

Multiresolution neural networks for tracking seismic horizons from few training images

Bas Peters¹, Justin Granek², and Eldad Haber¹

Abstract

Detecting a specific horizon in seismic images is a valuable tool for geologic interpretation. Because hand picking the locations of the horizon is a time-consuming process, automated computational methods were developed starting three decades ago. Until now, most networks have been trained on data that were created by cutting larger seismic images into many small patches. This limits the networks ability to learn from large-scale geologic structures. Moreover, currently available networks and training strategies require label patches that have full and continuous horizon picks (annotations), which are also time-consuming to generate. We have developed a projected loss function that enables training on labels with just a few annotated pixels and has no issue with the other unknown label pixels. We use this loss function for training convolutional networks with a multiresolution structure, including variants of the U-net. Our networks learn from a small number of large seismic images without creating patches. Training uses all seismic data without reserving some for validation. Only the labels are split into training/testing. We validate the accuracy of the trained network using the horizon picks that were never shown to the network. Contrary to other work on horizon tracking, we train the network to perform nonlinear regression, not classification. As such, we generate labels as the convolution of a Gaussian kernel and the known horizon locations that communicate uncertainty in the labels. The network output is the probability of the horizon location. We examine the new method on two different data sets, one for horizon extrapolation and another data set for interpolation. We found that the predictions of our methodology are accurate even in areas far from known horizon locations because our learning strategy exploits all data in large seismic images.

Introduction

Geologic horizons are of great importance for understanding geology and targeting resources such as hydrocarbon and water. Seismic imaging of the subsurface is the method of choice to obtain high-resolution images, from shallow to relatively large depths (see Figure 1 for an example).

Seismic data are collected as a function of a shot and a recording location. The raw seismic data can be converted to images with the vertical axes representing either depth or time. In this work, we assume that the raw seismic data are already converted into such images. In the field data examples, the seismic images are in terms of depth, and we will refer to depth subsequently for brevity. Traditionally, these seismic images are then manually interpreted by experts to identify and interpret the horizons. Hand picking the horizons in large seismic cross sections or 3D volumes can be very time consuming, especially where the resolution of the seismic image is suboptimal or the geology is more complex.

Previous work and related problems

Nonlearning-based automatic horizon tracking algorithms help seismic interpreters by tracking the horizon based on a few hand-picked reference locations; however, in more challenging environments and low-quality seismic images, these tools can produce poor results and require added user supervision.

Automatic horizon detection algorithms come in various flavors. Neural network-based methods have a long history for these applications. Early works by Harrigan et al. (1992), Veezhinathan et al. (1993), Liu et al. (1989), Huang (1997), Kusuma and Fish (1993), and Alberts et al. (2000) use multilayer perceptron or recurrent networks of a few layers. Neither the quantity or quality of data nor the computing power used in these early works is comparable with today's technology. Some of the earlier references are limited to work with one or a few time recordings (traces) at a time, thereby limiting the spatial information the networks can exploit. Wu and Zhang (2018) and Zhao (2018) use a convolutional autoencoder to perform

¹University of British Columbia, Vancouver, Canada. E-mail: bpeters@eoas.ubc.ca; ehaber@eoas.ubc.ca.

²Computational Geosciences Inc., Vancouver, Canada. E-mail: justin@compgeoinc.com.

Manuscript received by the Editor 29 November 2018; revised manuscript received 12 March 2019; published ahead of production 16 May 2019; published online 17 July 2019. This paper appears in *Interpretation*, Vol. 7, No. 3 (August 2019); p. SE201–SE213, 15 FIGS., 3 TABLES.

<http://dx.doi.org/10.1190/INT-2018-0225.1>. © 2019 Society of Exploration Geophysicists and American Association of Petroleum Geologists. All rights reserved.

segmentation of seismic images into a few regions. They pose the segmentation problem as a classification task in which the horizons delineate the boundary between the class regions. Their training data are randomly selected out of a seismic volume and are therefore an example of the interpolation of horizon locations. Di (2018) proposes to train on a large number of small annotated patches (an annotated patch means that the geologic interpretation, or label, is known for each pixel in the patch), for classifying seismic data volumes as an integrated geologic interpretation. A key difference from our work is that we work with the largest images practically possible (either limited by hardware constraints or by the spatial data set extent), such that we can exploit spatial information over large distances to help predictions. A comparison by Zhao (2018) confirms that training image-to-image leads to better predictions compared with predicting the class of the central pixel from a small patch. Image-to-image training means that the input for a network is a seismic image, and the output is an image in which each pixel contains information about whether or not a horizon is present. Contrary, image patch to central pixel training means that the input for a network is a patch of a larger seismic image, and the output is the probability of the horizon at only the central pixel of that patch.

There are also many algorithms for seismic horizon tracking that do not use neural networks. Some of them require data preprocessing. There are also algorithms that detect all horizons in a seismic image rather than one specific interface (Kusuma and Fish, 1993; Li et al., 2012). The numerous methods to compute a volume of horizons from an input seismic image automatically include the UVT transform (Mallet, 2004; Labrunye et al., 2009; Mallet et al., 2010), phase unwrapping (Stark, 2005; Wu and Zhong, 2012), slope-based flattening (Lomask et al., 2006; Parks, 2010; Wu and Hale, 2015b),

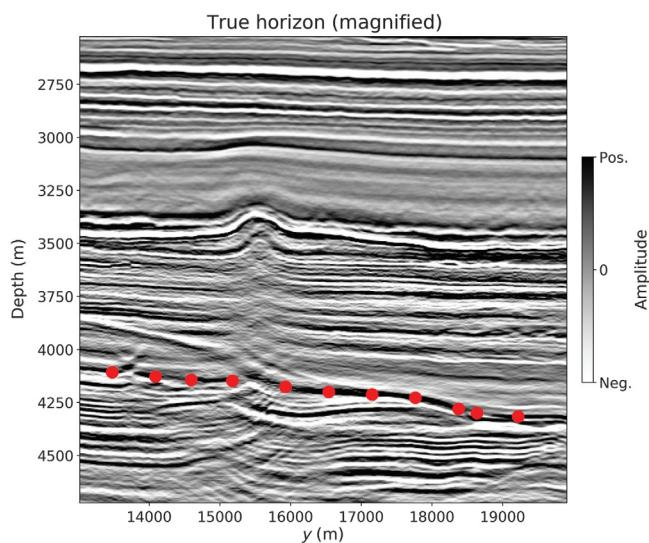


Figure 1. A seismic image in which the red dots indicate the x - y - z locations (seed points) of a horizon of interest. This image is a small part of the size we use for training.

and predictive painting (Fomel, 2010). Unconformities and faults are typical challenges for automatic horizon tracking because they are geologic boundaries in which horizons terminate or shift. To deal with these issues, Wu and Hale (2015a) propose to first automatically detect unconformities that are further used as boundary constraints to more accurately estimate reflection slopes and extract horizons near unconformities. To deal with faults in automatic horizon picking, some authors (Luo and Hale, 2013; Wu and Hale, 2016; Wu et al., 2016) propose to first undo the faulting in a seismic image by using automatically estimated fault throws and then to extract horizons from the unfaulted seismic image. Wu and Fomel (2018) propose to use slopes and multigrid correlations to extract horizons across faults automatically.

Our goal of interpolating or extrapolating a specific seismic horizon is different from the related problems of salt-body (Waldeleland et al., 2018; Shi et al., 2019), fault (Tingdahl and De Rooij, 2005; Araya-Polo et al., 2017), chimney detection (Meldahl et al., 2001), or multiple features (Alaudah et al., 2018). Recognizing and tracking a horizon presents a different challenge because every seismic image contains many horizons, but only a small specific subset is of interest. Thus, the goal is not only to learn how to recognize just any horizon but also how to find and track one particular horizon of interest.

New contributions

We provide a new approach to the horizon identification problem in seismic images. First, the multiscale nature of seismic data suits a recently proposed network architecture (Ronneberger et al., 2015) with 2D convolutional kernels, which has been shown to produce best-in-class performance for image segmentation in other fields such as medical imaging. Second, in contrast to most data sets using deep learning for image recognition (Deng et al., 2009; Krizhevsky and Hinton, 2009), our data set consists of a relatively small set of large images. To facilitate learning in such conditions, we introduce a partial loss function that enables training on partially labeled horizons. Our partial loss is different from methods that extract a small patch/cube around a label point (also known as a seed interpretation, Meldahl et al., 1999) and classify the data patch by patch, yielding one classified pixel at a time. The partial loss enables us to train on sparse labels directly, without extracting a patch around the label point.

