Seismic Facies Segmentation Using Deep Learning*

Daniel Chevitarese[1], Daniela Szwarcman[1], Reinaldo Mozart D. Silva[1], and Emilio Vital Brazil[1]

Search and Discovery Article #42286 (2018)**
Posted October 1, 2018

*Adapted from extended abstract prepared for oral presentation given at 2018 AAPG Annual Convention & Exhibition, Salt Lake City, Utah, May 20-23, 2018. Please view related article by the same authors in Search and Discovery Article #42285 (2018).
 DOI:10.1306/42286Chevitarese2018

[1]Research, IBM, Rio de Janeiro, Brazil (sallesd@ibm.com) .

Abstract

Seismic reflection surveying is the most used method to obtain subsurface information in the Oil & Gas exploration industry. with this data, one may determine structural and stratigraphic geometric features and potential hydrocarbon deposit locations. Even though it is paramount, seismic data interpretation is an extremely time-consuming and human-intensive task, mainly due to the ever-larger volumes of seismic data and the geological complexity present in the study areas. In response, computer-aid systems assisting geoscientists to interpret this large and complex data in a faster and more accurate manner represent vital importance for the development of the exploration industry. In fact, deep learning techniques are currently applied in several areas of science to support tasks that were considered human-centered, e.g., image classification, language translation, among others. In this work, we created a neural network topology to assist interpreters in the stratigraphic mapping of seismic images at the pixel-level resolution.

Our recent results have demonstrated that deep learning can distinguish among different facies helping the interpreter to process new seismic images. We also present a network that can classify parts of the image with high accuracy. Here, we extend this structure to create a neural network that can classify the seismic image at pixel level, producing an interpretation mask suggestion. First, we selected a trained convolutional neural network (CNN) with the highest accuracy on the classification task. Then, we modified the final part of the model to produce pixel-wise predictions. Next, we train our neural network using real seismic data from Netherlands North Sea in F3 block. Since these seismic images are somewhat large, we decided to break them into tiles corresponding to 20% of the entire image area. To generate the final prediction, we apply the network throughout the image. In our experiments, we achieved more than 97% of pixel accuracy, and our qualitative results show that the model could produce a mask very close to the actual interpretation.

Introduction

The interpretation of seismic images is one of the essential procedures in the task of locating oil fields. From the analysis of this data, experts may identify structural and stratigraphic features. Additionally, it can indicate physical properties of rock bodies in the subsurface and geographic limits of a target reservoir if combined with other sources of information. However, seismic interpretation and analysis is a time-consuming task that can overload human interpreters as the amount of geophysical information is continually increasing (Randen et al., 2000). In this scenario, researchers have proposed new computer-aided systems to support geoscientists in the interpretation job.

Several authors presented works related to the identification of structures such as salt bodies (Guillen et al., 2015) and faults (Zhang et al., 2014), and also siliciclastic deposits and carbonates Gao, 2011). The efforts to automate parts of the seismic facies analysis procedure rely mostly on computer vision algorithms, as seen in Shafiq et al. (2017); Wang et al. (2016), Mattos et al (2017), and Amin and Deriche (2016). More recently, deep learning techniques have been applied to seismic images classification (Chevitarese et al, 2018a, 2018b). In these studies, the system breaks seismic images into small patches of no more than 50 x 50 pixels, and the model can identify to which facies that patch belongs.

Following the deep learning models approach, we propose a fully convolutional neural network (Danet-FCN) that can distinguish different seismic facies with resolution equal to the pixel size. To train our model, we must provide seismic images along with masks indicating the horizons interpreted by an expert. After the training procedure, the model can classify a seismic image at the pixel level and can generate a suggestion mask for the interpreter. The idea is to create an assistant to optimize and enhance the accuracy of the interpretation procedure, by reducing the number of interpreted images necessary to infer the whole data cube.

This paper is divided into five sections. First, we review some related works. Next, we provide a brief description of the datasets used, followed by the method for creating and training our model. Then, we detail the experiments we performed and discuss the results. Finally, we present conclusions and future works.

Related Work

Here, we present some relevant studies found in the literature. However, to the best of our knowledge, this is the first work that applies a fully convolutional neural network to generate interpretation masks suggestions.

In Shelhamer et al.(2015), the authors propose a fully convolutional neural network architecture (FCN) for semantic segmentation. They adapted modern classification networks, such as AlexNet (Krizhevsky et al, 2012), VGG (Simonyan and Zisserman, 2012), and GoogLeNet (Szegedy et al., 2015) to perform segmentation tasks on classical datasets, e.g., PASCAL-VOC (Everingham et al., 2010). Our architecture was inspired by the FCN presented in Shelhamer et al.(2015).

One of the earliest works to use neural networks for seismic facies classification was presented by West et al. (2002). The authors combine image textural analysis with a probabilistic neural network (PNN) classification to quantitatively map seismic facies in three-dimensional data. Moreover, they extend the methodology to the interpretation of AVO attributes volumes, such as intercept and gradient. Despite the comprehensive investigation, the authors focus only on qualitative results.

Computer vision techniques have also been used to extract features from seismic images and help to automate parts of the seismic facies analysis procedure. Mattos et al. (2017) use local binary patterns (LBP), a type of visual descriptor used for classification in computer vision to extract features from seismic images. Their goal is to assist the interpreter in the task of manually selecting a region-of-interest (ROI) in a given image and searching in the remaining of the cube for the most similar regions.

In the seismic facies analysis context, Song et al. (2017) address issues regarding spatial continuity and seismic noise presence. The authors introduce the regularized fuzzy c-means (RegFCM) algorithm - a clustering technique - for unsupervised facies analysis. The unsupervised analysis does not depend either on geological or well-log information. According to the authors, the method uses the spatial location of seismic attributes as a constraint, which helps to locate boundaries and therefore enhances the continuity of the geological facies. The authors used a synthetic seismic cube simulating a carbonate reservoir in the F3-Block Netherlands dataset in their experiments.

