# SaltSeg: Automatic 3D salt segmentation using a deep convolutional neural network

Yunzhi Shi[1], Xinming Wu[1], and Sergey Fomel[1]


## Abstract

Salt boundary interpretation is important for the understanding of salt tectonics and velocity model building for seismic migration. Conventional methods consist of computing salt attributes and extracting salt boundaries. We have formulated the problem as 3D image segmentation and evaluated an efficient approach based on deep convolutional neural networks (CNNs) with an encoder-decoder architecture. To train the model, we design a data generator that extracts randomly positioned subvolumes from large-scale 3D training data set followed by data augmentation, then feed a large number of subvolumes into the network while using salt/nonsalt binary labels generated by thresholding the velocity model as ground truth labels. We test the model on validation data sets and compare the blind test predictions with the ground truth. Our results indicate that our method is capable of automatically capturing subtle salt features from the 3D seismic image with less or no need for manual input. We further test the model on a field example to indicate the generalization of this deep CNN method across different data sets.


## Introduction

In seismic interpretation and subsurface modeling, extracting geologic structures such as faults, unconformities, horizons, and salt bodies from 3D seismic data are critical. Conventional methods derive seismic attributes based on geologic, physical, and geometric principles. Interpreting salt from seismic often involves visual features including steeply dipping events and chaotic signals. Specifically, seismic attributes used for automatically interpreting salt boundaries include discontinuities (Asjad and Mohamed, 2015), textures (Wang et al., 2015), reflection dip or normal vector fields (Haukås et al., 2013), and salt likelihoods (Wu, 2016). Although automatic methods have been proposed for computing salt attributes and extracting salt boundaries from those attributes (Ramirez et al., 2016; Wu et al., 2018a), it remains a manual-intensive and time-consuming task in practice.

These attributes are designed with domain expertise knowledge and engineering; however, these attributes may not yet fully describe the complex noise-contaminated seismic data in real world (Marfurt and Alves, 2015). The recently developed machine-learning techniques enable computers to perform repetitive tasks, and unravel the relationships that contain useful patterns (Zhao, 2017). Ross and Cole (2017) review popular facies classification methods based on machine-learning algorithms. Deep neural networks (DNNs) are built on the premise that they can replicate a wide variety of nonlinear operator (universal approximation theorem, Csáji, 2001). Compared with traditional machine-learning algorithms, DNNs have the advantage that they extract useful features automatically via numerous hidden layers.

Convolutional neural networks (CNNs) are a specialization of DNNs by replacing matrix multiplications with convolution operators, to focus on learning the locality and spatial relationship between input image and output label. Huang et al. (2017) show that CNNs provide improved results over traditional methods such as support vector machines and logistic regression for identifying geologic faults in 3D seismic data. Araya-Polo et al. (2017) use prestack seismic data to identify faults directly without migrating the data to migrated images. Waldeland and Solberg (2017) train a CNN to perform pixel-by-pixel salt body classification. These experiments show the encouraging accuracy of CNNs in a variety of seismic processing and interpretation tasks.

To use the power of CNN in automatic salt interpretation, there have been patch-based studies that classify the seismic image as salt/nonsalt in a patch-by-patch fashion, and we assign the classification prediction to the central voxel of that patch. Di et al. (2018a) and Waldeland and Solberg (2017) propose a CNN architecture with fully connected layers attached after convolutional

[1]The University of Texas at Austin, Bureau of Economic Geology, John A. and Katherine G. Jackson School of Geosciences, University Station, Box X, Austin, Texas 78713-8924, USA. E-mail: yzshi08@utexas.edu (corresponding author); xinming.wu@beg.utexas.edu; sergey.fomel@beg.utexas.edu.

Manuscript received by the Editor 11 December 2018; revised manuscript received 15 February 2019; published ahead of production 7 April 2019; published online 28 May 2019. This paper appears in *Interpretation*, Vol. 7, No. 3 (August 2019); p. SE113–SE122, 10 FIGS., 1 TABLE.
http://dx.doi.org/10.1190/INT-2018-0235.1. 

layers to predict the classification using a softmax activation layer at the end. Wu et al. (2018b) also use a similar CNN-based pixel-wise classification method to predict fault existence and position in each image patch. However, patch-based methods are born with disadvantages in geobody interpretations because they are originally designed for object classification problems. Figure 1a demonstrates how patch-based detection methods work: For each pixel, the network will take a window, centered at the point of interest, as its input and classify the category to which this pixel belongs. The process will repeat by sliding the window across the image until all of the pixels are scanned. The disadvantage of these methods is twofold: First, a local window or cube is required to slide through the full data set to make a prediction at every pixel or voxel; second, it could be challenging for patch-based classification to delineate the boundary of geobodies with high resolution, for example, Figure 1a shows two window inputs with similar content but should be classified to different categories.

On the other hand, separating salt body from conformable seismic reflections is naturally an image segmentation task. Figure 1b shows an output example of the segmentation method; in the example, all pixels in the window input are classified to its category simultaneously and output together as a mask. Previous researchers (Lomask et al., 2007; Ramirez et al., 2016) discuss salt boundary extraction as a global image segmentation problem. Considering geobody interpretation as image segmentation addresses those disadvantages of patch-based methods. In computer vision area, image segmentation using deep-learning techniques is a topic being actively researched with promising progresses (Girshick, 2015; Ronneberger et al., 2015; Xie and Tu, 2015; Badrinarayanan et al., 2017; He et al., 2017; Ren et al., 2017). Zhao (2018) and Di et al. (2018b) present encouraging results using 2D encoder-decoder networks to separate different seismic facies including salt domes, low coherence, low amplitude dipping, high amplitude deformed, and compare with the patch-based method. Wang et al. (2018) show by 2D synthetic examples that it is possible to perform salt detection, even from prestack seismic data, via segmentation network. Wu et al. (2019) show that segmentation network can be highly effective and efficient for 3D seismic fault interpretations.

Therefore, in this paper, we propose to apply a deep CNN-based segmentation model to tackle 3D seismic salt interpretation automatically. We adopt the network architecture from U-net (Ronneberger et al., 2015) to build a 3D encoder-decoder network with skip connections. The network takes a seismic subvolume with certain size (receptive field) as input, and it outputs a salt probability subvolume with the same size. To train and validate the model, we use SEG Advanced Modeling (SEAM) Phase I synthetic data migrated image (Fehler and Keliher, 2011) as the input image and we extract a binary salt mask from the corresponding velocity model by thresholding and clipping. We split the data volume to a training part and a validation part; the training part is used to optimize the network, and the validation part can be used to test the generalization of the trained model via a blind test. During the training process, a data generator randomly crop and rotate a subvolume according to the size of network's receptive field. After a sufficient amount of training, we use the network to find the salt probability of all parts of the data, and we compare with the ground truth salt mask via several quantitative metrics. Furthermore, the model is applied to a field seismic data set and outputs a decent salt model prediction.

