Simplifying Horizon Picking Using Single-Class Semantic Segmentation Networks

Danilo Calhes*, Felipe K. Kobayashi†, Andrea Britto Mattos‡, Maysa M. G. Macedo‡ and Dario A. B Oliveira‡
*IBM
Rua Tutoia, 1157, Sao Paulo, Brazil
Email:danilo.calhes@ibm.com
†Federal University of ABC
Avenida dos Estados, 5001, Santo André, Brazil
Email:felipe.k@aluno.ufabc.edu.br
‡IBM Research
Rua Tutoia, 1157, Sao Paulo, Brazil
Email:abritto,mmacedo,dariobo@br.ibm.com

***Abstract*—Seismic image processing plays a significant role in geological exploration as it conditions much of the interpretation performance. The interpretation process comprises several tasks, and Horizon Picking is one of the most time-consuming. Thereat, several works proposed methods for picking horizons automatically, mostly focusing on increasing the accuracy of data-driven approaches, by employing, for instance, semantic segmentation networks. However, these works often rely on a training process that requires several annotated samples, which are known to be scarce in the seismic domain, due to the overwhelming effort associated with manually picking several horizons in a seismic cube. This paper aims to evaluate the simplification of the labeling process required for training, by using training samples composed of disconnected horizons tokens, therefore relaxing the requirement of annotating the full set of horizons from each training sample, as commonly observed in previous works employing semantic segmentation networks. We assessed two state-of-art neural networks for general-purpose domains (PSP-Net and Deeplab V3+) using public seismic data (Netherlands F3 Block dataset). Our results report a minor impact in the performance using our proposed incomplete token training scheme compared to the complete one, moreover, we report that these networks outperform the current state-of-art for horizon picking from small training sets. Thus, our approach proves to be advantageous for the interpreter, given that using partial results instead of providing a full annotation can reduce the user effort during the labeling process required for training the models.**

I. Introduction

[1]Machine Learning has been employed successfully in several industries and specifically for geophysical image analysis, where automatic approaches can support data acquisition, processing, and interpretation tasks. The delineation of geological structures and potential reservoirs is an essential step for seismic image analysis, in a process called Horizon Picking, and several works previously approached it from different perspectives.

First efforts to support horizon picking emerged many years ago, ranging from classic tracking approaches [1] to more sophisticated ones, for instance, using Kalman filter [2]. However, because most of those methods struggle at discontinuous structures (such as faults) [3], recent works employ deep learning-based alternatives and report good-quality results. Among noticeable achievements, Wu and Zhang [4] proposed a convolutional encoder-decoder neural network to handle horizon picking as a semantic segmentation problem, and Chevitarese *et. al* [5] proposed an optimized architecture for obtaining horizons in a similar schema.

In current industry practice, accurate horizon picking can take weeks or even months [6], and therefore many works focus on speeding up the manual picking process. Conversely, conventional deep learning models require a high amount of training data, adding a significant charge for labeling images in the seismic domain. In this context, recent works focused on horizon picking from small training sets. Peters *et. al* [7] started to focus on delineating geological features (such as horizons and geological units) using data with a small subset of pixels with known labels and investigated strategies for choosing new meaningful pixels to be labeled. Tschannen *et .al* [8] proposed a method that requires the experts to label only a few lines through the survey to yield good initial results, and offer ways to how interpreters can progressively improve the predictions by fine-tuning the network training. Shi *et. al* [9] proposed an unsupervised approach using a deep convolutional autoencoder network based on waveform patterns. Mattos *et. al* [10] described a post-processing method using 3D geometrical information that greatly reduced the required amount of training data with little loss on the outcome quality. This process was named *mesh optimization*.

In this work, we build on this literature and propose a flexible methodology that improves the efficiency of target horizons labeling for horizon picking based on semantic segmentation networks. Additionally, we report results overcoming the state-of-art for horizon picking using small training sets.

[1]The authors Calhes and Kobayashi contributed equally to this work.