Finally, Chevitarese et al. (2018b) designed deep neural models to classify seismic facies. The authors used patches of seismic images from Penobscot along with geoscientists' interpretation masks as training examples of different facies. The models achieved 97% of classification accuracy, which the authors claim to be the best result for this task reported so far. The present work can be considered as an extension of Chevitarese et al. (2018b), as we designed our architecture based on the best model presented in Chevitarese et al. (2018b).

Datasets

In this section, we describe the two seismic datasets used in our experiments: Penobscot, from Scotia Basin, and Netherlands data, from Central-Graben Basin – both available at Open Seismic Repository (dGB Earth Sciences 2017).

Penobscot cube contains 481 crosslines, and 601 inlines, with dimensions 601×1501 pixels and 481×1501 pixels, respectively. Experts previously interpreted the cube, resulting in a total of eight facies, which will be our classes. Netherlands cube consists of 951 crosslines of size 651×462 pixels, and 651 inlines of size 951×462 pixels. In this case, the expert interpretation resulted in a total of seven different classes. Figure 1(a) shows part of an inline slice from Penobscot and Figure 1(b) part of an inline slice from Netherlands cube, overlaid by their respective interpretation masks.

It is important to notice that we only used inline slices to generate our datasets and that we removed corrupted or poor-quality images from the edges of the cube, as indicated by an expert. After removing these images, we ended up with 459 inlines for Penobscot and 591 inlines for Netherlands data.

Methodology

The method to create and train our model comprises two parts: (1) create and train a feature extractor (FE); (2) append the segmentation block to FE and train the whole model.

Our feature extractor is a convolutional neural network that acts as a filter; it highlights the relevant information from the input image that intensifies the contrast between different textures. We assume that one may identify facies by their textural features as discussed in Mattos et al. (2017).

Neural networks are models that can learn several kinds of tasks. The learning process, which is called training, involves presenting examples of inputs to the network and compares the output it produces to the targets. Then the training mechanism adjusts its parameters to minimize the error between the targets and model's outputs. Since our feature extractor is a neural network, we need to create a specific training task that will make our model behave as mentioned before; it needs to highlight the image parts that are important to identify different seismic textures. We define this training task as a classification: by appending a classifier to our feature extractor, we can train the network to receive examples of seismic textures and predict the facies associated with that image. Figure 2(a) shows the feature extractor and the classifier structures.

Once we have the classifier trained, we can consider that it has learned how to distinguish seismic textures and our feature extractor is now able to highlight the essential parts of the image that will enable the segmentation block to distinguish between facies. At this moment, we can replace the classifier part with the segmentation block, as depicted in Figure 2(b). We then train the new network defining segmentation as our new training task. One can associate the segmentation task to a pixel-wise classification; we want the model to assign a class to each pixel of the input image. In this way, the desired output of the network will be a mask suggestion, that is, an image with the same size as the input, with every pixel classified as belonging to one class.

In both parts, to properly train the models, it is necessary to create two datasets: one for training and one for validation. The validation dataset is used to calculate model's performance during training so that we can select our final model based on these results. To separate a cube into train and validation sets, we followed these steps (Figure 3):

- Split the cube into ten parts.
- Take the first 70% of inline slices of each part; this will be the training set.
- The other 30% of inlines slices of each part will be the validation set.

We should mention that each training task requires a specific dataset preparation, as explained further in the text. For the classification step, we followed the idea presented in Chevitarese et al. (2018a, 2018b), in which the inputs to the network are small parts of seismic images – size 40×40. We must break the seismic images into small tiles because we need to provide examples to the network in which the majority of its area belongs to only one facies. Otherwise, the network would not be able to learn the differences between classes. Another consequence of this is that we cannot have good examples of the thinner layers using the chosen tile size. Therefore, we united some classes in both cubes. Referring to Figure 1, for Penobscot, we joined the orange layer with the dark green layer and the red layer with the yellow one. Similarly, for Netherlands data, we combined the orange layer with the dark green one.

We adopted the same preprocessing procedures described in Chevitarese et al. (2018a), such as quantizing the seismic images into 256 gray levels. Also indicated by Chevitarese et al. (2018a), we allowed 30% of class interference in a tile, that is, at least 70% of the pixels in a tile needs to belong to only one facies so it can be considered an example of that facies.

For the segmentation part, we could, in theory, provide to the network the whole seismic image as input. However, we decided to split the input image into tiles of size 80×120, as the model would increase significantly in the number of parameters and in training time.

Results and Discussion

In all experiments, we trained the networks for 200 epochs and used the same configuration for hyper-parameters such as learning rate, weight decay, and dropout as described in Chevitarese et al. (2018b). One epoch means that we presented all the training examples to the model and adjusted its parameters according to the training function. To prepare and run our experiments faster, we used our deep learning toolbox (Chevitarese et al., 2018b) both for training and testing our models on multiple GPUs. Our tool provides management for different types of environment, such as CPU-only (with or without Intel MKL), GPU and multi-GPU. We ran all the experiments using 8 CPU Intel Xeon, and 4 GPU K80. Table 1 shows the quantitative results of the two datasets.

Figure 4 shows some inline segmentation results from the Penobscot dataset, where different colors refer to different facies, and the white lines are the horizons. Figures 5, 6, and 7 show the qualitative results for the Netherlands cube.

Conclusions

In this work, we presented a technique to interpret seismic data using CNN. Indeed, our results demonstrated that if we have a proper classification, then we can perform a segmentation that is close to a human interpreter. This method has the potential to accelerate seismic interpretation process helping the geoscientists focus on the essential parts of a survey. As future work, we are planning to test the method to different datasets to better understand the technique limitation. We also will apply transfer learning (TL) techniques to use fewer labeled lines. TL approaches the difficulty to find a good starting point to adjust model's parameters for deep learning systems. The idea behind the TL technique is to use previous knowledge obtained from one task in another.

References Cited

Amin, A., and M. Deriche, 2016, Salt-dome detection using a codebook-based learning model: IEEE Geoscience and Remote Sensing Letters, v. 13/11, p. 1636-1640.