## Network architecture

The first semantic segmentation method using an encoder-decoder architecture is a fully convolutional network (Long et al., 2015). The encoder-decoder network consists of the stacking of multiple convolutional layers like the other CNNs; however, the difference is that instead of using a fully connected layer at the output to connect with categorical data, the encoder-decoder uses a convolutional layer to retain all spatial information and connect to multidimensional data. This allows for image-to-image segmentation rather than image-to-class classification in the case of ordinary CNN. Another important feature of encoder-decoder is the "bottleneck" architecture: The input data are gradually downsampled after passing through the encoder layers, and then they are upsampled layer-by-layer in the decoder section, as shown in Figure 2. The downsampling is achieved by selecting fewer pixels from the image feature according to a certain algorithm, e.g., max pooling or average pooling (Boureau et al., 2010), so that less significant

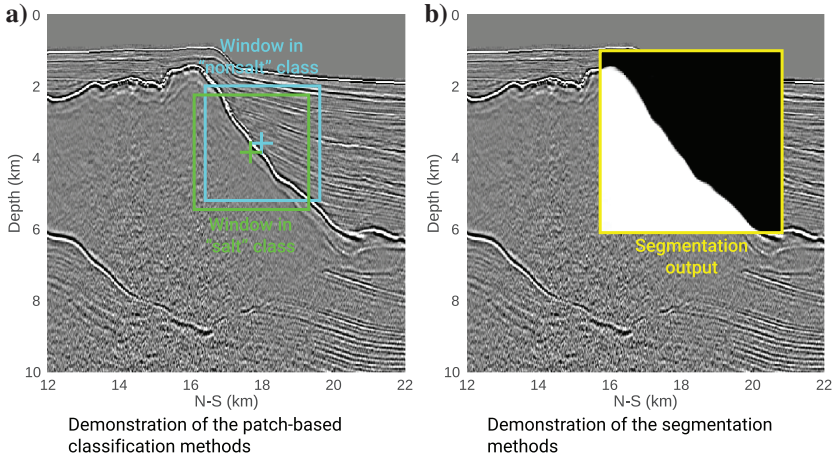



**Figure 1.** (a) Demonstration of the patch-based classification methods: For each pixel, the network will take a window, centered at the point of interest, as its input and classify the category to which this pixel belongs. The process will repeat by sliding the window across the image until all the pixels are scanned. (b) An example of the segmentation method. In this example, all pixels in the window input are classified to a category simultaneously and output together as a mask.

pixels are discarded and the significant ones are kept after downsampling. Conversely, the upsampling layers use the algorithm such as deconvolution (Noh et al., 2015), sometimes also referred to as transposed convolution (Dumoulin and Visin, 2016), to try to reconstruct the image feature to its original dimension. Training a DNN can be seen as looking for a parameterized transformation that transforms the input to the output using stochastic optimization; in the encoder-decoder network, the compression of dimension in the middle of the architecture enforces a sparse representation of such a transform to help to accelerate the training (Papyan et al., 2018). Due to the loss of information from downsampling in the encoder, Badrinarayanan et al. (2017) and Ronneberger et al. (2015) propose different techniques to alleviate this issue by reconnecting these layers from encoder to their counterparts in the decoder. SegNet (Badrinarayanan et al., 2017) is first proposed for traffic scene segmentation; in its decoder, the pooling indices from the encoder downsampling are reused to perform upsampling. U-net (Ronneberger et al., 2015) is first proposed for biomedical image segmentation; the architecture copies layers from the encoder and concatenate to the corresponding layers in the decoder. This type of connections to reuse of network features from the encoder is often referred to as "skip connections."

We adopt the U-net architecture to build a network for 3D salt segmentation. The convolution, pooling, and deconvolution are replaced with 3D operators. The receptive field defines the size of a data subvolume to be input into the network at once. The size of receptive field should be as large as possible so that the network can take global trend into account rather than breaking down to subvolumes; also, multiple predictions on adjacent subvolumes may result in conflicting predictions at boundary regions that require careful patching and merging. However, the size of receptive field is usually limited by hardware: Without sacrificing network complexity, a larger receptive field requires a larger processor memory. We set the receptive field size as $128 \times 128 \times 128$ voxels according to the memory size of our NVIDIA Titan Xp GPU. The input is a 3D image with a single channel: The seismic amplitude, therefore, has a size of $128 \times 128 \times 128 \times 1$. It is worth noting that this network is by no means limited to single-channel input; it is possible to append more channels, e.g., various seismic attributes, to the input to have multichannel input to improve the network capability. However, the additional channels may involve other information that is subject to errors or inaccuracies; in this work, we only use the seismic amplitude to simplify the analysis. The first part of the network, encoder, consists of four groups of convolutional layers, and all convolution layers have a kernel size of $3 \times 3 \times 3$. These convolution groups are followed by batch normalization (Ioffe and Szegedy, 2015), element-wise rectified-linear unit $\sigma(x) = \max\{0, x\}$ for nonlinearities, and max pooling to downsample by factor of two. Symmetric to the encoder, the second part of the network, decoder, upsamples the convolution groups by factor of two. At the end of the network, a single-channel convolutional layer followed by sigmoid activation outputs the binary probability (0 or 1). Because of the symmetry of the architecture, the output will have the same size of the input $128 \times 128 \times 128 \times 1$. Compared with the original U-net architecture in two dimensions, the 3D salt segmentation network includes many more parameters; to train it with generalizing capability, we also add a few dropout layers (Hinton et al., 2012) to the network. Figure 2 shows the illustration of our network architecture.