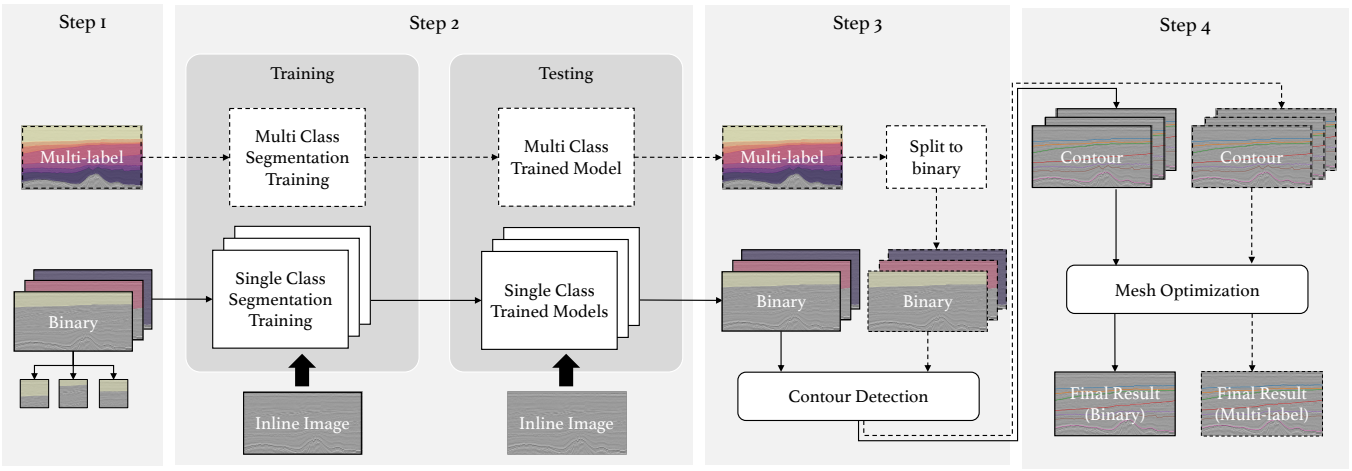


Fig. 1. Pipeline of the performed experiments, consisting of four main steps, demonstrating the differences between the multi-class approach (performed in previous works), and our single-class approach (described in this paper). The inner blocks represented in dashed lines describe the operations of the multi-label experiment while the inner blocks with contours as solid lines indicate our approach, based on binary labels. In this paper, both experiments were performed for comparing the obtained results.

II. Method

The acquisition of seismic data is made by placing receivers on the surface of the Earth or seafloor, and using a source (such as a dynamite shot or an air gun) to generate vibrations that travel into the Earth. The signal passes through strata with different seismic responses and returns to the surface, where it is recorded by the receivers as seismic volume. In the geophysics context, an inline is a seismic line within a 3D survey parallel to the direction in which the data were acquired, while a crossline is perpendicular to the direction in which the data were acquired [11].

The proposed method receives as input a set of inline (or crossline) images and outputs the corresponding horizon coordinates for all detected horizons. It consists of four distinct steps, depicted in Figure 1, which also displays the comparison of our pipeline based on a conventional multi-label approach (described in dashed lines) with our approach using binary labels (represented in solid lines).

The first task of each experiment is to build the training dataset (see Step 1 in the diagram). Unlike the conventional semantic segmentation multi-label approach, where all horizons coordinates from the same inline image must be fully annotated, our strategy creates the training set comprising binary tiles of individual horizons, where pixels above a horizon are labeled as 1 and pixels below the horizon are labeled as 0. Therefore, for N target horizons, we input N sets of tiles with binary labels. These tiles not necessarily need to cover the entire horizon coordinates, like the previous method does, and may be generated from discontinuous regions of the image, as well as extracted from random areas of the cube, allowing a much flexible training annotation.

In Step 2, we train the semantic segmentation network for the seismic data. Whereas the multi-label approach generates a single trained model, we train N networks (one per horizon) using tiles of seismic images, deriving an individual trained model for each horizon considered. Then, we test the whole seismic cube using the previously trained models.

In Step 3, the resulting images after the horizon prediction are subject to a contour detection process. Following the same experimental pipeline for both scenarios, for the multi-label case in particular, each segmented output must be first split into binary images before detecting the individual horizon contours.

Last, in Step 4, we post-process the derived concatenated contours by applying the mesh optimization process from Mattos *et. al* [10] mentioned earlier, and obtain the final picked horizons. The same post-processing methods are applied for both multi-class and single-class outputs (*i.e.*, contour detection and mesh optimization). Each of these steps are detailed in the following subsections.