Chevitarese, D.S., D. Szwarcman, R.M.G. e Silva, and E.V. Brazil, 2018a, Deep learning applied to seismic facies classification: A methodology for training: EAGE Saint Petersburg International Conference, April 9-12, 2018. Accepted for publication.

Chevitarese, D.S., D. Szwarcman, E.V. Brazil, and B. Zadrozny, 2018b, Efficient classification of seismic textures: 2018 International Joint Conference on Neural Networks (IJCNN), Rio de Janeiro, Brazil, July 8-13, 2018. Accepted for publication.

Everingham, M., L. Van~Gool, C. K. I. Williams, J. Winn, and A. Zisserman, 2010, The Pascal visual object classes (VOC) Challenge: International Journal of Computer Vision, v. 88/2, p. 303-338.

Gao, D., 2011, Latest developments in seismic texture analysis for subsurface structure, facies, and reservoir characterization: A review: *Geophysics*, v. 76/2, W1-W13. Website accessed September 18, 2018, https://library.seg.org/doi/10.1190/1.3553479.

GB Earth Sciences, 2017, Open Seismic Repository. Website accessed September 18, 2018, https://www.opendtect.org/osr/.

Guillen, P., G. Larrazabal, G. Gonzalez, D. Boumber, and R. Vilalta, 2015, Supervised learning to detect salt body: SEG Technical Program Expanded Abstracts 2015, p. 1826-1829.

Krizhevsky, A., I. Sutskever, and G. E. Hinton, 2012, Image net classification with deep convolutional neural networks, *in* F. Pereira, C.J.C. Burges, L. Bottou, and K.Q. Weinberger, editors, Advances in Neural Information Processing Systems 25: Curran Associates, Inc., p. 1097-1105.

Mattos, A.B. Mattos, R.S. Ferreira, R.M.D.G. e. Silva, M. Riva, and E. Vital Brazil, 2017, Assessing texture descriptors for seismic image retrieval: 30th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI), p. 292-299.

Randen, T., C. Signer, E. Monsen, J. Schiaf, et al., 2000, Three dimensional texture attributes for seismic data analysis: SEG Technical Program Expanded Abstracts 2000, p. 668-671.

Shafiq, M.A., Y. Alaudah, G. AlRegib, and M. Deriche, 2011, Phase congruency for image understanding with applications in computational seismic interpretation: 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), p. 1587-1591.

Shelhamer, E., J. Long, and T. Darrell, 2015, Fully convolutional networks for semantic segmentation. Proceedings/CVPR IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 12p. Website accessed September 18, 2018, https://arxiv.org/pdf/1605.06211.pdf].

Simonyan, K., and A. Zisserman, 2015, Very deep convolutional networks for large-scale image recognition: ICLR Conference, 14p. Website accessed September 18, 2018, https://arxiv.org/pdf/1409.1556.pdf.

Song, C., Z. Liu, H. Cai, Y. Wang, X. Li, and G. Hu, 2017, Unsupervised seismic facies analysis with spatial constraints using regularized fuzzy c-means: Journal of Geophysics and Engineering, v. 14/6, p. 1535.

Szegedy, C., W. Lium Y. Jia, P. Sermanet, et al., 2015, Going deeper with convolutions: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 9p.

Wang, Z., Z. Long, and G. AlRegib, 2016, Tensor-based subspace learning for tracking salt-dome boundaries constrained by seismic attributes: 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), p. 1125-1129.

West, B.P., S.R. May, J.E. Eastwood, and C. Rossen, 2002, Interactive seismic facies classification using textural attributes and neural networks: Leading Edge, v. 21/10, p. 1042-1049.

Zhang, C., C. Frogner, M. Araya-Polo, and D. Hohl, 2014, Machine learning based automated fault detection in seismic traces: 76th EAGE Conference and Exhibition 2014.