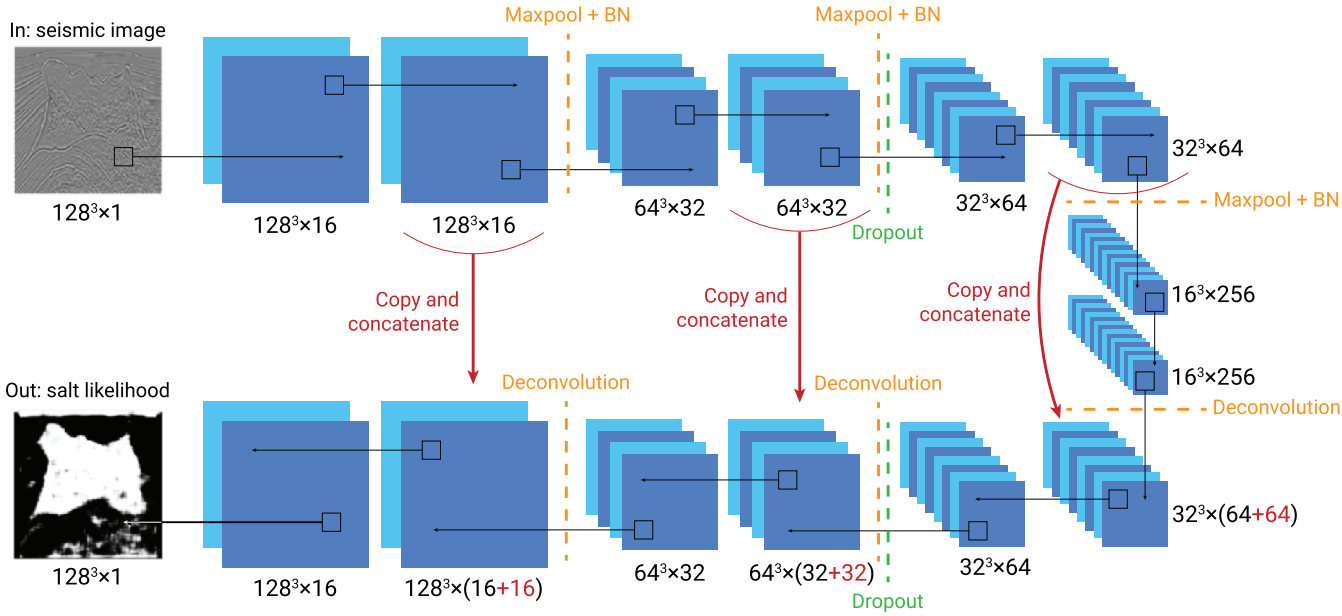



**Figure 2.** Illustration of the network architecture. "BN" is the abbreviation for batch normalization. In the second half of the network, the additional channels from the skip connections are highlighted in dark red. This illustration is not drawn to scale.

## Training the model

To train the model, we select the SEAM Phase I data set (Fehler and Keliher, 2011) as the training data, as shown in Figure 3a. The size of the data set is $1169 \times 1002 \times 751$. We split the data in half along the south–north axis; one half will provide the training data (0–11.69 km), and the other can be used after training as validation data (11.69–23.38 km). To acquire a ground truth salt mask to train the network model, we threshold the velocity model (Figure 3b) in the range of 4.3–4.4 km/s to generate a salt mask. According to the salt mask, we assign binary values to different regions to make a label volume: 0 represents the nonsalt region, and 1 represents the salt body. Figure 3c shows the salt mask as the training label, in which the cyan region represents the salt body and the other nonsalt region is rendered as transparent to show the seismic image background. Figure 3d shows a 3D mesh surface visualization of the ground truth salt model. In Figure 3, we can see that there are two major salt bodies in the volume: one in the shallow depth and another in a deeper depth.

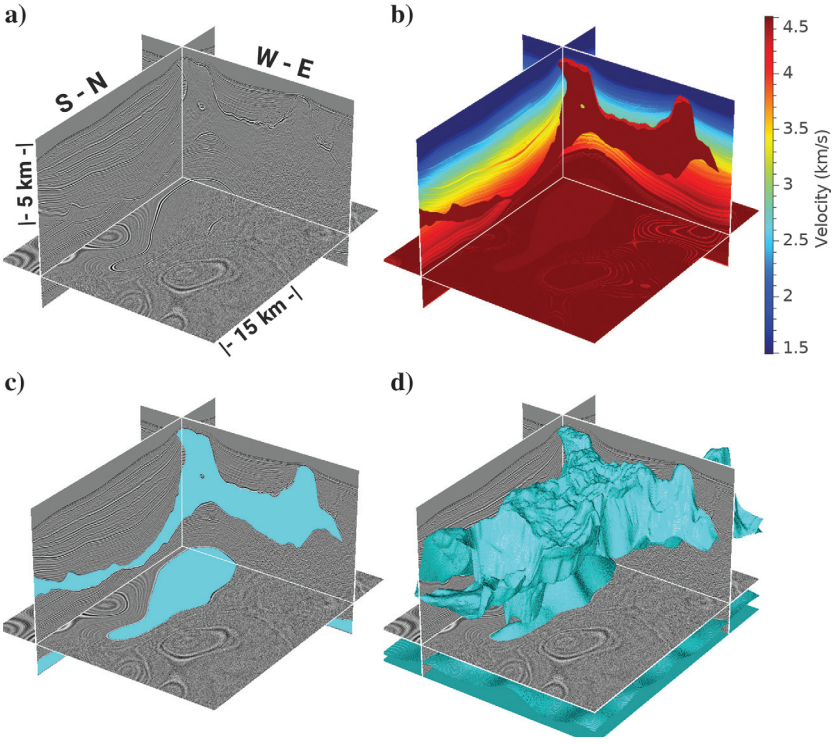



**Figure 3.** (a) SEAM Phase I data set as 3D training data. (b) The corresponding velocity model, from which we threshold the velocity value and extract the salt mask in (c). (c) Salt mask in cyan that is used as the ground truth training label. (d) The 3D mesh surface visualization of the ground truth salt model; these surfaces can represent the salt top and base boundaries.

### *Training data*

Because the network receptive field has a size of $128 \times 128 \times 128$, we randomly crop subvolumes with the same size from the training side of the data set volume. The training labels are also subvolumes cropped from the label volume at the corresponding locations. The position of training label must match the training image. All input images are normalized to have zero-mean and unit-variance distribution to ensure numerical stability in the training. Those training pairs are grouped to batches and input to the network. A larger batch size can result in a smooth and stable learning process; however, due to the size of our network, the batch size is often limited by the hardware memory. For example, the graphics processing unit (GPU) that we use (NVIDIA Titan Xp) has 12 GB graphic memory for the training, which would limit the maximum batch size to two; larger hardware memory could allow for more training samples in the batch.