A. Training sets creation

For creating the training sets used for all the experiments described in this paper, we follow the same sampling scheme adopted by Mattos *et. al* [10], and select every labeled inline at each 20 inlines from the seismic cube. To address both of our experimental designs, we created two different training sets: one with all target horizons annotated to implement the conventional semantic segmentation pipeline, and the other training set comprising tiles from binary images for each horizon, to validate our proposed approach. It is important to notice that, in the second scheme, each horizon data disregards completely the information of neighbouring horizons, which means that the network trained with samples of a single input horizon will simply learn to identify the upper and lower regions of that given horizon.

B. Semantic segmentation networks training

We explored two different state-of-art neural networks to segment the input inlines and derive the corresponding horizons. In all our experiments, the semantic segmentation networks indirectly identify the horizons by classifying pixels between horizons into different classes. The proposed single-class approach labels a given inline targeting a single horizon, where pixels above the detected region are labeled as 1, and pixels below as 0. For the baseline multi-class method, the selected inlines are fully labeled and assume values between 1 and 7 for the tested dataset (which contains 7 labelled horizons) and zero for the background.

For testing the whole seismic cube, *i.e.,* the remaining unlabeled inlines, we extract the tiles, submit them to the trained model and concatenate the results following the original shape structure. To avoid border effects, we used a stride strategy conserving the 64×64 kernel of predicted tiles, as depicted in Figure 2.

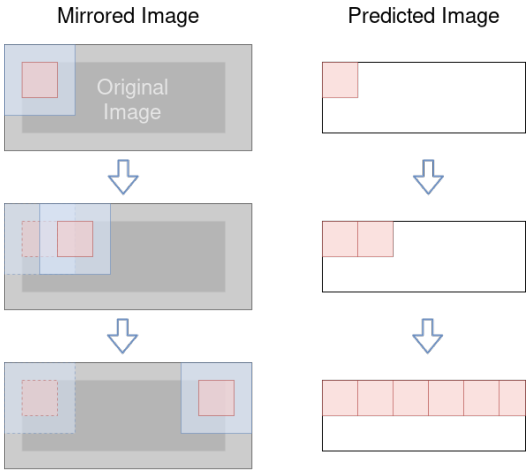


Fig. 2. Stride strategy to avoid border effects in the prediction. In dark gray, the original inline image, and in red the 64×64 kernel used in the final composition. Padding is represented by the lighter gray area.

C. Post-processing

For the multi-label experiment, because the trained network outputs a multi-class labeled image, we had to introduce an intermediate step so both experiments could follow the same pipeline. This step, denominated *Split to binary*, creates K binary images by merely identifying the respective K labels in the multi-class labeled image.

Since the semantic neural networks used in our experiments (for both single-class and multi-class scenarios) do not implement any horizon geometrical constraint, we expect their outcomes to deliver segmentation results that are not always consistent with the expected output, eventually presenting holes, discontinuities, and unrealistic connections between different horizons. To fix some of these inconsistencies, we propose simple image processing steps, presented in Figure 3, that enable finding the most likely interface points between the different predicted regions, and therefore derive the best possible horizon for that segmentation result.

We start by computing the contours for all the predicted images (*i.e.*, all inlines of the seismic cube) using a marching squares algorithm. Next, we select those results where a full contour is found and do not form the trivial straight-line marching squares solution. Then, we compute these contours' average height values and use them as a parameter for finding the most likely contour points in the other incomplete or ambiguous contours.

With a set of contour candidates, we discard those with more than 50% of its points in the same latitude to exclude the marching squares trivial solution. Then, we select the most elongate segments and check those with the closest average height compared to the previously estimated height value. Then, we finally derive a point cloud consisting of inline, crossline, and depth coordinates to be used in the mesh optimization step [10] for eliminating spurious geometrical noise.

III. Experiments

As described in Section II, we conducted our experiments considering two state-of-art semantic segmentation networks and two different training sets: a multi-class set with fully annotated target horizons, and a single-class set with annotation of independent individual horizons.

We used the Netherlands F3 dataset through all our experiments, carried out in the North Sea, Netherlands offshore. The data is publicly available and contains post-stack data, horizons, and well logs annotated by an expert [12]. For this work, we used 7 horizons in total, and the seismic cube has a total of 651 inlines. Figure 4 display one annotated inline of the Netherlands F3 dataset.

We removed mute trace regions, *i.e.*, regions where no valid signal is observed, using a simple pre-processing based on the mute region characteristic percentile. We enhanced the seismic cube contrast by mapping linearly the amplitude values from the [2^{th}, 98^{th}] range to [0,1].

