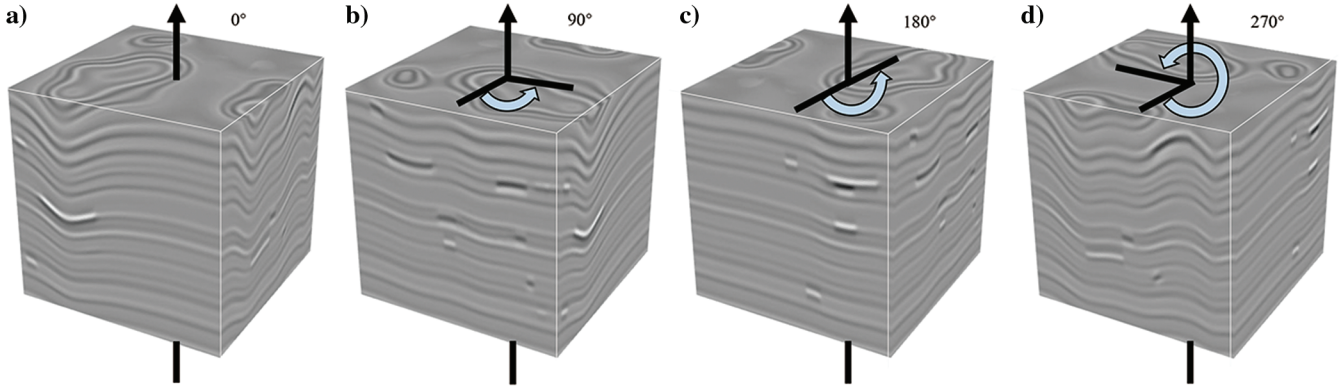


Figure 12. The process of data augmentation. (a–d) Samples rotated by 0°, 90°, 180°, and 270° around the vertical z-axis, respectively.

DOI:10.1190/geo2020-0572.1

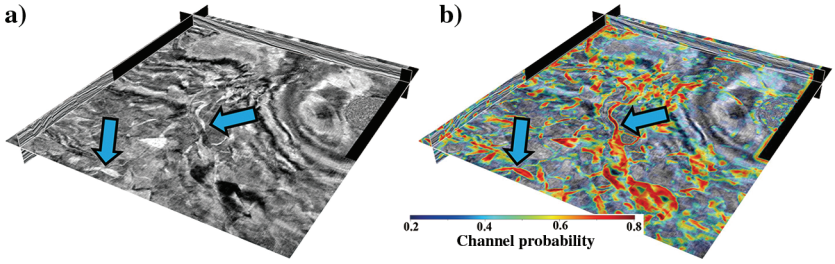


Figure 15. An overview of the work area in the Atchafalaya Bay. (a) A horizontal section in field seismic data from the Atchafalaya Bay. (b) The predictions using our CNN. As the blue arrows indicate, the channels are extracted by our CNN.