### *Data augmentation*

We use several data augmentations to improve the generalization of the model as shown in Figure 4. First, the cropping positions are all randomly selected within the volume. Furthermore, the cropping position is not limited to the inline-crossline direction; the horizontal window could be rotated randomly. Then, the two horizontal axes of the cropping window can be randomly transposed or reflected on any axis to create a different variety of subvolume. However, the $z$-axis (the depth axis) will not be transposed because of the vertical heterogeneity of seismic data. So effectively, the transformation is performed in $x$-$y$ 2D space and then replicated on all $z$-axis slices. These transformations are all valid based on the assumption that the salt features in the seismic image are independent to the seismic datum

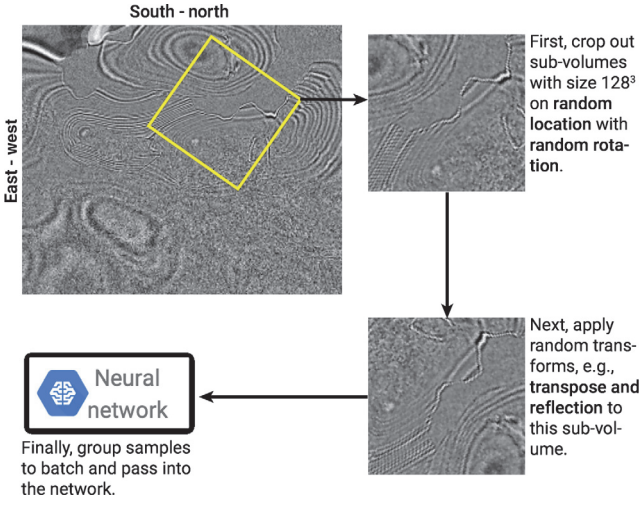



**Figure 4.** The illustration of the data augmentation workflow to improve the training data variety by applying random cropping and random transformations.

footprint. In fact, these data augmentation techniques are able to generate infinite subvolumes from a given seismic data volume.

At the beginning of training, all of the model parameters (the filters in the convolutional layers) are randomly initialized by the distribution described by He et al. (2015). We use the binary cross-entropy loss as the objective function and the adaptive momentum descent (Adam) as the optimization algorithm (Kingma and Ba, 2014) to iteratively update the model weights. We train the model for 100 epochs, where we define each epoch as 50 batches of randomly generated training samples and each batch contains two samples, so that the model is effectively trained with 10,000 random training samples cropped from the training data set. The training takes 17–19 h, with various setups, on a single NVIDIA Titan Xp GPU. Note that this is a one-time cost and can vary with the amount of training samples used in this process. At the end of training, the average pixel-wise accuracy reaches 98.2%.

## Model prediction

Because the network's receptive field size is smaller than the size of most data set volumes, we must predict the salt probability separately for multiple subvolumes and merge all the patches to the full volume dimension. For a local window with size $n_x \times n_y \times n_z$, we move the local window with step $(n_x - k_x, n_y - k_y, n_z - k_z)$ in which $k_x < n_x$, $k_y < n_y$, and $k_z < n_z$ are the size of the overlapping between two adjacent windows on each dimension, respectively. After the salt body probability $P(x, y, z)$ at location $(x, y, z)$ is output by the model, we apply a decaying-boundary mask $M(x, y, z; k_x, k_y, k_z)$ that has unit value at the central region and Gaussian-decaying boundaries on each dimension with $k_x, k_y, k_z$ as the decay radii. Thus, we can merge all the weighted predictions by this weighted summation:

$$P_{\text{full}} = \frac{\sum_{x,y,z} P(x, y, z) \cdot M(x, y, z; k_x, k_y, k_z)}{\sum_{x,y,z} M(x, y, z; k_x, k_y, k_z)}. \quad (1)$$

We can see from the following results that, with the above merging method, the boundary effect due to patch prediction is reduced to minimal.

### *Test on validation data*

We test the trained model on the full SEAM data set volume including the training and validation data sets to check if the model could generalize to the validation side or if it only overfits upon the training side. To keep consistent with the training stage, the zero-mean unit-variance normalization within each window is also applied in the test; however, the random transformations are all turned off.

Figure 5a shows the salt probability volume predicted by the trained model overlaid on the seismic image. The color in the image represents the salt probability values, in which the red stands for high salt probability and the blue represents lower probability. The probability image is also clipped at 0.1, so that any voxel with salt probability <0.1 would be transparent, showing the background seismic image. Figure 5b shows a 3D mesh surface visualization of the predicted salt model. The cyan surfaces represent the 3D contour in which the salt probability equals 0.5, so that these surfaces can represent the salt boundaries. Although the model is trained with half of the data set, it works pretty well to provide an accurate prediction of the salt positions. By comparing the network prediction in Figure 5 with the ground truth in Figure 3, we can observe that the network captures all major features of both salt bodies and the salt probability image matches with the ground truth model with a very sharp boundary. A few false detections scatter around the volume mostly due to the stochastic noise in the seismic image, e.g., the small false positive bodies on the bottom right and a false negative salt boundary on the very left of the volume. However, carefully investigating Figure 5a can show that these false detections have lower possibilities (relatively in blue) than the correct

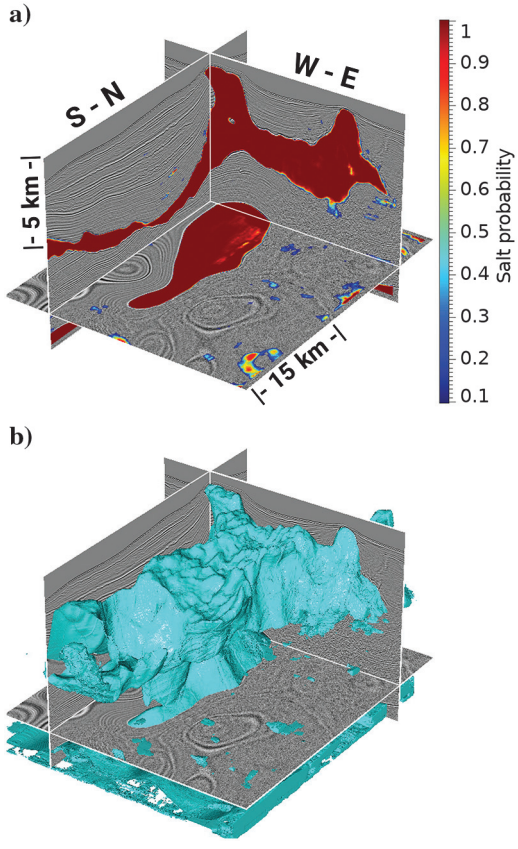



**Figure 5.** (a) The network prediction output on the full volume of SEAM data. The salt probability volume is displayed over the seismic image background. Color in the image represents the salt probability values, in which the red stands for high salt probability and the blue represents lower salt probability. The probability image is also clipped at 0.1 so that any voxel with salt probability <0.1 would be transparent, showing the background seismic image. (b) Similar to Figure 3d, shows a 3D mesh surface visualization of the predicted salt model.

detections (mostly in red), which means that we could use a higher threshold to filter out these false detections. Even though the network prediction is a bit more roughened at highly dipped flanks and has small scattered noisy false positives, it is possible to remove these inaccuracies with postprocessing; on the other hand, the top and base boundaries are captured fairly accurately.