Contrary to most work based on neural networks, we do not frame our problem as a classification task. Instead, we formulate nonlinear regression problems to learn how to map from a seismic image to the “horizon image.” Each pixel in the horizon image, roughly, corresponds to the probability of a horizon being at that pixel location. This is a convenient way to include uncertainty information on the horizon labels explicitly. Therefore, the network output is also an image that naturally conveys the uncertainty in the horizon depth estimates. Our labels do not need to be, technically, a probability. The nonlinear regression approach gives us

the freedom to choose the way we define labels. Note that a classification approach provides the probability map of a class, which corresponds to the probability of a geologic rock type at each location in the image. The horizon location follows from such information as the points in which the maximum class probability changes from one class to another; however, this does not directly provide the probability of the horizon at each location.

Because we train on large 2D seismic images, there is no need to create small image patches. We thereby avoid manual user input on the window/patch size, which would impact the results, as well as any artifacts resulting from a tiled solution to the problem. We choose to work on large 2D images rather than small 3D subvolumes because the network cannot learn large-scale geologic structure from small 3D subvolumes. Using large 3D seismic volumes as network input is currently out of reach because of the typical memory requirements of deep networks. The data set used consists of seismic images, and the algorithm is trained without the use of any other attribute images, wavelet information, or preprocessing that earlier work on horizon tracking used as supplemental input (see, e.g., Meldahl et al., 2001; Poulton, 2002; Leggett et al., 2003; Huang et al., 1997; Alberts et al., 2000).

Finally, because our approach uses regression to train for the depth of the horizon, special consideration is necessary when preparing labels for the problem. We introduce a novel parameterization of the training label information, which lends itself to more transparent handling of uncertainty information and a probabilistic interpretation of the predicted results. Due to the sparsity of the horizon labels, the resulting training set can be very unbalanced. We handle this problem by rebalancing the training set at each iteration via per-class random sampling and demonstrate the importance of this step for the result.

Putting the above ingredients together, we train a 2D convolutional image-to-image U-net to predict the probability of a specific horizon at each pixel in a large seismic image. The partial-loss function allows for training labels (images) that have only a few sparse picks of the horizon of interest, converted into a quantity akin to a probability, or, the inverse of the distance to the horizon.

Application to field data

We validate the new computational methods, loss function and learning strategy using seismic images from the North Sea and the Sea of Ireland. We demonstrate the effectiveness of our network architecture and new partial loss function, as well as investigate the difference between alternate problem setups, including

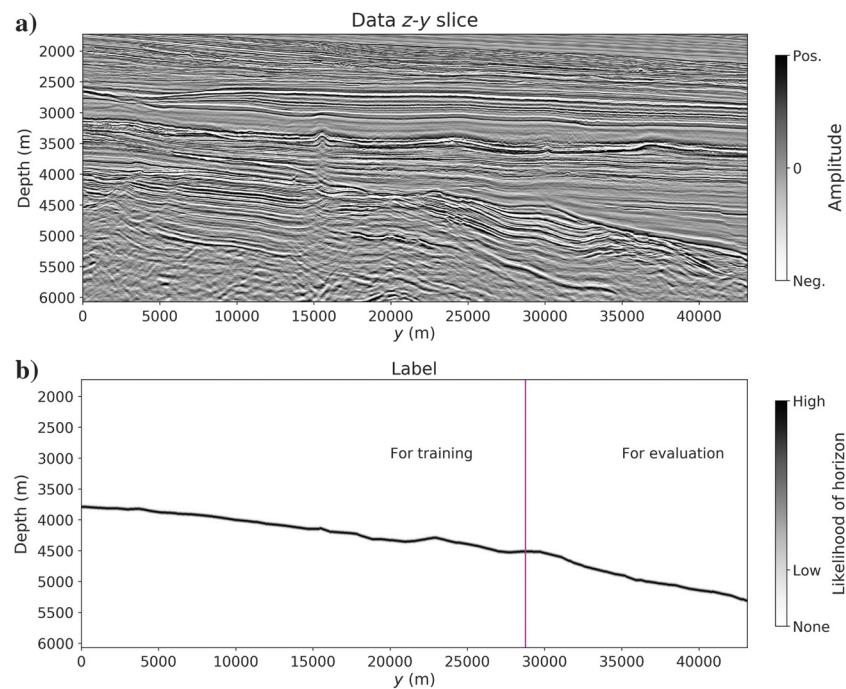


Figure 2. (a) A single seismic image (data) and (b) a single label image for horizon extrapolation. The part of the label image that is on the left of the vertical magenta line is used for training. The part that is on the right of the vertical line is for evaluation only. The label is the convolution of a Gaussian kernel with the horizon x - y - z locations.

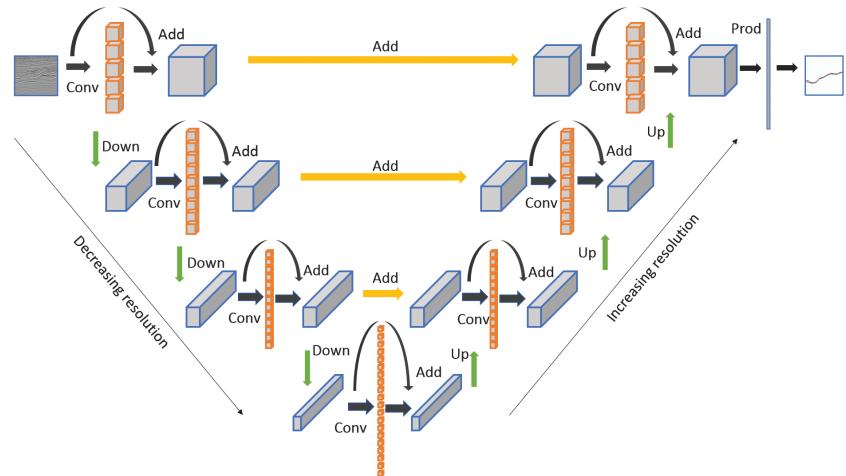


Figure 3. Schematic view of the U-net that we use in this work. The input is a seismic image, and the output is an image that shows the likelihood of the horizon. The loss function compares this output to a partial label image as in Figure 5b.

Algorithm 1. The SGD-based training algorithm with partial labels and binary random sample rebalancing.

Input:

```

 $\mathbf{y}^1, \mathbf{y}^2, \dots, \mathbf{y}^{n_{\text{examples}}}$  //data images
 $\mathbf{c}^1, \mathbf{c}^2, \dots, \mathbf{c}^{n_{\text{examples}}}$  //label images
 $\Omega^1, \Omega^2, \dots, \Omega^{n_{\text{examples}}}$  //known label pixels in each image
 $n_{\text{samp}}$  //number of samples per iteration
 $\gamma$  //set learning rate
for  $1:n_{\text{epochs}}$  do
  for  $1:n_{\text{examples}}$  do
     $\{\mathbf{y}, \mathbf{c}, \Omega\}$  //draw random data, label image, known label indices
     $\mathcal{B}_1$  //draw at random  $n_{\text{samp}}/2$  zero label indices from  $\Omega$ 
     $\mathcal{B}_2$  //draw at random  $n_{\text{samp}}/2$  nonzero label indices from  $\Omega$ 
     $\mathcal{A} = \mathcal{B}_1 \cup \mathcal{B}_2$ 
     $l_{\mathcal{A}}(f(\mathbf{y}, \theta), \mathbf{c}), \nabla_{\theta} l_{\mathcal{A}}(f(\mathbf{y}, \theta), \mathbf{c})$  //partial loss and gradient
     $\theta \leftarrow \theta - \gamma \nabla_{\theta} l_{\mathcal{A}}(f(\mathbf{y}, \theta), \mathbf{c})$  //update parameters
  End
End

```

Table 1. The downscale arm of the U-net used for the Sea of Ireland data.

Layer number	Feature size	Number of channels	kernel size
1	1088×2816	4	3×3
2	1088×2816	4	3×3
3	1088×2816	4	3×3
4	544×1408	6	3×3
5	544×1408	6	3×3
6	544×1408	6	3×3
7	272×704	8	3×3
8	272×704	8	3×3
9	272×704	8	3×3
10	136×352	12	3×3
11	136×352	12	3×3
12	136×352	12	3×3
13	68×176	16	3×3
14	68×176	16	3×3
15	68×176	16	3×3
16	34×88	24	3×3
17	34×88	24	3×3
18	17×44	32	3×3
19	17×44	32	3×3

Note: The upsampled arm is the same in reverse order, and it resuses the weights of the downscale arm, resulting in 37,336 parameters. The network output has the same size as the input image, and it represents the probability of the horizon of interest.