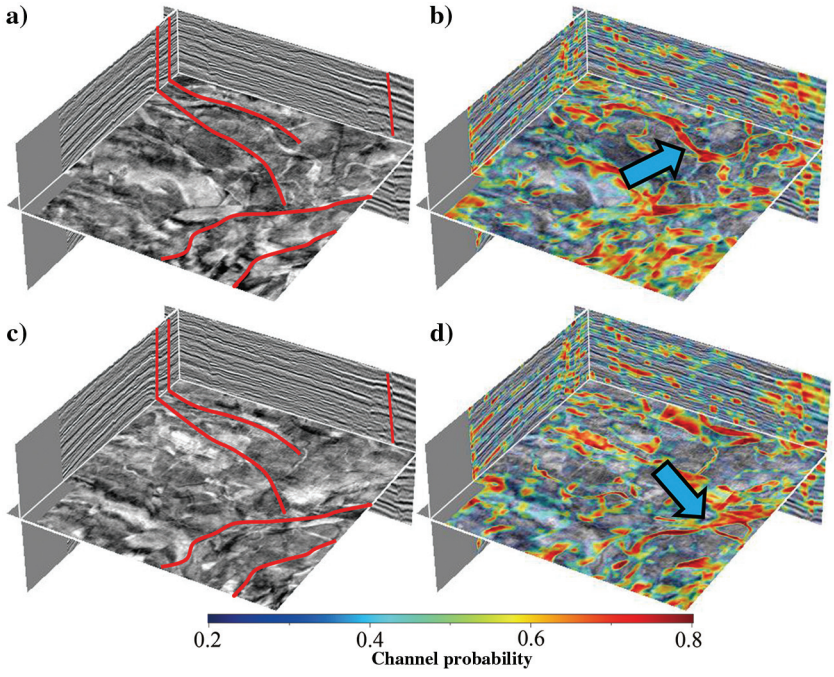
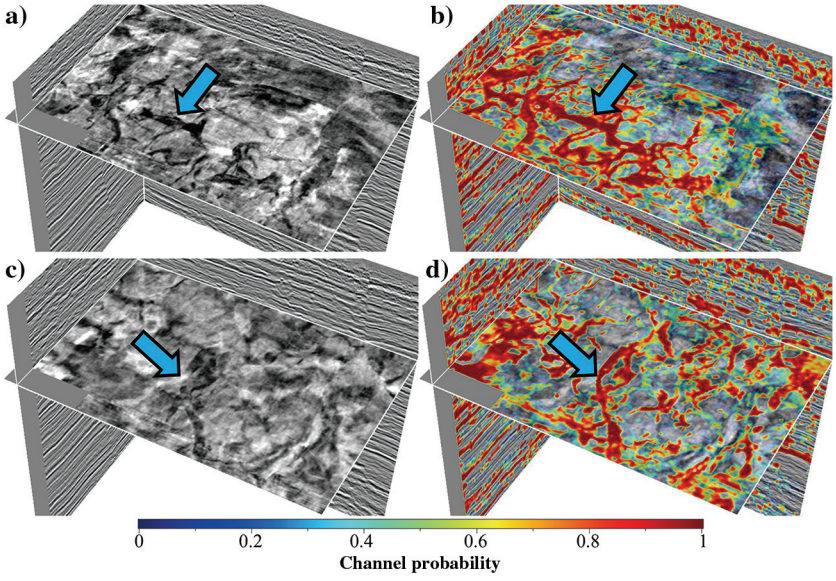


Figure 16. Field seismic images from (a and c) offshore Louisiana and (b and d) their corresponding predictions. Faults in these seismic images are noted by the red curves.

Figure 17. Two horizontal sections of field seismic images from (a and c) offshore Louisiana and (b and d) their corresponding predictions using our CNN. The blue arrows indicate the channels delineated by CNN.



are offset by multiple faults (as noted by the red curves in Figure 16) that are characterized by seismic discontinuities in the vertical and horizontal slices. Faults obscure and degrade the morphological characteristics of the buried meandering channels and therefore pose challenges to the channel delineation in the seismic image. However, our 3D CNN can still infer the branch-like meandering channels from the complex contextual information. Figure 16 shows a magnified view of the same example with different slices where the channels are also clearly highlighted, including those subtle channels with small widths and vertical thicknesses. This example shows that this CNN can effectively identify the channel bodies that are complicated by faults.

In a second example, we apply our 3D CNN to the field seismic image from offshore Louisiana as shown in Figure 17. The first row of Figure 17 shows that our CNN can sharply delineate the complex channel structure that is only partially visible on the seismic slice. The second row of Figure 17 shows the detection of meandering channels which are difficult to interpret directly on the seismic image.

Although trained on synthetic data sets, our 3D CNN can be generalized well to detect channels in field seismic images. This indicates that the structural and channel features simulated in the synthetic data sets are close to the actual geologic features apparent in the field seismic images. In addition, our CNN-based method is robust for highlighting channels that are complicated by faults and noise as shown in these examples. We further demonstrate this by comparing the CNN-based method with a conventional coherence method.