Figures 6 and 7 show four south–north line and west–east line sectional comparisons between the prediction result and ground truth overlaid on the seismic image background. These regions are selected to demonstrate different parts of the salt body. In these figures, the dashed red curve represents the network prediction result and the solid yellow curve represents the ground truth. In Figure 6, the dashed white line indicates the location where we split the training and validation data: The left side provides data to train the network, whereas the right side is only used in the validation test. Figure 6a shows a good match between the prediction and the truth. Figure 6b shows that the network model does not blindly follow the reflection energy boundary; instead, it extends out to match the salt body model as indicated by the arrow. In Figure 6c, there is a thin enclosing area missed by the network, some discrepancies at the salt base, and a few false positives underneath the salt base. In Figure 6d, this enclosing area is captured by the network; however, we can observe an extending false positive as indicated by the arrow. This may be due to the blurry flank coinciding with strong migration artifacts to the left. Figure 7a and 7b is located on the training side, and Figure 7c and 7d is located on the testing side. Figure 7a shows a good matching on the east–west line. Figure 7b shows an overall good matching, including the thin enclosing area, with a few scattering false detections. Figure 7c also shows a decent matching on the salt top and base surfaces. It is very interesting in Figure 7d that in spite of the very blurry seismic reflection and strong noise at the right flank and underneath the salt base, the network model still captures the salt boundary fairly close to the true model. Overall, with some scattering false detections at certain noisy areas, the network prediction matches the ground truth accurate and is robust enough to show the salt top and base surfaces, sometimes even small enclosings. In general,

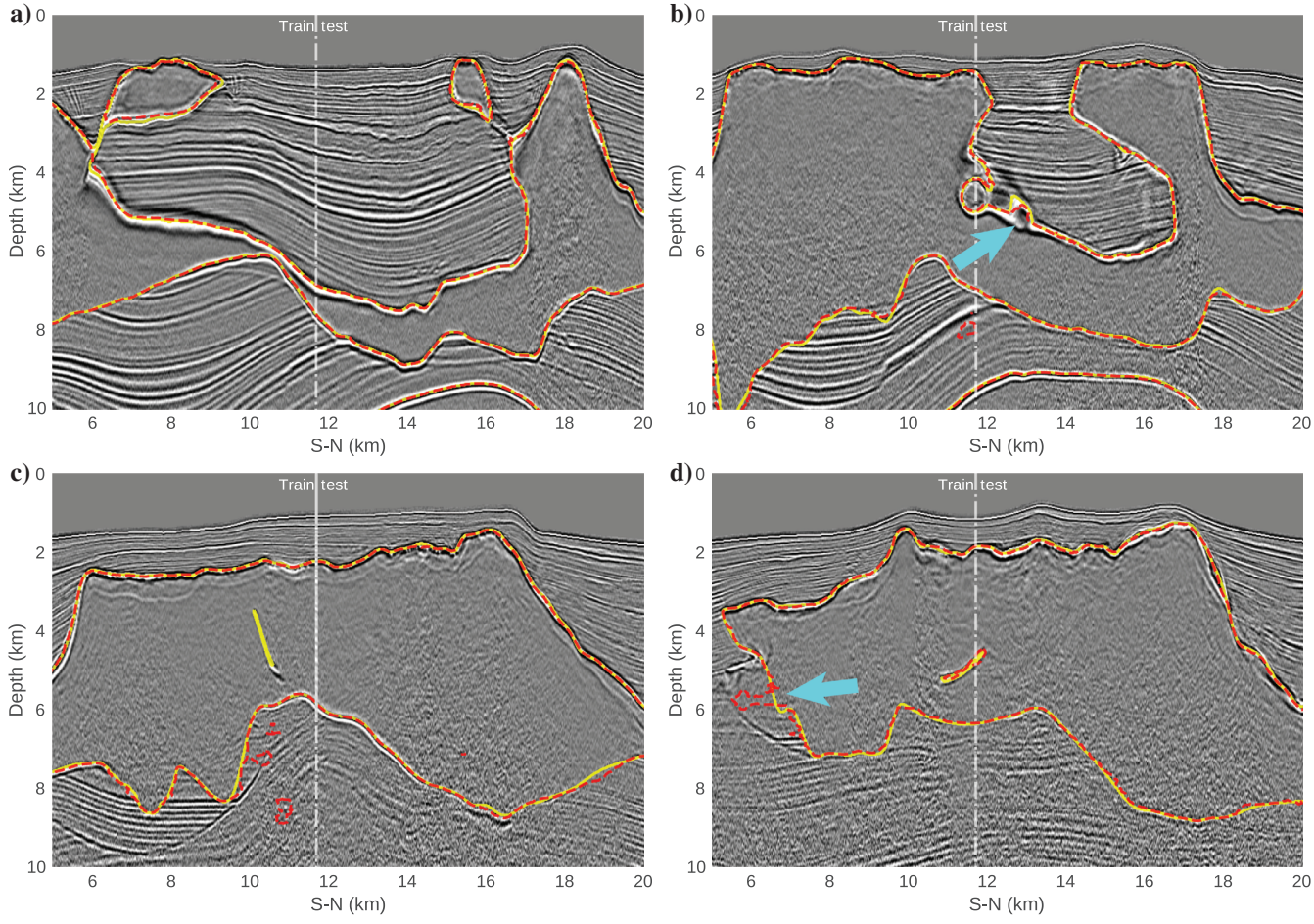



**Figure 6.** Four north–south line sectional comparisons between prediction and ground truth overlaid on the seismic image background. In these figures, the dashed red curve represents the network prediction result, and the solid yellow curve represents the ground truth. The dashed white line indicates the location where we split the training and validation data: The left side provides data to train the network, whereas the right side is only used in validation tests. These regions are selected to demonstrate different parts of the salt body.

the network performs better on prediction of the top than of the base; however, the sectional views show that the differences are not significant.

We computed some quantitative metrics on the validation part of the data set to verify the generalization performance of the network. These metric scores can demonstrate not only the pixel accuracy of the prediction but also the distribution of true positive, true negative, false positive, and false negative of the network prediction. Table 1 shows these metrics, the scores, and their definitions. In the definition, $T_P, T_N, F_P$, and $F_N$ represent the fraction of the true positive, true negative, false positive, and false negative predictions against the ground truth, respectively. These scores are all computed based on a 0.5 thresholding, which means that any voxel with salt probability $>0.5$ would be classified as salt, otherwise as nonsalt.

Admittedly, although the scores in Table 1 show decent performance on the validation test, salt detection in production has a very high demand for accuracy due to its impact on seismic imaging. Therefore, the model may not suffice as a final result but should serve as an initial reference for further interpretation.