We selected 32 inlines and labeled images and sampled a total of 750 tiles of 256×256 pixels at random per inline, comprising a total of 24.000 training samples. These 32 inlines were used for predicting the horizons at the remaining 651 inlines of the seismic cube. Considering that we had a total of 7 horizons, our single-class training set consisted of 7 training sets with binary annotations for each horizon, that derived each one a trained model to be used for testing the whole seismic cube. For the fully annotated training set, the total number of the training set was 7×750 per inline to enable a fair comparison, comprising a total of 168.000 training samples for this model. We further split each selected data into training, validation, and testing sets, with the proportion of 70%, 15%, and 15%, respectively.

For running the semantic segmentation task, we used two state-of-art semantic segmentation networks for general-domain purposes: PSP-Net [13] and Deeplab V3+ [14]. Although our initial goal was to address single-class *versus*

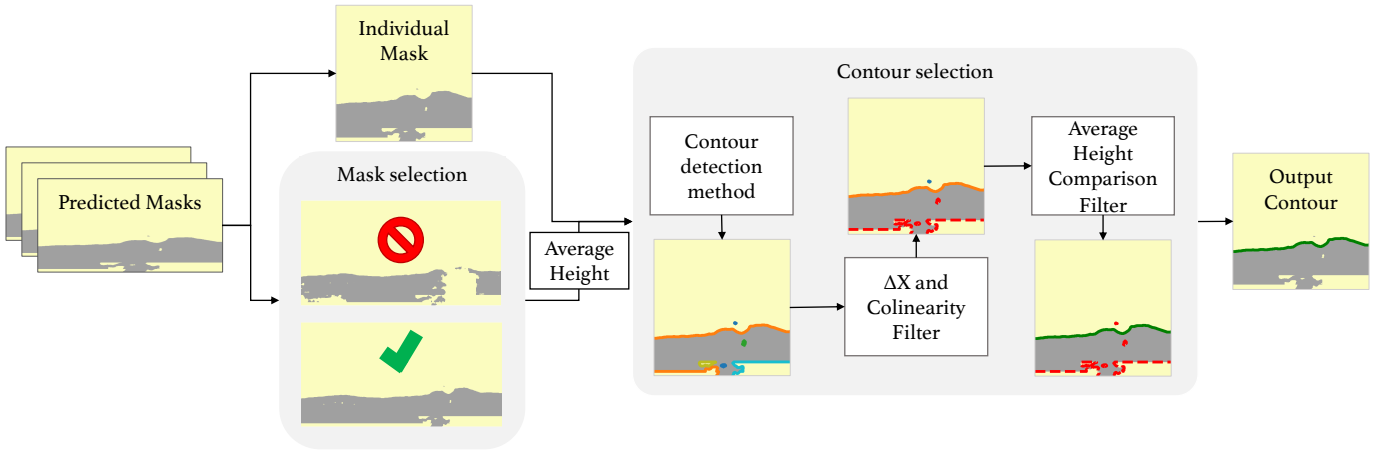


Fig. 3. Method for horizon contour detection. After a pre-selection of inlines without faults, the average height of the horizon is calculated and the marching squares method is performed to identify contour points. Then, the contour points which have the greatest coverage in the x-axis are selected and the contour points that have more than 50% of the found points in the same line are discarded. Finally, we sort the available contours by their average height, selecting the one that has the closest average height compared to the previously stored value. The colors of the contour points represent different contour detections. After each step, the rejected predictions are represented in dashed red lines and the output is represented with a green continuous line.

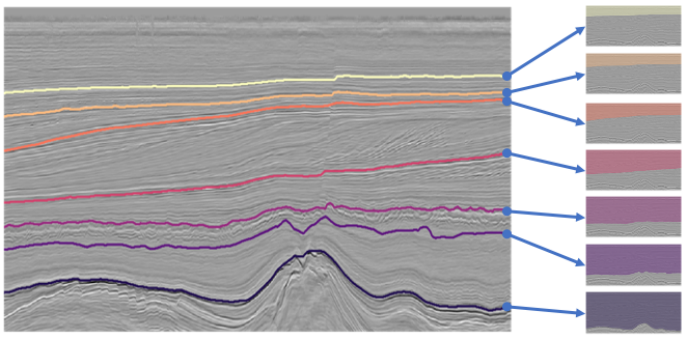


Fig. 4. Annotated inline sample of the considered dataset. Each horizon line presented in the left image corresponds to the binary mask presented in the right (used for extracting tiles for our single-class training experiments). The represented horizons, from top to bottom, are denoted *North Sea, Chalk, Rijnland, SSN, Altena, Germanic*, and *Zechstein*.

multi-class performance, we also wanted to evaluate how these networks would perform when solving the problem of horizon picking from small datasets. The kernel size (for the stride strategy displayed in Figure 2) was defined empirically: we evaluated different values and found the best results with a window of 64×64 pixels. It was also the best configuration in terms of execution time.