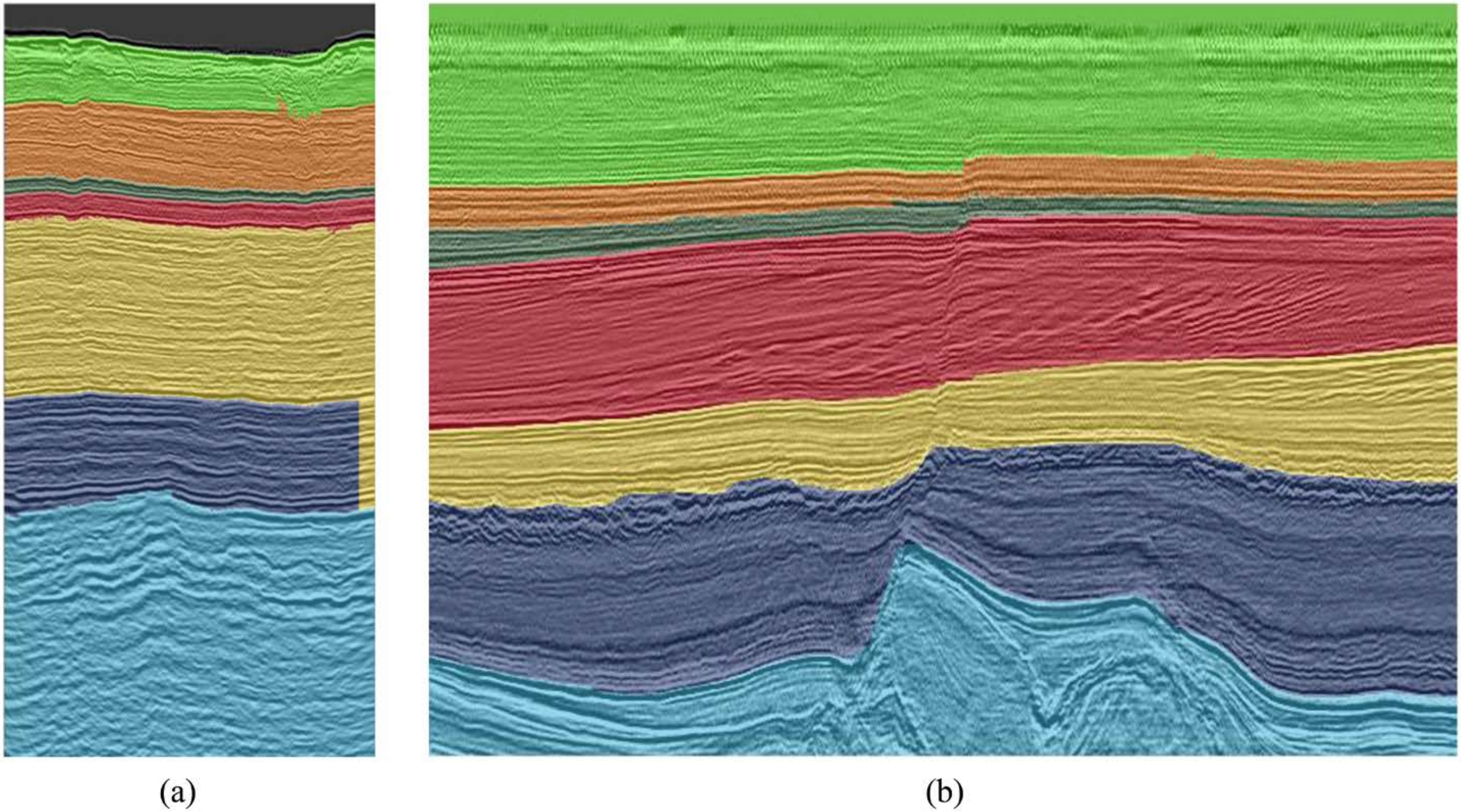


Figure 1. (a) Part of an inline of seismic data: example of Penobscot and its respective mask in colors. (b) Part of an inline of seismic data: example of Netherlands cube with its mask also in colors.

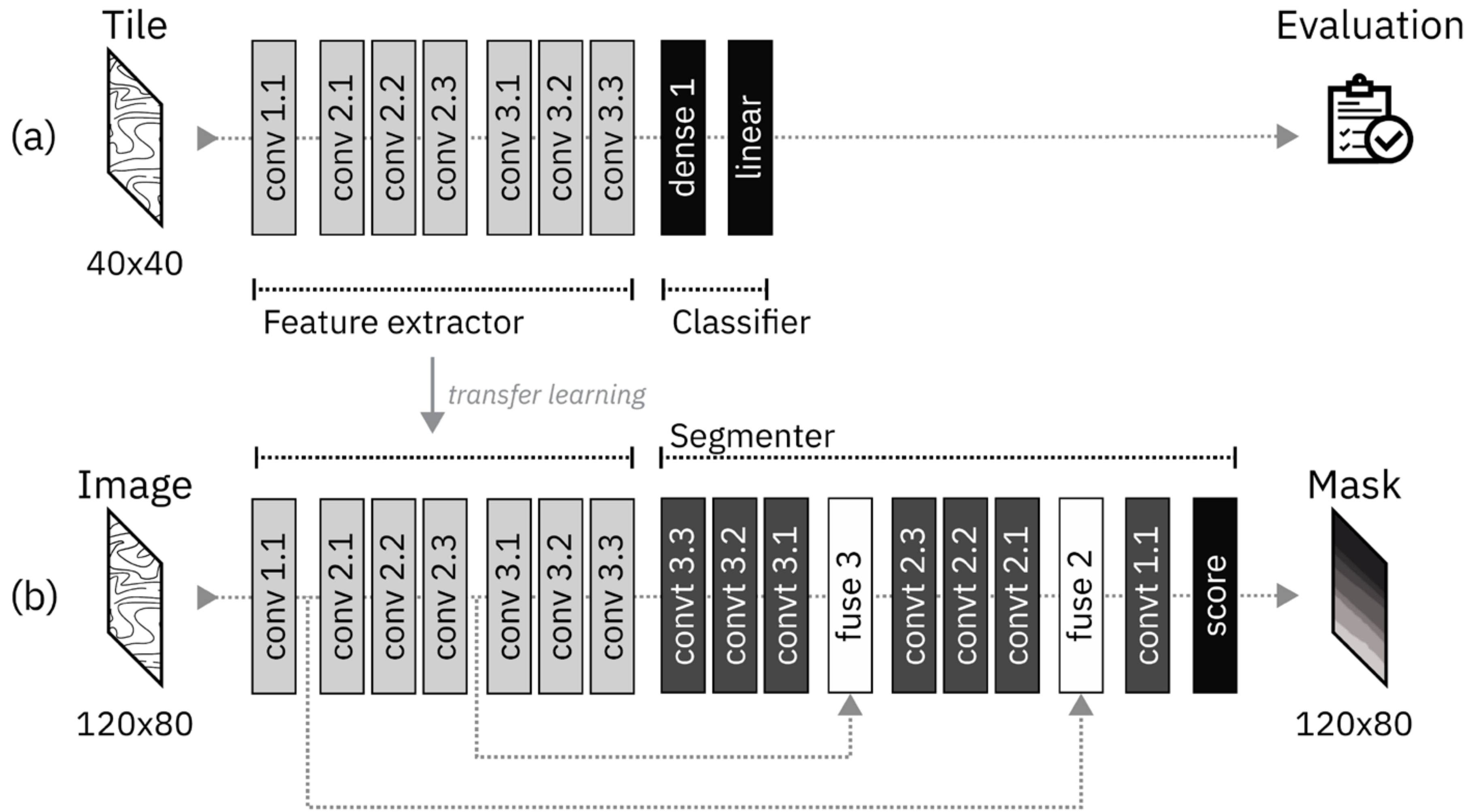


Figure 2. (a) Classifier structure, which comprises the feature extractor block with two fully connected layers (dense1 and linear) at the end. In the feature extractor, conv blocks represent a convolution operation followed by a ReLU activation and a batch normalization (Chevitarese et al., 2018b). (b) Semantic segmentation architecture: after the feature extractor, we append an upscale structure composed by transposed convolutions (convt) and a final score layer that is a convolution operation at the pixel level.