### *Test on field example*

We perform another test on the field data set, Netherlands off-shore F3 block seismic data, graciously provided by the Dutch government through TNO and dGB Earth Sciences. A subset of the full volume with size $651 \times 700 \times 240$ is cropped out to remove the uncovered border regions with incomplete traces and to contain three major salt bodies that appear at a deeper depth of the data set, as shown in Figure 8.

Figure 9 shows the salt probability volume and the 3D mesh surface visualization of the predicted salt model on the F3 data set. Unlike the previous synthetic example, these field data tend to be noisier and less

**Table 1. Validation metric scores.**

| | | |
|---|---|---|
| Accuracy | 0.9609 | $(T_P + T_N)/(T_P + T_N + F_P + F_N)$ |
| Precision | 0.9004 | $T_P/(T_P + F_P)$ |
| Recall | 0.9468 | $T_P/(T_P + F_N)$ |
| F1 score | 0.9230 | $(2 \times \text{Precision} \times \text{Recall})/(\text{Precision} + \text{Recall})$ |

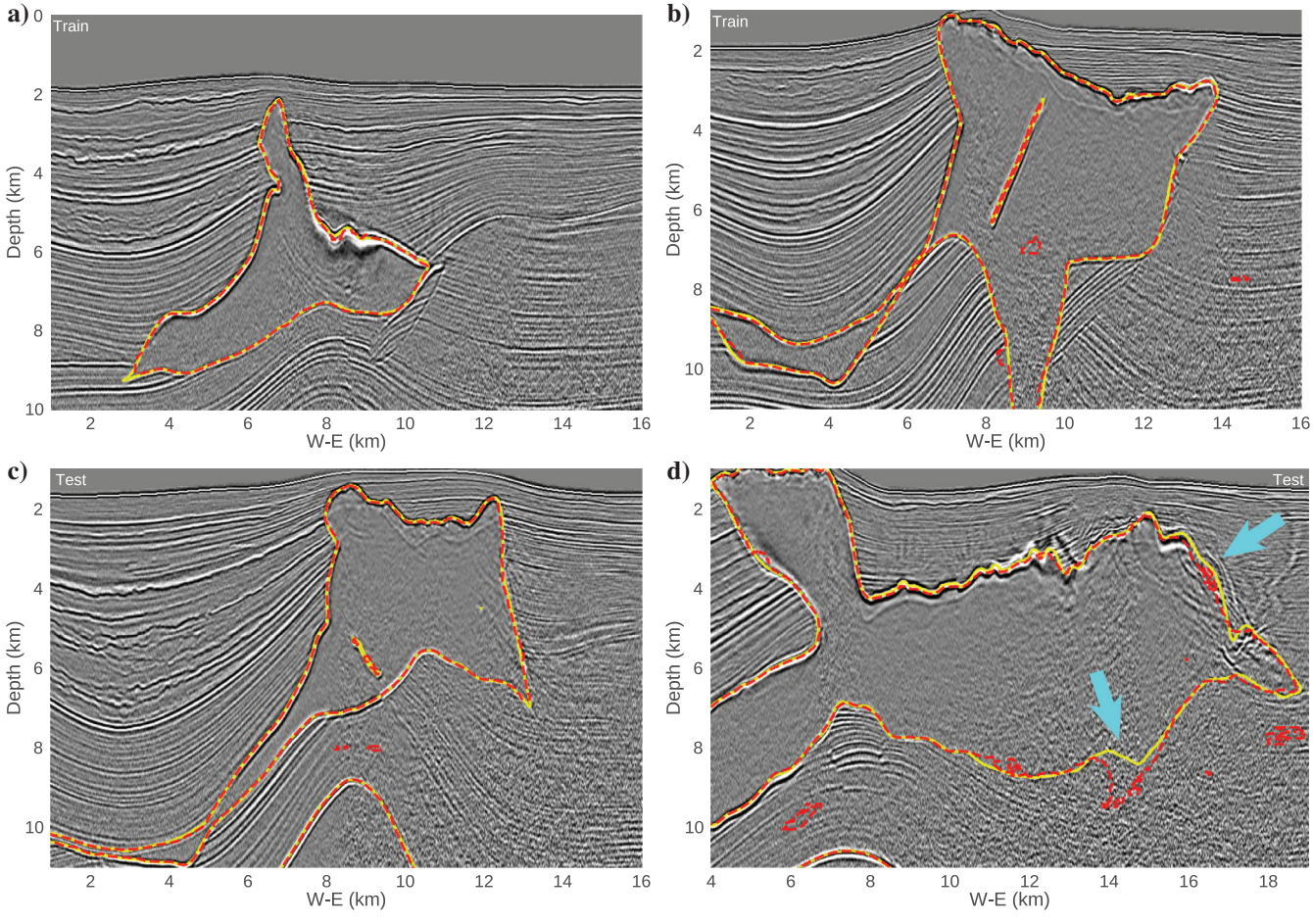



**Figure 7.** Four east–west line sectional comparisons between prediction and ground truth overlaid on the seismic image background. In these figures, the dashed red curve represents the network prediction result and the solid yellow curve represents the ground truth. (a and b) are located on the training side, and (c and d) are located on the testing side. These regions are selected to demonstrate different parts of the salt body.

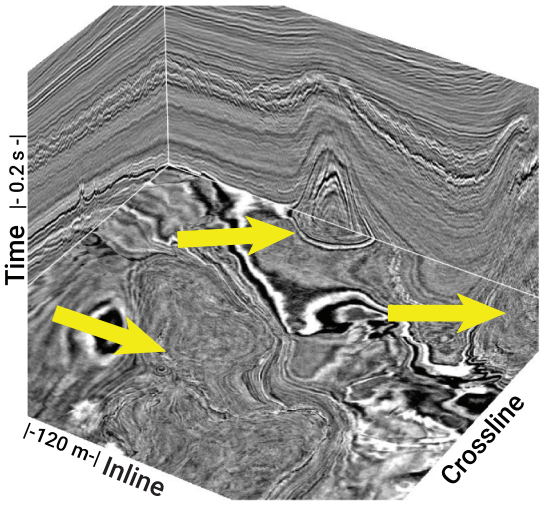



**Figure 8.** Netherlands off-shore F3 block seismic data. This data set is used to test the model generalization on the field survey. The three yellow arrows highlight the major salt bodies in this region.