As the metrics for evaluating the results, we used Mean Intersection-over-Union (mIoU) and accuracy, and the Jaccard distance and weighted categorical cross-entropy as loss functions. To optimize the networks, we used Adam optimizer with a learning rate of 0.001. We report the accuracy in our results using the distance in milliseconds to the corresponding horizons manually picked by the expert, as in Mattos *et. al* [10].

With a trained network, we evaluated the whole seismic cube using a sliding window scheme, preserving the kernel information of each tile prediction to mount the inlines result, as depicted in Figure 2 and presented in Section II. Then, we applied the contour detection pipeline and the mesh optimization method to derive the final horizon predictions, as previously described. Finally, we computed the distance between the obtained predictions and the ground truth data, and derived the mean distance value for each horizon and coverage area with a distance less than 8 milliseconds (*ms*), as in [10]. As the Netherlands F3 dataset resolution is 4*ms*, the latter metric selects the area within a 2-pixel range from the ground truth image.

IV. Results and Discussion

We present results of four experiments, using the two different semantic segmentation networks and the multi-class and single-class training data. We further compare these results with those presented by Mattos *et. al* [10], which represents the current state-of-art for horizon picking based on small training sets, using the same Netherlands F3 dataset and employing the same evaluation metrics (mean distance and coverage area). The network employed by this baseline work previously reported having outperformed the U-Net and FCN (Fully Convolutional Network) architectures, therefore, we wanted to check how PSP-Net and Deeplab V3+ would perform compared to it.

The obtained results of our experiments, as well as the ones provided by the baseline work, are reported in Table I and Table II, together with the corresponding loss functions employed[2]. There are two distinct metrics used in those results: the mean distance, which represents the average time difference, in milliseconds, between the predicted horizon line and the ground truth line, and the coverage area, which shows what percentage of the predicted lines has a time difference below a threshold, in this case 8*ms*. For the mean distance metric (see Table I), values closer to 0 are desirable, and for the metric computing the coverage area within 8*ms* (see Table II),

TABLE I
MEAN DISTANCE BETWEEN THE GROUND-TRUTH AND THE PREDICTED RESULTS (MILLISECONDS).

		Metric	Loss function	North Sea	Chalk	Rijnland	SSN	Altena	Germanic	Zechstein	Average
Single-class	**PSP-Net**	mIoU	Jaccard	2.51	2.46	**2.06**	**2.11**	3.4	5.36	**4.44**	3.19
	DeeplabV3+	Accuracy	Weighted	22.11	2.59	2.45	2.7	3.45	7.25	4.5	6.43
Multi-class	**PSP-Net**	mIoU	Jaccard	**2.32**	**2.39**	2.13	2.14	**3.1**	**5.25**	3.71	**3.01**
	DeeplabV3+	Accuracy	Weighted	29.86	19.78	2.81	3.11	3.65	6.24	7.3	10.39
Mattos *et. al* [10]		mIoU	Not informed	2.98	3.18	2.42	2.47	4.98	7.52	4.61	4.02

TABLE II
PERCENTAGE OF COVERAGE WITH DISTANCE UNDER 8*ms* BETWEEN THE GROUND-TRUTH AND THE PREDICTED RESULTS (%).

		Metric	Loss function	North Sea	Chalk	Rijnland	SSN	Altena	Germanic	Zechstein	Average
Single-class	**PSP-Net**	mIoU	Jaccard	95.83	96.51	**99.03**	98.79	87.5	**73.56**	56.07	86.76
	DeeplabV3+	Accuracy	Weighted	72.64	96.8	97.94	97.27	87.7	59.82	44.57	79.53
Multi-Class	**PSP-Net**	mIoU	Jaccard	**96.57**	**97.61**	98.99	**98.97**	**90.63**	73.29	**68.09**	**89.16**
	DeeplabV3+	Accuracy	Weighted	70.61	84.54	92.19	90.83	84.49	67.49	40.35	75.79
Mattos *et. al* [10]		mIoU	Not informed	91.93	91.19	94.57	93.06	74.37	57.03	66.37	81.22

values closer to 100% are ideal. In both tables, we display the achieved results for each individual horizon as well as the average values for the 7 horizons considered.