DOI:10.1190/geo2020-0572.1

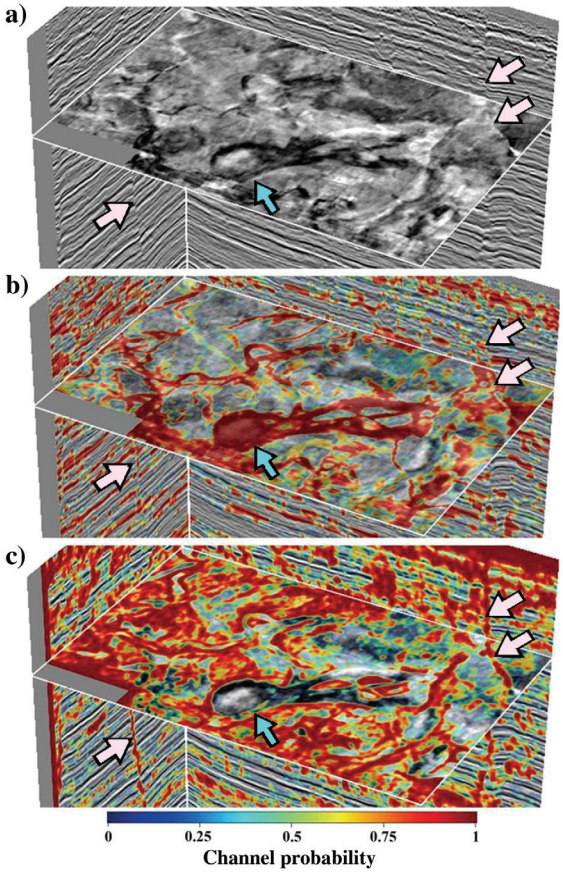


Figure 18. In a 3D field example from (a) offshore Louisiana, (b) our CNN method can more accurately detect narrow and wide channels and (c) is more robust to noise and faults than the conventional coherence. Faults and wide channels are marked with pink and cyan arrows, respectively.

The contrast between CNN and the conventional method

Figure 18b shows the channel predictions in the field seismic image (Figure 18a) with the CNN trained by only synthetic data sets. Compared to channel detection by coherence (Figure 18c), our CNN method can delineate the channel boundaries more sharply and highlight the channel bodies more accurately in the seismic image. In addition, our CNN can extract diverse channels with various sizes and shapes. However, the coherence-based method failed to detect the wide channels as denoted by the cyan arrows. Moreover, our CNN is more robust to noise and the miscalculation of other structures while the conventional coherence mistakes faults as channels as denoted by the pink arrows in Figure 18c.

We extract the target segment of this channel body for better observation in Figure 19. Figure 19a and 19b shows the position of this channel denoted by the cyan arrow in Figure 18. Figure 19c–19f shows the vertical section of this channel body. This figure shows that our CNN can highlight the wide channel bodies with a finite thickness.

DISCUSSION

Two field examples indicate that channel forms simulated in our proposed workflow are geologically reasonable to enable a CNN to achieve accurate channel interpretation. Although trained on the synthetic seismic images, the CNN can attain an outstanding performance in the field seismic images. However, further study shows that this method does not work well on some field seismic images that contain large-scale incised channel structures. We suppose that this problem is associated with training data sets rather than being a limitation of the CNN. This work mainly focuses on the simulation of the meandering channel models whose structural features are significantly different from the incised channel structures. Therefore, this CNN cannot sufficiently delineate the geologic characteristics of incised channels from meandering aggrading channels. This inspires us to further improve our simulated workflow and add the incised channel models in our synthetic training data sets.

Our method works well to identify channel bodies from 3D seismic volumes but still cannot distinguish the channel migration at different stages. Multistage channel interpretation is a highly challenging task even for manual interpretation. The CNN is a potentially effective way to solve this challenging problem by considering it as an instance segmentation problem. We can use the same simulation workflow to generate synthetic training data sets. Different from using a binary label image in this paper, we need to define a multiclass label image to indicate channels at different stages.