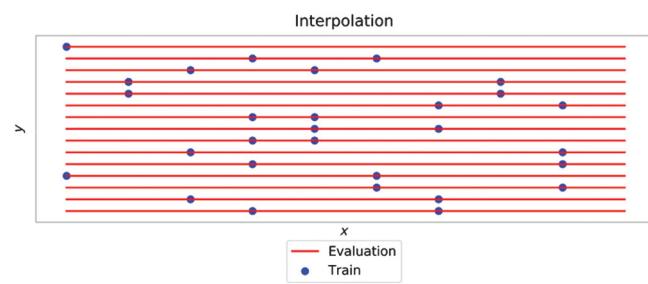
interpolation versus extrapolation and inline versus crossline predictions.

Label preparation and handling

The raw label data are x - y - z coordinates of the location of the horizon of interest. We can directly plot the x - y - z coordinates in an image, by assigning the number zero to nonhorizon locations and the number one to horizon locations. We found training and predictions from this type of training labels rather ineffective, and it does not include valuable information about the uncertainty of the horizon picks.

The horizon picks are either handpicked or obtained using an automatic horizon tracker with some human assistance and quality control. The selected x - y - z locations are therefore not completely accurate. Another source of label errors is the seismic image itself, from which the labels are generated. The quality of the seismic image deteriorates if there is noise in the data, incorrectly imaged/removed multiple reflections, or if a less-favorable acquisition design was used. Violations of the assumptions underlying the migration method that generates the seismic image also lead to low-quality images. A common assumption is that the geology is almost laterally invariant, i.e., slowly varying in the horizontal direction. More advanced imaging algorithms (e.g., reverse time migration) assume a background velocity model that is approximately a smoothed version of the true velocity model. Violations of the assumptions result in parts of the seismic image becoming blurred and continuous layers are broken up. The exact location of the horizon is ambiguous in these situations.

To reflect the uncertainty in the provided horizon labels, we add information about the uncertainty as follows: We convolve the horizon locations with a normalized Gaussian kernel. The resulting values are probabilities of the horizon location. The provided x - y - z location has the highest value, and the probability of a horizon tapers off as the distance from the x - y - z pick increases. In Figure 2b, we show an example of a data image and label for the case we are given a horizon and need to extrapolate it. These images are of the size that we use for training (1088×2816 pixels).

**Figure 4.** Plan view of horizon interpolation from seed points (scattered label points).

Network design

A key consideration in the prediction of a horizon in a seismic image is the network architecture. Most work done in the past uses very few network layers for the prediction. It has been shown that for many vision applications such networks can have a limited power of prediction (for some recent numerical experiments, see Jégou et al., 2017; Lin et al., 2017; Shah et al., 2018). Recent architectures are deep networks that can contain tens if not hundreds of layers. One such stable design is the residual network (ResNet, He et al., 2015), which is described as

$$\mathbf{y}_{j+1} = \mathbf{P}_j \mathbf{y}_j + g(\mathbf{y}_j, \boldsymbol{\theta}_j), \quad (1)$$

where \mathbf{y}_{j+1} is the tensor-valued network state at layer number $j + 1$, which depends on the state in the previous network layer, \mathbf{y}_j , and a nonlinear function g that consists of a hyperbolic tangent or a rectified linear unit. The term $\boldsymbol{\theta}$ is the parameter (the convolutional kernel) of each layer to be learned from the data and label images. The transformation \mathbf{P} is used to increase or decrease the number of channels of the network. For our problem, we start with a single channel (we process one seismic image at a time) and open the network to a few tens of channels. In case we have seismic images that are the result of imaging multiple offsets (e.g., near-stack, mid-stack, and far-stack), we can start with multiple input channels. Each input channel then contains a seismic image of the same piece of earth, but corresponding to data with different offsets.

Although ResNets have been very successful for image classification, they tend to be less accurate for segmentation problems. The main problem is scale; convolution is a local operator; therefore, the network can have difficulties learning features that span several scale lengths in either 1D, 2D, or 3D. The network in this paper uses only 2D convolutions because our input data and label images are 2D.

To resolve this problem, we have used a U-net (Ronneberger et al., 2015) structure. U-nets are similar to autoencoders in that they restrict (i.e., downsample the image) as they go deeper. The network has two “arms” (see Figure 3). In the downscale arm, equation 1 is used, with a small modification:

$$\mathbf{y}_{j+1} = \mathbf{R}\mathbf{P}_j\mathbf{y}_j + \mathbf{R}g(\mathbf{y}_j, \boldsymbol{\theta}_j), \quad (2)$$

where \mathbf{R} is a restriction operator that downsamples the image using a full weighting (Briggs et al., 2000). A full-

weighting operator averages multiple points on the fine grid to compute one point on the coarser (restricted) grid. We select such an operator, instead of simple restriction, because the transposed operator \mathbf{R}^\top allows us to move to finer grids without introducing “holes” (zeros).

Let \mathbf{y}_N be the image sampled on the lowest resolution. In the second arm of the network, the image is upsampled to its original size; that is, the image is interpolated starting with $\mathbf{z}_N = \mathbf{y}_N$ by equation:

$$\mathbf{z}_{j-1} = \mathbf{y}_{j-1} + \mathbf{R}^\top \mathbf{P}_j \mathbf{z}_j + g(\mathbf{R}^\top \mathbf{z}_j, \boldsymbol{\theta}_j^*), \quad (3)$$

where \mathbf{R}^\top is the transpose of the restriction operator. The difference with an autoencoder is the term \mathbf{y}_{j-1} , which is the skip connection in the network that adds intermediate results from the downscale arm to the upscale arm. To obtain symmetry for the two branches of the net, we choose the parameters $\boldsymbol{\theta}^*$ of the upgoing net to be the adjoints of the downgoing ones. In particular, we use the transpose of the convolutions of the weights

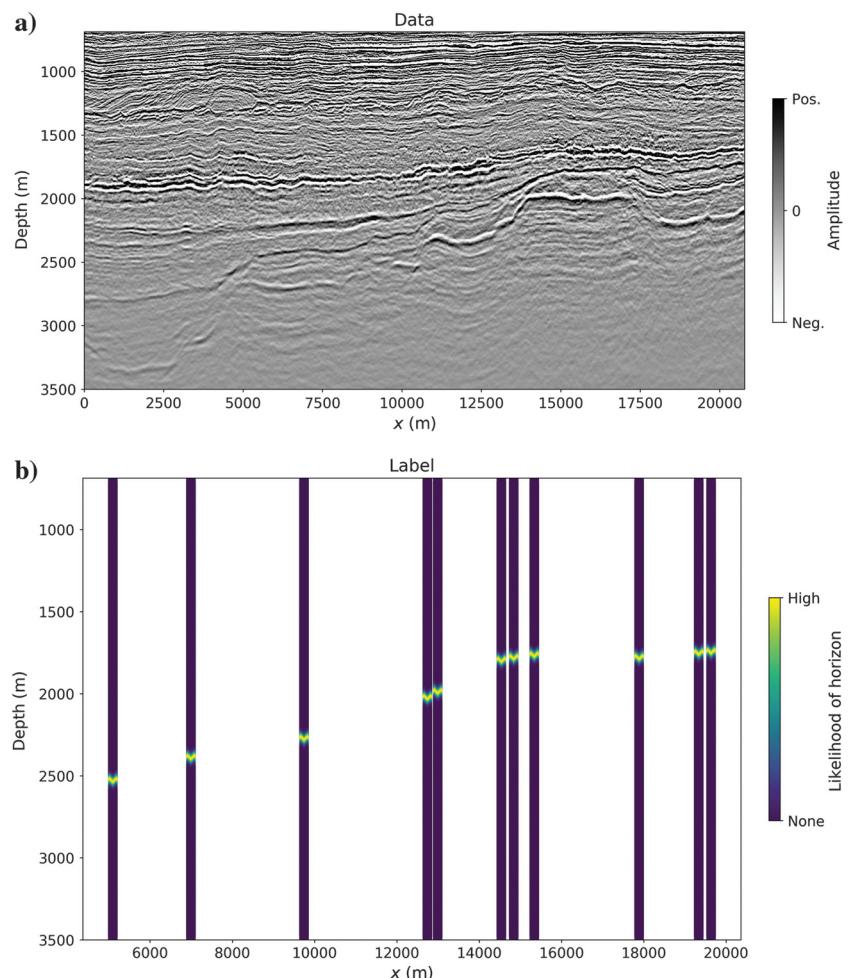


Figure 5. (a) The full seismic image is the input for the network. (b) The label is used in combination with the projected loss function that only computes the loss and gradient in the columns that have nonzero values. The nonzero label values are Gaussian kernels centered at the provided horizon x - y - z locations — the white areas do not have associated label values and are not used for training.

that are downgoing. Figure 3 illustrates the network architecture corresponding to equation 3. Table 1 lists the details of this network design that we use for the field-data examples.

The combination of low- and high-resolution features allows the network to communicate between different scales, which is crucial for our application in which reflectors have local and global features. Note that we do not use fully connected layers. We only work with convolutional kernels and restriction/upsampling operators, which are defined for any input data size. Our network can, therefore, be trained using seismic images and labels of varying size.