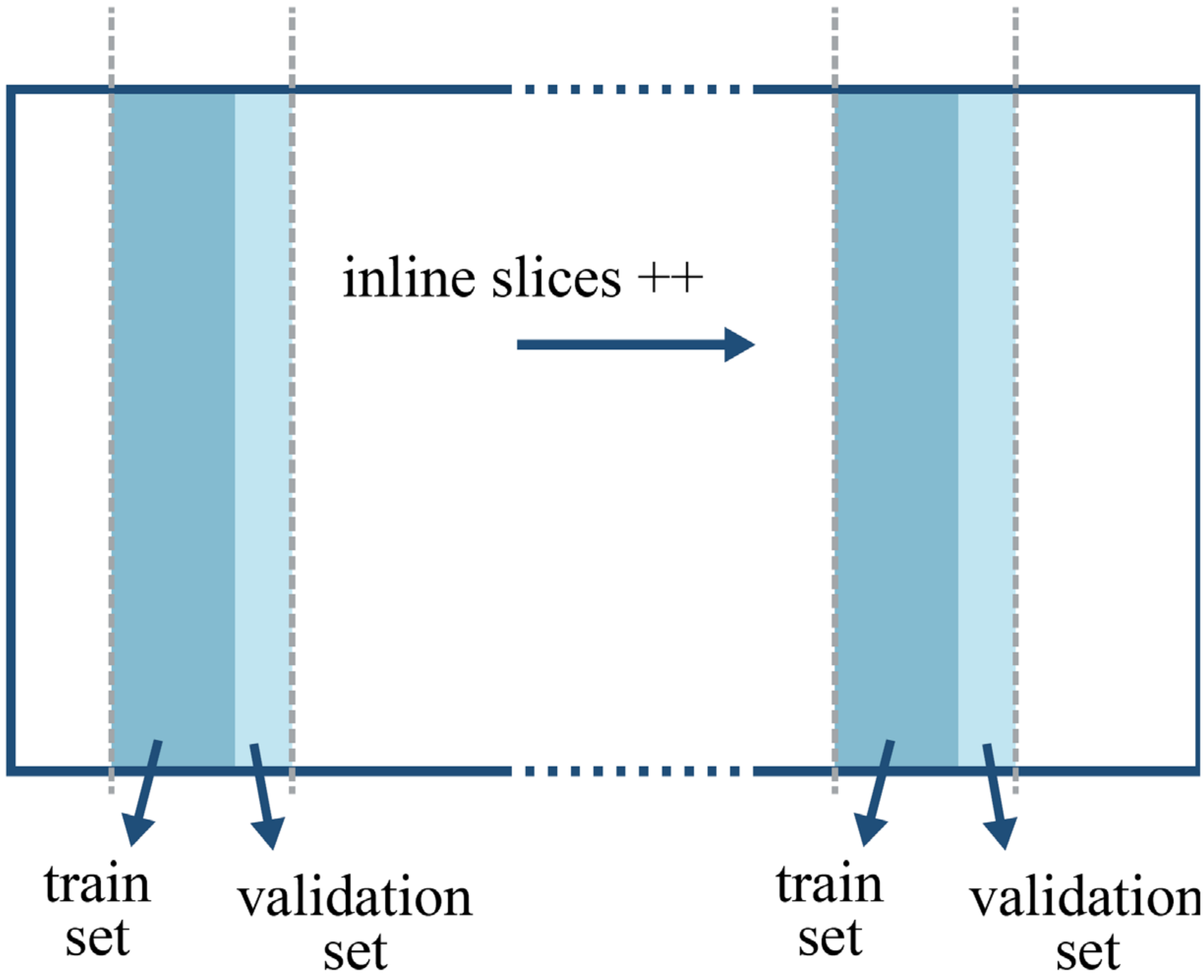


Figure 3. Cube splitting to separate inline slices into train and validation sets.