Comparing the mean distance to the ground truth (Table I), one can observe that PSP-Net network delivered the best results for both the single-class and the multi-class experiments, in comparison with the Deeplab V3+ network, for virtually all considered horizons. It also delivered an improvement on the distance error compared to the baseline work [10], from average values of 4.02*ms* to 3.01*ms* (an improvement of around 25%) for the multi-class scenario. Although this would be the most straightforward comparison, once the authors also employ multi-class annotation, our single-class experiment also delivered results that outperform the current state-of-art, from average values of 4.02*ms* to 3.19*ms* (an improvement of around 20%). Comparing the single-class and multi-class results, for PSP-Net, we observed a slight drop in performance (from average values of 3.01*ms* to 3.19*ms*), due to relaxing the need of fully annotating the target horizons. However, this drop (around 5%) seems to be fair for the benefit involved. Figure 5 shows the PSP-Net result for single-class for two different horizons.

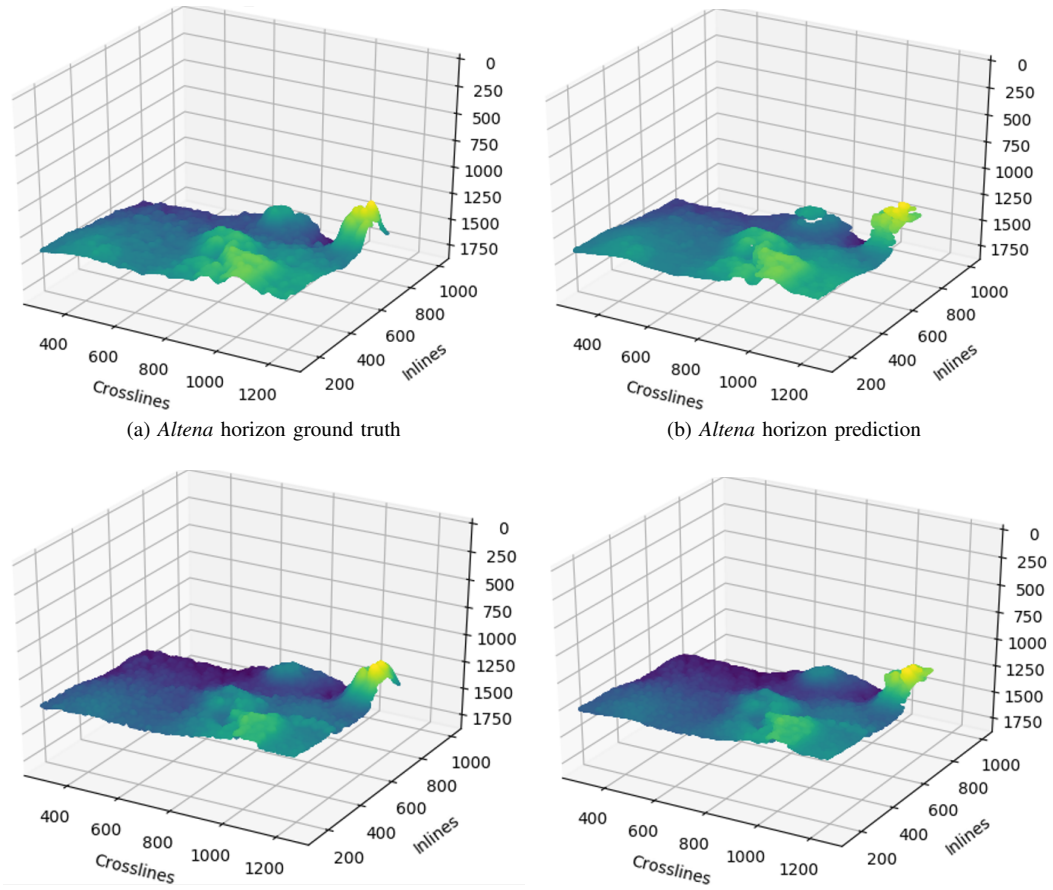
Interestingly, unlike PSP-Net, for Deeplab V3+, the single-class experiment outperformed the multi-class one, with average values of 6.43*ms* and 10.39*ms*, respectively, a counter-intuitive finding. In fact, the single-class experiments with Deeplab V3+ achieved better results for all horizons except for the Germanic group, in comparison with the output from the multi-class model. Also, although PSP-Net consistently performed better than Deeplab V3+, it is also important to notice that, compared to the previous baseline, the single-class experiment with Deeplab V3+ reported lower mean distance values for Chalk, Altena, Germanic, and Zechstein horizons. Corroborating with previous works that employ PSP-Net and Deeplab V3+ for seismic image processing, our results indicate that these networks, well established at the computer vision community for segmentation in general-purpose applications, may have good potential for the geophysical domain as well.