Partial loss function

Consider a network $f(\mathbf{y}, \boldsymbol{\theta}) : \mathbb{R}^N \rightarrow \mathbb{R}^N$ that maps from (vectorized) seismic images $\mathbf{y} \in \mathbb{R}^N$ of size $N = n_1 \times n_2$ to images of the same size that show the likelihood of the horizon. The network weights $\boldsymbol{\theta}$ are convolutional kernels, biases, and also include the vector $\mathbf{w} \in \mathbb{R}^{n_{\text{chan}} \times 1}$ (in classification settings, this would be a matrix). When we forward-propagate a seismic image through the network, the last convolutional layer outputs a tensor $\mathbf{X} \in \mathbb{R}^{N \times n_{\text{chan}}}$. The learned \mathbf{w} reduces \mathbf{X} to a vectorized image $\mathbf{s} \in \mathbb{R}^N$. The final network output is thus given by $\mathbf{s} = \mathbf{X}\mathbf{w}$, which is an image that shows the predicted horizon locations. We learn (update) variables $\boldsymbol{\theta}$.

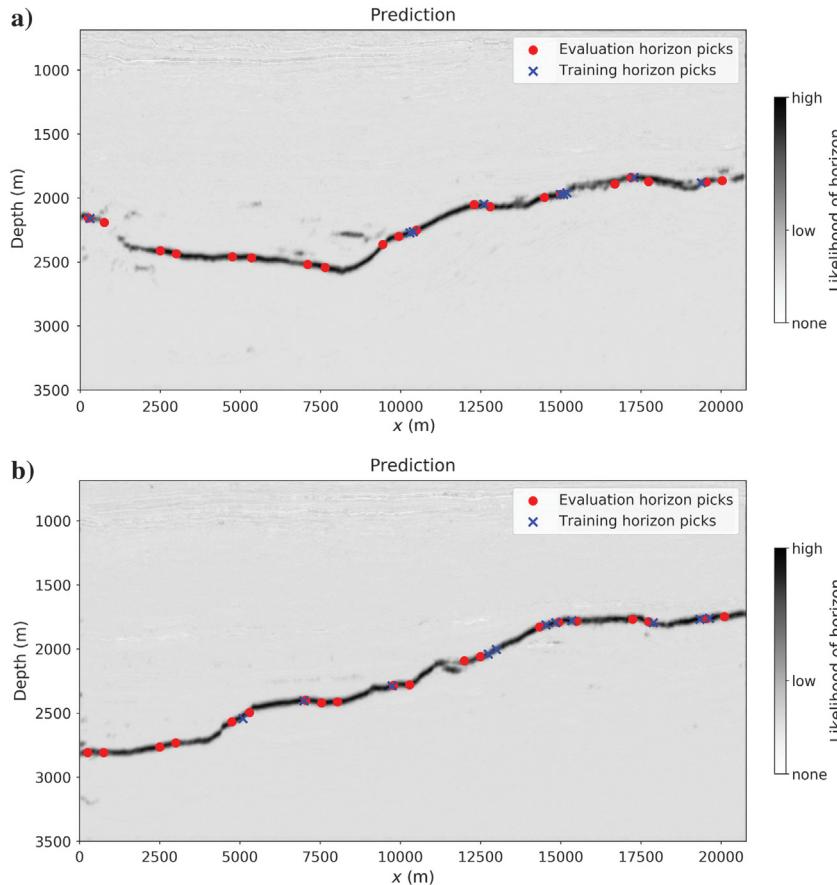


Figure 6. Predicted likelihood of the horizon for two different seismic slices.

The least-squares loss is defined as

$$l(f(\mathbf{y}, \boldsymbol{\theta}), \mathbf{c}) = \frac{1}{2} \|f(\mathbf{y}, \boldsymbol{\theta}) - \mathbf{c}\|_2^2, \quad (4)$$

where $\mathbf{c} \in \mathbb{R}^N$ is a vectorized label image. This is a separable function, so we can compute a partial loss over a selection of pixels as

$$l_\Omega(f(\mathbf{y}, \boldsymbol{\theta}), \mathbf{c}) = \frac{1}{2} \sum_{i \in \Omega} (f(\mathbf{y}, \boldsymbol{\theta})_i - \mathbf{c}_i)^2, \quad (5)$$

where Ω is the set of pixel indices in which we have labels. Note that this is subsampling of the prediction, $f(\mathbf{y}, \boldsymbol{\theta})$, which requires a full forward pass through the network. The gradient computation uses the loss at the points in Ω only.

Another interpretation of the partial loss that is more common in geophysical literature is in terms of a projection (see, e.g., Haber et al., 2000). We define $\mathbf{Q} \in \mathbb{R}^{n_{\text{samp}} \times N}$ as a projection matrix that projects onto the points in Ω ; i.e., \mathbf{Q} contains a subset of the rows of the identity matrix. We can then write the partial loss in equation 5 as

$$l_\mathbf{Q}(f(\mathbf{y}, \boldsymbol{\theta}), \mathbf{c}) = \frac{1}{2} \|\mathbf{Q}f(\mathbf{y}, \boldsymbol{\theta}) - \mathbf{c}_\mathbf{Q}\|_2^2, \quad (6)$$

where $\mathbf{c}_\mathbf{Q}$ are the partial labels. In this work, we use the ℓ_1 -norm, which is separable as well. The partial or projected ℓ_1 loss is defined as

$$l_\Omega(f(\mathbf{y}, \boldsymbol{\theta}), \mathbf{c}) = \sum_{i \in \Omega} |f(\mathbf{y}, \boldsymbol{\theta})_i - \mathbf{c}_i|. \quad (7)$$

The partial loss function enables us to train on partially known labels, as long as we know which pixels they are associated with, without labeling the whole seismic volume. The losses presented in this section share some similarity with point-wise annotations concepts in frameworks for object detection and recognition (Bearman et al., 2016).

Stochastic optimization using a projected loss function

Many neural network training strategies for classification of data sets that contain a large number of small ($\leq 128 \times 128$) images use random mini-batch stochastic gradient descent (SGD). At each iteration of SGD, the algorithm computes a gradient based on a small number of images and labels. For our applications, we typically only have access to a small number (≤ 100) of large images/labels ($\geq 1000 \times 1000$), sometimes even

only a single image. If we were to compute a gradient based on a single image/label, there is only a single gradient and no stochastic effects. It has long been observed that full gradient methods are not competitive to randomized and stochastic gradient-based optimization algorithms for nonconvex optimization in machine learning, particularly neural networks (Bottou and Bousquet, 2008). The subsampling of the image and label pixels as proposed in the previous section provides us with a stochastic optimization algorithm by using a random subset of the points in Ω at each iteration.

Rebalancing for the projected loss

Seismic horizon detection problems have labels in which most pixels have a value equal to zero, which means that there is no probability that the horizon is located at that pixel. In each column, there are only a few entries that have a nonzero label value. This imbalance (approximately 30 times more zero labels than nonzero in our numerical examples) leads to slower training and low-quality predictions. To mitigate these issues, we apply binary rebalancing and use an equal number of zero and nonzero pixel values.

In a randomized stochastic optimization algorithm, at each iteration, we draw n_{samp} randomly selected samples out of the set of known label pixels Ω . Binary rebalancing means there are $n_{\text{samp}}/2$ samples that have a label value equal to zero — denoted by the set \mathcal{B}_1 — and $n_{\text{samp}}/2$ samples that correspond to a nonzero label value — denoted by the set \mathcal{B}_2 . The union of the two subsets is $\mathcal{A} = \mathcal{B}_1 \cup \mathcal{B}_2$.

We summarize the stochastic optimization algorithm for training neural networks using a partial loss function in combination with binary sample rebalancing in Algorithm 1. The numerical examples show that balancing of zero and nonzero labels result in better predictions.

Field example of horizon tracking using neural networks

Our data consist of seismic images that are models of the reflectivity of the earth. The amplitude in the data relates to the elastic impedance contrast between the geologic layers. The raw data have been stacked or migrated into a large 3D model. We work with 2D slices. The labels are a combination of hand picking and algorithm-assisted horizon

tracking. The data and picks were previously generated as part of a commercial exploration project by an external company.

We present results for extrapolation by inline continuation, as well as interpolation from scattered horizon picks. The results also show the benefit of balancing the number of zero and nonzero labeled values in each random batch.