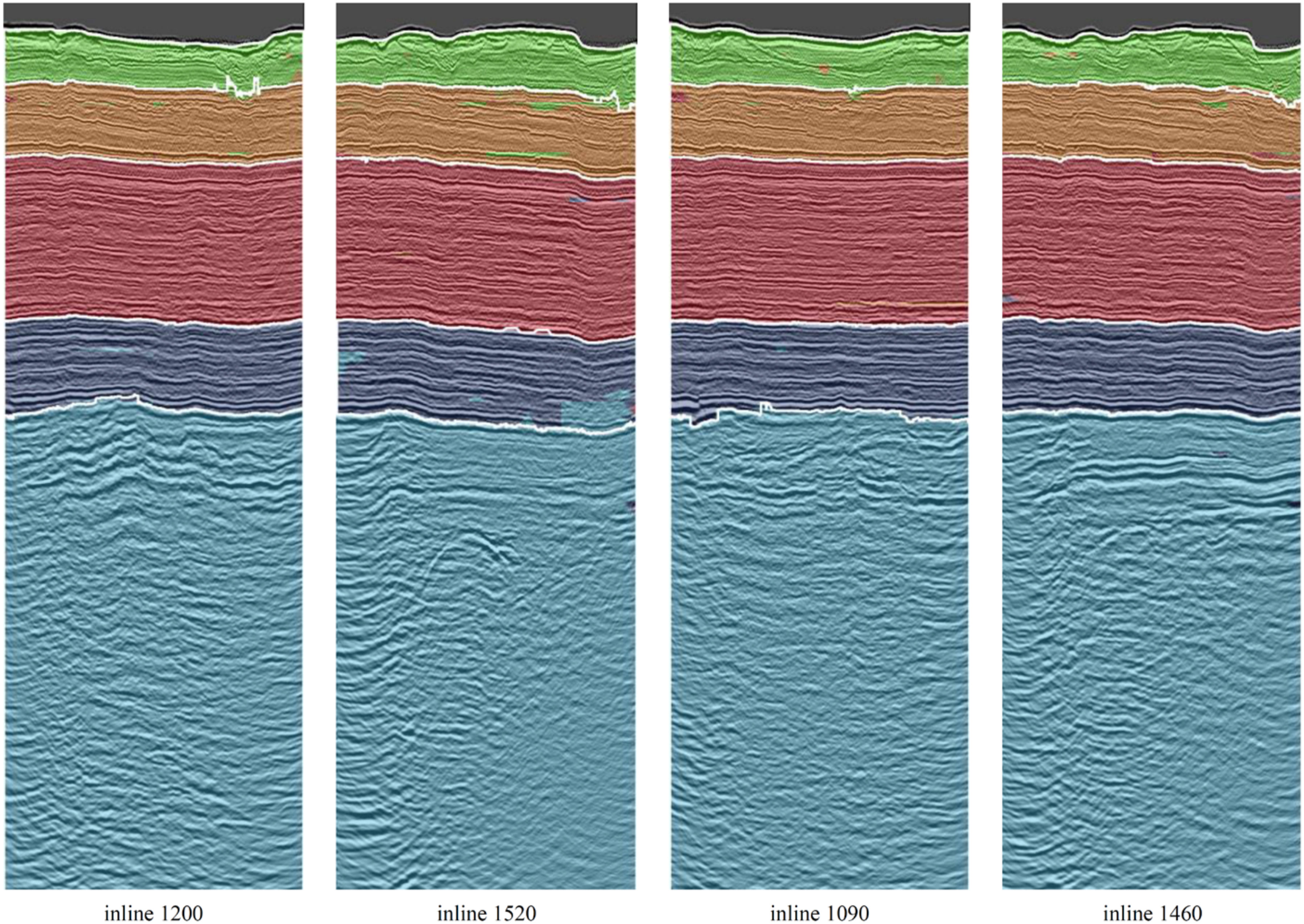
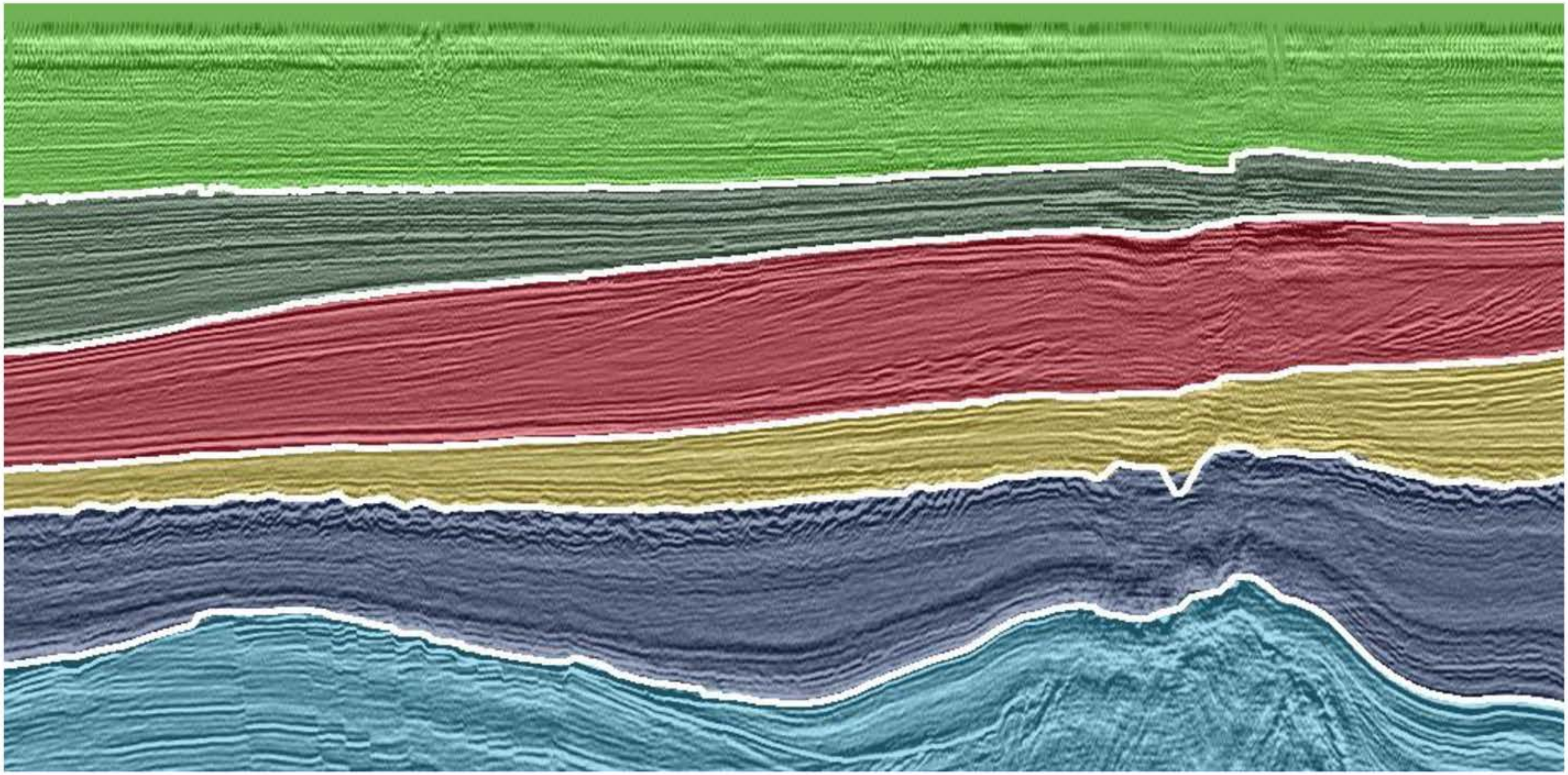