Considering the area coverage for distances within 8*ms* (Table II), PSP-Net also delivered the best results, consistently with the previous metric, for all considered horizons. Similar to the results observed for mean distance, the single-class experiment with Deeplab V3+ also outperformed the multi-class one (again, with exception for the Germanic horizon) and also achieved slightly better results compared to the baseline [10] for most horizons.

Curiously, for the topmost horizon (North Sea group), both single-class and multi-class versions of Deeplab V3+ presented poor results in comparison to the outputs produced by PSP-Net and the baseline work, considering both metrics. This behavior was unexpected due to relatively simple structure of the North Sea horizon (see Figure 4) and requires further investigation. Possible assumptions rely on the choice of the inlines composing the training dataset or on the fact that, being a more complex network, the parametrization of Deeplab V3+ is also more intricate, thus our configuration may be failing on handling a more trivial solution.

It is also important to notice that the two lower horizons (Germanic and Zechstein) often presented worst results in

[2]In both tables, *Jaccard* refers to the Jaccard Distance loss function and *Weighted* refers to the Weighted Categorical Cross-Entropy loss function.



(a) *Altena* horizon ground truth

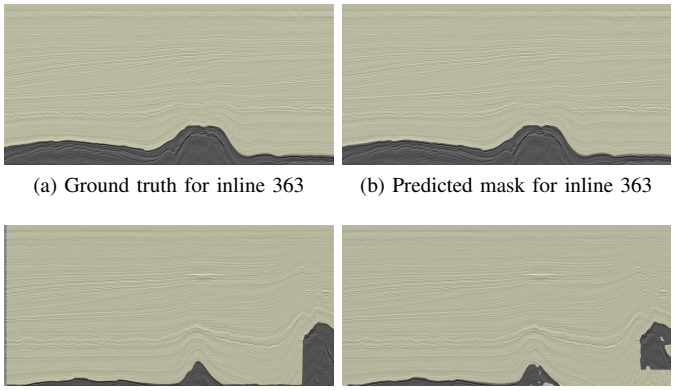
(b) *Altena* horizon prediction

(c) *Germanic* ground truth

(d) *Germanic* horizon prediction

Fig. 5. PSP-Net results for single-class classification for two different horizons, *Altena* and *Germanic*.

comparison to the other horizons, for all experiments and also for the baseline work. This behavior, however, is expected, due to the greater complexity of these horizons (again, refer to Figure 4 for exemplification). They both have a low average height and a wide amplitude variance, impacting the results. Also, Zechstein reaches the bottom edge of the image in some inlines, and is characterized by several salt domes, which also interfere in the structure of the remaining horizons, especially in Germanic, the adjacent one. As a consequence of this behaviour, the predicted mask could present some holes and discontinuities and so the contour detection algorithm will not predict correctly. Figure 6 depicts a case of a precise prediction and a case of a flawed prediction for Zechstein horizon using the single-class version of PSP-Net (using mIoU metric and Jaccard loss function). Although the predicted mask is feed to the post-processing pipeline and the contour detection method displayed in Figure 3, some errors are inevitably propagated to the subsequent steps.



(a) Ground truth for inline 363

(b) Predicted mask for inline 363

(c) Ground truth for inline 666

(d) Predicted mask for inline 666

Fig. 6. Two different inlines with segmentation results for the Zechstein horizon using PSP-Net. The top rows displays an example of a good prediction and the bottom row displays a defective one.

For PSP-Net, the best results achieved used mIoU as the evaluation metric and Jaccard distance as the loss function, while, for Deeplab V3+, the best results were obtained using accuracy as the evaluation metric and weighted categorical cross-entropy as the loss function. The convergence of the used metrics was achieved with 200 epochs, both for PSP-Net and DeepLab V3+ in the cases using mIoU metric. For accuracy, the convergence was achieved with 50 epochs for accuracy in both networks. Finally, for the multi-label case, both mIoU and accuracy reached convergence using 50 epochs in both networks.