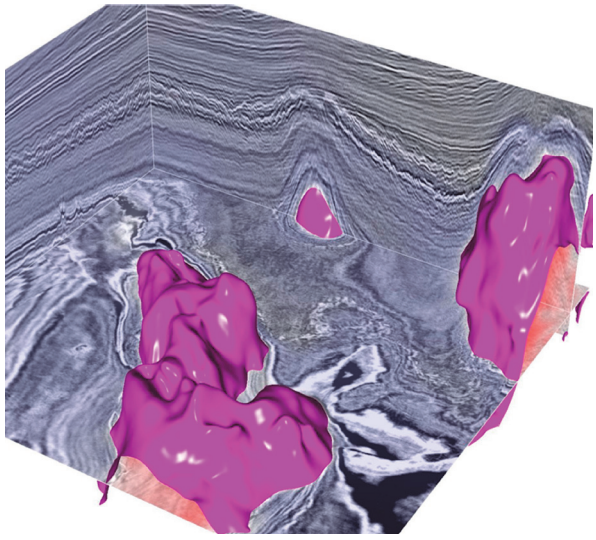


**Figure 10.** An enhanced view of the salt prediction via calculating a postprocessing attribute, salt indicator, described in Wu (2016). This postprocessing removes most of the scattered noisy low-probable salt bodies and snaps the salt boundaries to the nearby seismic reflection interfaces so that they appear more smooth and continuous.

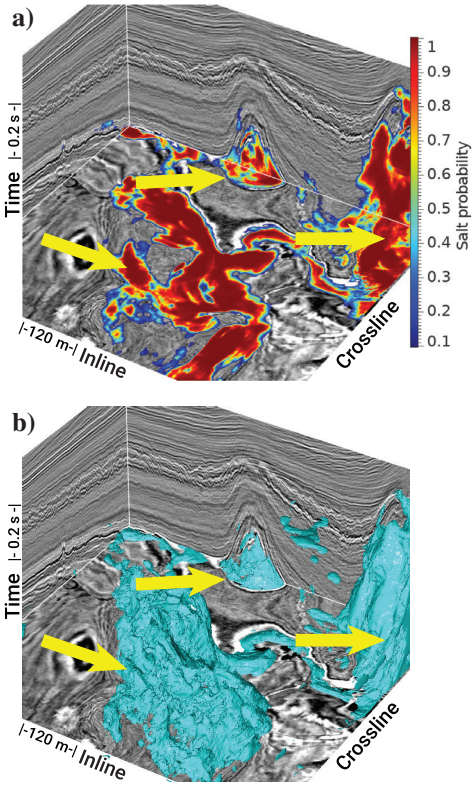



**Figure 9.** (a) The network prediction output on the Netherlands F3 data. The salt probability volume is displayed over the seismic image background. Note that in the center area, we can observe a strong false positive that incorrectly highlights a channel-like feature as salt body. (b) A 3D mesh surface visualization of the predicted salt model. Despite the limitation of training, the general structures of all the salt bodies are correctly highlighted.

apparent at salt boundaries. However, the result shows that all of the salt bodies appear in the prediction with noisy salt boundaries. The probability image in Figure 9a shows a rather gentle transition from low to high salt probability at the salt boundaries. It could be explained that the network is not trained with sufficient samples with noisy and ambiguous salt boundaries. In the center area of the figure, we can observe a strong false positive that incorrectly highlights a channel-like feature as a salt body; this implies that binary classification in complex field data is not enough and multifacies segmentation is a possible solution to such issue. Overall, the general structure of all of these salt bodies is still captured fairly well despite the limitations of the training. In addition, we calculate a postprocessing attribute based on the salt probability prediction via the salt indicator attribute described by Wu (2016) as shown in Figure 10. This postprocessing removes most of the scattered noisy low-probability salt bodies from the result and snaps the salt boundaries to the nearby seismic reflection interfaces so that they appear more smooth and continuous, so that the salt model could be enhanced.

In summary, the results show that the deep CNN model generalizes well in either synthetic or field data sets in terms of capturing subtle salt features from the seismic image. The implementation could also be very efficient with hardware acceleration on a GPU. By using one NVIDIA Titan Xp, computing the salt prediction via proposed model on a $1000^3$ volume would take less than 10 min.

## Conclusion

We propose a method for end-to-end automatic salt body detection in a seismic image based on a deep CNN model. The encoder-decoder architecture with skip

connections allows for extracting essential information from training data, thus resulting in high accuracy and great generalization across different types of data sets. The network can take a 3D single-channel seismic amplitude image as input and output a salt body probability volume by analyzing the visual features learned from the training samples. However, the model can take multichannel input with multiple seismic attributes as additional information. We design a general method to randomly generate training samples according to the receptive field size with several data augmentation methods, and we train the proposed network with effectively 10,000 random subvolumes. The prediction results show that the trained model can not only generalize to the synthetic validation data set, and the field data set with noisy salt boundaries as well, thus demonstrating the future potential of this efficient and effective tool for automatic geobody interpretations.


## Acknowledgments

We appreciate the financial support from sponsors of the Texas Consortium for Computational Seismology (TCCS). We also thank the NVIDIA GPU Grant Program for providing the Titan Xp GPU for the deep-learning computations.


## Data and materials availability

Data associated with this research are available and can be obtained by contacting the corresponding author.

## References


Araya-Polo, M., T. Dahlke, C. Frogner, C. Zhang, T. Poggio, and D. Hohl, 2017, Automated fault detection without seismic processing: The Leading Edge, **36**, 208–214, doi: 10.1190/tle36030208.1.

Asjad, A., and D. Mohamed, 2015, A new approach for salt dome detection using a 3D multidirectional edge detector: Applied Geophysics, **12**, 334–342, doi: 10.1007/s11770-015-0512-2.

Badrinarayanan, V., A. Kendall, and R. Cipolla, 2017, SegNet: A deep convolutional encoder-decoder architecture for scene segmentation: IEEE Transactions on Pattern Analysis and Machine Intelligence, **39**, 2481–2495.

Boureau, Y.-L., J. Ponce, and Y. LeCun, 2010, A theoretical analysis of feature pooling in visual recognition: Proceedings of the 27th International Conference on Machine Learning, 111–118.

Csáji, B. C., 2001, Approximation with artificial neural networks: M.Sc. thesis, Etvs Lornd University.

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

Di, H., Z. Wang, and G. AlRegib, 2018b, Real-time seismic-image interpretation via deconvolutional neural network: 88th Annual International Meeting, SEG, Expanded Abstracts, 2051–2055, doi: 10.1190/segam2018-2997303.1.

Dumoulin, V., and F. Visin, 2016, A guide to convolution arithmetic for deep learning: ArXiv preprint arXiv:1603.07285.

Fehler, M. C., and P. J. Keliher, 2011, SEAM phase I: Challenges of subsalt imaging in Tertiary basins, with emphasis on deepwater Gulf of Mexico: SEG.

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

Haukås, J., O. R. Ravndal, B. H. Fotland, A. Bounaim, and L. Sonneland, 2013, Automated salt body extraction from seismic data using the level set method: First Break, **31**, 35–42.