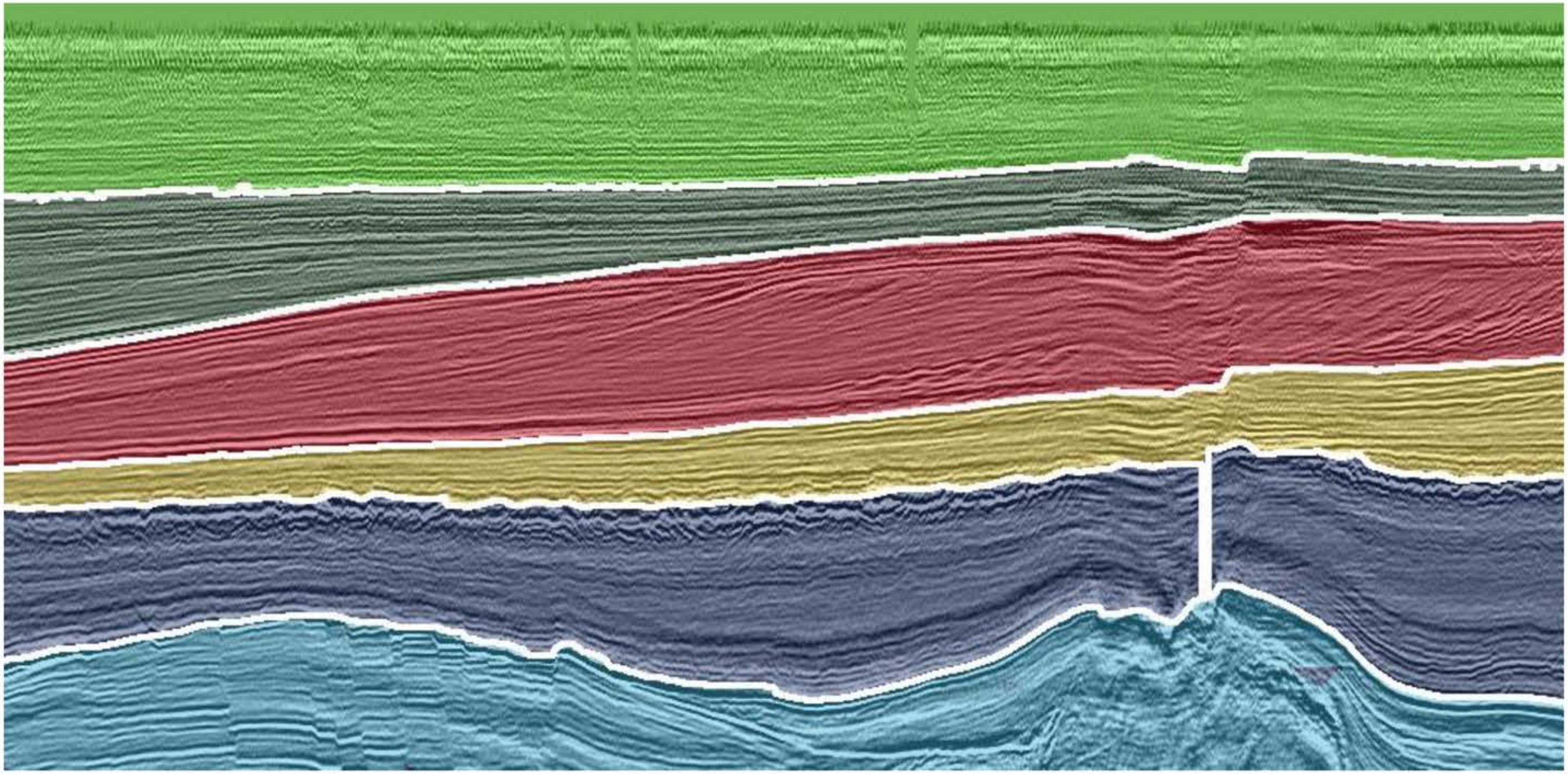