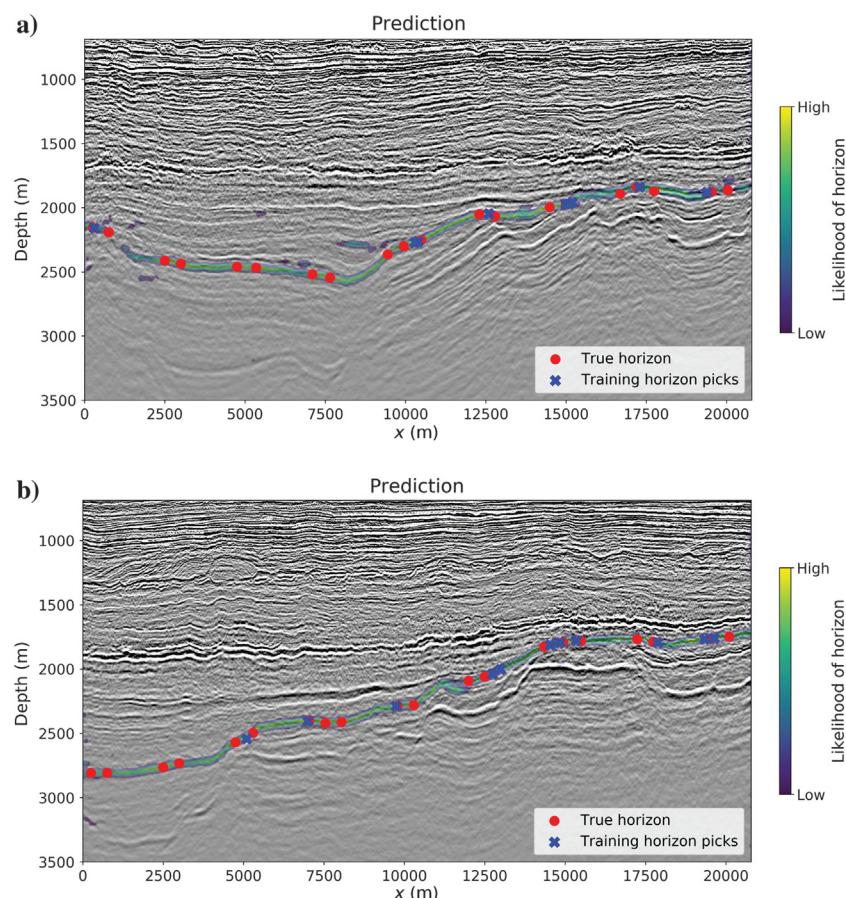


Figure 7. Predicted likelihood of the interface for two different slices overlaid on the seismic data.

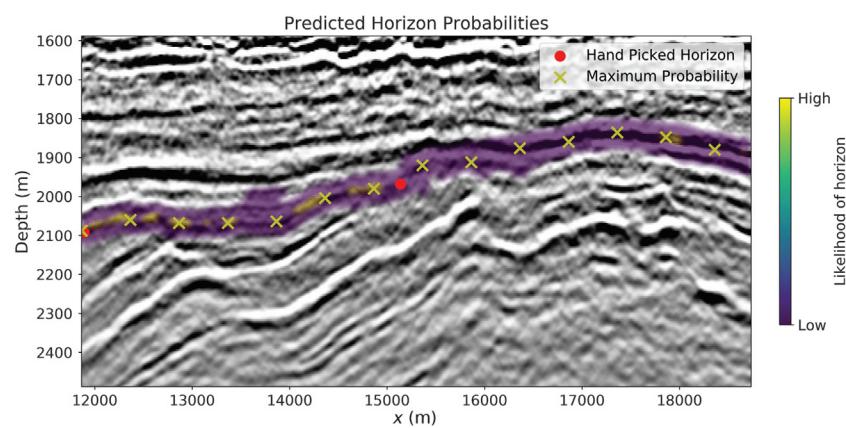


Figure 8. Magnified version of Figure 7b.

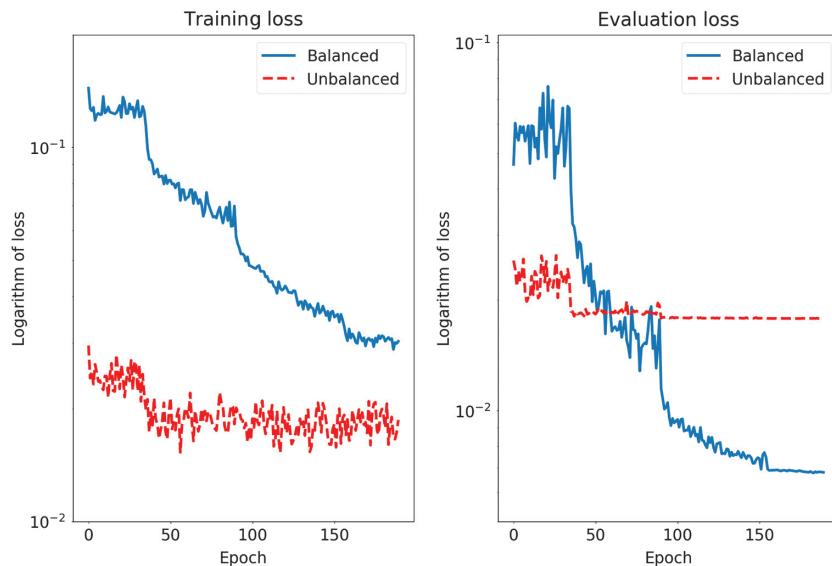


Figure 9. Training and evaluation loss for training with and without rebalancing the number of zero and nonzero label samples. The training loss is much smaller without rebalancing because most of the label samples have a value equal to zero. The evaluation loss is measured over all available labels, and it is evident that rebalancing leads to better evaluation performance.

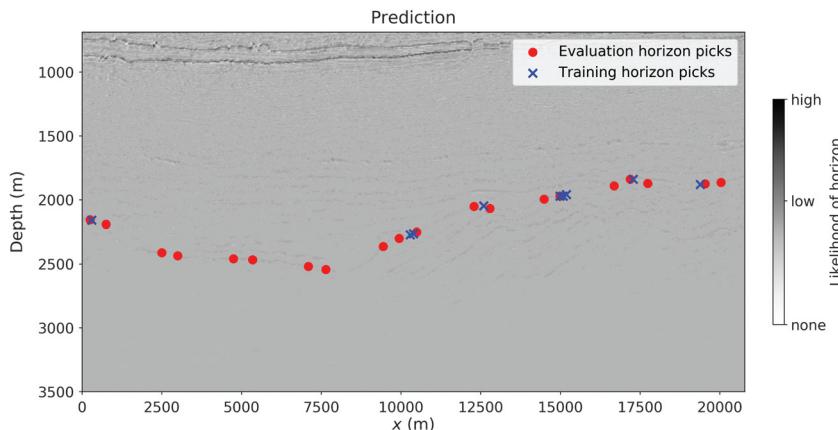
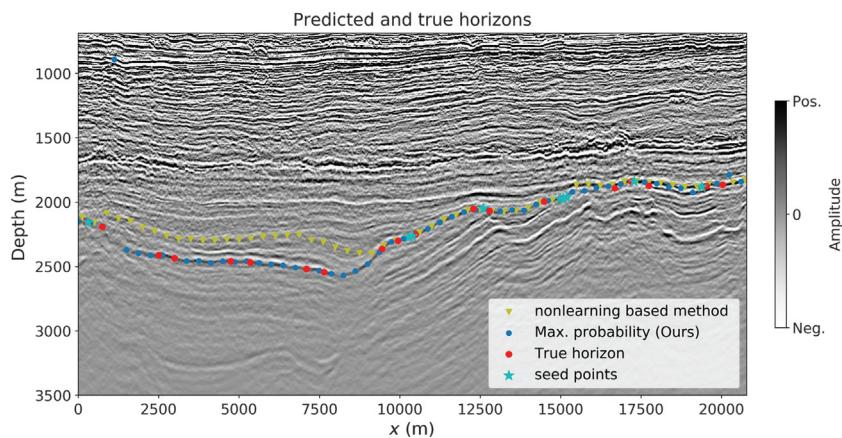


Figure 10. Prediction when the network is trained without rebalancing the number of zero and nonzero label values. This is the prediction only, not overlaid on a seismic image. The figure also shows the red dots and the blue crosses that represent the training and evaluation horizon picks and not the prediction.

Figure 11. Comparing the results from a non-learning-based method that uses least-squares local slopes with multigrid correlations, with our method in which we display the maximum predicted probability only (see Figure 7a).



We use the same network design for both examples and train the two networks using the projected ℓ_1 -loss as defined in equation 7. The initialization of the network parameters is random.

Horizon interpolation

Hand picking horizon locations is a time-consuming task. Many interpreted horizons have sparse spatial sampling as a result. In this case, we want to interpolate the picks to obtain continuous horizon surfaces, as shown in Figure 4. To be able to train on just a few labeled points in large images, we need a loss function that measures the loss at the labeled points only, but not at the other parts of the image. For a seismic horizon image, this means that we compute the loss based on the columns that have a horizon label (Figure 5b). In each of these columns, there is one horizon location; the other column entries are labels that indicate there is no horizon. The columns in which we do not have any labeled information are excluded from training by the ℓ_1 projected loss function as defined in equation 7; the network trains on all seismic data but only part of the label images.