In our workflow, we have simulated only simple and small-scale channels. We currently approximate the channel structures and fill the channel body in the same sedimentary cycle with the same impedance. To simulate larger and more complicated channel features, we need to set more complicated impedance/reflectivity patterns within the channel bodies (e.g., spatially varying).

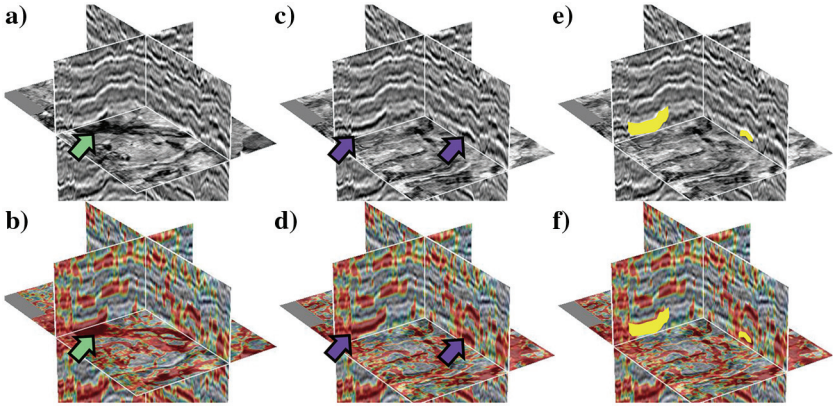


Figure 19. Field seismic images from (a, c, and e) offshore Louisiana and (b, d, and f) the corresponding predictions using our CNN. The horizontal and vertical slices of the channel structures are denoted by (a and b) green arrows and (c and d) purple arrows. The channel body in the vertical section is highlighted in yellow in (e and f).

Due to the complexity and diversity of field data, there is still a gap between the synthetic and field seismic data sets. This indicates another aspect in which our method can be improved. Improvements on synthetic seismic images can make the predictions more stable and robust. In future investigations, it might be possible to apply transfer learning in channel interpretation to reduce the difference between the training and field seismic data.

Considering the memory limitation in training a 3D CNN with larger 3D data sets, we can replace the deeper ResNet-101 with ResNet-50 in the DCNN module which, however, may degrade the performance of the network. This is a trade-off between the graphic processing unit memory and performance.

CONCLUSION

We have presented a workflow to automate the channel interpretation in 3D seismic volumes by using a CNN, in which the channel interpretation is considered as a volumetric segmentation problem. The designed CNN is modified from DeepLabv3+, which was used for semantic segmentation in the 2D images. We use ResNet-101 as the backbone network to extract the features. To adapt the network to this problem, we replace the 2D layers with 3D ones and apply the trilinear interpolation in the upsampling operation.

The main limit in applying the CNN for the channel interpretation is the absence of training data sets. To solve this problem, we have proposed a numerical workflow to simulate realistic meandering channels and integrate them with simulated structural deformation to impart folded structures into channel-hosted models. By randomly adjusting the parameters in the simulation, we can automatically create numerous structure models with various channels and folded structures. From the structural models, we further automatically generate synthetic seismic volumes and the corresponding channel label volumes to train our CNN for channel interpretation.

Although trained on only synthetic seismic volumes, the CNN works well to detect the meandering channels and extracts channel bodies from field seismic images. This suggests that the meandering channels simulated in numerical simulations are realistic enough for CNN to learn the attributes of channel forms and achieve realistic channel interpretation.

ACKNOWLEDGMENTS

This research was supported by the National Science Foundation of China under grant no. 41974121. We would like to thank Z. Hu for his constructive reviewing and suggestions about the CNN architecture. Field seismic data for Atchafalaya Bay are available on the USGS website.

DATA AND MATERIALS AVAILABILITY

Data associated with this research are available and can be obtained by contacting the corresponding author. Our implementation of the meandering channel simulation is based on Z. Sylvester's open-source codes on GitHub (https://github.com/zsylvester/meanderpy).