He, K., G. Gkioxari, P. Dollár, and R. Girshick, 2017, Mask R-CNN: Proceedings of the IEEE International Conference on Computer Vision, 2980–2988.

He, K., X. Zhang, S. Ren, and J. Sun, 2015, Delving deep into rectifiers: Surpassing human-level performance on imagenet classification: Proceedings of the IEEE International Conference on Computer Vision, 1026–1034.

Hinton, G. E., N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, 2012, Improving neural networks by preventing co-adaptation of feature detectors: ArXiv preprint arXiv:1207.0580.

Huang, L., X. Dong, and T. E. Clee, 2017, A scalable deep learning platform for identifying geologic features from seismic attributes: The Leading Edge, **36**, 249–256, doi: 10.1190/tle36030249.1.

Ioffe, S., and C. Szegedy, 2015, Batch normalization: Accelerating deep network training by reducing internal covariate shift: International Conference on Machine Learning, 448–456.

Kingma, D. P., and J. Ba, 2014, Adam: A method for stochastic optimization: ArXiv, abs/1412.6980.

Lomask, J., R. G. Clapp, and B. Biondi, 2007, Application of image segmentation to tracking 3D salt boundaries: Geophysics, **72**, no. 4, P47–P56, doi: 10.1190/1.2732553.

Long, J., E. Shelhamer, and T. Darrell, 2015, Fully convolutional networks for semantic segmentation: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 3431–3440.

Marfurt, K. J., and T. M. Alves, 2015, Pitfalls and limitations in seismic attribute interpretation of tectonic features: Interpretation, **3**, no. 1, SB5–SB15, doi: 10.1190/INT-2014-0122.1.

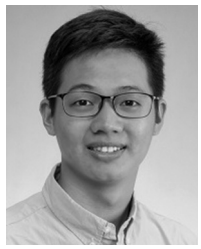
Noh, H., S. Hong, and B. Han, 2015, Learning deconvolution network for semantic segmentation: Proceedings of the IEEE International Conference on Computer Vision, 1520–1528.

Papyan, V., Y. Romano, J. Sulam, and M. Elad, 2018, Theoretical foundations of deep learning via sparse representations: A multilayer sparse model and its connection to convolutional neural networks: IEEE Signal Processing Magazine, **35**, 72–89, doi: 10.1109/MSP.2018.2820224.

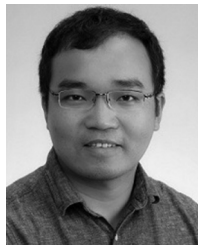
Ramirez, C., G. Larrazabal, and G. Gonzalez, 2016, Salt body detection from seismic data via sparse representation: Geophysical Prospecting, **64**, 335–347, doi: 10.1111/1365-2478.12261.

Ren, S., K. He, R. Girshick, and J. Sun, 2017, Faster R-CNN: Towards real-time object detection with region proposal

networks: IEEE Transactions on Pattern Analysis and Machine Intelligence, **39**, 1137–1149.
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.
Ross, C. P., and D. M. Cole, 2017, A comparison of popular neural network facies-classification schemes: The Leading Edge, **36**, 340–349, doi: 10.1190/tle36040340.1.
Waldeland, A., and A. Solberg, 2017, Salt classification using deep learning: 79th Annual International Conference and Exhibition, EAGE, Extended Abstracts, doi: 10.3997/2214-4609.201700918.
Wang, W., F. Yang, and J. Ma, 2018, Automatic salt detection with machine learning: 80th Annual International Conference and Exhibition, EAGE, Extended Abstracts, doi: 10.3997/2214-4609.201800917.
Wang, Z., T. Hegazy, Z. Long, and G. AlRegib, 2015, Noise-robust detection and tracking of salt domes in postmigrated volumes using texture, tensors, and subspace learning: Geophysics, **80**, no. 6, WD101–WD116, doi: 10.1190/geo2015-0116.1.
Wu, X., 2016, Methods to compute salt likelihoods and extract salt boundaries from 3D seismic images: Geophysics, **81**, no. 6, IM119–IM126, doi: 10.1190/geo2016-0250.1.
Wu, X., S. Fomel, and M. Hudec, 2018a, Fast salt boundary interpretation with optimal path picking: Geophysics, **83**, no. 3, O45–O53, doi: 10.1190/geo2017-0481.1.
Wu, X., L. Liang, Y. Shi, and S. Fomel, 2019, FaultSeg3D: Using synthetic datasets to train an end-to-end convolutional neural network for 3D seismic fault segmentation: Geophysics, **84**, no. 3, K1–K9, doi: 10.1190/geo2018-0120.1.
Wu, X., Y. Shi, S. Fomel, and L. Liang, 2018b, Convolutional neural networks for fault interpretation in seismic images: 88th Annual International Meeting, SEG, Expanded Abstracts, 1946–1950, doi: 10.1190/segam2018-2995341.1.
Xie, S., and Z. Tu, 2015, Holistically-nested edge detection: Proceedings of the IEEE International Conference on Computer Vision, 1395–1403.
Zhao, T., 2017, Machine assisted quantitative seismic interpretation: Ph.D. thesis, University of Oklahoma.
Zhao, T., 2018, Seismic facies classification using different deep convolutional neural networks: 88th Annual International Meeting, SEG, Expanded Abstracts, 2046–2050, doi: 10.1190/segam2018-2997085.1.



**Yunzhi Shi** received a B.S. (2015) in geophysics from the University of Science and Technology of China (USTC). He interned twice at BP in Houston, TX. He has been working at TCCS as a Ph.D. student since 2015. He currently focuses on deep-learning applications on interpretation tasks including fault detection, salt body classification, and stratigraphic correlation.



**Xinming Wu** received a Ph.D. in geophysics from the Colorado School of Mines, where he was a member of the Center for Wave Phenomena. He interned twice at DrillingInfo in Littleton, Colorado. From August 2015, he has been working at TCCS with S. Fomel, first as a visiting Ph.D. student and now as a postdoctoral fellow. He received the Best Paper Award in GEOPHYSICS in 2016 (coauthored by D. Hale) and the Top 30 SEG Papers Award in 2017 (coauthored by S. Bader and S. Fomel). He is mainly interested in seismic interpretation, image processing, machine learning, subsurface modeling, and geophysical inversion.

**Sergey Fomel** received a Ph.D. (2001) in geophysics from Stanford University and previously worked at the Russian Institute of Geophysics and the Lawrence Berkeley National Laboratory. He received the J. Clarence Karcher Award from SEG in 2001, Best SEG Poster Presentation Award in 2007, and the Conrad Schlumberger Award from the EAGE in 2011.