The training data (Figure 5a) are full 2D inline slices of size 704×1664 pixels ($\approx 4000 \times 40,000$ m), without windowing or splitting into patches. The label images are known only at on average nine random locations per slice, provided by an industrial partner. We convolve the horizon location with a Gaussian kernel (in the vertical direction only) to assign an uncertainty to the hand-picked location. All other entries in the same column have a value equal to zero,

which indicates the horizon does not occur at that location (see Figure 5b).

Training starts with 90 epochs and a learning rate of 0.1. During each epoch, we process the seismic and label image pairs in random order. Every iteration of an epoch thus uses a single seismic image and a single label image. Out of the approximately $9 \times 704 = 6336$ known label pixels, we randomly select 150 samples per iteration. As a result, not all label pixels are shown to the network during each epoch. We distribute the samples between zero and nonzero values equally. Note that the Gaussian kernel that we convolve with the horizon x - y - z locations has a width of 31 pixels, so there are on average 9×31 nonzero label values per image. Train-

ing continues with another 65 epochs, and the learning rate is reduced by a factor of 10. The third and last training stage is 35 epochs, where we again reduce the learning rate by a factor of 10.

Figure 6a and 6b displays the prediction for two slices. Figure 7a and 7b shows the same information using color coding for the predicted probability and overlaid on the seismic image. The magnified version in Figure 8 shows more details.

The results in Figures 6a and 7a show excellent predictions compared to a nonlearning-based method. The network output displays the probability of a horizon directly, and no additional postprocessing was applied. The average of nine picks per slice is not a lower limit or a recommended number. Getting good predictions using fewer picks is possible. We point out that we could train more to reduce the evaluation loss (see Figure 9). We also did not use any data augmentation, which could benefit the training in the case of fewer label points.

With regard to the balancing procedure outlined in an earlier section, the loss function logs in Figure 9 clearly show that not balancing the number of zero and nonzero label points during each SGD iteration leads to a worse evaluation loss. Figure 10 shows a prediction from training without balancing, which is not close to the desired output in any way.

To provide a better understanding of the quality of the results, we compare the results presented so far, to methods for horizon tracking that are not based on learning. We want to emphasize that the following comparison is not one-to-one. We define the training picks (seed points) for the horizon as a distributed quantity that reflects the probability of the horizon being at that location, and the results provided by our network are of

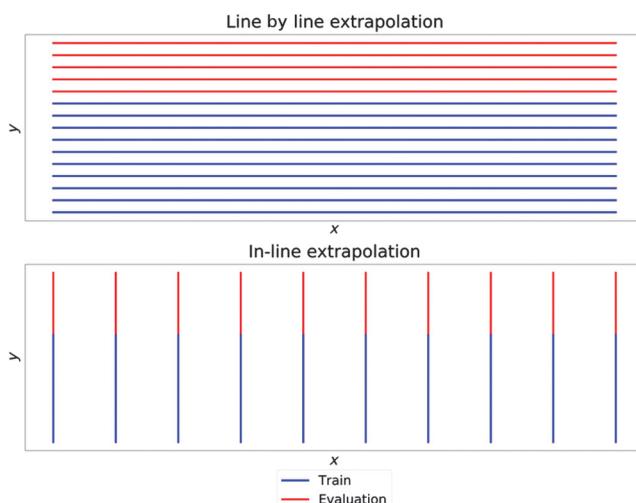


Figure 12. Plan view of two different types of horizon extrapolation.

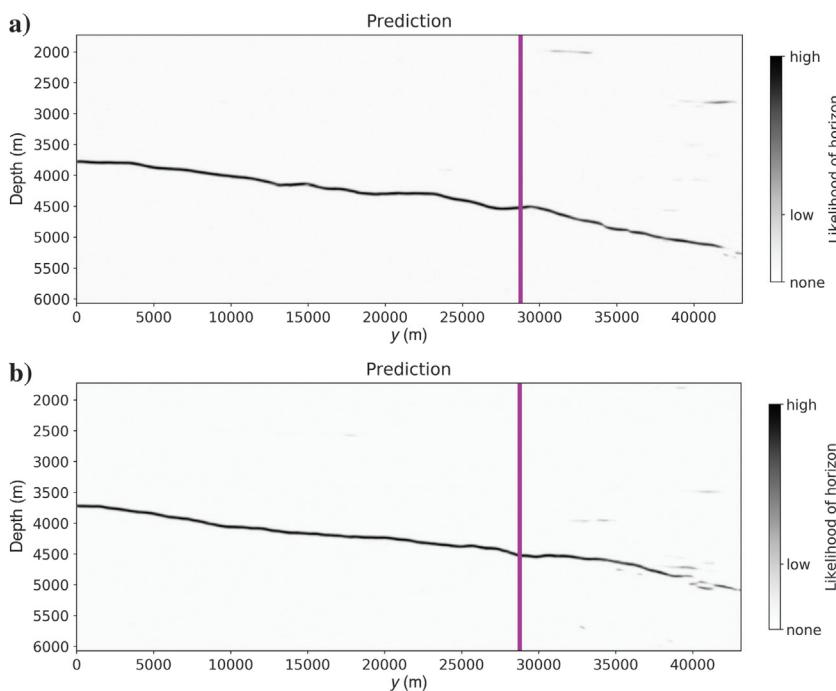


Figure 13. Neural-network predictions for horizon extrapolation based for two seismic images. The part of the figures left of the vertical line is the prediction in which we have training labels. There were no labels on the right of the vertical line.

the same type. Methods not based on learning mostly define one or a few seed points, from which the algorithms generate the horizon as a single line without a sense of uncertainty.

We compare our results with the three methods implemented by the software package (Wu, 2019) that corresponds to the paper by Wu and Fomel (2018). There are three methods implemented that track a horizon: (1) predictive horizons from a single seed point with local reflection slopes from a structure tensor, (2) least-squares fitting of horizons to local slopes from a structure tensor and multiple control points, and (3) least-squares fitting of horizons to local slopes with additional information from slopes on multiple grids. Method 3 can use multiple control points and provide excellent results. In Figure 11, we compare our results from Figure 7a with method 3 from Wu and Fomel (2018). We observe that both methods perform well in the vicinity of the horizon picks. However, in the area with a large gap in the seed points, only our method tracks the correct horizon. It is exactly this setting in which we need multiresolution neural networks: low-quality seismic image with large gaps in the labels. Because our network operates on multiple scales simultaneously, it can learn to predict small and large-scale features. Moreover, learning-based methods also benefit from the other slices and their corresponding horizon picks in the large 3D volume, something that nonlearning methods cannot easily do.

Horizon extrapolation

Points that indicate the x - y - z locations of a horizon are also called horizon picks. Given a collection of picks in an area, we can try to extrapolate the horizon away

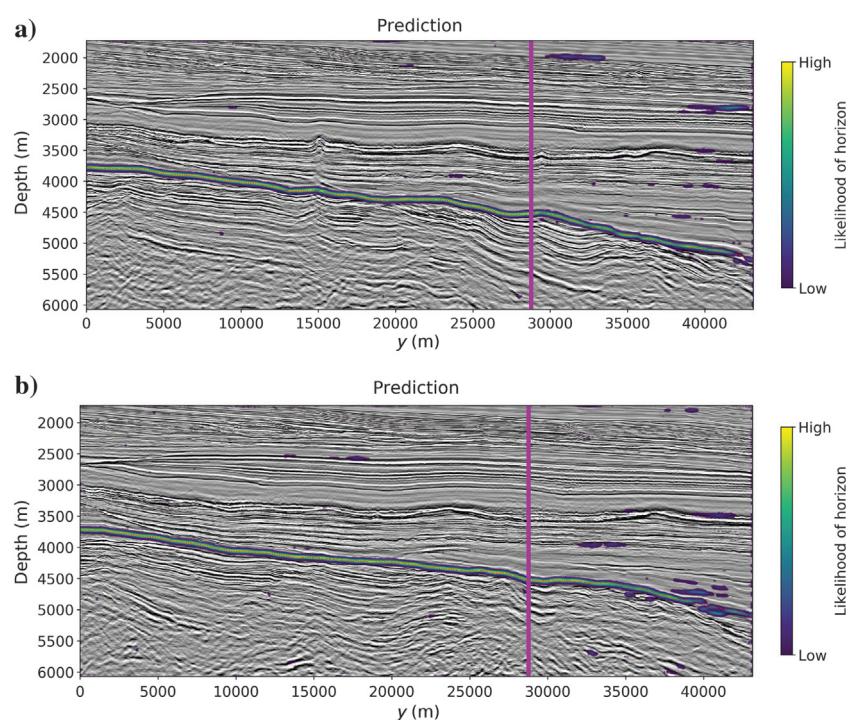
Figure 14. Color-coded predictions overlaid on the seismic data for two slices of training and evaluation data. The part of the figures left of the vertical line is the prediction on training data; on the right is the prediction on the evaluation data.

from the known locations. Much historical industrial work produced large quantities of horizon picks that we can use for training. A potential challenge is that the extrapolation can be in areas with different geology than where the training picks are.