REFERENCES

Badrinarayanan, V., A. Kendall, and R. Cipolla, 2017, Segnet: A deep convolutional encoder-decoder architecture for image segmentation: IEEE Transactions on Pattern Analysis and Machine Intelligence, **39**, 2481–2495, doi: 10.1109/TPAMI.2016.2644615.

Bolya, D., C. Zhou, F. Xiao, and Y. J. Lee, 2019, Yolact: Real-time instance segmentation: Proceedings of the IEEE International Conference on Computer Vision, 9157–9166.

Bridge, J. S., and R. S. Tye, 2000, Interpreting the dimensions of ancient fluvial channel bars, channels, and channel belts from wireline-logs and cores: AAPG Bulletin, **84**, 1205–1228.

Chen, L.-C., G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, 2017a, DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFS: IEEE Transactions on Pattern Analysis and Machine Intelligence, **40**, 834–848, doi: 10.1109/TPAMI.2017.2699184.

Chen, L.-C., G. Papandreou, F. Schroff, and H. Adam, 2017b, Rethinking atrous convolution for semantic image segmentation: arXiv preprint arXiv:1706.05587.

Chen, L.-C., Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, 2018, Encoder-decoder with atrous separable convolution for semantic image segmentation: Proceedings of the European Conference on Computer Vision, 801–818.

Dai, J., K. He, Y. Li, S. Ren, and J. Sun, 2016, Instance-sensitive fully convolutional networks: Proceedings of the European Conference on Computer Vision, 534–549.

Di, H., and G. AlRegib, 2020, A comparison of seismic saltbody interpretation via neural networks at sample and pattern levels: Geophysical Prospecting, **68**, 521–535, doi: 10.1111/1365-2478.12865.

Di, H., M. A. Shafiq, Z. Wang, and G. AlRegib, 2019, Improving seismic fault detection by super-attribute-based classification: Interpretation, **7**, no. 3, SE251–SE267, doi: 10.1190/INT-2018-0188.1.

Di, H., Z. Wang, and G. AlRegib, 2018, Deep convolutional neural networks for seismic salt-body delineation: Presented at the Annual Convention and Exhibition, AAPG.

Gersztenkorn, A., and K. J. Marfurt, 1999, Eigenstructure-based coherence computations as an aid to 3-D structural and stratigraphic mapping: Geophysics, **64**, 1468–1479, doi: 10.1190/1.1444651.

Girshick, R., 2015, Fast R-CNN: Proceedings of the IEEE International Conference on Computer Vision, 1440–1448.

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

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

Hart, B. S., 2008, Channel detection in 3-D seismic data using sweetness: AAPG Bulletin, **92**, 733–742, doi: 10.1306/02050807127.

He, K., G. Gkioxari, P. Dollár, and R. Girshick, 2017, Mask R-CNN: Proceedings of the IEEE International Conference on Computer Vision, 2961–2969.

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

Howard, A. D., and T. R. Knutson, 1984, Sufficient conditions for river meandering: A simulation approach: Water Resources Research, **20**, 1659–1667, doi: 10.1029/WR020i011p01659.

Ikeda, S., G. Parker, and K. Sawai, 1981, Bend theory of river meanders — Part 1: Linear development: Journal of Fluid Mechanics, **112**, 363–377, doi: 10.1017/S0022112081000451.

Kingma, D. P., and J. Ba, 2014, Adam: A method for stochastic optimization: arXiv preprint arXiv:1412.6980.

Li, F., Q. Jie, B. Lyu, and K. J. Marfurt, 2017, Multi-spectral coherence: Interpretation, **6**, no. 1, T61–T69, doi: 10.1190/INT-2017-0112.1.

Marfurt, K., R. Kirlin, S. Farmer, and M. Bahorich, 1998, 3-D seismic attributes using a running window semblance algorithm: Geophysics, **63**, 1150–1165, doi: 10.1190/1.1444415.

Marfurt, K. J., V. Sudhaker, A. Gersztenkorn, K. D. Crawford, and S. E. Nissen, 1999, Coherency calculations in the presence of structural dip: Geophysics, **64**, 104–111, doi: 10.1190/1.1444508.