Figure 4. Results for Penobscot.



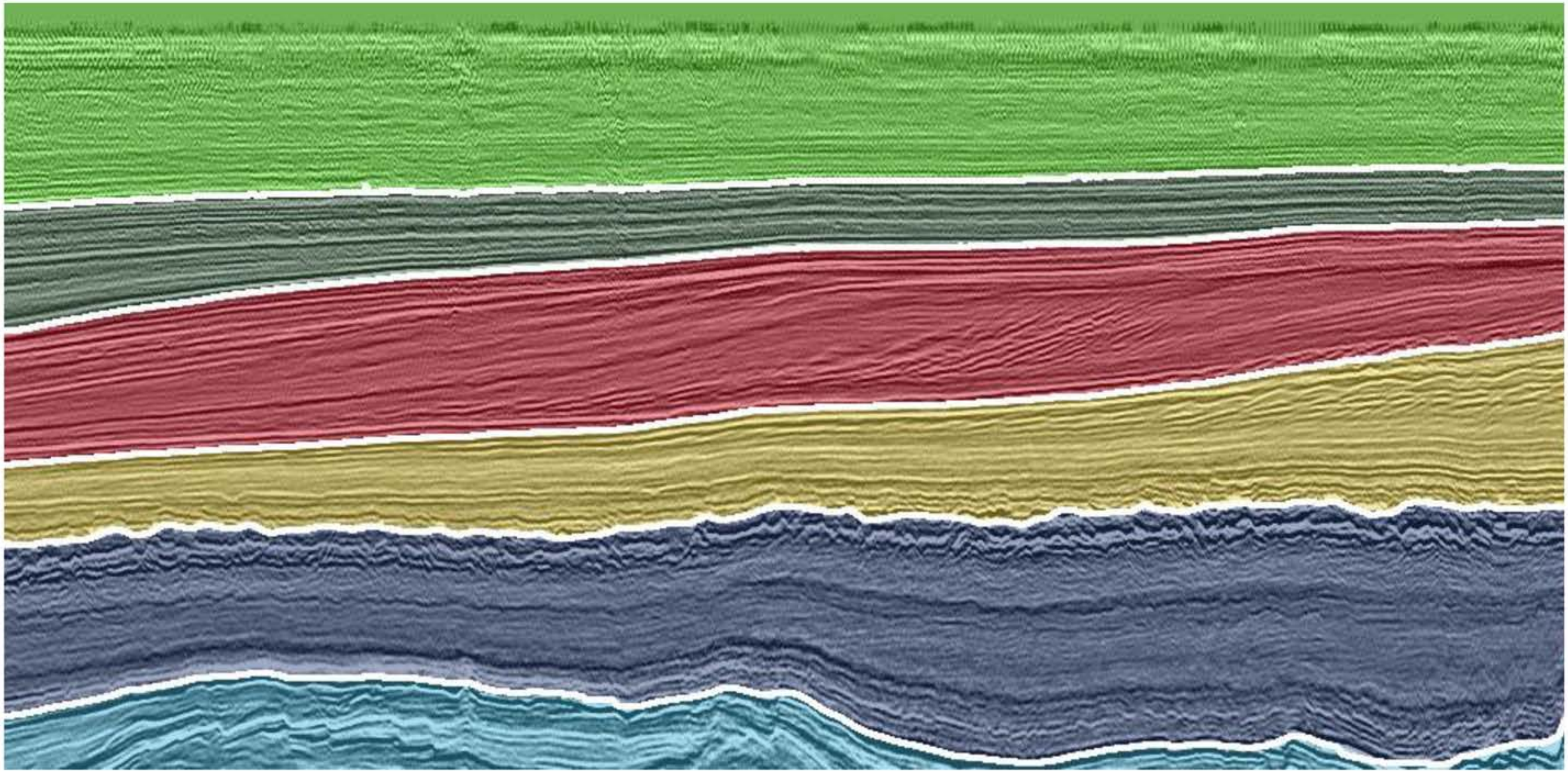
inline 160

Figure 5. Results for Netherlands – inline 160.



inline 120

Figure 6. Results for Netherlands – inline 120.



inline 500

Figure 7. Results for Netherlands – inline 500.

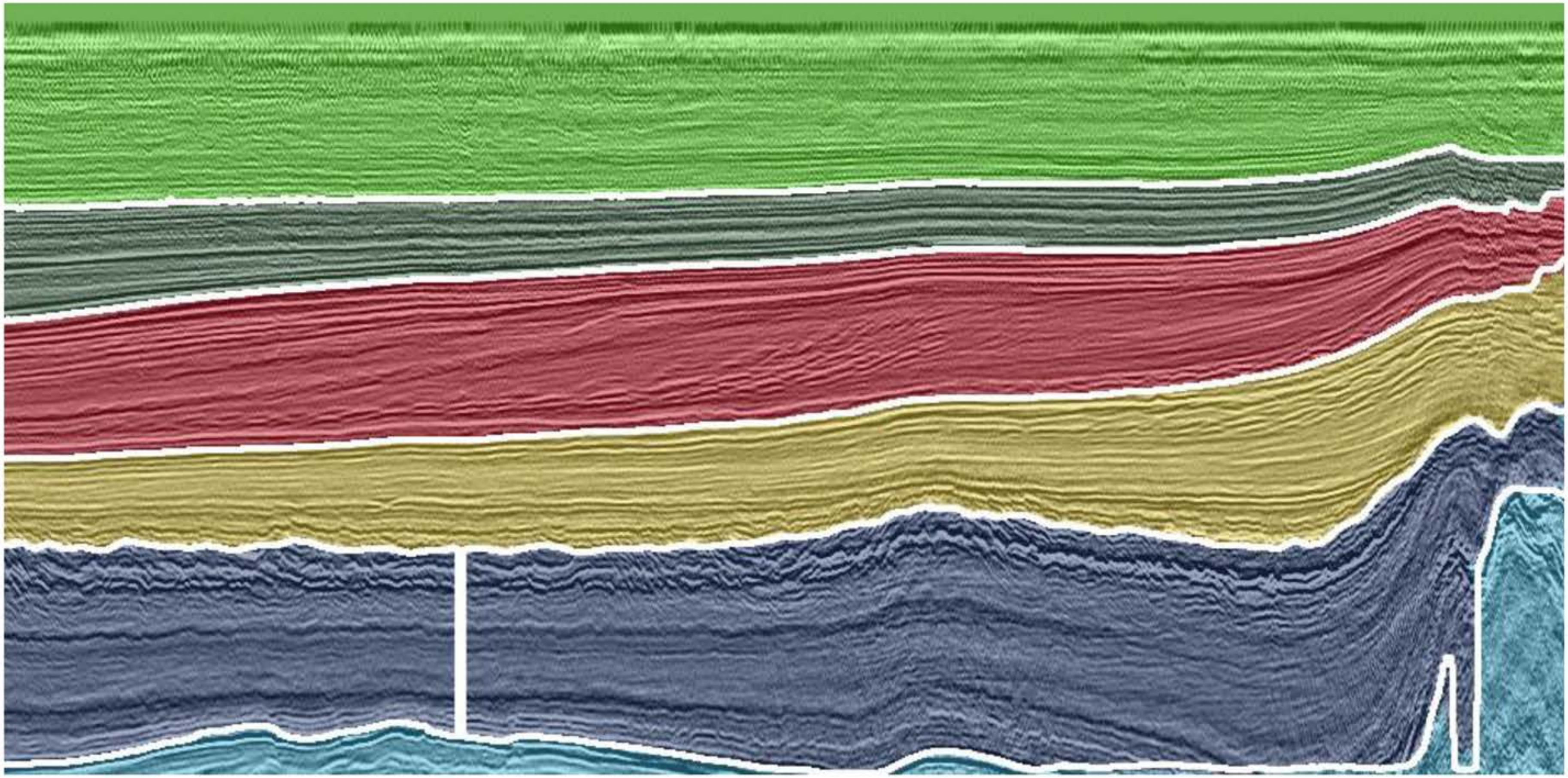

inline 640

Figure 8. Results for Netherlands – inline 640.

Datasets	Mean IOU	Time to train (200 epochs)	Number of training examples
Penobscot	0.9805	3h43m	29,624
Netherlands	0.9825	9h51m	231,840

Table 1. Mean intersection over union (IOU) for the trained models in the validation datasets.