There are multiple types of extrapolation, two of which are shown in Figure 12. Perhaps most similar to standard classifications or segmentation tasks on data sets containing many small images (e.g., Modified National Institute of Standards and Technology database [MNIST], Canadian Institute For Advanced Research [CIFAR]), is to train on one set of images, then apply the trained network and classifier on another test set of images. We call this line-by-line or slice-by-slice learning. A slice refers to a 2D slice from a 3D tensor. The second strategy extrapolates a horizon inline. The training procedure sees the full data (seismic image), but the label is only partially known.

Line-by-line versus inline extrapolation

We provide some insight about which of the two types of horizon extrapolation is preferable. First, it is important to realize that the interpretation of seismic images is different from other problems, such as segmentation of images from video for self-driving vehicles. That application has prerecorded video/images along with segmented labels available for training. The testing data arrive in real time, and segmentation needs to happen in a short amount of time. In our case, the complete seismic 3D volume is available at the time of training. It is only the labels that are incomplete. Therefore, we would like to use all training data, together with the labels corresponding to a part of the training data. Inline extrapolation keeps several slices separately for



testing, so the network never has access to those seismic images. Contrary, inline extrapolation trains on the full seismic slices, but it sees only part of the labels (see Figure 2b). Because we will use a deep neural network with multiple convolutional layers and subsampling/upsampling stages, the data from the area without labels will influence the prediction in the area where we do have labels. For this reason, inline extrapolation has the capability to use all of the data, and we focus on this method in the remainder of this paper.

For training, we use just 24 images of size 1088×2816 pixels. There are three training stages. We start with 40 epochs and a learning rate of 0.1, fol-

lowed by 30 epochs with the learning rate reduced by a factor of 10. The last stage is another 20 epochs where we reduce the learning rate by another factor of 10.

Figure 2b shows an example of the labels and data. The evaluation part of each data image is approximately 1/3, which is the extrapolation distance of interest to an industrial partner. In Figure 13b, we display the predictions for two slices. The prediction on the right of the vertical line shows that we generally predict a continuous line, but it is difficult to see how accurate the prediction is. By color coding the predicted likelihood of the horizon in Figure 14b, we see that the incorrectly highlighted areas have a much lower probability than the correct horizon locations. We also observe that our prediction on the training part is almost perfect. Figure 15b shows a magnified section that better illustrates the relation between the predicted probability and the seismic image.

Conclusion

In this work, we provided a new look at the problem of detecting horizons in seismic images using neural networks. Specifically, we addressed extrapolation away from previously interpreted horizons, as well as the interpolation of a small number of scattered hand-picked horizon locations. The proposed networks, loss function, and learning strategies to overcome issues that limit the success of automatic interpretation using neural networks. We use deep networks with a multiresolution structure to train on a small number of large seismic images that take large-scale geologic structures into account, in the sense that information propagates over long distances on multiple scales. This is not directly possible using standard network-based learning methods that train on small image patches. We proposed a projected loss function that enables training on label images with only a few annotated pixels. Generating such labels is easier and faster than working with conventional label images that need complete labeling of a full image or patch. The standard practice of splitting data and labels into training and test sets is no longer necessary when we train with the projected loss. In seismic imaging, we have access to all data during training. It is the labels that are incomplete. Our networks train on all available seismic images, and we compute the loss and gradient based on a small number of known label pixels. The data in areas without corresponding labels are still seen by the network, and because the network has multiple layers on multiple resolutions, the information influences the predictions and misfit at locations where we do have horizon picks. Application of the proposed network, loss function, and learning strategy to horizon extrapolation and interpolation showed that our methods provide accurate predictions and uncertainty estimates close to and farther away from known horizon locations. Our experiments so far were restricted to sedimentary geologic settings in the North Sea and the Sea of Ireland. The proposed methods make automatic horizon detection possible using fewer horizon picks and take all available seismic data into account.

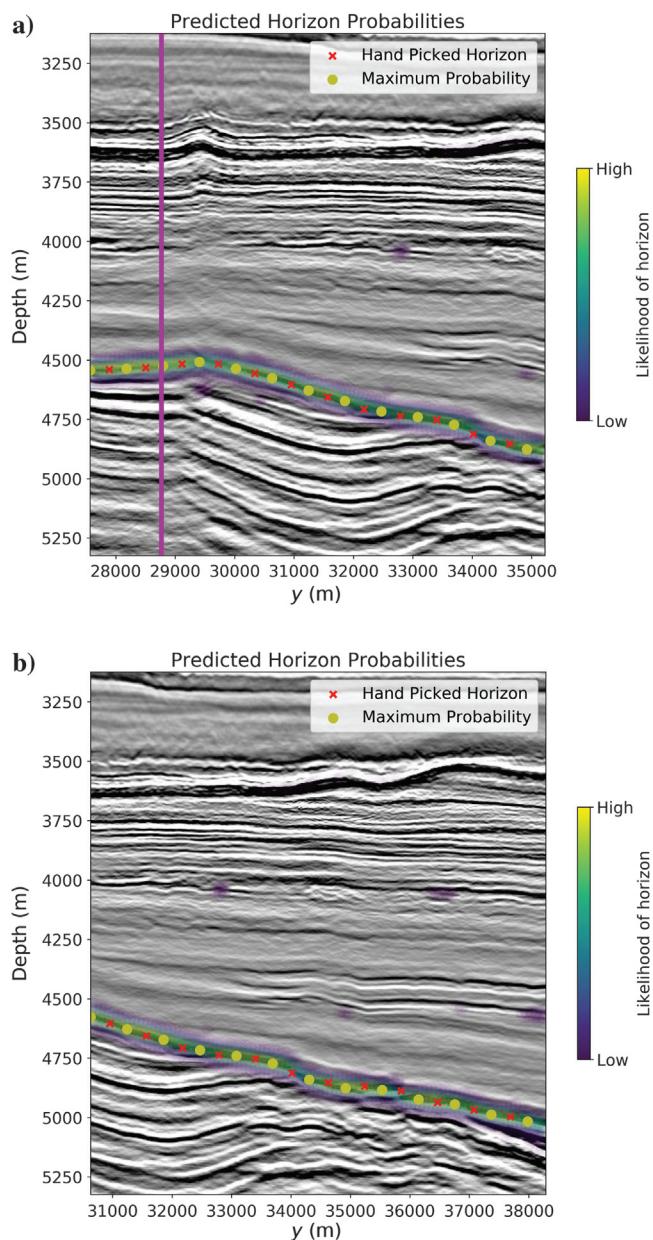


Figure 15. Magnified sections of Figure 14b. Figure shows the predicted probabilities, maximum predicted probability, and the horizon locations that are hand picked by an industrial partner.