Nanson, G. C., and E. J. Hickin, 1983, Channel migration and incision on the Beatton River: Journal of Hydraulic Engineering, **109**, 327–337, doi: 10.1061/(ASCE)0733-9429(1983)109:3(327).

Payenberg, T. H. D., and S. C. Lang, 2003, Reservoir geometry of fluvial distributary channels — Implications for Northwest Shelf, Australia, deltaic successions: The APPEA Journal, **43**, 325–338, doi: 10.1071/AJ02017.

Pham, N., S. Fomel, and D. Dunlap, 2019, Automatic channel detection using deep learning: Interpretation, **7**, no. 3, SE43–SE50, doi: 10.1190/INT-2018-0202.1.

Redmon, J., S. Divvala, R. Girshick, and A. Farhadi, 2016, You only look once: Unified, real-time object detection: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 779–788.

Ren, S., K. He, R. Girshick, and J. Sun, 2015, Faster R-CNN: Towards real-time object detection with region proposal networks: Advances in Neural Information Processing Systems, 91–99.

Renhai, P., Z. Li, and Z. Hongli, 2009, 3-D seismic identification and characterization of ancient channel morphology: Journal of Earth Science, **20**, 858–867, doi: 10.1007/s12583-009-0072-6.

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

Shi, Y., X. Wu, and S. Fomel, 2019, Saltseg: Automatic 3D salt segmentation using a deep convolutional neural network: Interpretation, **7**, no. 3, SE113–SE122, doi: 10.1190/INT-2018-0235.1.

Suarez, Y., K. J. Marfurt, and M. Falk, 2008, Seismic attribute-assisted interpretation of channel geometries and infill lithology: A case study of Anadarko Basin Red Fork channels: 78th Annual International Meeting, SEG, Expanded Abstracts, 963–967, doi: 10.1190/1.3063798.

Sylvester, Z., P. Durkin, and J. A. Covault, 2019, High curvatures drive river meandering: Geology, **47**, 263–266, doi: 10.1130/G45608.1.

Triezenberg, P., P. Hart, and J. Childs, 2016, National archive of marine seismic surveys (NAMSS): A USGS data website of marine seismic-reflection data within the US exclusive economic zone (EEZ): United States Geological Survey Data Release.

Wallet, B., and C. Bradley, 2016, Attribute expression of channel forms in a hybrid carbonate turbidite formation: Interpretation, **4**, no. 2, SE75–SE86, doi: 10.1190/INT-2015-0108.1.

Wu, X., 2017, Directional structure-tensor-based coherence to detect seismic faults and channels: Geophysics, **82**, no. 2, A13–A17, doi: 10.1190/geo2016-0473.1.

Wu, X., Z. Geng, Y. Shi, N. Pham, S. Fomel, and G. Caumon, 2020, Building realistic structure models to train convolutional neural networks for seismic structural interpretation: Geophysics, **85**, no. 4, WA27–WA39, doi: 10.1190/geo2019-0375.1.

Wu, X., L. Liang, Y. Shi, and S. Fomel, 2019, FaultSeg3D: Using synthetic data sets to train an end-to-end convolutional neural network for 3D seismic fault segmentation: Geophysics, **84**, no. 3, IM35–IM45, doi: 10.1190/geo2018-0646.1.

Xie, E., P. Sun, X. Song, W. Wang, X. Liu, D. Liang, C. Shen, and P. Luo, 2019, Polarmask: Single shot instance segmentation with polar representation: arXiv preprint arXiv:1909.13226.

Zhang, Y., P. Lu, H. Yu, and S. Morris, 2019, Enhancement of seismic imaging: An innovative deep learning approach: arXiv preprint arXiv:1909.06016.

Zhao, T., and P. Mukhopadhyay, 2018, A fault-detection workflow using deep learning and image processing: 88th Annual International Meeting, SEG, Expanded Abstracts, 1966–1970, doi: 10.1190/segam2018-2997005.1.

Biographies and photographs of the authors are not available.