V. Conclusions

In this work, we presented an approach for simplifying training data annotation for horizon picking based on semantic segmentation networks, which allows to reduce the annotation effort dramatically.

We proposed to create the training sets for semantic segmentation networks from individual horizon annotations instead of requiring inlines with full annotation for all target horizons under consideration, which enables the expert to pick individual horizons in the most convenient inlines or crosslines. We defined a training set for each horizon individually and consequently trained a specific model for each horizon, compiling their results at a post-processing stage that tackle geometrical inconsistencies among each derived point cloud. We further evaluated two commonly used semantic segmentation architectures that report very competitive results in the computer vision community. We increased the performance achieved by a state-of-art work for horizon picking using small training sets, both in multi-class (around 25% of improvement for the mean distance) and in our proposed single-class segmentation (around 20% of improvement for the mean distance).

Thinking about the day-to-day of expert Geologists, using this proposed method, they will be able to reduce the amount of points annotated for each horizon that they would like to segment.

As further research, we intend to explore the post-processing step to remove geometrical inconsistencies and fix them. Another future work is running experiments based on annotation from well-established automatic tools, as well as new experiments using other datasets so that the generalization of the network can be confirmed. We also intend to run experiments where the experts would create horizon picking annotations considering the more convenient inlines and horizons with the automatic tools they usually have in hand. That would enable us to evaluate a real scenario application of the proposed methodology.

References

[1] C. A. Woodham, W. A. Sandham, and T. S. Durrani, "3-d seismic tracking with probabilistic data association," *Geophysics*, vol. 60, no. 4, pp. 1088–1094, 1995.

[2] X. Deng and Z. Zhang, "Automatic multihorizons recognition for seismic data based on kalman filter tracker," *IEEE Geoscience and Remote Sensing Letters*, vol. 14, no. 3, pp. 319–323, 2017.

[3] Y. Lou, B. Zhang, T. Lin, and D. Cao, "Seismic horizon picking by integrating reflector dip and instantaneous phase attributes," *Geophysics*, vol. 85, no. 2, pp. O37–O45, 2020.

[4] H. Wu and B. Zhang, "A deep convolutional encoder-decoder neural network in assisting seismic horizon tracking," *arXiv preprint arXiv:1804.06814*, 2018.

[5] D. Chevitarese, D. Szwarcman, R. G. e Silva, and E. V. Brazil, "Deep learning applied to seismic facies classification: a methodology for training," in *Saint Petersburg 2018*, 2018.

[6] B. Zhang, J. Qi, Y. Lou, H. Fang, and D. Cao, "Generating seismic horizon using multiple seismic attributes," *IEEE Geoscience and Remote Sensing Letters*, pp. 1–5, 2020.

[7] B. Peters, J. Granek, and E. Haber, "Automatic classification of geologic units in seismic images using partially interpreted examples," in *81st EAGE Conference and Exhibition 2019*, 2019.

[8] V. Tschannen, M. Delescluse, N. Ettrich, and J. Keuper, "Extracting horizon surfaces from 3d seismic data using deep learning," *Geophysics*, vol. 85, no. 3, pp. N17–N26, 2020. [Online]. Available: https://doi.org/10.1190/geo2019-0569.1

[9] Y. Shi, X. Wu, and S. Fomel, "Waveform embedding: Automatic horizon picking with unsupervised deep learning," *Geophysics*, vol. 85, no. 4, pp. WA67–WA76, 2020.

[10] A. B. Mattos, D. Civitarese, D. Szwarcman, M. Oliveira, S. Zaytsev, D. G. Semin, and D. A. Oliveira, "Enabling robust horizon picking from small training sets," *IEEE Transactions on Geoscience and Remote Sensing*, 2020.

[11] "The Schlumberger Oilfield Glossary," http://www.glossary.oilfield.slb.com/, accessed: August 2021.

[12] R. M. Silva, L. Baroni, R. S. Ferreira, D. Civitarese, D. Szwarcman, and E. V. Brazil, "Netherlands dataset: A new public dataset for machine learning in seismic interpretation," *arXiv preprint arXiv:1904.00770*, 2019.

[13] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[14] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.