Data and materials availability

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

References

- Alaudah, Y., S. Gao, and G. AlRegib, 2018, Learning to label seismic structures with deconvolution networks and weak labels: 88th Annual International Meeting, SEG, Expanded Abstracts, 2121–2125, doi: [10.1190/segam2018-2997865.1](https://doi.org/10.1190/segam2018-2997865.1).
- Alberts, P., M. Warner, and D. Lister, 2000, Artificial neural networks for simultaneous multi horizon tracking across discontinuities: 75th Annual International Meeting, SEG, Expanded Abstracts, 651–653, doi: [10.1190/1.1816150](https://doi.org/10.1190/1.1816150).
- 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](https://doi.org/10.1190/tle36030208.1).
- Bearman, A., O. Russakovsky, V. Ferrari, and L. Fei-Fei, 2016, What's the point: Semantic segmentation with point supervision: *Computer Vision*, 549–565.
- Bottou, L., and O. Bousquet, 2008, The tradeoffs of large scale learning: *Advances in Neural Information Processing Systems*, **20**, 161–168.
- Briggs, W., V. Henson, and S. McCormick, 2000, *A multigrid tutorial*, 2nd ed.: SIAM.
- Deng, J., W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, 2009, ImageNet: A large-scale hierarchical image database: Presented at the IEEE Conference on Computer Vision and Pattern Recognition.
- Di, H., 2018, Developing a seismic pattern interpretation network (SpiNet) for automated seismic interpretation: arXiv preprint arXiv:1810.08517.
- Fomel, S., 2010, Predictive painting of 3D seismic volumes: *Geophysics*, **75**, no. 4, A25–A30, doi: [10.1190/1.3453847](https://doi.org/10.1190/1.3453847).
- Haber, E., U. M. Ascher, and D. Oldenburg, 2000, On optimization techniques for solving nonlinear inverse problems: *Inverse Problems*, **16**, 1263–1280, doi: [10.1088/0266-5611/16/5/309](https://doi.org/10.1088/0266-5611/16/5/309).
- Harrigan, E., J. R. Kroh, W. A. Sandham, and T. S. Durrani, 1992, Seismic horizon picking using an artificial neural network: IEEE International Conference on Acoustics, Speech, and Signal Processing, 105–108.
- He, K., X. Zhang, S. Ren, and J. Sun, 2015, Deep residual learning for image recognition: CoRR, abs/1512.03385.
- Huang, K., 1997, Hopfield neural network for seismic horizon picking: 75th Annual International Meeting, SEG, Expanded Abstracts, 562–565, doi: [10.1190/1.1885963](https://doi.org/10.1190/1.1885963).
- Huang, K.-Y., C.-H. Chang, W.-S. Hsieh, S.-C. Hsieh, L. K. Wang, and F.-J. Tsai, 2005, Cellular neural network for seismic horizon picking: 9th International Workshop on Cellular Neural Networks and Their Applications, 219–222.
- Jégou, S., M. Drozdzal, D. Vazquez, A. Romero, and Y. Bengio, 2017, The one hundred layers Tiramisu: Fully convolutional densenets for semantic segmentation: IEEE Conference on Computer Vision and Pattern Recognition Workshops, 1175–1183.
- Krizhevsky, A., and G. Hinton, 2009, Learning multiple layers of features from tiny images: Technical report, Citeseer.
- Kusuma, T., and B. C. Fish, 1993, Toward more robust neural network first break and horizon pickers: 75th Annual International Meeting, SEG, Expanded Abstracts, 238–241, doi: [10.1190/1.1822449](https://doi.org/10.1190/1.1822449).
- Labrunye, E., C. Winkler, C. Borgese, J. Mallet, and S. Jayr, 2009, New 3D flattened space for seismic interpretation: 79th Annual International Meeting, SEG, Expanded Abstracts, 1132–1136, doi: [10.1190/1.3255052](https://doi.org/10.1190/1.3255052).
- Leggett, M., W. A. Sandham, and T. S. Durrani, 2003, Automated 3-D horizon tracking and seismic classification using artificial neural networks: Springer, 31–44.
- Li, L., G. Ma, and X. Du, 2012, New method of horizon recognition in seismic data: *IEEE Geoscience and Remote Sensing Letters*, **9**, 1066–1068, doi: [10.1109/LGRS.2012.2190039](https://doi.org/10.1109/LGRS.2012.2190039).
- Lin, G., A. Milan, C. Shen, and I. Reid, 2017, Refinenet: Multi-path refinement networks for high-resolution semantic segmentation: IEEE Conference on Computer Vision and Pattern Recognition, 5168–5177.
- Liu, X., P. Xue, and Y. Li, 1989, Neural network method for tracing seismic events: 75th Annual International Meeting, SEG, Expanded Abstracts, 716–718, doi: [10.1190/1.1889749](https://doi.org/10.1190/1.1889749).
- Lomask, J., A. Guittot, S. Fomel, J. Claerbout, and A. A. Valenciano, 2006, Flattening without picking: *Geophysics*, **71**, no. 4, P13–P20, doi: [10.1190/1.2210848](https://doi.org/10.1190/1.2210848).
- Luo, S., and D. Hale, 2013, Unfaulting and unfolding 3D seismic images: *Geophysics*, **78**, no. 4, O45–O56, doi: [10.1190/geo2012-0350.1](https://doi.org/10.1190/geo2012-0350.1).
- Mallet, J.-L., 2004, Space-time mathematical framework for sedimentary geology: *Mathematical Geology*, **36**, 1–32, doi: [10.1023/B:MATG.0000016228.75495.7c](https://doi.org/10.1023/B:MATG.0000016228.75495.7c).
- Mallet, J.-R., S. Jayr, and P. Neri, 2010, New modelling technology delivers consistency: *First Break*, **28**.
- Meldahl, P., R. Heggland, B. Bril, and P. de Groot, 1999, The chimney cube, an example of semiautomated detection of seismic objects by directive attributes and neural networks — Part 1: Methodology: 69th Annual International Meeting, SEG, Expanded Abstracts, 931–934, doi: [10.1190/1.1821262](https://doi.org/10.1190/1.1821262).
- Meldahl, P., R. Heggland, B. Bril, and P. de Groot, 2001, Identifying faults and gas chimneys using multiatributes and neural networks: *The Leading Edge*, **20**, 474–482, doi: [10.1190/1.1438976](https://doi.org/10.1190/1.1438976).
- Parks, D., 2010, Seismic image flattening as a linear inverse problem: M.Sc. thesis, Colorado School of Mines.
- Poulton, M. M., 2002, Neural networks as an intelligence amplification tool: A review of applications: *Geophysics*, **67**, 979–993, doi: [10.1190/1.1484539](https://doi.org/10.1190/1.1484539).
- Ronneberger, O., P. Fischer, and T. Brox, 2015, U-net: Convolutional networks for biomedical image segmentation: *Medical Image Computing and Computer-Assisted Intervention*, 234–241.

- Shah, S., P. Ghosh, L. S. Davis, and T. Goldstein, 2018, Stacked u-nets: A no-frills approach to natural image segmentation: arXiv preprint arXiv:1804.10343.
- 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](https://doi.org/10.1190/int-2018-0235.1).
- Stark, T. J., 2005, Generation of a 3D seismic “wheeler diagram” from a high resolution age volume: 75th Annual International Meeting, SEG, Expanded Abstracts, 782–785, doi: [10.1190/1.2148275](https://doi.org/10.1190/1.2148275).
- Tingdahl, K. M., and M. De Rooij, 2005, Semi-automatic detection of faults in 3D seismic data: Geophysical Prospecting, **53**, 533–542, doi: [10.1111/j.1365-2478.2005.00489.x](https://doi.org/10.1111/j.1365-2478.2005.00489.x).
- Veezhinathan, J., F. Kemp, and J. Threet, 1993, A hybrid of neural net and branch and bound techniques for seismic horizon tracking: Proceedings of the ACM/SIGAPP Symposium on Applied Computing: States of the Art and Practice, 173–178.
- Waldegaard, A. U., A. C. Jensen, L.-J. Gelius, and A. H. S. Solberg, 2018, Convolutional neural networks for automated seismic interpretation: The Leading Edge, **37**, 529–537, doi: [10.1190/tle37070529.1](https://doi.org/10.1190/tle37070529.1).
- Wu, X., 2019, University of Texas at Austin, <https://github.com/xinwucwp/mhe>, accessed 5 March 2019.
- Wu, H., and B. Zhang, 2018, A deep convolutional encoder-decoder neural network in assisting seismic horizon tracking: arXiv preprint arXiv: 1804.06814.
- Wu, X., and S. Fomel, 2018, Least-squares horizons with local slopes and multigrid correlations: Geophysics, **83**, no. 4, IM29–IM40, doi: [10.1190/geo2017-0830.1](https://doi.org/10.1190/geo2017-0830.1).
- Wu, X., and D. Hale, 2015a, 3D seismic image processing for unconformities: Geophysics, **80**, no. 2, IM35–IM44, doi: [10.1190/geo2014-0323.1](https://doi.org/10.1190/geo2014-0323.1).
- Wu, X., and D. Hale, 2015b, Horizon volumes with interpreted constraints: Geophysics, **80**, no. 2, IM21–IM33, doi: [10.1190/geo2014-0212.1](https://doi.org/10.1190/geo2014-0212.1).
- Wu, X., and D. Hale, 2016, Automatically interpreting all faults, unconformities, and horizons from 3D seismic images: Interpretation, **4**, no. 2, T227–T237, doi: [10.1190/INT-2015-0160.1](https://doi.org/10.1190/INT-2015-0160.1).
- Wu, X., S. Luo, and D. Hale, 2016, Moving faults while unfaulting 3D seismic images: Geophysics, **81**, no. 2, IM25–IM33, doi: [10.1190/geo2015-0381.1](https://doi.org/10.1190/geo2015-0381.1).
- Wu, X., and G. Zhong, 2012, Generating a relative geologic time volume by 3D graph-cut phase unwrapping method with horizon and unconformity constraints: Geophysics, **77**, no. 4, O21–O34, doi: [10.1190/geo2011-0351.1](https://doi.org/10.1190/geo2011-0351.1).
- 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](https://doi.org/10.1190/segam2018-2997085.1).

Biographies and photographs of the authors are not available.